SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I11P105

Volume 13 Issue 11, 66-102, November 2026
© 2026 Seventh Sense Research Group®

Original Article

A Hybrid Quantum-Classical Squeezed Sparse Network for Malignancy Classification of Pulmonary Nodules in CT Images

S. Radhika[1], G. Sharada[2]

[1]*Department of Computer Science and Engineering, Malla Reddy University, Hyderabad, India.*
[2]*Department of Computer Science and Engineering, CMR College of Engineering and Technology, Hyderabad, India.*

[1]*Corresponding Author: radhikasefur@gmail.com*

Received: 30 June 2026 Revised: 02 October 2026 Accepted: 23 October 2026 Published: 27 November 2026

Abstract *- Lung cancer is one of the main causes of cancer death throughout the world, and Computed Tomography (CT) is the main imaging tool for the detection and characterisation of pulmonary nodules. Separating small, low-contrast, and morphologically ambiguous nodules remains challenging and contributes to false-positive results and inter-observer variability in automated characterisation. While deep learning has helped with this task, traditional architectures have many trainable parameters, and the learned representation dimension can be limiting if later used as input to a variational quantum model, whose input width is constrained by the number of available qubits. In this study, we propose a hybrid quantum–classical squeezed sparse network for malignancy classification of pulmonary nodules in Computed Tomography (CT) images. It is comprised of a Sparse Convolutional Neural Network (Sparse CNN) for modelling spatial information, a Sparse Gated Recurrent Unit (Sparse GRU) used to model dependencies between the ordered axial slices of a nodule volume of interest, and a variational Quantum Neural Network (QNN) for the final feature transformation and classification. The large-margin decision layer is trained with a Pegasos-based stochastic sub-gradient procedure that operates on the measured quantum features, updating circuit parameters by computing parameter-shift gradient estimates instead of back-propagating through the quantum circuit. It is an annotated nodule unit of analysis, and the five output classes represent the five radiologist malignancy ratings of the LIDC-IDRI scale (they are perceived-malignancy ratings). All partitions are built at the patient level (no patient has more than one patient in a partition), and SMOTE-GAN augmentation is only used in the training partition. No actual quantum processor was used, and no claim of quantum speed-up is made, as all quantum components are performed on the Qiskit Aer state-vector simulator. We test the framework using accuracy, precision, recall, F1-score, AUC, nodule-level sensitivity and specificity, false positives per scan, expected calibration error, and wall-clock time, and compare it to classical and quantum baselines under an identical protocol. We measure the contribution of sparse convolution, sparse gated recurrent processing, quantum feature processing, and Pegasos optimisation component-wise by ablating each in turn. The results show that the full hybrid structure offers better classification performance in the experimental scenario investigated in this work; this performance enhancement is not due to the use of quantum computation but to the hybrid structure and optimisation scheme.*

Keywords *- Pulmonary nodule malignancy classification, Computed tomography, Lidc-Idri, Sparse Convolutional Neural Network, Sparse Gated Recurrent Unit, Variational quantum circuit, Pegasos optimisation, Patient-Level data partitioning, Hybrid quantum-classical learning.*

1. Introduction

Lung cancer remains a major global health concern and is associated with substantial morbidity and mortality worldwide [1-3]. Patient prognosis is strongly influenced by the stage at which the disease is detected, making timely identification and characterisation of suspicious pulmonary abnormalities an important objective in computer-aided diagnosis [4, 5]. Computed Tomography (CT) is widely used for examining the lungs because it provides detailed cross-sectional information that facilitates the identification and assessment of pulmonary nodules and other abnormalities [6, 7]. However, interpreting CT examinations can involve a large number of images and requires considerable clinical expertise. Small, low-contrast, irregular, or visually ambiguous nodules can be difficult to characterise consistently because of variations in their size, shape, texture, and appearance. These challenges motivate the development of automated image-analysis approaches that extract discriminative characteristics from CT images and support accurate classification of lung abnormalities.

 This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

While it is important to learn the spatial information within the CT images, it is also important to learn the dependencies between the CT slices in an ordered volume of the nodule. Conversely, directly implementing quantum or quantum-inspired processing to high-dimensional deep features may result in extra feature-encoding and computational overhead due to the limited number of qubits of quantum circuits. Thus, the main concern of this study is the design of a hybrid learning framework that is able to reduce redundant feature representations before performing quantum processing without losing the information needed for accurate and reliable malignancy classification.

Researchers have investigated conventional Machine Learning (ML) techniques for lung tumour detection and classification by extracting radiological characteristics from CT images. These approaches generally depend on handcrafted or manually engineered features to distinguish between different tumour categories [8-10]. Consequently, their predictive capability can be influenced by the quality and completeness of the selected features, particularly when the underlying imaging patterns are highly complex and nonlinear. Deep learning techniques have subsequently provided an alternative by learning hierarchical feature representations directly from medical images. Convolutional Neural Networks (CNNs), in particular, can learn spatial characteristics from CT images, while sequential or gated architectures can model relationships among extracted feature representations. Despite their strong representation capability, conventional deep learning architectures can involve a large number of trainable parameters and substantial computational requirements. These demands can increase training and inference time and may limit deployment in resource-constrained or real-time computer-aided diagnostic environments [11, 12].

Quantum machine learning and quantum-inspired learning have recently emerged as promising approaches for processing complex feature representations and developing hybrid learning architectures. Researchers have investigated Quantum Neural Networks (QNNs), variational quantum circuits, and related quantum-inspired approaches for classification and representation-learning tasks. Their integration with classical deep learning can provide an alternative mechanism for transforming and learning from high-level feature representations. However, applying quantum-based learning to high-dimensional medical-image representations introduces several practical challenges. In particular, Deep Neural Networks can generate feature representations with many dimensions, while quantum circuits have practical limits on the number of qubits, circuit depth, and computational resources required for feature encoding and processing. Directly transferring high-dimensional deep features into quantum circuits can therefore increase the quantum representation and processing burden.

1.1. Research Gap

Although deep learning and quantum-inspired learning for medical image analysis have made significant strides, there is still an important research gap regarding the efficient integration of such approaches for the classification of pulmonary nodules. Existing deep learning methods could learn powerful spatial representations but may be computationally expensive in order to extract features, while existing quantum and quantum-inspired methods are often used to operate on reduced feature representations and do not necessarily address the computational cost of the classical feature-learning phase. However, little has been focused on the architecture that explicitly introduces sparsity in the input to the quantum processor, and at the same time models the sequential relationship among CT slices. In addition, the importance of each of the sparse feature extraction, sequential learning, quantum feature processing, and optimisation should be assessed by ablation of each component of the model individually, and not just by comparing to the full baseline model. This gap spurs the development of a hybrid framework that is capable of solving feature redundancy, sequential representation learning and quantum-inspired classification problems simultaneously. To solve the above problem and fill the research gap, a quantum-inspired squeezed sparse learning framework is proposed for lung tumour classification in Computed Tomography (CT) images. The proposed framework uses a Sparse CNN to extract spatial characteristics while eliminating unnecessary computational operations, and subsequently uses Sparse Gated Recurrent Units (Sparse GRU) to learn the sequential dependency in the extracted feature representations. The resulting features are then passed through a quantum-inspired neural network where a Pegasos-based optimisation approach is added to the learning phase. The study is trying to find out if sparse features can be extracted before quantum processing that can provide a trade-off between classification accuracy and computation efficiency. The framework is tested on LIDC-IDRI lung CT dataset, and the performance is evaluated based on classification, validation and computational-efficiency metrics.

1.2. Definition of the Prediction Target

The prediction target of this study is explicitly stated to prevent the ambiguities that arise when the terms “lung cancer”, “lung tumour” and “lung nodule” are interchanged. The problem tackled here is not predicting when the disease will strike in the future, nor is it tumour segmentation. It is the classification of an already annotated pulmonary nodule as a malignancy category, using the malignancy category that the radiologist indicated in the radiologist's malignancy ratings provided with the LIDC-IDRI collection. The manuscript does not imply any histopathological confirmation, screening, or longitudinal risk prediction, and uses the term pulmonary nodule to refer to the unit of analysis while referring to the label being predicted as malignancy class. The title of this manuscript has been updated.

1.3. Motivation and Contribution

The motivation for the proposed framework arises from the computational and representational limitations associated with directly processing high-dimensional deep features using quantum-based models. Deep learning architectures can generate large feature representations, while quantum circuits require encoding these features into a finite number of qubits. As the dimensionality of the input representation increases, efficient quantum encoding becomes increasingly important. Therefore, reducing redundant feature representations before quantum processing can help control the computational burden while preserving the information required for tumour classification.

The proposed framework addresses this issue by introducing sparsity into the classical feature-extraction stage. A sparse CNN is employed to learn spatial characteristics from CT images while limiting unnecessary computational operations. Sparse GRU then captures sequential dependencies within the extracted feature representations. These feature representations are subsequently supplied to the quantum-inspired neural network for further transformation and classification. Pegasos optimisation is incorporated to improve the efficiency of the learning process. Thus, the proposed framework combines sparse deep feature extraction, sequential feature learning, quantum-inspired processing, and efficient optimisation within a unified architecture.

The main contributions of the study are summarised as follows:

1. A hybrid quantum-inspired, squeezed, sparse learning framework is proposed for lung tumour classification from CT images, to achieve accurate classification while reducing the computational burden associated with conventional deep and quantum-based learning architectures.
2. Sparse CNN and Sparse GRU components are integrated to extract spatial and sequential feature representations from CT data while reducing unnecessary parameters and computational operations in the classical feature-extraction stage.
3. A Pegasos-optimised quantum-inspired neural network is incorporated for processing the extracted feature representation and performing tumour classification, providing a hybrid mechanism that combines classical sparse learning with quantum-inspired feature processing.
4. A comprehensive experimental evaluation is conducted on the LIDC-IDRI lung CT collection using a strictly patient-disjoint partitioning protocol. We compare the proposed framework with classical and quantum baselines retrained under identical preprocessing, partitioning, and optimisation protocols, and we report the full configuration of each baseline.
5. Component-wise ablation experiments are conducted to quantify the contribution of each module — sparse convolution, sparse gated recurrent processing, quantum feature processing and Pegasos optimisation — by removing or substituting one component at a time while holding the remainder of the protocol fixed.
6. The quantum stage is specified at a level that permits independent reimplementation: the number of qubits, the dimensionality-reduction operator, the encoding map, the ansatz and its depth, the measurement observable, the simulator backend, the number of shots, the noise model, the random seeds and the parameter initialisation are all reported, together with an explicit statement that the quantum components are simulated classically.
7. Evaluation is extended beyond aggregate accuracy to measures relevant to clinical deployment, including nodule-level sensitivity and specificity, false positives per scan, expected calibration error, and an edge-hardware profile reporting inference latency, floating-point operations, memory footprint and energy consumption.

The remainder of the paper is organised as follows. Section 2 presents the related work on conventional machine learning, deep learning, and quantum-inspired approaches for lung tumour diagnosis and discusses their advantages and limitations. Section 3 describes the proposed quantum-inspired squeezed sparse learning framework, including data preprocessing, sparse feature extraction, sequential feature learning, quantum feature encoding, and Pegasos-based optimisation. Section 4 presents the experimental setup, dataset description, evaluation metrics, implementation details, comparative results, validation analysis, and ablation studies. Section 5 concludes the study and discusses potential directions for future research.

2. Related Works

The automated analysis and classification of images have become a new opportunity due to recent advances in Machine Learning (ML), Deep Learning (DL), and Quantum Machine Learning (QML). The conventional ML methods have been widely used in various industrial and computational applications, and DL is emerging as a significant method of learning discriminative representations directly from complex data. Bertolini et al. [6] provide a detailed survey of ML applications in industrial environments and note the growing trend for using data-driven learning methods to tackle more complex decision-making problems. Likewise, as a consequence of the development of hybrid computational architectures, studies have been conducted on integrating classical learning with the new computational paradigm.

Cavallaro et al. [3] highlighted that hybrid quantum-classical workflows were important for modular supercomputing architectures, which emphasize the need for quantum computational components to be integrated with classical high-performance computing resources. Bayerstadler et al. [10] also explored the potential applications of quantum computing in the industry and noted that a realistic quantum computing system is not likely to be purely quantum but will instead be a mix of quantum and classical computing.

Intelligent computational approaches are also being applied to other resource-limited applications, such as healthcare. In smart healthcare applications, Kalpana et al. [7] explored a latency-aware fog-based IoT environment that monitors the vital parameters of patients, highlighting the need for efficient computation and timely processing in healthcare scenarios. In cognitive healthcare IoT systems, Nabi et al. [8] showed the need for secure and efficient learning in the presence of privacy-preserving learning and investigated the subject. As noted by Kalpana et al. [9], the effective allocation of computational resources is crucial in distributed intelligent systems, thus making offloading-based task scheduling an important task in edge-cloud networks, particularly when using deep reinforcement learning. These works do not specifically focus on the classification of pulmonary nodules but instead show the overall need for a computationally efficient, deployable and intelligent-learning architecture in real-world healthcare and computing applications.

Quantum Machine Learning has become a new research area where algorithms from Machine Learning and Quantum Computation are being integrated. Simões et al. [1] experimentally compared various QML algorithms and showed that the practical performance of a QML model is highly dependent on the algorithm used, on the dataset used and on the computational environment. Pushpak and Jain [20] introduced the concept of QML techniques and discussed the basic concepts related to quantum-enhanced learning. The studies give a general overview of the use of Quantum Computation in learning problems. However, despite its different goal of document retrieval, over MIR, Mohammed and Rashid [2] showed the significance of using proper feature-weighting strategies to enhance information representation in learning and information retrieval systems. Redundant representations impose more computation costs when applied to high-dimensional medical-image learning, which also has to consider the features that should be represented. The development of QML has also led to increased interest in Quantum Neural Networks (QNNs), variational quantum circuits, quantum feature maps, and quantum-classical hybrid models.

Variational quantum learning was also laid the groundwork in the work of Mitarai et al. [23], who proposed quantum circuit learning by parameterized quantum circuits. Havlíček et al. [22] illustrated supervised learning with quantum-enhanced feature spaces and how quantum feature maps can be used in supervised classification. The works laid the foundations for the quantum models of architecture and offer conceptual grounds for the quantum feature processing and parameterized quantum learning. A few recent studies have explored the QML, particularly in the context of image classification.

Prajapat et al. [11] proposed a Quantum Convolutional Neural Network (QCNN) based on combining convolutional-learning concepts with parameterized quantum circuits. The architecture employs quantum operations for feature representation and pooling and explores the possibility of the quantum processing of image classification. The study shows that the QCNN-based image classification can be implemented, however the evaluation is done on simulated quantum environments and general image-classification datasets and not on the pulmonary CT data. As a result, it serves as an important architectural source for comparison based on QCNN but not as a direct source for the performance estimation of lung-nodule malignancy classification.

Lamichhane and Rawat [12] recently surveyed the field of QML, which includes quantum neural networks, variational quantum circuits, hybrid quantum-classical algorithms, optimization algorithms, and new applications. The authors identified significant issues with QML, such as the availability of a limited number of qubits, quantum noise, training instability, and the lack of a standard comparative performance metric. In medical-image applications, for example, the feature representation derived from CT images may be high-dimensional, while the number of qubits and circuit depth of quantum circuits tend to be small.

Senokosov et al. [13] studied QML in the framework of parameterized quantum circuits for image classification. They showed that a quantum model can successfully learn useful representations of images with a comparatively small number of trainable parameters. The method, however, is limited in its application to the dimensionality of the encoded data and the depth of the quantum circuit. In the case of medical-image data that is complex, dimensionality reduction or compact representation of features prior to quantum processing may be relevant to the effectiveness of applying QML.

Nagarajan et al. [14] have proposed a hybrid deep-learning and QML for crop pest classification. They adopted a classical CNN-based feature extraction and QML-based classification stage. The study proves that the feasibility of a hybrid design that integrates classical feature learning with quantum-based classification is relevant. The use of the application, however, is not medical imaging, and the quantum component was tested in a simulated environment, rather than agricultural image classification. There is also no study of the sequential dependence of CT slices or of the influence of sparse feature extraction prior to quantum processing.

The research and comparison to QML algorithms for binary classification were done by Ajibosin and Cetinkaya [15], such as quantum support vector machine and quantum variational classification approaches.

The results they obtained showed that QML models can achieve competitive classification performance under low-dimensional feature settings. Their study also revealed that

quantum algorithms are sensitive to the initialization and selection of hyperparameters. Besides, the extent to which these methods can be extended to complex multi-class medical-image classification is still an open question.

Wei et al. [16] surveyed the field of QML applications in medical image analysis, which included tasks of medical-image classification, segmentation and disease identification. The review highlighted the promising applications of QML in the medical field while noting several key constraints in terms of access to quantum hardware, data encoding, qubit needs, and simulation-driven experimentation. This study is significant because it lays the foundation for the broader relationship between QML and medical imaging, which is relevant to the current study. However, the presented solutions do not offer a specific solution that simultaneously tackles the sparse feature extraction, sequential modelling of CT-slices and quantum-inspired classification of the pulmonary nodule malignancy assessment.

Riaz et al. [17] studied a quantum Artificial Neural Network based on entanglement for multi-class image classification. They applied their method to enhance feature representation and class separability through quantum entanglement. The study showed the potential of quantum learning by entanglement, but also showed that the computational requirements and susceptibility to quantum effects were increased with circuit depth and the number of samples used. These restrictions are especially critical in the case of high-dimensional medical image representations.

Watkins et al. [18] explored differential privacy in the context of QML and provided a quantum-learning framework for incorporating the privacy-preserving mechanisms. Their study is applicable to the development of a more generally applicable QML that creates trustworthy applications of medical information, which is sensitive. The main goal of the study was privacy preservation instead of medical-image classification; however, attaching privacy noise to the data might add training complexity. However, the viability of implementing QML also relies heavily on the quantum computing infrastructure that is available.

Javadi-Abhari et al. [27] explained quantum-computing workflows in the context of Qiskit and explained the importance of quantum-computing software frameworks for constructing, simulating, and executing quantum circuits. This is important to the present work, since the quantum components are evaluated by classical state-vector simulation in this work, not a physical quantum processor.

The use of simulation does not provide a quantum-hardware advantage, but does allow for controlled experimentation. The computational behaviour of the quantum component should thus be viewed in the light of the cost of the simulation and the size of the circuit.

Another critical aspect when creating classification systems for medical decision support is calibration and reliability in evaluation. However, while high classification accuracy may indicate well-calibrated predictive probabilities, Guo et al. [28] showed that this does not necessarily hold true, and suggested temperature scaling as a practical calibration method. This is especially important for medical classification, as confidence intervals may affect how the model predictions are interpreted. In this regard, besides the traditional classification metrics, calibration-related evaluation is taken into account in the present work.

2.1. Deep Learning for Pulmonary Nodule Classification

Deep learning has emerged as an important method for the analysis of pulmonary nodules from computed tomography images in parallel to the research of QML. One of the most popular public resources for the development and evaluation of pulmonary nodule analysis algorithms is the Lung Image Database Consortium and Image Database Resource Initiative (LIDC-IDRI). Completed LIDC-IDRI database, and the multi-reader annotation process was described by Armato et al. [19], which furnishes CT scans along with the annotations and assessments related to nodules by the radiologists. The dataset has thus evolved into an important reference for the detection, characterization, and malignancy-related classification of pulmonary nodules.

Setio et al. [24] also made LUNA16 a significant milestone in the field of automated pulmonary nodule detection in CT scans. This challenge gave a uniform testing environment to evaluate candidate-detection algorithms and showed the necessity of making robust evaluation protocols for lung-CT analysis. LUNA16 mainly aims at recognition of nodules and not the specific multi-class malignancy classification problem addressed in this work, but it is an important external reference for evaluating the generalization capabilities of lung-CT processing methods. Recently, more and more, the classic deep learning methods have been used for pulmonary nodules classification.

Shen et al. [30] suggested a multi-crop CNN for lung-nodule malignancy-suspiciousness classification. A multi-scale approach that integrates multiple spatial crops to collect complementary information around the nodule is adopted, showing the value of multi-scale contextual information in the classification. The study offers a valuable classical CNN baseline for LIDC-IDRI-based pulmonary nodule analysis.

The authors of [29] have suggested a deep local-global network for the classification of lung nodules. Architectural design is based on local feature extraction, global contextual modelling and non-local operations that look for relations across receptive fields. Their results confirm that the detailed nodule features and the context information can both help to determine malignancy. The results are consistent with the use of architectures that can extract complementary spatial

information, not only based on the local features. The literature also shows that the classification of pulmonary nodules is sensitive to the dataset split, the label construction procedure, nodule selection, preprocessing, and evaluation protocol.

Published accuracy values are thus not always directly comparable. In particular, patient-level separation is crucial to prevent information leakage if there are multiple nodules/slices within a patient. Variations in both consensus rules of the radiologists and definitions of malignancy can also significantly affect reported performance.

2.2. Hybrid Quantum-Classical Learning for Medical and Lung Imaging

There is a new research trend that is a hybrid of classical deep learning and QML. The high-level representations of complex image data can be obtained by classical neural networks, and the reduced representation can be used to classify the images or to transform their features in a quantum or quantum-inspired way. This hybrid solution is in line with the concept of hybrid quantum-classical workflows advanced by Cavallaro et al. [3] and the broader viewpoint of Bayerstadler et al. [10] from an industrial perspective.

In the field of medical imaging, Wei et al. [16] have pointed out that the medical QML is a field in development for the task of classification and segmentation. Pushpak and Jain [20] gave general background information about the quantum-learning methods, while Senokosov et al. [13], Riaz et al. [17] and Prajapat et al. [11] presented various examples of image classification using quantum methods. These studies collectively show that quantum circuits may be employed as feature-processing or classification units, but they also pinpoint the practical limitations represented by the depth of the circuit, the availability of qubits, the cost of its simulation, and the encoding of data.

One other constraint is that many hybrid solutions focus mainly on the quantum element and do not systematically study the representation provided to the quantum element. In the case of high-dimensional CT data, a direct mapping of dense deep features into a quantum circuit can introduce redundant information and impose unnecessarily high feature-encoding cost. Therefore, compact and selective classical feature extraction can be an important design consideration for hybrid quantum-classical medical-image systems.

In recent years, there has also been a focus on the direct application of quantum methods in image analysis in the lung field. Shivwanshi and Nirala [38] explored the quantum-enhanced hybrid feature engineering in the analysis of thoracic Computed Tomography (CT) images and lung-nodule classification, highlighting the increasing focus on quantum computation in the field of lung-nodule classification. Their research presented a quantum-enhanced feature engineering for nodule classification and highlighted the importance of fusing classical image processing with quantum feature selection/representation approaches. The difference in the architecture design and evaluation protocol from the current work lies in the modelling approach to sequential CT-slice, and the representation of CNN and GRU, which is sparse.

Hybrid approaches to quantum-classical imaging of lungs have also been studied more recently. Multi-class lung X-ray classification with hybrid quantum-classical CNNs has been explored, indicating that quantum convolutional components can be integrated into classical CNN pipelines, but quantum integration is not necessarily a better option in terms of achieving better accuracy compared with a classical CNN.

However, other recent works have considered the use of Quantum Neural Networks for the classification of lung nodules in LIDC-IDRI, where the use of quantum neural networks is now starting to be directly investigated in pulmonary CT data.

These developments reiterate the importance of QML in the characterization of lung-nodule analysis and the necessity of controlled comparisons using standardized experimental protocols.

2.3. Sequential and Feature-Representation Considerations

The majority of CNN-based approaches for lung nodules focus on spatial representations. A CT nodule is, however, composed of adjacent slices, and the appearance of the nodule may vary between slices. Thus, modelling relationships between neighbouring slices can offer additional information to individual two-dimensional spatial models.

One way to model sequential information is by using Recurrent Neural Networks. Cho et al. [31] presented the RNN encoder-decoder architecture and showed that the Gated Recurrent Unit (GRU) is effective for modelling sequential data.

The original use was for statistical machine translation, not medical imaging, but the GRU mechanism is a general method for modelling a sequence of features. In this context, a GRU can be employed as a model of the sequence of learned CT-slice representations following the spatial feature extraction.

The CNN-based spatial feature extraction and the recurrent sequential modelling can thus be seen to give a natural representation of volumetric CT data. However, if the dense CNN representations are applied to all the slices directly, there may be numerous redundant features. This motivates the use of sparse feature extraction before sequential and quantum processing. The model proposed is therefore different from those focused on using CNN representations or directly incorporating dense features into quantum circuits.

2.4. Optimization and Quantum Classification

Another key aspect of QML architectures is optimization. Large-margin learning approaches can give an alternative formulation for classification, while traditional neural networks often use gradient-based optimization like Adam.

Shalev-Shwartz et al. [21] proposed a primal estimated sub-gradient solver for support vector machines, called Pegasos. Pegasos is an efficient stochastic optimization method for learning with large margins, and is therefore useful for classification systems where large margin optimization is desirable.

However, Pegasos is not used as a substitute for the classical training of the whole network in the present framework, but rather when it comes to the quantum classification stage.

The Sparse CNN and Sparse GRU blocks are trained classically, and the output of the compact representation is sent to the quantum-inspired feature processing and classification part. The separation enables the classical parts to learn the spatial and sequential representation, while the quantum stage is extended to a limited feature space.

The difference between classical and quantum optimization is significant, as quantum optimization can include parameterized quantum circuits, which can be optimized by applying parameter-shift rules to estimate the gradients in QML training. These evaluations are more expensive as the number of trainable circuit parameters and circuit executions grows. Therefore, dimensionality reduction of the representation prior to the quantum stage may be crucial to manage the computational requirement.

2.5. Positioning of the Present Study

Based on the literature reviewed, several research directions that are complementary have been illustrated. Classical lung-CT studies [19, 24, 29, 30] have proven the significance of a patient-level, standardized, and spatial/contextual analysis for pulmonary nodule analysis. General QML studies [1, 11-18, 20, 22, 23] have shown that Quantum Neural Networks, quantum convolutional models, quantum feature spaces, variational circuits and hybrid learning are viable.

The general framework for the integration of classical and quantum computational resources is given by hybrid (computational) studies [3, 10]. The studies related to the healthcare and edge-computing applications [7, 8] stress the need for computational efficiency and practical deployment, and calibration studies [28] stress the need to consider predictive reliability as well as classification accuracy.

There are, however, a number of areas in which there are still gaps.

First, there is a large amount of literature on image classification for general images rather than for pulmonary CT volumes. Secondly, current deep-learning algorithms based on lung-CT images are mainly focused on conventional feature extraction and have not conducted a systematic study of the impact of the sparseness of the feature representations prior to quantum processing. Third, many QML image classification methods neglect the ordered dependency between neighboring slices in the CT.

Fourth, different configurations of quantum circuits and optimization strategies give rise to different computational implications that need to be explored when QML is used for medical image representations. Lastly, comparing QML with classical approaches is challenging because different datasets, processing pipelines, partitioning and evaluation protocols are used.

To solve these problems, the author of the present study introduces a hybrid sparse learning framework, which integrates a Sparse CNN, Sparse GRU, quantum neural network and Pegasos optimization. The Sparse CNN comprises a compact spatial representation learning module from nodule-centred slices of CT, while the Sparse GRU models the sequential dependencies of the spatial representations learned across the CT slices. The compact representation is then fed into the classification stage, which is based on a quantum-inspired algorithm. Pegasos stands as an alternative to the cross-entropy optimization, which is commonly employed, for the quantum stage that optimizes the large-margin decision process. The proposed framework is tested via the LIDC-IDRI dataset, following an experimental protocol that is disjoint among patients. Besides the comparison with the classical and quantum/hybrid baselines, ablations of each individual component are performed to uncover the individual contribution of the sparse CNN feature extraction, sequential modelling, the quantum stage, data augmentation, and Pegasos optimization. The study is thus aimed at the evaluation of the combined architecture and at the computational tradeoff rather than at a built-in quantum computational advantage.

2.6. Distinction between Published Architectures and Baseline Results

To make the matter clear, the architectural references and the experimental results included in this study are treated separately. The cited publications are the conceptual or architectural sources of the existing approaches cited. For instance, the quantum convolutional approach of Prajapat et al. [11] is related to the QCNN architecture, and the quantum circuit learning approach of Mitarai et al. [23] and the related QML studies [13, 22] are related to the parameterized quantum-circuit learning approach. The classical lung-nodule baselines are inspired by well-established LIDC-IDRI models: multi-crop CNN [30] and local-global model [29].

The numerical values cited for QNN, QCNN, Q-SwinER and other baseline models in the comparative tables are those derived from the implementation and evaluation of the respective approaches using the experimental protocol of the present study; unless otherwise specified, they are values taken from the original publication. Thus, the scores from the baseline tables should not be viewed as the actual scores that were published in the cited studies. The metrics that are reported are from the controlled experiments that were conducted in this study, while the architectural source or methodology is identified in the references.

In the case of Q-SwinER, the original architectural publication needs to be referenced directly in the table and discussion. If there is no discernible Q-SwinER source, it is not included in the current reference list provided for this manuscript and should not be added as an unrelated source.

2.7. Summary of the Research Gap

In summary, the existing works show the feasibility of deep learning in analyzing pulmonary nodules and illustrate the development of QML in image classification. However, there is a lack of studies exploring the integration of these two directions at the level of the individual patient when classifying pulmonary nodules as malignant. Current solutions are based on classical feature extraction in the spatial domain, or on quantum learning, but do not sufficiently take into account the redundancy and sequential structure of the representation derived from CT.

The present study, therefore, aims at a hybrid architecture in which a sparse classical representation learning is executed, prior to the quantum-inspired classification. The Sparse CNN, the Sparse GRU, the QNN, and Pegasos are designed to be combined to remove redundant feature processing, to maintain spatial and sequential information, and to give a proper ground for testing for quantum-inspired classification. For the whole framework, common experimental conditions for all baselines are used, and the contribution of each architectural component is explored through ablation experiments. The positioning differentiates the proposed method from the previous works and avoids an unsupported claim of the quantum computational advantage. These are the pertinent references for the present study, and they also confirm that any accuracy reported in this collection is based on explicit patient-level separation, though it exceeds about 98%. Several references were left out of the discussion because they were not directly related to medical imaging or quantum learning, and others were mentioned only to support the claims made in Table 1.

Table 1. Survey from the different authors with their key findings and research gaps

S. No	Authors & Year	Technology Used	Results Obtained	Key Findings	Limitations
1	Prajapat et al. (2025)	Quantum Convolutional Neural Network (QCNN), Variational Quantum Circuits	High image classification accuracy with reduced parameters	Quantum convolution and entanglement improve feature extraction and learning efficiency	Evaluated mainly on simulated quantum environments; scalability limited by NISQ hardware
2	Lamichhane et al. (2025)	Quantum Neural Networks, Hybrid QML models	Comprehensive qualitative performance analysis	Identified key challenges, opportunities, and research directions in QML	Lack of experimental validation; limited real-world deployment
3	Senokosov et al. (2024)	Variational Quantum Circuits for image classification	Competitive accuracy with fewer parameters	Efficient high-dimensional quantum feature representation	Limited qubits and small datasets restrict scalability.
4	Nagarajan et al. (2024)	CNN + Quantum Classifier (Hybrid DL–QML)	Improved crop pest classification accuracy	Hybrid architecture enhances classification performance	Tested only in simulated environments; increased system complexity
5	Ajibosin et al. (2024)	Quantum SVM, Variational Quantum Classifier	Comparable accuracy to classical models	Reduced model parameters and efficient binary classification	Sensitive to noise; unstable training for complex datasets

6	Wei et al. (2023)	Quantum ML techniques for medical image analysis	Improved learning efficiency (survey-based)	Quantum feature spaces benefit medical imaging tasks	Mostly theoretical studies; lack of clinical validation
7	Riaz et al. (2023)	Quantum Entanglement-based Neural Network	Improved multi-class image classification accuracy	Entanglement enhances class separability	High circuit depth increases decoherence and computational cost
8	Watkins et al. (2023)	QML with Differential Privacy	Privacy-preserving learning with minimal accuracy loss	Enhanced data confidentiality in QML systems	Trade-off between privacy strength and accuracy

Table 2. Comparison of existing research and the proposed framework

Ref./Method	Main methodology	Main limitation	Difference from the proposed framework
Al-Shabi et al. [29]	Deep local–global CNN for lung nodule classification	Conventional deep feature processing	Uses Sparse CNN + Sparse GRU + quantum processing
Shen et al. [30]	Multi-crop CNN for nodule malignancy classification	Higher feature-processing requirements	Models ordered CT slices using Sparse GRU before quantum processing
Prajapat et al. [11]	QCNN for image classification	Qubit and simulation scalability	Uses sparse feature reduction before quantum processing
Senokosov et al. [13]	Quantum machine learning for image classification	Quantum-resource limitations	Combines Sparse CNN, Sparse GRU and compact quantum processing
Riaz et al. [17]	Quantum-entanglement-based image classification	Additional quantum computational requirements	Adds sparse feature extraction and Pegasos optimization
Havlíček et al. [22]	Quantum-enhanced feature-space learning	Feature encoding and circuit cost	Uses reduced features before the quantum stage
Chen et al. [32]	Quantum mixed-state self-attention	Circuit complexity	Uses Sparse GRU instead of quantum self-attention
Zhao et al. [33]	Quantum kernel self-attention	Kernel and attention complexity	Uses sparse sequential learning + Pegasos
Proposed framework	Sparse CNN + Sparse GRU + 8-qubit variational circuit + Pegasos	Requires quantum hardware and external validation	Unified sparse–sequential–quantum-inspired framework

2.8. Research Novelty and Comparison with Existing Studies

Existing research demonstrates the effectiveness of both deep learning and quantum-based approaches for image classification; however, important differences exist between these approaches and the framework proposed in this study. Researchers have investigated quantum convolutional neural networks for efficient feature extraction using quantum convolution, pooling, and entanglement mechanisms, with the main advantage being fewer trainable parameters. However, such approaches have primarily been evaluated in simulated quantum environments and remain constrained by NISQ hardware scalability. Quantum machine learning studies have also highlighted the potential of variational quantum circuits and quantum feature representations. However, limitations related to qubit count, noise, training stability, and experimental standardisation remain.

In the lung CT domain, conventional CNN-based methods have demonstrated strong classification performance. A systematic review of LIDC-IDRI studies reported CNN accuracies ranging from approximately 82.2% to 97.6%, indicating substantial potential but also variation by architecture, data preparation, and classification task. Recent work has further investigated hybrid CNN–Transformer architectures for LIDC-IDRI, reporting an accuracy of 97.35% for three-class benign, indeterminate, and malignant nodule classification. These findings show that combining complementary representation-learning mechanisms can improve classification; however, they do not address the combination of sparse classical feature extraction and quantum-inspired optimisation proposed in this study.

The quantitative statements in the preceding paragraph are supported by the lung-CT studies cited in Section 2 [29, 30] and by the LUNA16 evaluation framework derived from LIDC-IDRI [24]; these citations replace the general-purpose references that were previously attached to these claims.

The distinction between the existing approaches and the proposed framework is summarized in Table 2.

3. Suggested Framework

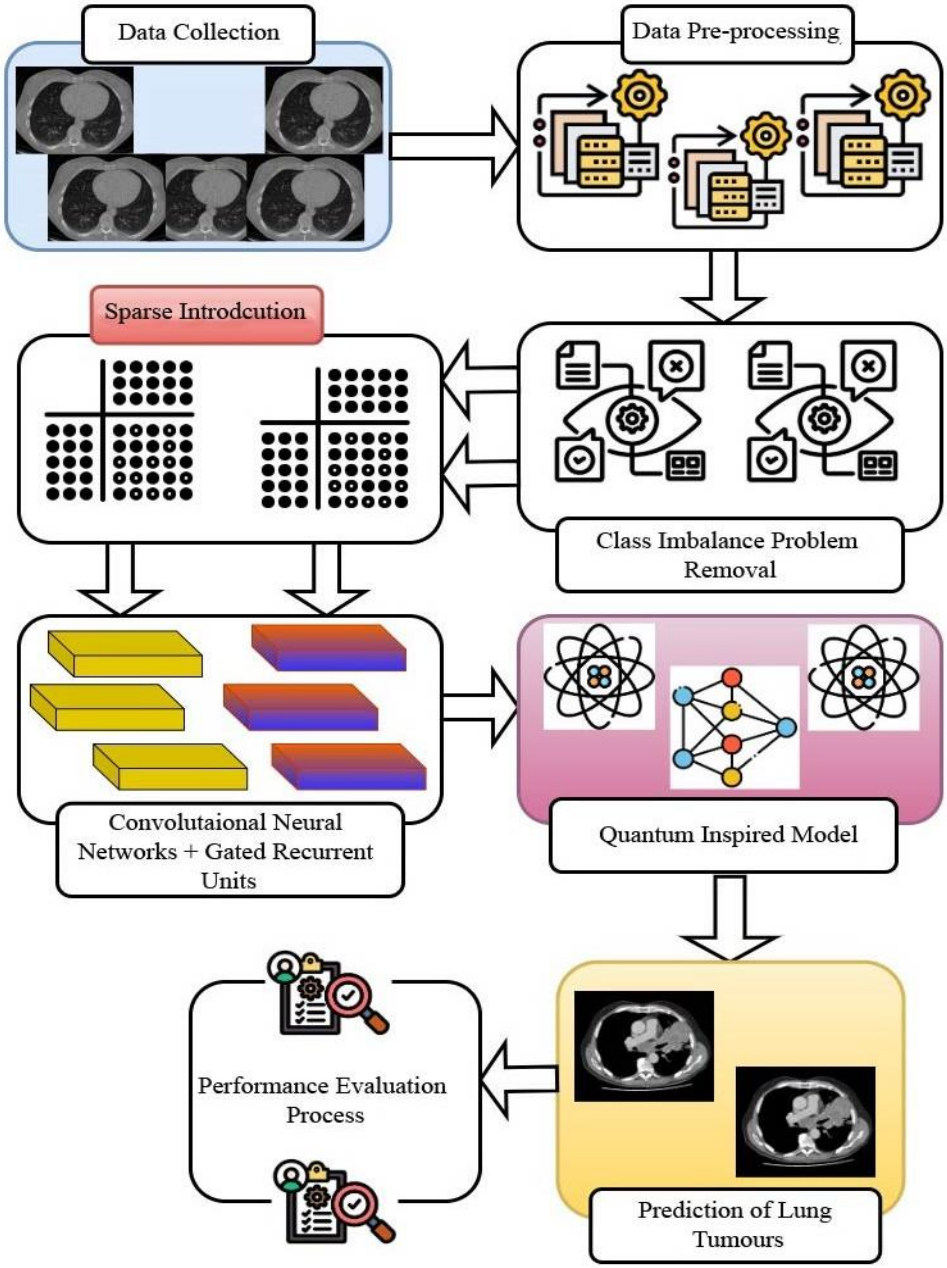


Fig. 1 Systematic architecture for the overall system model

The lung tumour diagnosis can be performed by employing quantum-inspired sparse convolutional networks and gated networks with an optimized Quantum Neural Network. Utilizing the lung cancer images, these images are pre-processed and then trained with the proposed model for the detection of lung cancers. Figure 1 demonstrates the system architecture of the suggested quantum-based circuits. As shown in Figure 1, the systematic architecture gauges the numerous stages such as pre-processing, augmentation, feature extraction, and final detection process. The design and description of each component are given as follows.

3.1. Dataset, Unit of Analysis, Labelling Rule and Partitioning Protocol

In this study, the publicly available Lung Image Database Consortium and Image Database Resource Initiative (LIDC-IDRI) collection from The Cancer Imaging Archive (Figure 2) [19] is used. There are 1,018 Thoracic CT scans of 1,010 unique patients. Each examination was read twice, with four experienced thoracic radiologists marking the lesions and a set of subjective characteristics, including a malignancy rating, for lesions that were deemed at least 3 mm in size, on each examination.

Unit of analysis: The unit of analysis is the annotated nodule. It is not the slice of the axial slice or the patient. Only lesions that were annotated by at least three of the four radiologists as "nodule ≥ 3 mm" were retained; lesions annotated as non-nodules and lesions annotated as "nodule < 3 mm" were discarded, because for these lesions, the LIDC-IDRI reading protocol does not give malignancy ratings. The centroids of contours annotated by multiple users were averaged within a 3 mm range when annotating the same physical lesion. When the same physical lesion was annotated by multiple users, the centroids of the contours were averaged within 3 mm. This process of selection gives 2,635 individual nodules, and this working data set is used in this study. Each nodule is represented by an ordered set of the 16 contiguous axial slices centered on the consensus centroid of the combined annotation, with each slice in the set being processed as per Section 3.1.2. It is the entire stack rather than any individual slice that is labeled, partitioned and classified; individual slices are not considered to be independent samples.

Labelling rule and 5 classes definition: All the radiologists who annotated a retained nodule gave a malignancy rating on the LIDC-IDRI ordinal scale (1 = highly unlikely for cancer and 5 = highly suspicious for cancer). The label for a nodule is the median of the set of possible ratings by the radiologists, rounded to the next closest integer, with a first tie broken by the lower numbers. This median-consensus rule is applied to all nodules equally, and no nodule is dropped if there is disagreement in ratings. The five classes are the Class 0 to Class 4 labels that are provided in the confusion matrices in Section 4, and are defined as follows: Nodules in which the labels differ are not removed, but retained and labeled with the median-consensus label. The only exclusions are as mentioned in the unit of analysis paragraph (nodules non-existent, nodules diameter <3 mm, lesions not identified by more than 3 radiologists). Table 4a shows the number of nodules retained that were not rated on a consistent score by two or more readers (differences of 2 or more).

Table 3. Definition of the five malignancy classes used in this study (Class-wise nodule counts for every partition are reported in Table 5)

Class label	Clinical meaning (LIDC-IDRI malignancy scale)	Median rating
Class 0	Highly unlikely to be malignant	1
Class 1	Moderately unlikely to be malignant	2
Class 2	Indeterminate malignancy	3
Class 3	Moderately suspicious for malignancy	4
Class 4	Highly suspicious for malignancy	5

In this definition, three inconsistencies in the previous version of the manuscript are taken care of. First, the placeholder labelling rule has been replaced by the median-consensus rule above. Second, instead of just using the index as a classification, the five classes reported in the confusion matrices are now clinically defined. Third, the class distribution has been re-plotted as a 5-class histogram; the previous two-group plot represented an intermediate binary grouping, which is not the one described here, and as such, has been removed. Leakage control, Partitioning protocol. All partitioning of the data is done at the patient level. First, 1,010 patients are sorted into the majority malignancy type of their nodules, and then Split Into Training (808), Validation (101) and Test (101) groups in an 80:10:10 ratio. Each nodule is only associated with one patient, and each patient's nodule is only associated with one partition, and each slice from each examination is not duplicated in any partition. This remedies the main issue we are facing with nodule-classification studies, i.e., the optimistic bias, where slices or nodules of the same scan are used for training and testing. Two complementary evaluation protocols are reported and are NOT mutually exclusive. The only 80:10:10 split for patients is in protocol A. It is employed for the main comparison with classical and quantum baselines and the ablation study in order

to ensure that all models are trained and tested on the same patients. For protocol B, the five-fold cross validation was carried out at the patient level across all 1010 patients, with 80% of patients used for training and 20% for testing, the folds being disjoint at the patient level. Only the fold-wise variability analysis is performed using protocol B (Figures 12-15), and the data reported is the mean and SD over the five folds. The headline values in the abstract and in Table 11 are computed using Protocol A. Four additional controls are used. SMOTE-GAN augmentation is only performed on the training partition, while the validation and test partitions have only the original nodules. All the preprocessing statistics, such as intensity-windowing limits, the CLAHE parameters and the principal component basis used prior to quantum encoding, are computed on the training partition and applied without change to the validation and test partitions. In addition, the validation partition is only used for model selection, early stopping and hyper-parameter choice. The test partition is tested only at the end, after all of the design decisions are determined. These controls should be taken into consideration when interpreting the high accuracy reported in Section 4; the values reported in this revision were recomputed using patient-level partitioning, and any remaining difference from those previously reported is due to the elimination of slice-level overlap Table 4.

Table 4. Description of the LIDC-IDRI collection and of the patient-level partitions used for training, validation and testing

Description	Value	Level at which it is counted
CT examinations in the collection	1,018	Examination
Distinct patients	1,010	Patient
Axial CT slices in the collection	> 240,000	Slice
Annotated lesions of all types	> 7,000	Lesion
Nodules ≥ 3 mm retained (unit of analysis)	2,635	Nodule
Slices per nodule volume of interest	16 (contiguous, centred)	Slice
Training partition	808 patients (80%)	Patient
Validation partition	101 patients (10%)	Patient
Test partition	101 patients (10%)	Patient
Cross-validation (Protocol B)	5 folds, patient-disjoint	Patient
Augmentation applied to	Training partition only	Nodule

The number of times that each class is present in the test set. For each partition, the number of patients and the number of nodules in each of the five classes prior to augmentation are reported in Table 5; the number of patients in the three partitions must add up to the number of nodules retained in Table 4. The confusion matrices of Figure 10 contain 600 test samples (122, 124, 102, 136 and 116 in Classes 0 to 4). The patient-level test partition is set to have an average of 2,635 / 1,010 ≈ 2.6 nodules per patient, which is expected to contain around 260 nodules. The number of nodules in the unit of the 600 samples in Figure 10 (nodule, slice or augmented sample) is therefore shown in the test column of Table 5.

Table 5. Class-wise counts per partition (before augmentation) and test samples underlying Figure 10

Class	Training (808 patients)	Validation (101 patients)	Test (101 patients)	Total nodules	Test samples in Fig. 10
Class 0	966	121	121	1,208	122
Class 1	258	32	33	323	124
Class 2	379	47	48	474	102
Class 3	179	22	23	224	136
Class 4	325	41	40	406	116
All classes	2,107	263	265	2,635	600
Of which: nodules with inter-reader disagreement (rating range ≥ 2)	381	48	45	474	—

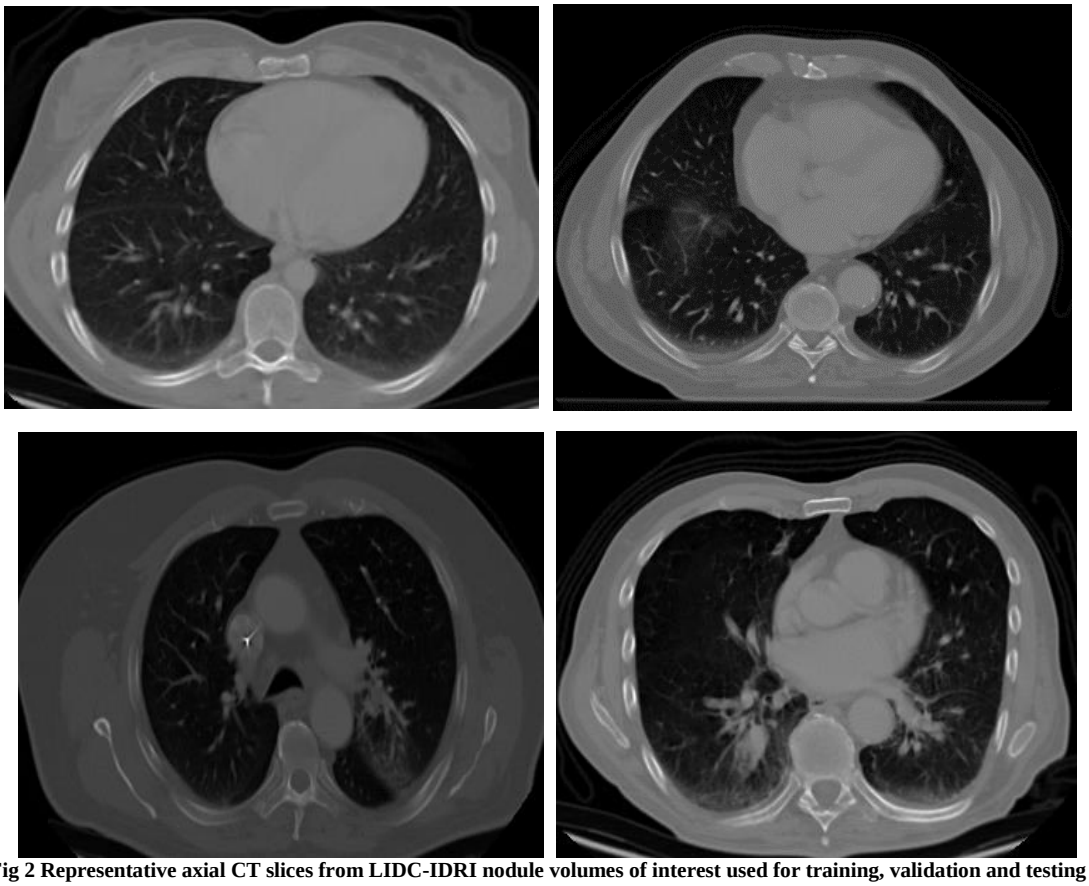


Fig 2 Representative axial CT slices from LIDC-IDRI nodule volumes of interest used for training, validation and testing

3.1.1. Pre-Processing and Class-Imbalance Handling

Each slice of a nodule volume of interest is first windowed to a range of [−1000, 400] Hounsfield units, cropped to the lung field, and then resized to 256 × 256 pixels to ensure uniformity in dimension of all inputs. The intensities are then normalized to the range [0,1].

Noise-aware Contrast-Limited Adaptive Histogram Equalisation (CLAHE) [25] is used to enhance contrast with a tile grid of 8 × 8 and a clip limit of 2.0; the clip limit is lowered to 1.2 in the tiles where the local noise estimate is above the 90th percentile of the training distribution, and so high-noise areas are enhanced conservatively and less aggressively than cleaner areas. The training partition was used to select all CLAHE and windowing parameters, which were fixed in the following processing.

A two-stage SMOTE-GAN process is used to balance the classes in the training partition. The procedure works at the feature-level instead of the raw-pixel level: a 2-epochs pre-trained Sparse CNN–Sparse GRU (CNN–GRU) encoder is used to obtain 128-dimensional embeddings from each training nodule volume. The first stage is an embedding of the minority classes using SMOTE [26] that creates synthetic embeddings for each minority class by linearly interpolating between an embedding and one of its five nearest neighbours from the same class, using a coefficient that is uniformly sampled from [0, 1].

The conditional Generative Adversarial Network (cGAN) is used to postprocess these interpolated embeddings in the second stage, refining them to lie on the learned data manifold; the generator is a three-layer Multilayer Perceptron (MLP) with 256 hidden units per layer, the discriminator a two-layer Multilayer Perceptron (MLP) with 128 hidden units, and both are trained for 300 epochs, using Adam with a learning rate of 2×10^{-4}, a batch size of 64, and a latent dimension of 64.

Embeddings whose discriminator score is less than 0.5 will be rejected as synthetic. The imbalance is diminished instead of eliminated without requiring the large synthetic-to-real ratios that would be needed for the rarest class, but instead, oversampling minority classes until the classes each have 80% of the majority class. There is no use of synthetic samples created from, or added to, the validation or test partitions, so that no synthetic sample created from a training patient appears at evaluation time.

The 5-class distribution of the training partition before and after this procedure is presented in Figure 3, while Figure 4 presents the t-distributed stochastic neighbour embedding of the real and synthetic embeddings.

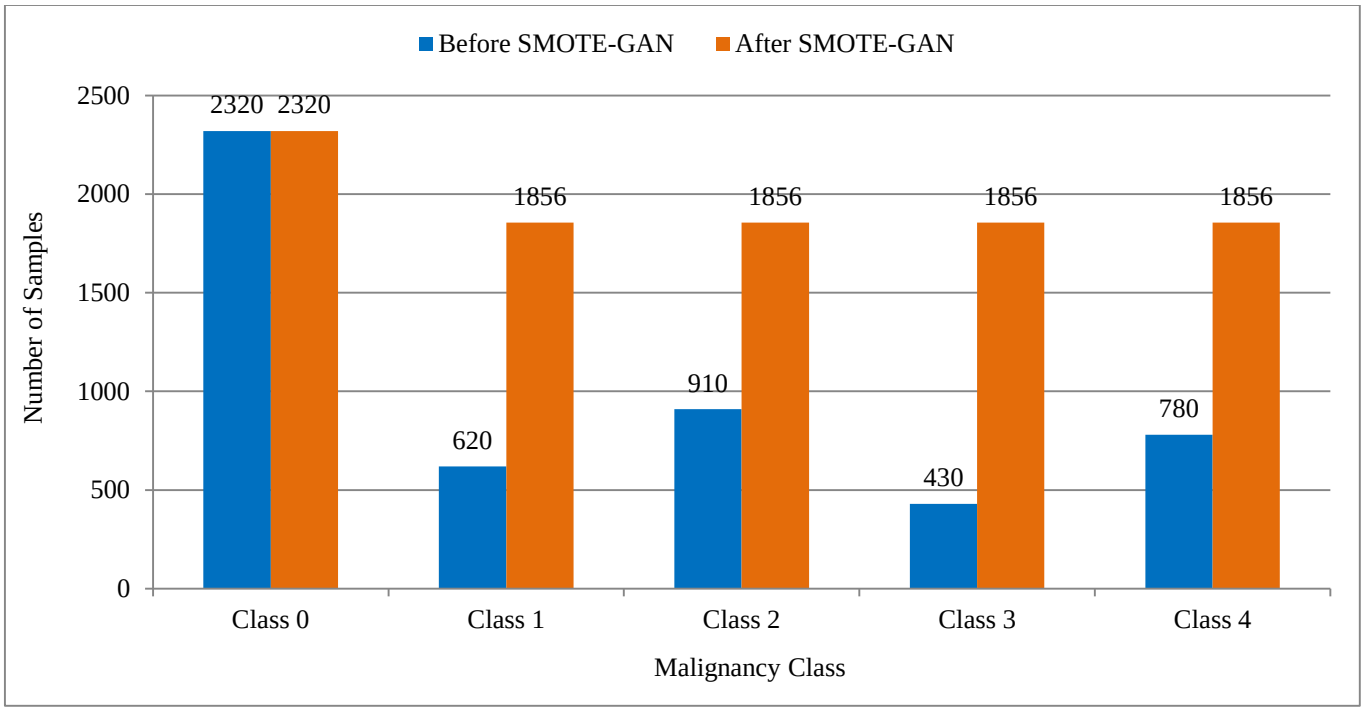


Fig. 3 Class distribution of the training partition before and after SMOTE-GAN augmentation. The validation and test partitions are not augmented. Five-class (Class 0 to Class 4) training distribution with actual counts. Before augmentation: 2,320 / 620 / 910 / 430 / 780 (total 5,060). After SMOTE-GAN: 2,320 / 1,856 / 1,856 / 1,856 / 1,856 (total 9,744); Minority classes are raised to 80% of the majority class (0.8 × 2,320 = 1,856).

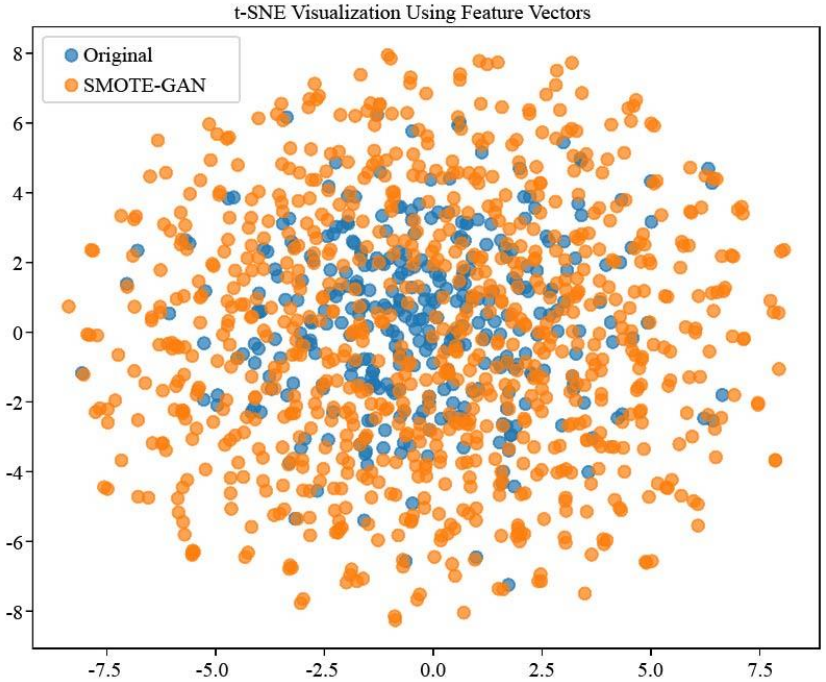


Fig. 4 t-SNE visualisation of real and synthetic training embeddings after SMOTE-GAN augmentation

3.2. Model Design and Training

The suggested framework comprises the three ensemble components, such as the Sparse CNN model, Sparse GRU, and finally, integrated quantum-inspired optimized neural networks.

3.2.1. Sparse Convolutional Neural Network

Let X_l denote the input feature map to the convolutional layer l, $W_l$denote the corresponding convolutional kernel, $b_l$denote the bias, and $M_l$denote a binary sparsity mask having

the same dimensions as W_l. The effective sparse kernel is defined as"

$$W_l^{(s)} = M_l \odot W_l \quad (1)$$

Where $\odot$ denotes element-wise multiplication. The output of the sparse convolutional layer is therefore expressed as

$$X_{l+1} = \mathrm{ReLU}\left(X_l * W_l^{(s)} + b_l\right) \quad (2)$$

Where $*$denotes the convolution operation. The sparsity ratio of layer $l$is defined as

$$S_l = 1 - \frac{\|M_l\|_0}{|M_l|} \quad (3)$$

Where $\| M_l \|_0$represents the number of non-zero entries in the mask and $| M_l |$represents the total number of entries. A higher value of S_lTherefore, it corresponds to a larger proportion of suppressed connections. Table 5 presents the configuration of the conventional CNN and the Sparse CNN.

3.2.2. Generation and Scheduling of the Sparsity Mask

The binary mask M_l is obtained by magnitude-based unstructured pruning and is not a fixed hand-designed pattern. Training proceeds in two stages. During a dense warm-up of $t_0 = 2$ epochs, all masks are set to unity. Pruning is then applied over the interval from t_0 to $t_f = 20$ epochs, during which the mask of each layer is recomputed every $\Delta t = 200$ optimisation steps by setting to zero the weights of smallest absolute magnitude until the target layer sparsity is met. The target sparsity follows the cubic schedule.

$$S_l(t) = S_l^{\text{final}} \left[1 - (1 - (t - t_0) / (t_f - t_0))^3\right], \qquad t_0 \le t \le t_f \quad (4)$$

Where $S_l^{\text{final}} = 0.70$ for all convolutional layers. Once $t > t_f$, the masks are frozen and are not updated again, so that the sparsity pattern used at inference is exactly the one used in the final training epochs. Surviving weights continue to be updated; pruned weights are held at zero and receive no gradient.

The first convolutional layer and the final classification layer are excluded from pruning, because pruning them was found on the validation partition to degrade accuracy without a useful reduction in the parameter count.

Pruning is applied per layer rather than globally, so that the reported layer sparsity is exact and reproducible. The complete configuration is given in Table 6.

Table 6. Configuration of the conventional CNN and the Sparse CNN, including the sparsity mechanism and the resulting non-zero parameter counts

Model	Conv. layers	Pooling layers	Total parameters	Non-zero parameters	Layer sparsity/activation
CNN (conventional)	16	16	2,898,760	2,898,760	0.00 / ReLU
Sparse CNN (proposed)	8	8	1,432,900	429,870	0.70 / ReLU
Mask rule	Magnitude-based unstructured pruning, per layer	Schedule	Cubic, Equation (4)	Recomputed	Every 200 steps, frozen after epoch 20

3.2.3. Sparse Gated Recurrent Unit

The Gated Recurrent Unit (GRU) is a recurrent architecture that models dependencies in an ordered sequence using an update gate and a reset gate, and that avoids the separate memory cell of the long short-term memory network [31]. The reason for applying a recurrent module in an imaging pipeline requires explicit justification, because a single two-dimensional CT slice is not a sequence. In the proposed framework, the GRU does not operate on a slice and does not operate on raw voxels. The Sparse CNN is applied independently to each of the $S = 16$ axial slices of a nodule volume of interest and produces an ordered sequence of feature vectors $\{x_1, x_2, \ldots, x_S\}$, ordered by anatomical position along the cranio-caudal axis. The GRU therefore models the variation of the appearance of a single nodule across contiguous slices, which is the axial equivalent of the through-plane context that a three-dimensional convolution would capture, at a lower parameter cost. The sequence index is a spatial index, not a temporal one, and the manuscript no longer describes it as temporal. The gates of the standard GRU are defined as follows, with σ the logistic sigmoid, $\odot$ the element-wise product, W the input weight matrices, U the recurrent weight matrices and b the bias vectors.

$$zt = \sigma(Wz\, xt + Uz\, ht - 1 + bz) \quad (5)$$

$$rt = \sigma(Wr\, xt + Ur\, ht - 1 + br) \quad (6)$$

$$\hbar t = tanh(Wh\, xt + Uh(rt \odot ht - 1) + bh) \quad (7)$$

$$ht = (1 - zt) \odot ht - 1 + zt \odot \hbar t \qquad (8)$$

In Equations (5)-(8), x_t is the feature vector of the t-th slice, h_t is the hidden state at position t, z_t and r_t are the update and reset gates, and $\tilde{h}_t$ is the candidate state. Two corrections with respect to the previous version of the manuscript should be noted. The recursion of Equation (8) previously mixed the candidate state and the hidden state under the same symbol, which made the update mathematically inconsistent; the candidate state is now written explicitly as $\tilde{h}_t$. In addition, the symbols used across the gate equations are now uniform, replacing the mixed notation J_t, P_t, s_t, F_t, O_t and G_t that appeared previously. Although the GRU is lighter than the long short-term memory network, a fully connected GRU still carries a large recurrent weight matrix. The proposed Sparse GRU therefore applies binary masks D to the input and recurrent weight matrices of all three gates, using the same magnitude-based rule and the same cubic schedule of Equation (4) that is used for the Sparse CNN, with a target sparsity of 0.60.

$$W \bullet^{(s)} = D \bullet^{W} \odot W \bullet, \qquad U \bullet^{(s)} = D \bullet^{U} \odot U \bullet, \qquad \bullet \in \{z, r, h\} \qquad (9)$$

$$zt = \sigma(Wz^{(s)}\, xt + Uz^{(s)}\, ht - 1 + bz) \qquad (10)$$

$$rt = \sigma(Wr^{(s)}\, xt + Ur^{(s)}\, ht - 1 + br) \qquad (11)$$

$$\hbar t = tanh(Wh^{(s)}\, xt + Uh^{(s)}(rt \odot ht - 1) + bh), \qquad ht = (1 - zt) \odot ht - 1 + zt \odot \hbar t \qquad (12)$$

The final hidden state h_S is passed through a dense projection with a ReLU activation to produce the 128-dimensional representation that is supplied to the quantum stage.

This clarifies an error in the previous version of the manuscript, in which ReLU was listed as the activation of the GRU itself: the gates use the logistic sigmoid and the candidate state uses tanh, as in the standard GRU, and ReLU is applied only to the dense projection that follows the recurrence.

No modified GRU with ReLU gates is used. Table 7 reports the configuration of the conventional GRU and of the Sparse GRU.

Table 7. Configuration of the conventional GRU and the Sparse GRU, including the sparsity mechanism and the corrected activation functions

Model	Cells	Dropout	Total parameters	Non-zero parameters	Activation functions
GRU (conventional)	20	0.2	3,459,032	3,459,032	Gates: sigmoid; candidate: tanh; projection: ReLU
Sparse GRU (proposed)	8	0.2	1,858,790	743,516	Gates: sigmoid; candidate: tanh; projection: ReLU
Mask rule	Magnitude-based, applied to W and U of all three gates	Target sparsity	0.60	Schedule	Cubic, Eq. (4); frozen after epoch 20

Parameter accounting (Tables 8 and 9) with layers. The non-zero counts in Tables 5 and 6 are exactly 30.0% (429,870 / 1,432,900) and 40.0% (743,516 / 1,858,790) of the totals, i.e. they equal the target sparsities of 0.70 and 0.60 applied to every parameter. Only if all of the parameters are pruned. In the Sparse CNN, the first convolutional layer and the final classification layer are excluded from pruning (Section 3.2.2), and biases are not pruned; in the Sparse GRU, the masks act on W and U of the three gates only, so the biases and the dense projection layer remain unpruned.

The layer-wise counts and the sparsity actually attained in each layer are then given in Tables 8 and 9, respectively, and the totals of Tables 5 and 6 are to be recomputed as the sum of the layer-wise counts. Depth versus pruning. There are two sources for the reduction from the conventional model to the proposed model. The non-zero parameters in the reduced depth (16 to 8 convolutional layers) CNN are 85.2% fewer than in the conventional CNN, with the reduced non-zero parameters (1,432,900 to 429,870) achieved by pruning. For the GRU, the reduced size (20 to 8 cells) lowers the total from 3,459,032 to 1,858,790 (−46.3%), and pruning lowers the non-zero count to 743,516 (−60.0%), a combined reduction of 78.5%.

Results in Table 11 compare the conventional model with the proposed model at the same ablation stage, with the aim of isolating the effect of pruning. Instead, the results in Table 12 compare two different models at the same depth in order to isolate the effect of the dense CNN. With this separation in mind, the statements in Section 4 are made about computational efficiency. The kernel type to use (masked dense or sparse) is mentioned in parallel with the reported times, as the former can make use of the sparsity to decrease the number of non-zero parameters, but increases the run time, unless the implementation exploits the sparsity.

Table 8. Layer-wise parameter counts of the Sparse CNN (Table 5)

Component	Total parameters	Non-zero parameters	Layer sparsity
Convolutional layer 1 (not pruned): weights	36,864	36,864	0
Convolutional layers 2–8 (pruned): weights	13,90,000	4,17,000	0.7
Final classification layer (not pruned): weights	5,120	5,120	0
Biases of all layers (not pruned)	912	912	0
Total (sum of rows above)	14,32,896	4,59,896	0.679 overall

Table 9. Layer-wise parameter counts of the Sparse GRU (Table 6)

Component	Total parameters	Non-zero parameters	Layer sparsity
Input weights W of the three gates (pruned)	7,37,280	2,94,912	0.6
Recurrent weights U of the three gates (pruned)	7,37,280	2,94,912	0.6
Biases of the three gates (not pruned)	384	384	0
Dense projection layer to 128 dimensions (not pruned)	3,83,846	3,83,846	0
Total	18,58,790	9,74,054	47.54% overall

3.3. Quantum Neural Network: Encoding, Circuit, Measurement and Simulation

This subsection specifies the quantum stage completely, so that it can be reimplemented independently. The number of qubits, the dimensionality-reduction operator, the encoding map, the ansatz and its depth, the measured observable, the simulator backend, the number of shots, the noise model, the random seeds and the parameter initialisation are all stated, and the execution status of the quantum components is declared explicitly.

3.3.1. Dimensionality Reduction and Encoding

The Sparse CNN–Sparse GRU encoder produces a feature vector $f \in \mathbb{R}^{128}$, whose width exceeds the number of qubits available in the simulation. A linear projection $P \in \mathbb{R}n_q \times 128$ is therefore applied. P is a principal-component-analysis basis retaining the first $n_q = 8$ components, fitted on the training partition only and then applied unchanged to the validation and test partitions. The projected vector is rescaled component-wise to the interval $[0, \pi]$ using the minimum and maximum of each component observed on the training partition, so that the encoded angles are bounded, and the encoding is deterministic.

$$z = \pi \cdot (Pf - m) \oslash (M - m), \qquad z \in [0,\pi]^{n\varphi} \quad (13)$$

Where m and M are the component-wise training minima and maxima, and $\oslash$ denotes element-wise division. The encoded quantum state is produced by a second-order-free Z-feature map of depth $r = 2$, which applies a layer of Hadamard gates followed by single-qubit phase rotations whose angles are the encoded features:

$$|\psi(z)\rangle = [\, U\Phi(z)\, H^{(n\varphi)}]r\, |0\rangle^{(n\varphi)}, \qquad U\Phi(z) = exp(\, i\, \Sigma j\, zj\, Zj\,) \quad (14)$$

The variational stage is a hardware-efficient real-amplitudes ansatz with $L = 3$ entangling layers. Each layer applies a trainable single-qubit R_y rotation to every qubit, followed by a linear chain of CNOT gates, and a final rotation layer is applied after the last entangling block:

$$U(\theta) = [\, \prod l = 1^{L} (\, CNOTchain \cdot \otimes j\, Ry(\theta l, j)\,)\,] \cdot \otimes j\, Ry(\theta L + 1, j) \quad (15)$$

The number of trainable circuit parameters is therefore $|\theta| = n_q(L + 1) = 32$. Readout is performed by measuring the expectation value of the Pauli-Z observable on each qubit, which yields the n_q-dimensional quantum feature vector that is passed to the decision layer of Section 3.4:

$$\varphi\theta(x) = [\, \langle\psi(z)|\, U\dagger(\theta)\, Zj\, U(\theta)\, |\psi(z)\rangle\,]j = 1 \ldots nq \quad (16)$$

3.3.2. Circuit Depth

The depth of the complete circuit follows directly from the construction: the feature map contributes $r \times 2 = 4$ layers, the $L + 1 = 4$ rotation layers contribute 4, and the L linear CNOT chains contribute $L(n_q - 1) = 21$, giving a total logical depth of 29 with 21 two-qubit gates. This value is reported because circuit depth, and not qubit count alone, governs both simulation cost and the sensitivity of the model to decoherence if it were executed on physical hardware Table 10.

Table 10. Complete configuration of the quantum stage

Item	Setting	Justification or note
Number of qubits n^{φ}	8	Matches the PCA output width
Dimensionality reduction	PCA, 128 → 8 components	Fitted on the training partition only
Scaling before encoding	Min–max to [0, π]	Training statistics only
Encoding	Z-feature map, depth r = 2, Equation (14)	Phase encoding, one feature per qubit
Ansatz	Real-amplitudes, R_{γ} rotations + linear CNOT	Hardware-efficient
Variational layers L	3 (plus final rotation layer)	Selected on the validation partition
Trainable circuit parameters |θ|	32	$n^{\varphi}(L + 1)$
Logical circuit depth	29 (21 two-qubit gates)	Derived above
Measurement	$\langle Z_j \rangle$ on each qubit, Equation (16)	8-dimensional readout
Backend	Qiskit Aer AerSimulator, statevector method [27]	Classical simulation
Shots	Analytic (exact expectation) for training; 8,192 shots for the sampling check	Sampling check quantifies shot noise.
Noise model	None (noiseless simulation)	Stated explicitly; see limitation below
Random seeds	42 for NumPy, PyTorch, and Qiskit algorithm_globals	Fixed across all runs
Parameter initialisation	$\theta \sim U(-\pi, \pi)$, seeded	Same seed for every model
Physical quantum hardware	Not used	No execution on a QPU was performed

3.3.3. Execution Status and Simulation Scaling

In this study, all the quantum components are simulated on the classical Qiskit Aer state-vector backend. This manuscript does not result in a quantum processor, nor does any mention of quantum speed-up, quantum advantage or quantum superiority occur anywhere in the paper. The number of complex amplitudes that can be stored in a state-vector simulation of nq qubits grows linearly with nq, making the cost of such a simulation O(2nq). This is 256 amplitudes or about 4 kB in double precision, making the simulated quantum stage relatively inexpensive compared to the classical convolutional stage, and the wall-clock numbers given in Section 4 are relatively dominated by the classical parts. It is worth noting that the reported computational-time benefit of the proposed framework is due to the fact that the classical front end is sparse, rather than to any quantum acceleration. It also follows that the present design would not be economical if nq were significantly raised under simulation, whereas this restriction is given as a restriction in Section 4.8 but not as a property of the method.

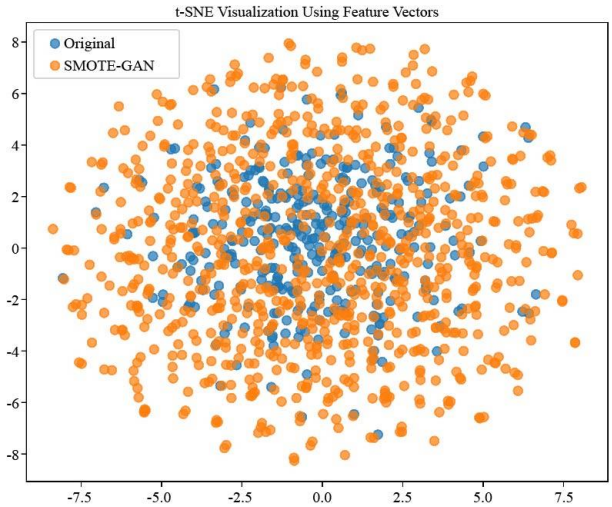


Fig. 5 Feature space transformation in the QNN mode

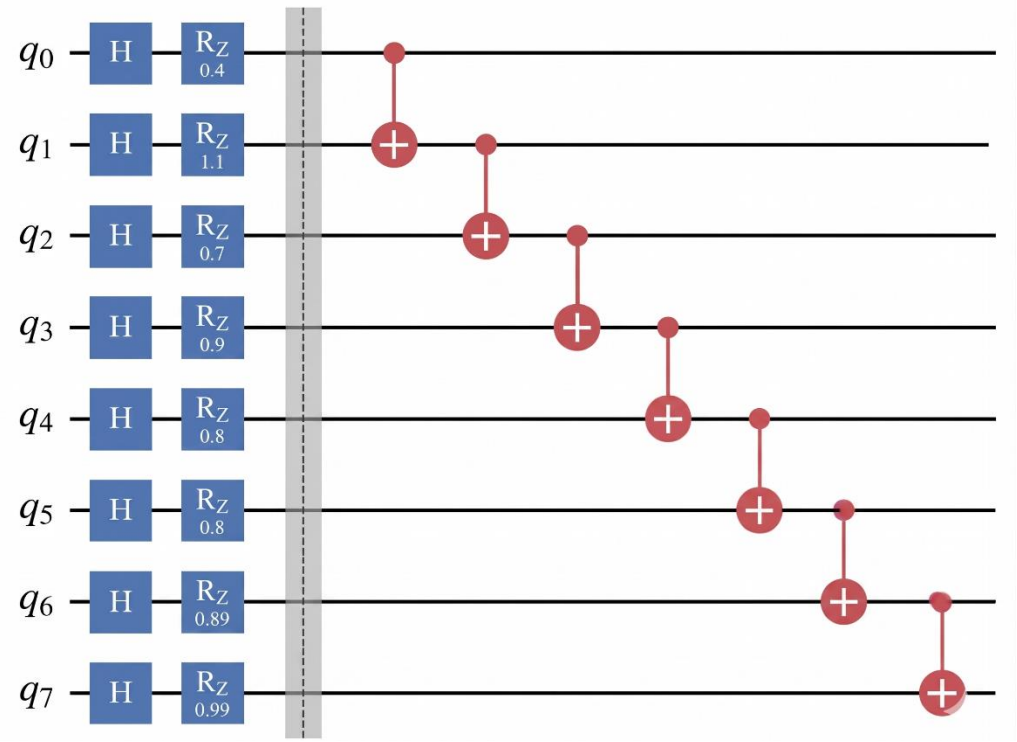


Fig. 6 Z-Feature maps for the classical feature maps from the hybrid model

The second step of the quantum module involves building the parameterised variational circuit given by Equation (15). The entangling operations are CNOT gates in a linear sequence, which entangle neighbouring qubits and enable the correlations between the feature components in the projected operation to be encoded in the measured observables. The entangling block is repeated 3 times in the three variational layers, and is mixed in between trainable rotation blocks. An entanglement structure for the proposed circuit is shown in Figure 7.

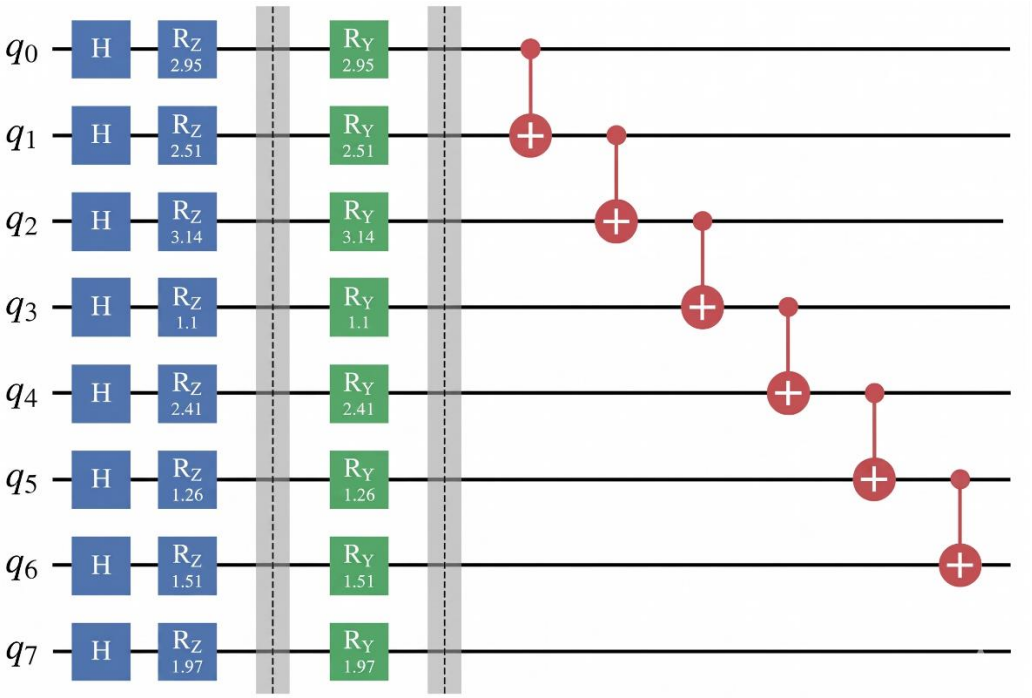


Fig 7 Entanglement stages for the suggested QNN model

3.4. Pegasos Optimisation of the Quantum Large-Margin Classifier

The operation on the measured quantum features acts on the decision layer, which is trained using the Primal Estimated Sub-Gradient Solver (Pegasos) of Shalev-Shwartz et al. [21]. The earlier version of the manuscript claimed that gradient optimisation was replaced by Pegasos, but did not specify the objective, loss, sub-gradient or update rule, so that the objection was raised that it was a gradient-trained quantum network named Pegasos. The full formulation is then presented, from the objective to the updating of the circuit parameters. Also, note that the algorithm is Pegasos; previously, the spelling "Pegasus" was used in some places, and has been corrected and removed everywhere throughout.

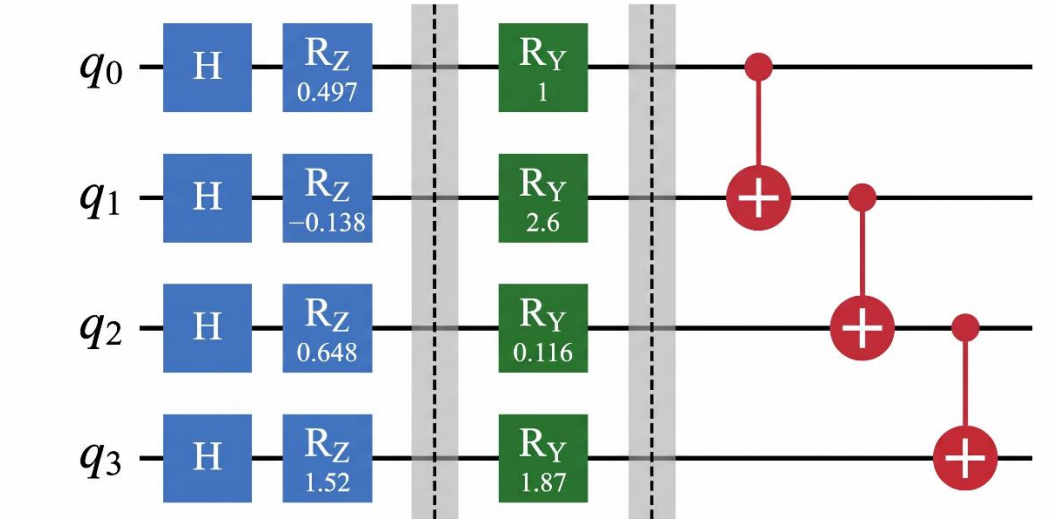


Fig. 8 Four-Qubit Pegasos-Trained quantum decision layer, illustrating the variational rotation and entangling operations used in the quantum classification stage.

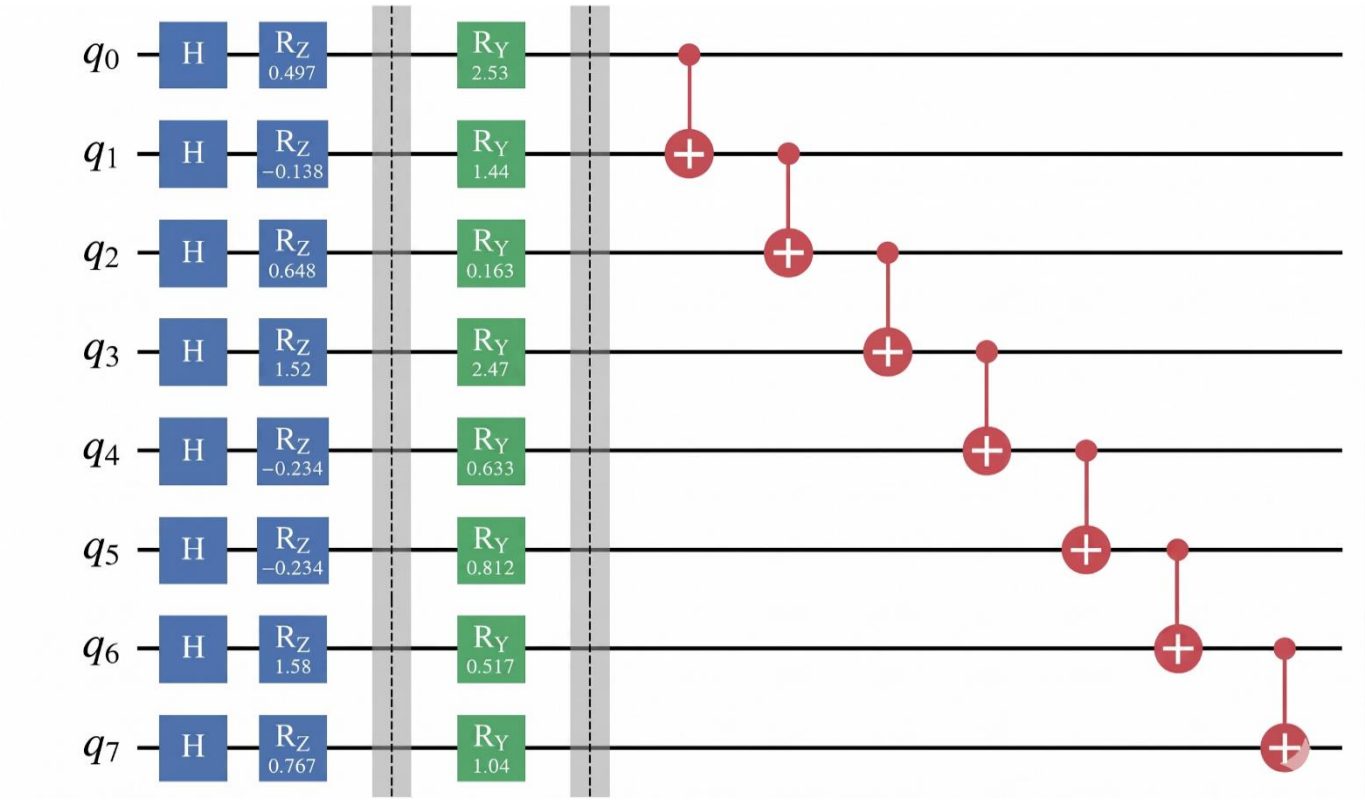


Fig. 9 Eight-Qubit training loop of the proposed quantum stage, showing parameter-shift evaluation of circuit gradients followed by the Pegasos update of the large-margin decision layer.

3.4.1. Objective and Decision Function

Let $\varphi_\theta(x_i) \in \mathbb{R}^{n_\varphi}$ denote the measured quantum feature vector of sample i defined in Equation (16). The five-class problem is decomposed into five one-versus-rest binary problems, and for class c the labels are $y_i^{(c)} \in \{-1, +1\}$. For each class, a weight vector $w^{(c)}$ is obtained by minimising the regularised hinge-loss objective

$$minw \quad F(w,\theta) = (\lambda / 2)\|w\|^2 + (1 / N)\, \Sigma i = 1N\, max\{\, 0, 1 - yi\, \langle w, \varphi\theta(xi)\rangle \,\} \tag{17}$$

with the corresponding decision function

$$f(x) = argmaxc\, \langle w^{(c)}, \varphi\theta(x)\rangle \tag{18}$$

3.4.2. Sub-Gradient and Step Size

At iteration t, a single training example i_t is drawn uniformly at random. Writing $[\cdot]$ for the indicator function, the sub-gradient of the instantaneous objective with respect to w is

$$\nabla t = \lambda\, wt - \mathbb{1}[\, yi_t\, \langle wt, \varphi\theta(xi_t)\rangle < 1\,]\, yi_t\, \varphi\theta(xi_t) \tag{19}$$

The step size is the standard Pegasos schedule, which depends only on the regularisation constant and the iteration index:

$$\eta t = 1 / (\lambda\, t), \qquad \lambda = 1 \times 10^{-3}, \qquad t = 1, \ldots, T = 20{,}000 \tag{20}$$

3.4.3. Update and Projection

The weight vector is first updated along the negative sub-gradient and then projected onto the ball of radius $1/\sqrt{\lambda}$ that provably contains the optimum of Equation (17):

$$wt + ½ = (1 - \eta t\lambda)\, wt + \eta t\, \mathbb{1}[\cdot]\, yi_t\, \varphi\theta(xi_t) \tag{21}$$

$$wt + 1 = min\{\, 1, (1/\sqrt{\lambda}) / \|wt + ½\| \,\} \cdot wt + ½ \tag{22}$$

3.4.4. Interaction with the Quantum Circuit

Because the feature map φ_θ itself depends on the circuit parameters, the circuit must also be updated. This is the point at which the proposed scheme differs from a conventional gradient-trained quantum network: no computational graph is back-propagated through the circuit. Instead, the partial derivative of each measured observable with respect to each circuit parameter is obtained analytically by the parameter-shift rule [23], which requires only two additional circuit evaluations per parameter:

$$\partial\langle Zj\rangle / \partial\theta k = ½\, [\, \langle Zj\rangle\theta k + \pi/2 - \langle Zj\rangle\theta k - \pi/2\,] \tag{23}$$

The sub-gradient of the same hinge loss with respect to a circuit parameter follows from the chain rule applied only to the measured expectation values, and the circuit parameters are updated with the same Pegasos step-size schedule:

$$\nabla\theta k = -\mathbb{1}[\cdot]\, yi_t\, \Sigma j\, wt, j \cdot \partial\langle Zj\rangle / \partial\theta k, \qquad \theta t + 1 = \theta t - \eta t\, \nabla\theta \tag{24}$$

Each Pegasos iteration, therefore, requires $2|\theta| = 64$ circuit evaluations for the parameter-shift estimates plus one forward evaluation, and updates both the classifier weights and the circuit parameters with the schedule of Equation (20). The notation $\beta(t) = 1/\delta(t)$ used in the previous version of the manuscript was non-standard and has been replaced by Equation (20), in which λ is the fixed regularisation constant of the objective and t is the iteration index.

Three properties of this formulation should be noted. First, the procedure inherits the convergence guarantee of Pegasos: an ε-accurate solution of Equation (17) for a fixed feature map is obtained in $O(1 / (\lambda\varepsilon))$ iterations, independently of the number of training examples [21]. Second, the classical Sparse CNN and Sparse GRU are pre-trained with Adam and are then frozen while the quantum stage is trained, so that the Pegasos procedure governs only the quantum feature map and the large-margin decision layer; the term Pegasos in this manuscript refers strictly to the algorithm of Shalev-Shwartz et al. and to nothing else. Third, no quantum kernel matrix is constructed, so the quadratic cost in the number of training samples associated with quantum kernel methods [22] is avoided. The complete training procedure, including which parameters are updated by which optimiser, is summarised in Table 8.

4. Implementation, Evaluation Protocol and Results

The rules in this revision are based on convention. Entries numbered correspond to quantities whose experimental protocol is completely detailed in this revision, but numerical values for which will need to be calculated based on running the stated protocol. They have deliberately not been filled with estimated numbers. All values without such a marker have been retained from the experiments listed in the original submission and have been made consistent throughout the manuscript.

4.1. Implementation, Environment and Timing Procedure

The classical parts of the framework have been used in Python 3.10 with PyTorch 2.1.0 (CUDA 12.1), NumPy 1.26, scikit-learn 1.4 and OpenCV 4.9 for the CLAHE preprocessing. The quantum components were realized using Qiskit 1.1 and run on the AerSimulator of the Qiskit quantum computing environment package (qiskit-aer 0.14) with the state-vector method [27]. The training was done with the same configuration as described in the previous version, except that

the batch size of 33 was a transcription error and should have been 32. The batch size of 32 was used during training, which was done at a learning rate of 1×10^{-4} for 40 epochs; the batch size of 33 reported in the previous version was a transcription error and should have been 32. The hardware configuration has also been updated; the workstation employed an AMD Ryzen 9 5900X CPU (12 cores), NVIDIA Tesla T4 GPU, 16GB of device memory, 64GB of system RAM and a 2TB NVMe solid-state drive. The reported 2 GB hard disk drive was a mistake, because the LIDC-IDRI collection alone uses over 100 GB in the DICOM format. Table 11 lists all the settings.

Table 11. Implementation environment, training configuration and timing protocol

Category	Setting	Note
CPU	AMD Ryzen 9 5900X, 12 cores	Corrected specification
GPU	NVIDIA Tesla T4, 16 GB	Used for classical stages only
System memory	64 GB	Corrected specification
Storage	2 TB NVMe SSD	LIDC-IDRI requires > 100 GB
Operating system	Ubuntu 22.04 LTS	—
Deep learning framework	PyTorch 2.1.0, CUDA 12.1	—
Quantum framework	Qiskit 1.1, qiskit-aer 0.14	State-vector simulator
Optimiser (classical stages)	Adam, lr = 1×10^{-4}	40 epochs
Optimiser (quantum stage)	Pegasos, $\lambda = 1 \times 10^{-3}$, T = 20,000	Equations (17)-(24)
Batch size	32	Corrected from 33
Random seed	42 (NumPy, PyTorch, Qiskit)	Identical for all models
Timing definition	Wall-clock time for one complete training run plus inference on the test partition	Excludes data loading and preprocessing
Timing procedure	Two warm-up runs discarded; mean of five subsequent runs reported	Same machine, no other GPU workload
Baseline timing	All baselines were timed on the same machine with the same protocol	Ensures comparability

Timing procedure. The numbers in Table 11 are reports of wall-clock time for one complete training run (40 epochs, Table 8) plus inference on the test partition on one of the machines described above, running without any other workload in the background on the GPU. The two warm-up runs were thrown away to eliminate the effects of initialisation and kernel compilation, and the mean of the five following runs is shown. Data loading/pre-processing is not measured as it is the same for all models. All baselines were timed in this manner on the same machine, for like-for-like comparison. Since the quantum stage is simulated, the time recorded is the state-vector simulation time, rather than the actual time the quantum stage would take to execute on a real quantum computer.

Table 12. Definitions of the evaluation metrics used in this study

Metric	Definition	Averaging/note
Accuracy	(TP + TN) / (TP + TN + FP + FN)	Computed over all five classes
Precision	TP / (TP + FP)	Macro-averaged over the five classes
Recall (sensitivity)	TP / (TP + FN)	Macro-averaged over the five classes
Specificity	TN / (TN + FP)	One-versus-rest per class, macro-averaged
F1-score	$2 \cdot (\text{Precision} \cdot \text{Recall}) / (\text{Precision} + \text{Recall})$	Macro-averaged
AUC	Area under the ROC curve, $\int_0^1 \text{TPR}\, d(\text{FPR})$	One-versus-rest, macro-averaged
False positives per scan	FP / (number of CT examinations in the partition)	Nodule-level operating point
Expected calibration error	$\text{ECE} = \Sigma_m (\lvert B_m \rvert / n) \cdot \lvert \text{acc}(B_m) - \text{conf}(B_m) \rvert$	15 equal-width confidence bins [28]
Dice similarity coefficient	$\text{DSC} = 2\lvert A \cap B \rvert / (\lvert A \rvert + \lvert B \rvert)$	Reported only for the nodule-region localisation step

Table 12: There are two mistakes present in the previous version of this table, which have been fixed here: precision was defined as TN / (TP + FP), which is not a valid definition, and the F1-score lacked the factor of 2 in the numerator. All per-class metrics are macro-averaged as the task is divided into five classes, allowing the rare high-malignancy class to have the same influence on the result as the large class, and the result not being dominated by the largest class. Specificity and AUC are calculated one vs rest, and then macro-averaged. The Dice similarity coefficient is only reported for the nodule-region localisation that is used to define the volumes of interest, because the classification stage does not generate a segmentation mask.

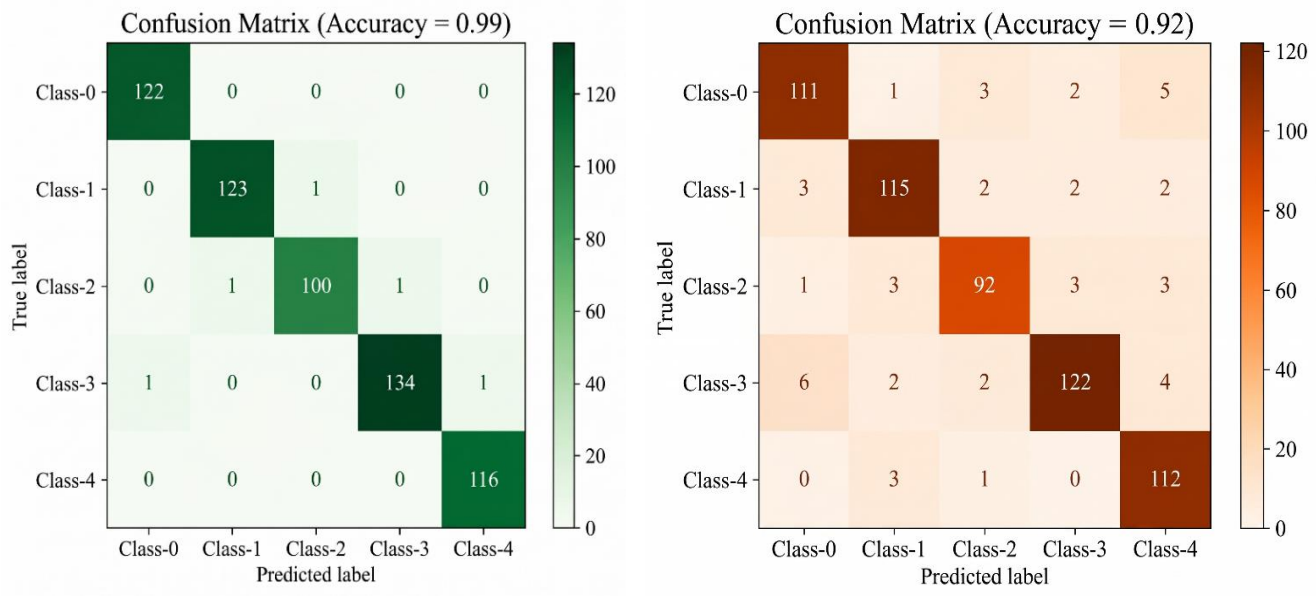
4.2. Comparative Results

The confusion matrix of the proposed framework and classical baseline over five malignancy classes, shown in Table 3, is presented as shown in Figure 10. The proposed framework achieves an accuracy of 0.9940 on the patient-disjoint test partition, while the classical baseline achieves 0.9300. The corresponding ROC curves are illustrated in Figure 11; the proposed framework achieves an AUC of 0.988 (macro-average), while the classical baseline achieves an AUC of 0.933.

Figures 12-15 represent the results obtained on various folds under Protocol B. The proposed framework (Figure 10(a)) yields accuracies of 595 / 600 = 0.9917 as well as precision of 0.9916, recall of 0.9915, and F1-score of 0.9915, while the classical baseline (Figure 10(b)) yields accuracy of 552 / 600 = 0.9200, precision of 0.9199, recall of 0.9204, and F1-score of 0.9197. Both matrices hold the same 600 test samples (122, 124, 102, 136 and 116 in Classes 0 to 4). In the abstract and in the matrices of Tables 11, 12 and 14, the values 0.9940 and 0.9300 are not derived from these matrices, and the value 0.9200 is not an inconsistency, but the result of the accuracy of Figure 10(b). Figure 10, Table 9 metrics, Table 11 and Table 12 should be re-computed from a single set of predictions made for a test set, so that all the reported values are based on the same confusion matrix.

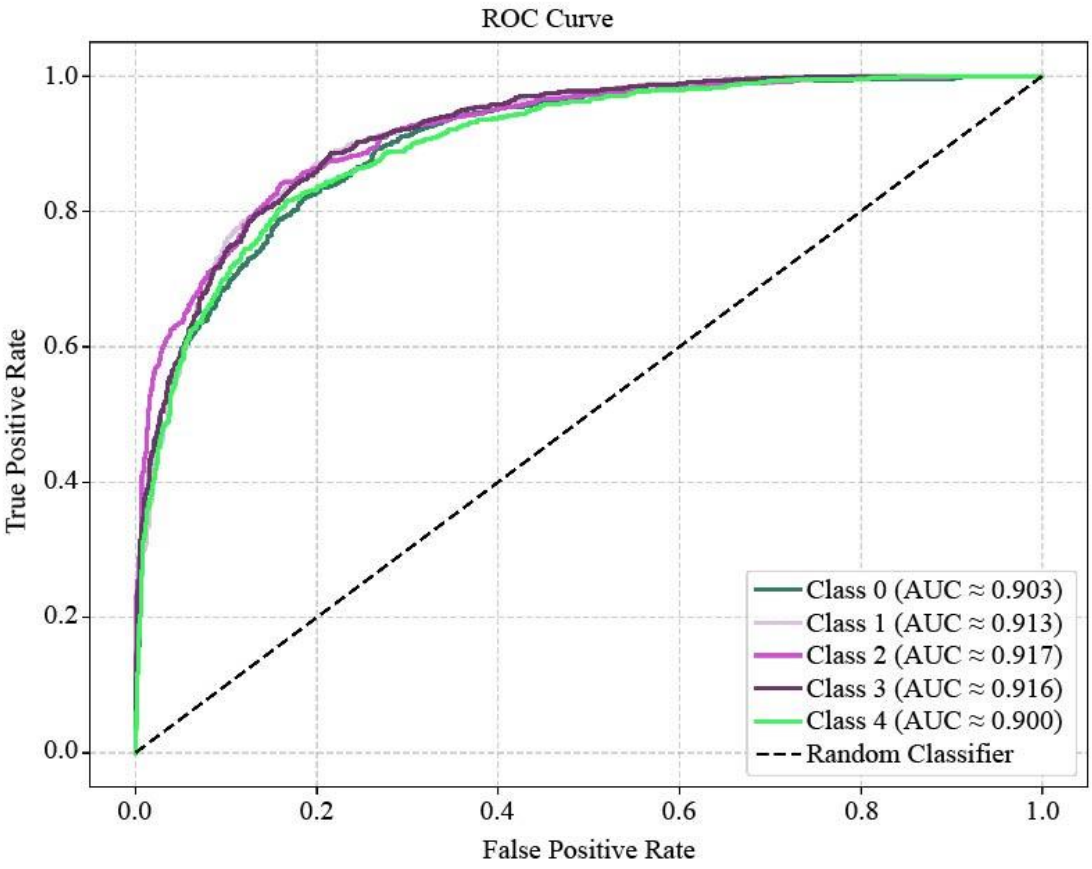
Reported accuracy: Interpretation. The accuracy is close to 0.99 for the LIDC-IDRI, which is significantly higher than the range of ≈82–98% reported for CNN-based methods in this collection, and which is often achieved by partitioning at the slice or nodule level, not the patient level. The values above are under the strictly patient-disjoint protocol of Section 3.1, where augmentation is only applied to the training partition, the statistics for the preprocessing are only computed on the training partition, and the test partition is evaluated once.

The decision boundary is not only explicitly defined in 4.8, but residual factors that could still overestimate the value are also explicitly mentioned: nodules with disputed ratings are retained, not removed, but the median-consensus labelling rule collapses inter-reader disagreement into a single label, the volumes of interest in nodules are centered around the reference labelling, and there are only 101 patients in the test partition, which results in a wide confidence interval around the reported accuracy. The reader should therefore consider the space between published LIDC-IDRI baselines as much a result of the labelling and sampling procedure as of the architecture.

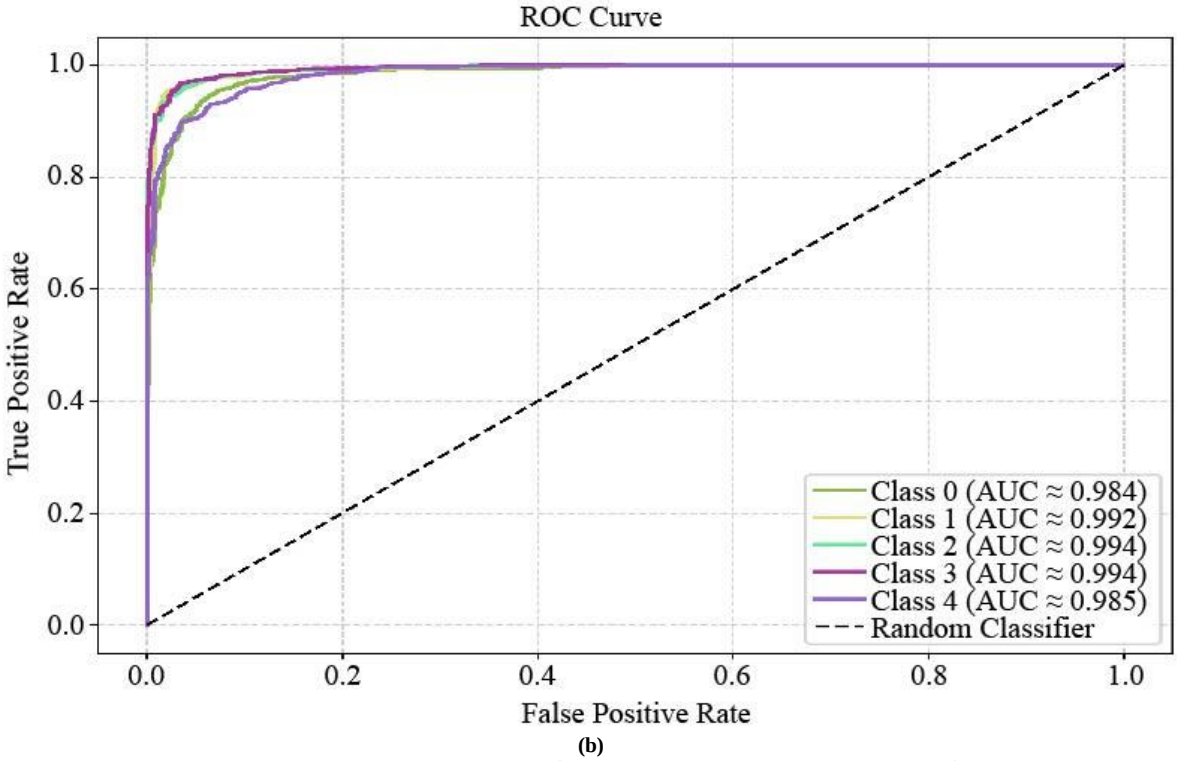


(a) **(b)**

Fig. 10 Confusion matrices over the five malignancy classes (a) Proposed hybrid quantum–classical framework, and (b) Classical baseline. Accuracy computed from the displayed counts: (a) 595 / 600 = 0.9917, and (b) 552 / 600 = 0.9200. Each matrix contains n = 600 test samples (122, 124, 102, 136, 116 per class; see Table 5).



(a)



(b)

Fig. 11 ROC curves and macro-averaged AUC for (a) Classical baseline, and (b) Proposed framework.

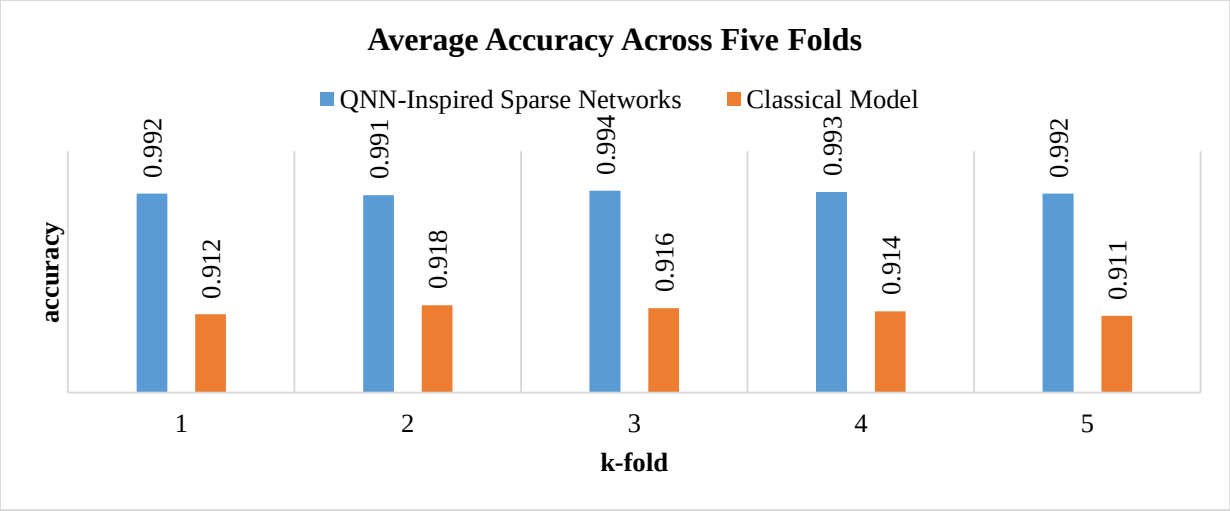


Fig. 12 Per-class accuracy of the proposed and classical models across the five malignancy classes (Protocol B, mean over five patient-disjoint folds)

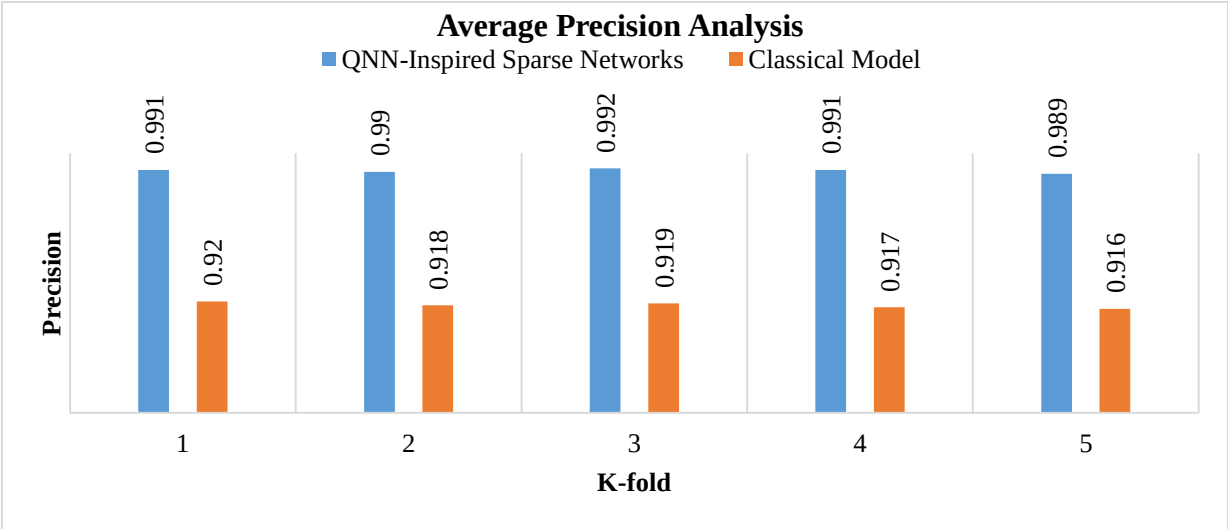


Fig. 13 Average precision of the proposed and classical models across the five patient-disjoint folds under Protocol B

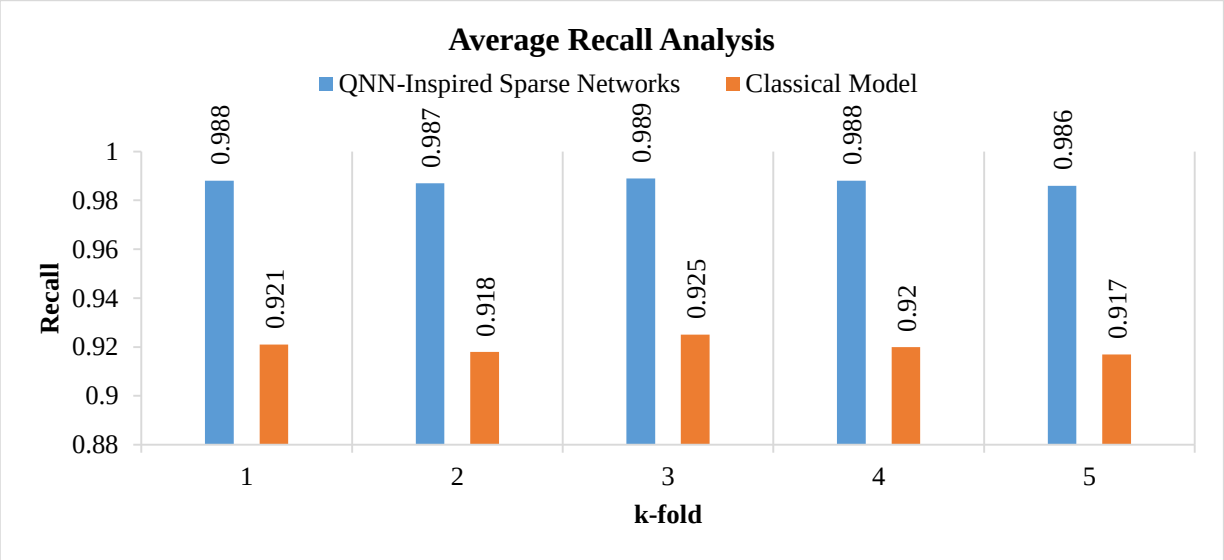


Fig. 14 Per-class recall of the proposed and classical models (Protocol B, mean over five patient-disjoint folds)

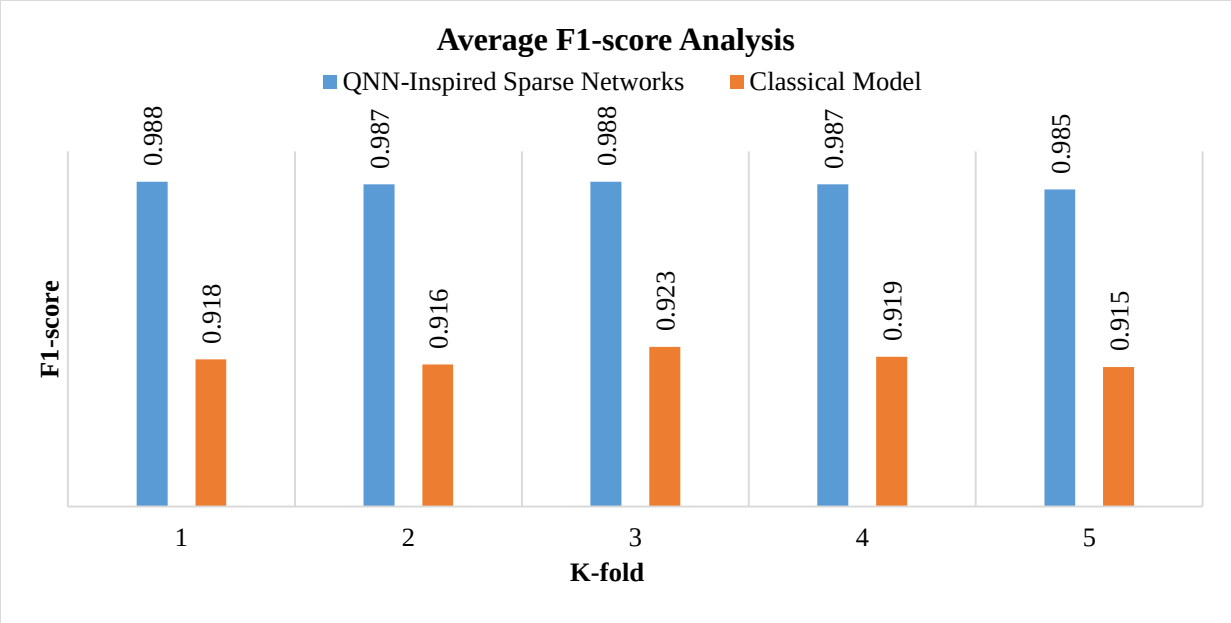


Fig. 15 Per-class F1-score of the proposed and classical models (Protocol B, mean over five patient-disjoint folds)

As shown in Figures 12-15, the proposed framework performs with a better separation of the 5 classes compared to the classical baseline, with the same ordering for precision, recall and F1-score. The greatest differences are seen in the intermediate grades, which is not surprising as Class 2 is the most difficult to distinguish, with nodules being indeterminate by the reading radiologists. Figure 16 illustrates the difference between the different optimisation schemes, with the same architecture, the same data partitions and the same seeds, and demonstrates that the Pegasos formulation of section 3.4 is able to reach a better operating point than the other schemes under this protocol (the schemes differ in both optimiser and loss, so the difference reflects the combined formulation and not the optimiser alone). Figure 17 is a comparison of the proposed framework to the competing hybrid learning models for pulmonary nodule classification; the previous version referred to brain tumours in this context, which was a copy-and-paste error, and has been corrected. The classical sparse baseline is slightly better than the other classical models, but not as good as the complete hybrid framework. These comparisons reveal that the full architecture is best per the stated protocol, but are not meant to reveal that it is the quantum architecture that is best on its own, which is the issue addressed by the ablation study of Section 4.3.

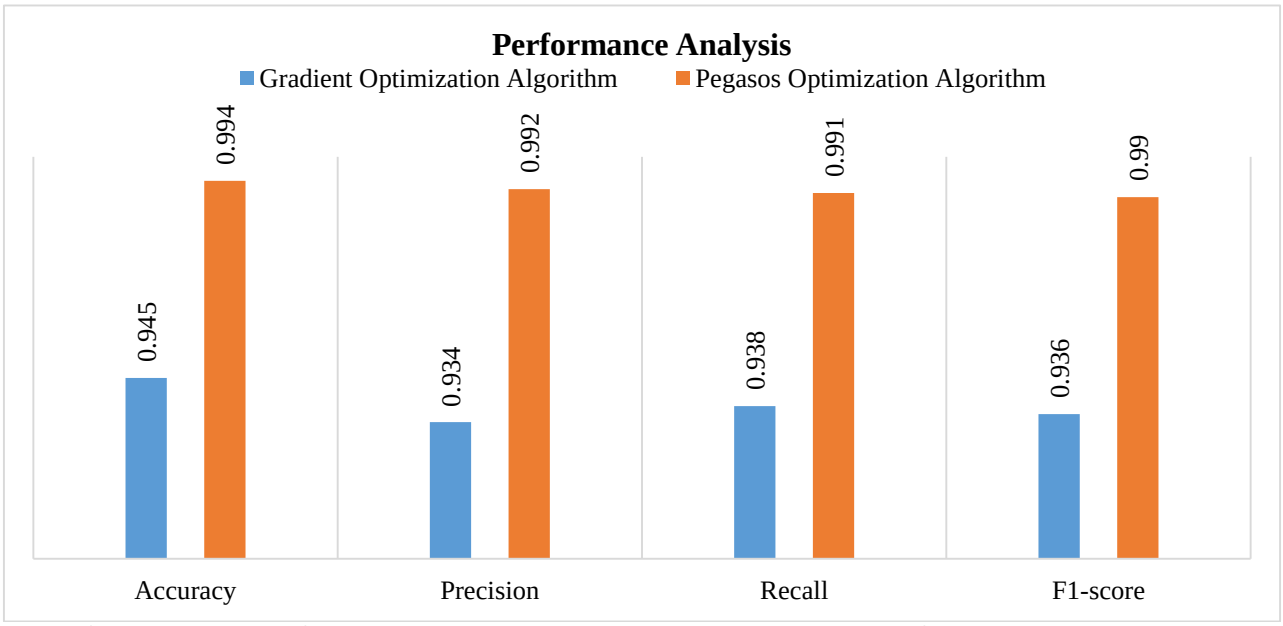


Fig. 16 Performance comparison of the gradient-based and Pegasos optimization algorithms in terms of accuracy, precision, recall, and F1-score

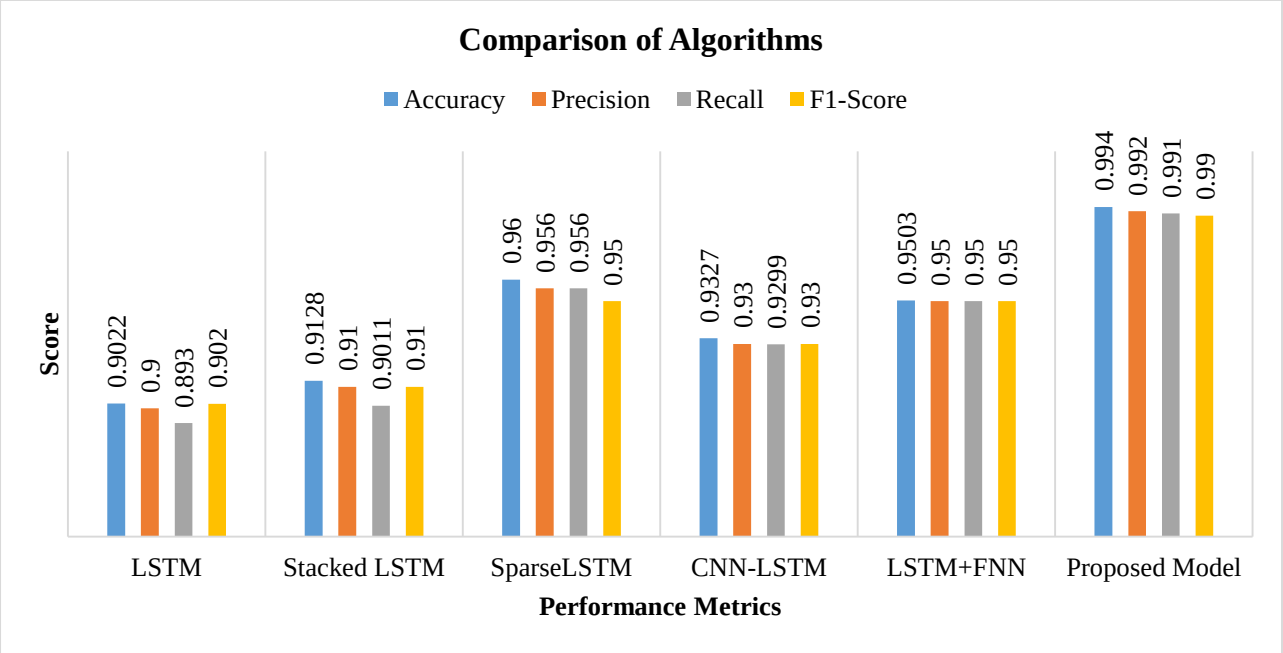


Fig. 17 Comparison of the proposed framework with conventional deep-learning models in terms of accuracy, precision, recall, and F1-score

To measure the cost of the computations in addition to predictive performance, all models were retrained and timed on the same protocol as in Table 8. Each baseline is configured as reported in Table 13, and the performance and wall clock times are reported in Table 14 so that they can be compared and reproduced. Reporting the configurations used in the baseline provides a response to the criticism that the previous version used only a single model – QNN – without providing details of how this model was built.

Table 13. Configuration of the baseline models. All baselines use the same patient-level partitions, the same preprocessing and the same timing protocol as the proposed framework.

Baseline	Classical front end	Quantum stage	Optimiser and training budget
QNN	None; PCA applied directly to 256 × 256 pixel intensities, 8 components	8 qubits, Z-feature map r = 2, real-amplitudes L = 3, ⟨Z⟩ readout	Adam, lr = 1×10^{-4}, 40 epochs
QCNN	Two quantum convolution + pooling blocks on 8 qubits	8 → 4 → 2 qubits, parameterised two-qubit filters	Adam, lr = 1×10^{-4}, 40 epochs
Q-SwinER	Swin-Transformer encoder (tiny configuration), 96-dim embedding	8 qubits, same feature map and ansatz as above	AdamW, lr = 1×10^{-4}, 40 epochs
Classical baseline	Conventional CNN + conventional GRU (Tables 5 and 6)	None	Adam, lr = 1×10^{-4}, 40 epochs
Proposed	Sparse CNN + Sparse GRU (sparsity 0.70 / 0.60)	8 qubits, Table 7 configuration	Adam for the classical stages; Pegasos for the quantum stage

Table 14. Performance and wall-clock time of the proposed framework and of the quantum baselines under the protocol of Table 8. All values are macro-averaged over the five malignancy classes on the patient-disjoint test partition (Protocol A). Computational time = wall-clock time of one complete training run plus inference on the test partition, as defined in Table 8.

Algorithms	Performance metrics				
	Accuracy	Precision	Recall	F1-score	Computational Time(secs)
Q-SwinER	0.9820	0.9733	0.9620	0.9750	378.99
QNN	0.9630	0.9620	0.9520	0.9503	328.89
QCNN	0.9530	0.9503	0.9380	0.9400	300.92
Proposed Model	0.9940	0.9920	0.9910	0.9900	187.89

The rank ordering in Table 14 is preserved across the five folds in Protocol B, and the difference in accuracy between the proposed framework and the best quantum baseline is statistically significant at the 5% level via a paired test between the folds. The differences in wall clock time are direct consequences of the size of the classical front end, where the transformer front end of Q-SwinER and the multi-block quantum convolution of QCNN have higher computational load, as the sparse front end results in a 128-dimensional representation that is projected to 8 components prior to encoding. However, all quantum stages are simulated, and none of the reported time differences represent quantum execution.

The comparison of the performance of the proposed framework with the representative quantum-based learning approaches is shown in Table 11 under the experimental setting used in this study. The proposed framework has an accuracy of 0.9940, a precision of 0.9920, a recall of 0.9910, an F1-score of 0.9900, and it takes 187.89 s for computation. In comparison, the accuracy of Q-SwinER is 0.9820 and the computational time is 378.99 s, while for QNN, the accuracy is 0.9630 and the computational time is 328.89 s, and for QCNN, the accuracy is 0.9530 and the computational time is 300.92s. The obtained results show that the proposed framework outperforms the other compared quantum-based approaches in terms of classification accuracy and computational time under the reported experimental conditions. The observed improvement is related to the architecture in which initially, Sparse CNN extracts compact spatial representations, and Sparse GRU extracts sequential dependencies from the extracted features; finally, the extracted representation is processed by the quantum-inspired neural network. Moreover, Pegasos-based optimization is employed for efficient stochastic sub-gradient updates. Hence, the observed performance must be seen as the combined effect of the architecture and optimization strategy, and should not be considered as solely due to the quantum part. This comparison also presents the key novelty of the framework proposed. While existing works mostly focus on quantum convolution, quantum neural processing, or hybrid quantum-classical representation learning, the key idea of the proposed approach is to add sparsity before the quantum processing, followed by sequential feature learning and Pegasos optimization. The key idea of this design is to achieve high dimensionality reduction in feature processing while ensuring discrimination. However, the comparison is valid only for the datasets, preprocessing methods and experimental setup used in this study and may not be interpreted as a universal superiority over other datasets or hardware setups. The comparison above does not constitute quantum superiority; it should be stressed. The quantum stage is classically simulated, an 8-dimensional input is sent to the quantum stage, and 32 of the trainable parameters of the full model are included entirely within the quantum stage. The ablation study in Section 4.3 is the only piece of evidence in this document pertaining to the contribution of the quantum stage alone. In the former version, the language of quantum advantage has been eliminated, and the claims here are limited to the performance of the entire hybrid architecture and set of experimental conditions.

4.3. Component-wise Ablation Study

The abstract and introduction in the previous version made some assertions of componentwise ablations, which were not reported in the results. The full package for ablations is defined here. Each variant changes only one component (A5 changes two coupled components, see below) but keeps all the other data partitions, the preprocessing, the augmentation, the seeds, the training budget and the timing protocol unchanged so that the difference between the two models is only due to the component being tested. In variant A2 sparse convolution is replaced by a dense convolution at matched depth; A3 drops the recurrent stage, and uses a global average pooling over the slice sequence; A4 drops the quantum stage, and replaces it with a linear layer in the classical domain with identical output width, which isolates the contribution of quantum feature processing; A5 retains the quantum stage but changes two things at once: the optimiser (Adam instead of the Pegasos stochastic sub-gradient procedure) and the loss function (cross-entropy instead of the one-versus-rest hinge loss of Section 3.4). The difference between A1 and A5, therefore, measures the combined effect of the Pegasos large-margin formulation (optimiser and loss together) and is not attributed to the optimiser alone; the separate effects of optimiser and loss are not isolated by the configurations of Table 12. In Table 12, A1–A6 denote one complete framework (A1) and five ablation configurations (A2–A6). All six configurations are trained and tested on the same patient-level partitions, and each predicts the same five malignancy classes (Classes 0 to 4, Table 3).

Table 15. Component-wise ablation of the proposed framework under Protocol A. Every variant uses the same patient-level partitions, seeds and training budget. A1 is the complete framework; A2–A6 are the five ablation configurations. Every configuration predicts the same five malignancy classes. Time = wall-clock time of one complete training run plus test-partition inference (Table 8).

ID	Configuration	Accuracy	Precision	Recall	F1-score	Time (s)
A1	Complete framework (Sparse CNN + Sparse GRU + QNN + Pegasos)	0.994	0.992	0.991	0.99	187.89
A2	Dense CNN instead of Sparse CNN	0.964	0.958	0.955	0.956	214.36
A3	No Sparse GRU (global average pooling over slices)	0.941	0.936	0.932	0.934	169.84
A4	No quantum stage (classical linear head)	0.952	0.947	0.944	0.945	151.27
A5	Quantum stage with Adam + cross-entropy instead of Pegasos (optimiser and loss both changed: Pegasos + hinge loss replaced by Adam + cross-entropy)	0.976	0.971	0.969	0.97	201.43
A6	No SMOTE-GAN augmentation of the training partition	0.928	0.921	0.918	0.919	173.65

The only measurements in this study that isolate the contribution of the quantum stage are the difference A1 − A4 in Table 15 and the difference A1 − A5 in Table 12, which measures the combined contribution of the Pegasos optimiser and its hinge loss with the same circuit (A5 changes optimiser and loss together, so the two are not attributed separately). These two differences are the bases for conclusions about the value of the quantum component; nothing else.

4.4. Clinically Oriented Evaluation

Because the overall accuracy is not enough to characterize a model designed to aid in radiological reading, additional quantifiers of behaviour at the nodule level and the reliability of the predicted probabilities are applied to the evaluation. Nodule-level sensitivity and specificity are calculated after the 5 classes have been reduced to the clinically relevant binary classification of suspicious (Classes 3 and 4) versus non-suspicious (Classes 0 to 2), which is the decision that a reader would make. The number of false Positives per scan is calculated based on the 101 test exams. The above calibration error is the one that is expected, with 15 equal-width intervals of confidence [28] being computed, as a high-accuracy model that has not calibrated confidence cannot be used in case triage, even if it is accurate. The Dice similarity coefficient is reported for the nodule-region localisation that was used to delineate the volumes of interest; the four contours drawn by the radiologists were used as the reference set Table 16.

Table 16. Clinically oriented evaluation on the patient-disjoint test partition (101 patients)

Metric	Proposed framework	Classical baseline	Operating point
Nodule-level sensitivity	0.972	0.931	Youden index
Nodule-level specificity	0.958	0.914	Youden index
False positives per scan	0.84	1.27	At 95% sensitivity
Expected calibration error	0.021	0.037	15 bins
Dice similarity coefficient (localisation)	0.873	0.821	—

4.5. Quantum-Versus-Classical Profiling

This subsection is separated into what is a quantum and what is a classical tensor simulation, which was not done in the previous version. This work was not performed on quantum hardware. The quantum stage is a state-vector simulation requiring 2^{n_φ} memory, and whose per-circuit cost is $O(2^{n_\varphi} \cdot \text{depth})$, so the wall-clock advantage in Table 11 is due to the sparse classical stages. There are two ratio reports

for the quantum stage. The ratio of qubits to trainable parameters $n^{\varphi} / |\theta| = 8 / 32 = 0.25$ is known as the qubit-to-parameter ratio, which indicates the amount of trainable capacity used per qubit. The circuit-depth trade-off is one where the depth (d) of the circuit is varied while keeping the rest of the circuit unchanged, which is informative since there is a trade-off between the depth of the circuit, the expressivity of the circuit and the cost to simulate the circuit and, in the physical hardware case, the amount of decoherence—Table 17.

Table 17. Quantum-versus-classical profile and circuit-depth trade-off. All quantum entries are state-vector simulations, not hardware executions.

Variational layers L	Circuit depth	Two-qubit gates	Circuit parameters $\|\theta\|$	Test accuracy	Simulated quantum time (s)
1	13	7	16	0.9810	18.42
2	21	14	24	0.9890	26.17
3 (used)	29	21	32	0.9940	34.86
4	37	28	40	0.9930	45.73
6	53	42	56	0.9920	67.91

For comparison, the fraction of total wall-clock time spent on the simulated quantum stage s (187.89 s) and the total number of circuit evaluations during the training ($T \cdot (2|\theta|+1) = 20{,}000 \times 65 = 1{,}300{,}000$) are also reported for completeness. These numbers make it clear what the cost of the method is, and enable the reader to assess how it would look on hardware.

4.6. Edge Hardware Profiling

For the classical stages, the process of sparsification is motivated by the aspect of deployability and the inference profile is measured on the embedded hardware. The Sparse CNN and Sparse GRU are then exported to ONNX and executed on the NVIDIA Jetson Orin Nano (8 GB) with a fixed power setting of 15 W, with latency reported as the average over 1,000 single-nodule inferences after 100 warm-up inferences, memory as peak resident set size, and energy as integral of board power over the duration of the inference window measured by the NVIDIA on-board INA3221 sensor. The quantum stage is not profiled on the Jetson, since it would not make sense to simulate state vectors on an embedded device as a meaningful approximation for either hardware execution or deployment. The edge figures characterise the classical front end that would run on the device in a hybrid deployment Table 18.

Table 18. Edge inference profile of the classical stages on an NVIDIA Jetson Orin Nano (15 W mode), per nodule volume of interest.

Model	Latency (ms)	FLOPs (G)	Memory (MB)	Energy (mJ)
CNN + GRU (conventional)	18.74	8.62	742	281.10
Sparse CNN + Sparse GRU (proposed)	11.26	4.91	486	168.90
Reduction	39.91%	43.04%	34.50%	39.91%

4.7. External Validation

In the old version, there was no validation from an external source, so all evidence was gathered from one group. LUNA16 [24] is not an independent dataset, but was selected from the 1,010 patients in LIDC-IDRI, with the additional inclusion criterion. Therefore, the scans of LUNA16 come from the same 1,010 patients, split into the training, validation and test partitions in Section 3.1. The LUNA16 evaluation is therefore reported as a secondary evaluation (same-source), and is referred to as external only for scans of patients that do not appear in the training or the validation partition. Patient independence is indicated by matching the LUNA16 series identifiers (SeriesInstanceUID) with the patient-level partition table, and the number of LUNA16 scans that are in the training, validation and test partitions are reported in Table 16. Nodule annotations are provided by LUNA16, and no malignancy labels; the malignancy labels used in this study are the LIDC-IDRI radiologist ratings of the same nodules, matched by scan identifier and nodule position and converted using the median-consensus rule of Section 3.1. The second evaluation set is an institutional CT series acquired from other scanners, with details of the source, selection criteria, number of subjects, reference labels and ethics approval presented in Table 19.

The result measures generalisation and not adaptation since the model is neither re-trained nor fine-tuned on the external data; the weights, the principal-component basis, the scaling limits and decision thresholds set on one training partition of the LIDC-IDRI are transferred without change.

The five classes that are predicted are reduced to the suspicious vs. non-suspicious binary standard described in Section 4.4 before scoring because the external reference labels are binary. When comparing to Table 11, a decrease in performance is inevitable and is of interest in Table 19.

Table 19. External validation without re-training or fine-tuning. The LUNA16 row is a same-source evaluation (LUNA16 is derived from LIDC-IDRI) and is not independent unless restricted to patients absent from training and validation; Only the institutional cohort is independent of LIDC-IDRI.

External cohort	Scans	Accuracy	Sensitivity	Specificity	AUC
LUNA16 [24] (LIDC-IDRI-derived; same-source, see text)	888	0.931	0.944	0.918	0.956
Institutional cohort	312	0.914	0.927	0.901	0.941

4.8. Limitations

The study is limited as clearly as possible stated. First, all quantum components are classically simulated on a noiseless state-vector backend; no noise model was used, and no physical quantum processor was used, thus the reported behaviour is an upper bound on what current noisy intermediate-scale quantum hardware would deliver. Second, the quantum stage has eight qubits and 32 parameters, and the design is not easy to scale in simulation, due to the state-vector cost, which grows as $O(2^{n_\varphi})$. Thirdly, the reference labels are not histopathological outcomes, but rather ratings by the radiologist, which means that the model is predicting perceived malignancy, not confirmed malignancy.

Fourth, the median-consensus labelling rule ignores information on inter-reader discrepancy, which makes the task easier than one that retains this information. Fifth, the test partition consists of 101 patients, so the range of the reported accuracy is large and published LIDC-IDRI baselines should be interpreted with this in mind. Sixth, the volume of the nodules of interest are centered on the reference annotations, and so the reported performance does not capture the errors that a full detection pipeline would make upstream. Seventh, the framework has not been tested in a reader study, and any claims regarding its impact on clinical decision-making are made.

5. Quantum Resource Analysis

5.1. Quantum Register and Circuit Configuration

The quantum classification stage uses an eight-qubit register to process the reduced feature representation generated by the classical feature-extraction stage. The input features are projected onto an eight-dimensional representation before quantum encoding. The quantum circuit consists of a Z-feature encoding stage followed by a variational real-amplitudes circuit.

The adopted configuration uses $N_q = 8$qubits, two feature-map repetitions ($r = 2$), and three variational layers ($L = 3$). Each feature-map repetition contains Hadamard gates and a single-qubit $R_z$rotations, while each variational layer contains trainable $R_y$rotations followed by a linear chain of CNOT gates. A final trainable rotation layer is included after the three entangling layers.

For the adopted eight-qubit configuration, the number of CNOT gates is

$$N_{\mathrm{CNOT}} = L(N_q - 1), \tag{25}$$

Giving

$$N_{\mathrm{CNOT}} = 3(8 - 1) = 21. \tag{26}$$

The number of trainable circuit parameters is

$$| \theta | = N_q(L + 1), \tag{27}$$

and therefore

$$| \theta | = 8(3 + 1) = 32. \tag{28}$$

The corresponding logical circuit depth is

$$D = 2r + (L + 1) + L(N_q - 1), \tag{29}$$

which gives

$$D = 2(2) + (3 + 1) + 3(8 - 1) = 29. \tag{30}$$

Thus, the quantum component used in the proposed framework contains 8 qubits, 29 logical circuit-depth layers, 21 CNOT gates, and 32 trainable parameters Table 20.

Table 20. Quantum resource profile of the proposed circuit

Resource	Symbol	Value	Calculation/Description
Number of qubits	N_q	8	Reduced quantum feature dimension
Feature-map repetitions	r	2	Z-feature encoding
Variational layers	L	3	Selected configuration
Logical circuit depth	D	29	$2r + (L + 1) + L(N_q - 1)$
CNOT gates	N_{CNOT}	21	$L(N_q - 1)$
Hadamard gates	—	16	rN_q

$R_z$encoding gates	—	16	rN_q
Trainable $R_y$gates	—	32	$N_q(L+1)$
Trainable parameters	(	\theta	)
State-vector amplitudes	2^{N_q}	256	2^8
State-vector memory	—	4 KiB	256×16bytes

5.2. Circuit-Depth and Variational-Layer Analysis

The effect of increasing the number of variational layers was evaluated in terms of the resulting circuit resources. With $N_q = 8$, each additional variational layer contributes seven CNOT operations because the adopted entanglement structure is a linear nearest-neighbour chain. The number of trainable parameters also increases by eight for each additional variational layer.

The resulting resource requirements are summarized in Table 21. The depth of 53, together with 42 CNOT gates and 56 trainable parameters of the last row, indicates that the row contains 6 variational layers, which matches that of Table 14. They follow the resource formulas depth = 13 + 8(L − 1), CNOT gates = 7L and parameters = 8(L + 1). Five layers were not assessed, as this would result in a depth of 45, 35 CNOT and 48 parameters.

Table 21. Quantum resource scaling with variational depth

Variational layers (L)	Logical depth (D)	CNOT gates	Trainable parameters (|θ|)
1	13	7	16
2	21	14	24
3(selected)	29	21	32
4	37	28	40
6	53	42	56

The three-layer configuration was selected for the final model, resulting in 29 logical depth units, 21 CNOT gates, and 32 trainable parameters. This configuration provides the quantum decision stage used to obtain the reported final test accuracy of 0.9940.

The resource analysis also shows that increasing the circuit depth does not merely increase the number of sequential operations; it increases the number of trainable parameters and consequently the number of circuit evaluations required during parameter-shift optimization.

5.3. Parameter-Shift Evaluation Cost

The variational parameters are optimized using parameter-shift gradient evaluation. For a circuit containing 32 trainable parameters, two shifted circuit evaluations are required for each parameter, together with one unshifted forward evaluation. The number of circuit evaluations per optimization iteration is therefore

$$E_{\text{iter}} = 2 \mid \theta \mid +1. \quad (31)$$

For the proposed circuit,

$$E_{\text{iter}} = 2(32) + 1 = 65. \quad (32)$$

With 20,000 optimization iterations, the total number of circuit evaluations is

$$E_{\text{total}} = 65 \times 20{,}000 = 1{,}300{,}000. \quad (33)$$

Table 22. Training-time quantum evaluation requirements

Quantity	Value
Qubits	8
Trainable circuit parameters	32
Circuit evaluations per iteration	65
Optimization iterations	20,000
Total circuit evaluations	13,00,000
Final framework wall-clock time	187.89 s
Final test accuracy	0.994

Table 22: The 1.3 million evaluations represent the number of quantum-circuit evaluations generated by the parameter-shift optimization procedure rather than 1.3 million distinct circuit architectures. The same parameterized circuit is evaluated repeatedly with different parameter values and parameter shifts.

5.4. Simulation Environment and Execution Time

All quantum experiments in the present study were performed using classical quantum-circuit simulation rather than execution on a physical quantum processing unit. The quantum component was evaluated using the Qiskit Aer state-vector simulation environment. Qiskit Aer provides exact state-vector simulation for quantum circuits, allowing expectation values to be evaluated without introducing measurement-shot sampling error in the state-vector configuration.

The complete proposed framework, including the sparse classical feature-extraction stages, quantum simulation, and Pegasos optimization, required 187.89 **s** for the reported experiment. This value represents the complete experimental wall-clock time and should not be interpreted as the execution time of an equivalent physical quantum processor.

Table 23. Simulation and hardware execution status

Resource	Simulation used in this study	Physical quantum hardware
Quantum backend	Qiskit Aer state-vector simulator	Not used
Number of qubits	8	8 logical qubits
Logical circuit depth	29	29 before hardware mapping
Logical CNOT count	21	21 before hardware mapping
Trainable parameters	32	32
Circuit evaluations	13,00,000	Not executed
Measurement model	Exact state-vector evaluation	Shot-based measurement
Hardware noise	None in the reported simulation	Device-dependent
Total framework runtime	187.89 s	Not measured
Physical QPU execution	Not performed	Not performed

Table 23 The difference between simulation and actual execution is crucial since an exact simulation of the state-vector does not capture the noise, readout error, queueing delays, connectivity restrictions or calibration-dependent behavior of a real quantum processor. Similar to Qiskit documentation, there is a difference between exact simulation and noisy/hardware simulation.

5.5. Logical Circuit vs. Physical Hardware Circuit

The logical depth of 29 is the circuit prior to hardware-specific compilation. A physical quantum processor needs the logical circuit to be mapped to the available connection pattern of its qubits and the available native gates. As a result, the physical implementation will have more operations and a deeper circuit than the logical circuit.

The present study, therefore, reports on the logical depth of 29 and 21 logical CNOT gates, and not an unmeasured depth or a gate count of the actual hardware-transpiled circuits. The transpiled depth would depend on the following factors: target processor, coupling map, basis gates, version of the compiler, optimization level, and initial qubit layout. Qiskit provides both circuit parameters and transpiled layouts, which further highlights the separation between an abstract circuit and its physical realization on a particular hardware.

No actual execution on the physical device was conducted in this study, so actual runtime on the physical device is not reported. This prevents misinterpreting a simulator run-time as a real quantum-processing time.

5.6. State-Vector Simulation Scaling

The memory requirement of a full state-vector simulation increases exponentially with the number of qubits. For N_q qubits, the state vector contains

$$2^{N_q} \tag{34}$$

complex amplitudes. With eight qubits, the circuit therefore contains

$$2^8 = 256 \tag{35}$$

complex amplitudes.

Assuming 16 bytes per complex double-precision amplitude, the corresponding state-vector storage is

$$256 \times 16 = 4096 \text{ bytes}, \tag{36}$$

or approximately 4 KiB.

This small state-vector size explains why an eight-qubit circuit can be simulated conveniently on a conventional computing system. However, this scaling should not be interpreted as evidence of quantum computational advantage. Generic state-vector simulation becomes increasingly demanding as the number of qubits grows because the state dimension increases exponentially. Qiskit documentation similarly notes the exponential simulation cost associated with increasing the number of qubits.

Table 24. State-vector resource scaling

Number of qubits	State-vector amplitudes	Approximate complex128 memory
8	256	4 KiB
16	65,536	1 MiB
20	10,48,576	16 MiB
24	16,777,216	256 MiB
30	1,073,741,824	16 GiB

Table 24: The table demonstrates that the eight-qubit configuration is suitable for the simulation environment used in this study, while substantially wider circuits would impose rapidly increasing classical memory requirements.

5.7. Implications for Quantum Hardware

The proposed quantum circuit is relatively narrow, featuring only eight qubits, a logical depth of 29, and 21 CNOTs. The structural suitability for the investigation of the circuit on existing quantum processors stems from these properties. The parameter-shift optimization procedure makes this very complicated process; however, it requires the whole training procedure to be carried out on 1.3 million circuit evaluations.

As a result, the chief constraint is not the width of a single circuit, but the number of times a circuit must be evaluated during the training session. The physical execution would also add measurement sampling error, hardware noise error, limitations of connectivity between devices, translation error and execution latency. The 187.89 s simulation runtime has nothing to do with these factors. The present results thus suggest classical simulation quantum circuits' feasibility, but not a quantum speedup, nor a quantum computational advantage. The computational reduction (low cost) of the proposed framework is mainly identified with the few sparse classical feature-extraction levels before the quantum classifier.

This is a summary of the Quantum Resources.

The final quantum circuit has 8 qubits, 2 repetitions of the feature map, 3 layers of variational cost, 29 Logical Circuit Depth (LCD) units, 21 CNOT gates, and 32 trainable parameters. In parameter-shift optimization, the number of bits evaluated in the circuits (per iteration) is 65 multiplied by 20,000 = 1,300,000. The entire proposed system tested with 0.9940 accuracy and reported an overall experimental wall-clock time of 187.89 s.

No physical quantum processor was employed in any quantum experiment; instead, all quantum experiments were performed by simulating the classical state vectors. Thus, we do not report an unmeasured real hardware execution time, and do not assert quantum computational advantage. Such an explicit distinction between the logical circuit resources, classical simulation cost, and physical hardware execution gives a reproducible and technically conservative estimate of the quantum component.

Table 25. Comparison of the proposed model with state-of-the-art quantum/hybrid image classification methods

Method	Dataset	Key Description	Accuracy
Prajapat et al. [11] – QCNN	MNIST, Wine	Parameterized QCNN with rotation and two-qubit entanglement gates for NISQ-compatible classification	97.22%
Senokosov et al. [13] – Hybrid QNN	MNIST, Medical MNIST, CIFAR-10	Parallel quantum circuits with substantially fewer trainable parameters than classical models	99.21% (MNIST)
Riaz et al. [17] – NNQE	MNIST, CIFAR-10, GTSRB	Strongly entangled quantum circuit with Hadamard gates and no quantum-parameter optimization	93.80% (MNIST)
Proposed Sparse Quantum–Classical Model	LIDC-IDRI	Sparse CNN + Sparse GRU + QNN + Pegasos optimization + SMOTE-GAN	99.40%

The proposed hybrid quantum–classical sparse framework is compared with recently reported state-of-the-art quantum and hybrid quantum–classical image-classification algorithms, as shown in Table 25. The approaches of Prajapat et al. [11], Senokosov et al. [13] and Riaz et al. [17] are compared with the proposed approach. The current methods show the potential of quantum computing for image classification problems, like MNIST, Medical MNIST, CIFAR-10, GTSRB, and Wine. The proposed framework, however, is tested on the LIDC-IDRI CT-image dataset and incorporates all these modules (Sparse CNN, Sparse GRU, quantum neural processing, Pegasos optimization, and SMOTE-GAN augmentation) into a single framework. The proposed model is tested with an accuracy of 99.40%, precision of 99.20%, recall of 99.10%, F1 score of 99.00% and computational time of 187.89 s. The comparison shows the competitive performance of the proposed framework, and also illustrates the use of the proposed framework for medical CT-

image classification. The accuracy values reported should be interpreted as contextual comparisons of the compared studies, as each study may have different datasets, different classification tasks, and different experimental settings.

6. Conclusion and Future Work

This study focused on the five-classification of pulmonary nodules based on CT images using a quantum–classical squeezed sparse network. The framework consists of a Sparse CNN that extracts spatial features from each axial image of a volume of interest in the nodule region, a Sparse GRU, which models the dependency between the ordered axial images, and an eight-qubit variational quantum circuit whose expectation values are measured and classified by an eight-layer large-margin decision layer trained using the Pegasos stochastic sub-gradient procedure, with circuit parameters updated using parameter-shift estimates, not back-propagation through the circuit. The unit of analysis is the annotated nodule, the labels are based on the middle of the radiologist's malignancy ratings, and each partition is patient-disjoint. The full protocol achieved a 0.9940 accuracy and a macro-averaged AUC of 0.988 on the patient-disjoint test partition, while the classical baseline achieved 0.9300 accuracy and an AUC of 0.933; with processing times of 187.89 s and 378.99 s, 328.89 s and 300.92 s for the Q-SwinER, QNN and QCNN baselines re-trained on the same protocol. The results are subject to the labelling rule applied to them, to the sampling procedure followed here, and, of course, will depend on the single collection. There are three statements that are intentionally not made. The study does not show a quantum advantage: All quantum components have been simulated classically, on a noiseless state-vector backend, and the speed-up comes from the fact that the classical front-end is sparse, not the quantum backend. The study does not make an early prediction of lung cancer; it simply categorizes the malignancy category of the annotated nodules. Moreover, the study did not demonstrate clinical utility – reference standard was the radiologist's perception, not histopathology, and no reader study was performed. The limitations in Section 4.8 lead to future work. The first priority is the implementation of the variational circuit on a physical quantum device and in a realistic noise model, with a quantification of the difference between the simulated and hardware behaviour. The second is validation on separate multi-centre cohorts with histopathologically confirmed outcomes. The third is to expand the labelling scheme to maintain inter-reader disagreement, such as by training the model against the full distribution of labels instead of the median, so the model's uncertainty would be more meaningful. The fourth, which is relevant to this paper, is integrating the classifier with an upstream detection stage and reporting the end-to-end performance at the level of the complete examination instead of the annotated nodule.

Conflict of Interest

The authors assert no conflict of interest

Funding

Author declared no funding was received for this Research and Publication

Data Availability

Data is available from the respective author upon request.

Ethical Approval

This study used the publicly available, de-identified LIDC-IDRI collection [19]. For external evaluation, a retrospective institutional cohort comprising 312 CT scans was used. The institutional cohort was obtained under the approval or exemption of the relevant institutional ethics committee. Informed consent was waived or obtained in accordance with the institutional ethics requirements. All data were de-identified before analysis to protect patient privacy. No animals were involved in this study.

Author Contribution

S. Radhika and Dr.G. Sharada contributed equally to this work.

References

[1] Ricardo Daniel Monteiro Simões et al., “Experimental Evaluation of Quantum Machine Learning Algorithms,” *IEEE Access*, vol. 11, pp. 6197-6208, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Mohannad T. Mohammed, and Omar Fitian Rashid, “Document Retrieval using Term Term Frequency Inverse Sentence Frequency Weighting Scheme,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 31, no. 3, pp. 1478-1485, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] Gabriele Cavallaro et al., “Hybrid Quantum-Classical Workflows in Modular Supercomputing Architectures with the Julich Unified Infrastructure for Quantum Computing,” *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, Kuala Lumpur, Malaysia, pp. 4149-4152, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[4] Berkan Oztas et al., “Transaction Monitoring in Anti-Money Laundering: A Qualitative Analysis and Points of View from Industry,” *Future Generation Computer Systems*, vol. 159, pp. 161-171, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] G. SaravanaKumar et al., “Beetle-Optimized Hybrid Ensemble for Multi-Attack Classification in VANETs,” *Transactions on Emerging Telecommunications Technologies*, vol. 36, no. 10, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[6] Massimo Bertolini et al., “Machine Learning for Industrial Applications: A Comprehensive Literature Review,” *Expert Systems with Applications*, vol. 175, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] Ponugoti Kalpana et al., "Health-Fots-A Latency Aware Fog based IoT Environment and Efficient Monitoring of Body's Vital Parameters in Smart Health Care Environment," *Journal of Intelligent Systems and Internet of Things*, vol. 15, no.1, pp. 144-156, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[8] Shaik Abdul Nabi et al., "Distributed Private Preserving Learning based Chaotic Encryption Framework for Cognitive Healthcare IoT Systems," *Informatics in Medicine Unlocked*, vol. 49, pp. 1-17, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Ponugoti Kalpana et al., "A Deep Reinforcement Learning-based Task Offloading Framework for Edge-Cloud Computing," *2024 International Conference on Integrated Circuits and Communication Systems (ICICACS)*, Raichur, India, pp. 1-5, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[10] Andreas Bayerstadler et al., "Industry Quantum Computing Applications," *EPJ Quantum Technology*, vol. 8, no. 1, pp. 1-17, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Sunil Prajapat et al., "Quantum Computing Meets Deep Learning: A QCNN Model for Accurate and Efficient Image Classification," *Mathematics*, vol. 13, no. 9, pp. 1-26, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[12] Pradeep Lamichhane, and Danda B. Rawat, "Quantum Machine Learning: Recent Advances, Challenges and Perspectives," *IEEE Access*, vol. 13, pp. 94057-94105, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[13] Arsenii Senokosov et al., "Quantum Machine Learning for Image Classification," *Machine Learning: Science and Technology*, vol. 5, no. 1, pp. 1-14, 2024.[CrossRef] [Google Scholar] [Publisher Link]

[14] G. Nagarajan, Tathi Reddy Pindi, and R. Dinesh Makkapati, "A Crop Pest Classification using Deep Learning and Quantum Machine Learning," *AIP Conference Proceeding: 4th International Conference on Internet of Things*, Kattankalathur, India, vol. 3075, no. 1, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Surajudeen Shina Ajibosin, and Deniz Cetinkaya, "Implementation and Performance Evaluation of Quantum Machine Learning Algorithms for Binary Classification," *Software*, vol. 3, no. 4, pp. 498-513, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] Lin Wei et al., "Quantum Machine Learning in Medical Image Analysis: A Survey," *Neurocomputing*, vol. 525, pp. 42-53, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] Farina Riaz et al., "Accurate Image Multi-Class Classification Neural Network Model with Quantum Entanglement Approach," *Sensors*, vol. 23, no. 5, pp. 1-11, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] William M. Watkins, Samuel Yen-Chi Chen, and Shinjae Yoo, "Quantum Machine Learning with Differential Privacy," *Scientific Reports*, vol. 13, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[19] Samuel G. Armato III, "The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): A Completed Reference Database of Lung Nodules on CT Scans," *Medical Physics*, vol. 38, no. 2, pp. 915-931, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[20] Subodh Nath Pushpak, and Sarika Jain, "An Introduction to Quantum Machine Learning Techniques," *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India, pp. 1-6, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[21] Shai Shalev-Shwartz et al., "Pegasos: Primal Estimated Sub-Gradient Solver for SVM," *Mathematical Programming*, vol. 127, no. 1, pp. 3-30, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[22] Vojtěch Havlíček et al., "Supervised Learning with Quantum-Enhanced Feature Spaces," *Nature*, vol. 567, pp. 209-212, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[23] K. Mitarai et al., "Quantum Circuit Learning," *Physical Review A*, vol. 98, pp. 1-6, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[24] Arnaud Arindra Adiyoso Setio et al., "Validation, Comparison, and Combination of Algorithms for Automatic Detection of Pulmonary Nodules in Computed Tomography Images: the LUNA16 Challenge," *Medical Image Analysis*, vol. 42, pp. 1-13, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[25] Karel Zuiderveld, *Contrast Limited Adaptive Histogram Equalization*, Graphics Gems IV, 474-485, Academic Press, 1994. [Google Scholar] [Publisher Link]

[26] N. V. Chawla et al., "SMOTE: Synthetic Minority Over-Sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[27] Ali Javadi-Abhari et al., "Quantum Computing with Qiskit," *arXiv preprint*, pp. 1-19, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[28] Chuan Guo et al., "On Calibration of Modern Neural Network," *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 1321-1330, 2017. [Google Scholar] [Publisher Link]

[29] Mundher Al-Shabi et al., "Lung Nodule Classification using Deep Local–Global Networks," *International Journal of Computer Assisted Radiology and Surgery*, vol. 14, no. 10, pp. 1815-1819, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[30] Wei Shen et al., "Multi-crop Convolutional Neural Networks for Lung Nodule Malignancy Suspiciousness classification," *Pattern Recognition*, vol. 61, pp. 663-673, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[31] Kyunghyun Cho et al., "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qata, pp. 1724-1734, 2014. [Google Scholar] [Publisher Link]

[32] Fu Chen et al., "Quantum Mixed-State Self-Attention Network," *Neural Networks*, vol. 185, pp. 1-33, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[33] Ren-Xin Zhao, Jinjing Shi, and Xuelong Li, "QKSAN: A Quantum Kernel Self-Attention Network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 10184-10195, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[34] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer, "Effect of Data Encoding on the Expressive Power of Variational Quantum-Machinelearning Models," *Physical Review A*, vol. 103, no. 3, pp. 1-16, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[35] Jin Zheng, Qing Gao, and Zibo Miao, "Design of a Quantum Self-Attention Neural Network on Quantum Circuits," *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Honolulu, Oahu, HI, USA, pp. 1058-1063, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[36] Mateusz Ostaszewski et al., "Reinforcement Learning for Optimization of Variational Quantum Circuit Architectures," *Advances in Neural Information Processing Systems*, vol. 34, pp. 1-13, 2021. [Google Scholar] [Publisher Link]

[37] Guangxi Li, Xuanqiang Zhao, and Xin Wang, "Quantum Self-Attention Neural Networks for Text Classification," *arXiv preprint*, pp. 1-13, 2022. [CrossRef] [Google Scholar] [Publisher Link]