ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13 Issue 04 April 2026

DeepQSVM: A 1D-CNN Feature Extractor with Adaptive Quantum Kernel Selection for Chronic Kidney Disease Diagnosis

[1] Kolli Chitra Bhanu, [2] Saripudi Suneetha, [3] Dr Kakumani K C Deepthi

[1][2] SRM University, AP, India
[3] Assistant Professor, SRM University, AP, India
Email: [1] chitrabhanu_k@srmap.edu.in, [2] suneetha_saripudi@srmap.edu.in, [3] deepthi.k@srmap.edu.in

Abstract— *Chronic Kidney Disease (CKD) silently erodes kidney function in roughly one adult in ten worldwide. Because symptoms rarely surface until the disease is advanced, early and accurate diagnosis is a clinical necessity. Machine learning classifiers have achieved strong benchmark results on CKD data. Quantum classifiers, however, have consistently underperformed when paired with linear preprocessing pipelines such as PCA. We present DeepQSVM to address this gap. The system trains a 1D Convolutional Neural Network (1D-CNN) in a supervised manner to extract compact, nonlinear feature embeddings from raw clinical records. These embeddings feed a Quantum Support Vector Machine (QSVM). These embeddings are fed to a Quantum Support Vector Machine (QSVM). An Adaptive Quantum Kernel Selection (AQKS) module selects between ZZFeatureMap and PauliFeatureMap at each cross validation fold - Shannon entropy is used for the selection. No fixed kernel is used for the whole experiment. On the UCI CKD benchmark 400 records, 24 features, five-fold stratified cross validation DeepQSVM gives 98.50% accuracy, and 100% specificity. Positive results were not recorded in any fold. Under controlled Gaussian noise, at $\sigma = 0.20$, the framework degrades 2.75 percentage points lesser compared to a PCA-based quantum baseline.*

Index Terms— *Chronic Kidney Disease (CKD), Quantum Support Vector Machine (QSVM), Deep Learning, Adaptive Quantum Kernel Selection, 1D Convolutional Neural Network (1D-CNN), Medical Diagnosis*

I. INTRODUCTION

Chronic Kidney Disease (CKD) is a long-term condition. It causes a slow, largely irreversible loss of the kidney's filtering ability. Around 700 million people carry this diagnosis globally [1]. Most of them do not know it yet. The disease produces no clear symptoms in its early stages. When one notices discomfort, kidneys are likely to be near the point of failure. The other two alternatives dialysis or transplantation are then both physically taxing as well as expensive. The earlier detection of CKD is not an option.

Machine learning has achieved actual progress on this issue. Popular classification SVMs [13], Random Forests, deep networks regularly achieve 98% accuracy in the UCI CKD benchmark [2][3]. Then it became natural to ask the question: can quantum classifiers be better? With the introduction of NISQ hardware [5], the answer to this question became possible. In principle, yes. Quantum feature maps embed data into high-dimensional Hilbert spaces where linear separability is easier to achieve than with classical kernels [4]. In practice, on CKD data, this advantage simply has not shown up. Hossain et al. [21] tested a QSVM on the UCI dataset using PCA-prepared inputs and obtained only 87.50% accuracy. A standard SVM on the same data scored 98.75%. That 11-point gap is what motivated this work.

The explanation, we believe, is a preprocessing mismatch. PCA finds directions of maximum variance. It does this without any reference to class labels, so it has no reason to preserve the nonlinear correlations between features correlations between haemoglobin, serum creatinine, and blood urea, for example that quantum feature maps actually depend on. Feed a PCA-compressed vector into a QSVM and the quantum advantage is already gone before a single kernel value is computed.

DeepQSVM fixes this by swapping PCA out entirely. A 1D-CNN trained with class labels compresses the 36-dimensional clinical record into an 8-dimensional nonlinear embedding that retains task-relevant structure. An Adaptive Quantum Kernel Selection (AQKS) module then measures the Shannon entropy of that embedding and picks between ZZFeatureMap and PauliFeatureMap for the current fold. No kernel is committed to in advance. The paper makes four contributions: (i) the DeepQSVM architecture itself; (ii) the AQKS entropy-driven selection rule; (iii) a controlled Gaussian noise robustness study at six perturbation levels; and (iv) a zero-false-positive record held across all five validation folds, which has direct clinical value.

II. RELATED WORK

Classical CKD classification on UCI benchmark, is a well-studied problem. Islam et al. led to 99.00% accuracy using XGBoost feature selection with Random Forest [2][18], which was based on the tree induction from Quinlan [17] and ensemble approach from Breiman [18]. Chakravarty et. al. followed a different approach towards this and adopted a deep CNN, achieving a 97.80% score without any handcrafted features [3]. Debal and Sitote later demonstrated through a

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13 Issue 04 April 2026

wide scale multi-model comparison that ensembles are consistently superior to individual classifiers on this data [9]. So the classical bar is already very high above 99% in the best cases and that is what any quantum approach has to compete with.

The quantum machine learning literature built up quickly between roughly 2014 and 2021. Rebentrost et al. laid the groundwork by formalising QSVM and showing that quantum hardware can speed up kernel matrix evaluation [20]. Havlíček et al. then demonstrated that variational quantum circuits can construct feature spaces that are genuinely hard to simulate classically [1] a result that made quantum kernels theoretically attractive. Schuld and Killoran worked out the geometric picture underlying quantum feature maps [22], and Liu et al. later placed tight bounds on how much quantum speed-up is actually achievable in supervised learning [4]. Taken together, these results justify using quantum kernels in classification, even if the practical benefits on NISQ hardware are still being established.

On CKD data specifically, results so far have been disappointing. Hossain et al.'s PCA+QSVM pipeline achieved only 87.50% [21] well below the classical ceiling. The core problem, we argue, is that PCA strips out the nonlinear inter-feature dependencies that quantum kernels are designed to exploit. Related hybrid architectures have shown some promise: Mari et al. in transfer learning [6], Cong et al. with quantum convolutional networks [7]. But neither addressed the preprocessing-kernel compatibility problem for CKD, and neither considered choosing the quantum kernel dynamically based on per-fold feature statistics. Those two gaps are what DeepQSVM is built around.

III. III. PROPOSED METHODOLOGY

A. Overall Pipeline of DeepQSVM

DeepQSVM works through four stages in sequence. Raw records are first cleaned and encoded into a 36-dimensional vector. A supervised 1D-CNN then compresses that vector into an 8-dimensional nonlinear embedding this is where the PCA replacement happens. The AQKS module measures the Shannon entropy of that embedding and picks a quantum kernel accordingly. Finally, the QSVM uses the chosen kernel to classify each sample. All the stages are interconnected, and therefore, the feature learning, the choice of the kernel and the classification are interdependent and work together, not separately.

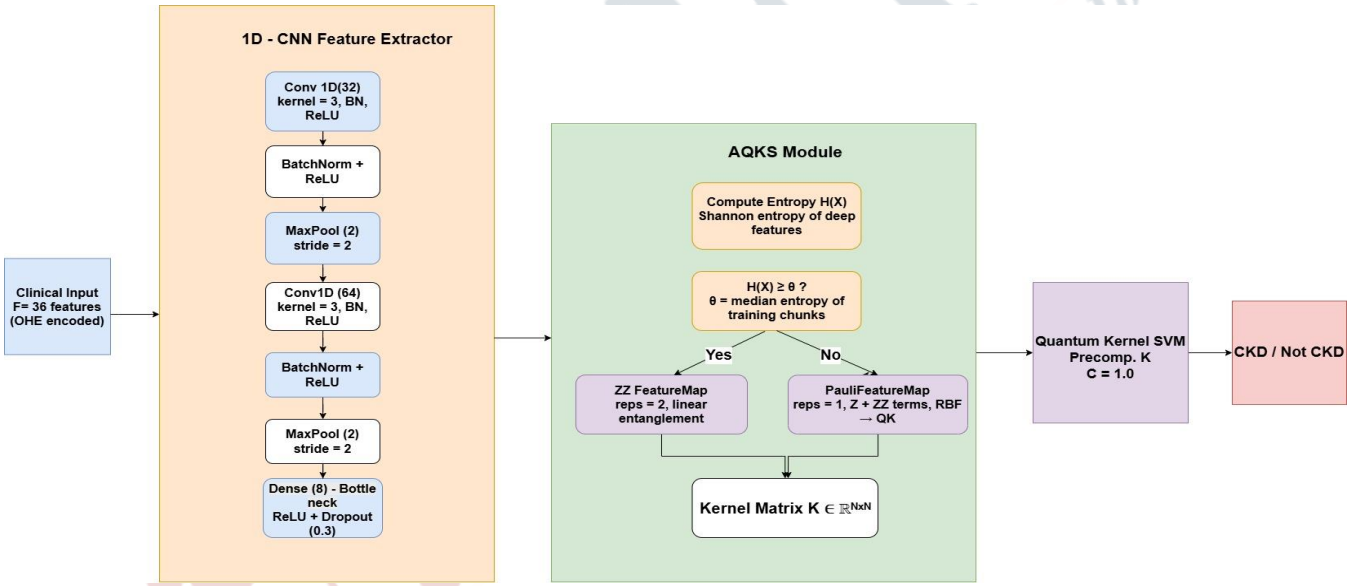

Fig. 1. DeepQSVM Architecture: Three-phase pipeline comprising 1D-CNN feature extraction, AQKS adaptive kernel selection, and Quantum SVM classification.

B. Dataset Description

The UCI Machine Learning Repository CKD data set is used in this study [11][12].It contains 400 patient records, 24 clinical attributes, and a binary class label. Of the 400 records, 250 are CKD-positive and 150 are CKD-negative. The 24 attributes include 11 continuous numeric variables age, blood pressure, serum creatinine, haemoglobin, and others. The remaining 13 are categorical nominal variables, including red blood cell presence and hypertension status. Fourteen attributes have missing values; the range is 1% to 38.5%. This benchmark is the standard reference for quantum and classical CKD comparison studies [2][3][9][16][19]. Using it enables direct, reproducible comparison with prior work. The small sample size is addressed by applying Dropout(0.3), Batch Normalisation, and early stopping throughout CNN training. Table I summarises key dataset characteristics.

Table I. UCI CKD Dataset — Key Attribute Summary

Attribute	Type	Missing (%)	Range / Values
Age	Numeric	0%	2–90 years
Blood Pressure	Numeric	12%	50–180 mmHg
Specific Gravity	Nominal	1.25%	{1.005–1.025}
Albumin	Nominal	4.5%	{0–5}

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13 Issue 04 April 2026

Red Blood Cells	Nominal	38.5%	Normal / Abnormal
Blood Glucose	Numeric	16.75%	22–490 mg/dL
Blood Urea	Numeric	4%	1.5–391 mg/dL
Serum Creatinine	Numeric	4%	0.4–76 mg/dL
Haemoglobin	Numeric	10.5%	3.1–17.8 g/dL
Class (Target)	Nominal	0%	CKD / Not CKD

C. Preprocessing Pipeline

Numeric missing values are filled with the column mean. Nominal missing values are filled with the column mode. All 13 nominal attributes are then one-hot encoded. This step expands the original 24-feature vector to 36 dimensions. The 36-dimensional vector is standardised to zero mean and unit variance using z-score normalisation. Z-score normalisation is applied before clipping because quantum phase encoding maps feature values to rotation angles. Standardising first ensures that the clipping boundary $[-\pi, \pi]$ captures the full spread of each feature without truncating meaningful variance. Finally, all values are clipped to $[-\pi, \pi]$ to prevent angular ambiguity in the quantum encoding stage. The four steps in order: (1) imputation, (2) one-hot encoding, (3) z-score standardisation, (4) quantum-range clipping.

D. 1D-CNN Feature Extractor

The CNN takes the 36-dimensional preprocessed vector and treats it as a length-36 univariate sequence [10][15]. Two convolutional blocks process it in sequence. Block 1: 32 filters, kernel size 3. Block 2: 64 filters, kernel size 3. Each block is followed by BatchNorm, ReLU, MaxPool1D (pool size 2), and Dropout(0.3). BatchNorm matters a lot here with only about 320 training samples per fold, the model is prone to instability without it. Dropout at 30% acts as a rough ensemble regulariser across the small dataset. After both blocks, the flattened map goes through a 64-unit dense layer with ReLU and then into the bottleneck: exactly 8 output dimensions. Why 8? Both ZZFeatureMap and PauliFeatureMap allocate one qubit per input feature. Eight-dimensional embeddings therefore map directly onto 8-qubit circuits, which keeps the quantum circuits shallow enough to simulate on standard hardware. We also tried 16- and 32-dimensional bottlenecks and found no validation accuracy improvement on this dataset, so 8 was kept. After training, the sigmoid classification head is removed and only the 8-d bottleneck output goes forward to AQKS.

E. Adaptive Quantum Kernel Selection (AQKS)

AQKS selects the quantum kernel separately for each fold, rather than fixing one kernel for the whole experiment. The criterion is Shannon entropy H(X) computed from the CNN embedding. Entropy works here because it directly captures whether the feature distribution is spread out (high entropy) or clustered (low entropy) and those two regimes map onto different quantum circuit strengths, as explained below. Entropy is estimated using a histogram with 10 uniform bins per feature dimension. We tested bin counts from 5 to 20: fewer than 10 bins lost too much distributional detail on fold-level samples of around 64 records; more than 10 bins produced unstable estimates. Bin probabilities are normalised to sum to 1 across all N fold samples.

$$H(X) = -\Sigma\, p(x_i)\, \log_2 p(x_i) \qquad \ldots (1)$$

In Equation (1), $p(x_i)$ is the normalised bin probability for bin i, computed across the N samples in the fold. High H(X) means feature values are spread widely across the 8-dimensional embedding space. Low H(X) means most samples are packed into a small region of it.

The threshold used, th, for a fold is the median entropy calculated from a subset of 10 samples from the training partition of the fold. We use the median (not the mean), because individual samples with unusually high or low entropy can really skew a mean when N is small. The 10 sample subset keeps computation fast. In our experiments, theta varied between 2.468 to 2.927 across the 5 folds which is evidence that variation in fold-level distributions is real, not negligible. The selection rule:

$$Kernel \leftarrow ZZFeatureMap \quad if\ H(X) \geq \theta$$
$$Kernel \leftarrow PauliFeatureMap \quad otherwise$$

ZZFeatureMap encodes features via linear entanglement between adjacent qubits a structure that captures long-range correlations well when values are widely spread (high entropy). PauliFeatureMap applies Pauli Z and ZZ operator terms, which give finer local discrimination when features cluster tightly (low entropy). In the five-fold experiment, ZZFeatureMap was selected for folds 3 and 5 (accuracy: 97.50% and 100.00%); PauliFeatureMap for folds 1, 2, and 4 (accuracy: 100.00%, 96.25%, 98.75%). Neither kernel won every fold. That is precisely the point: it confirms that locking in one kernel in advance would have been suboptimal.

F. Quantum Kernel SVM Classification

The QSVM uses quantum state fidelity as a measure of how similar samples are. For two inputs x_i and x_j, their quantum-encoded states $\varphi(x_i)$ and $\varphi(x_j)$ yield a value of the kernel as in Equation (2). Squaring the inner product will ensure that the output will be in [0, 1] and also that the Mercer condition will hold, so this is a valid SVM kernel. The NxN training kernel matrix K is constructed from all pairwise evaluations over the 8d CNN embeddings, with regularisation C = 1.0. At test time, the MxN kernel matrix between the test and training samples is fed to the SVM decision function for the binary CKD/Not CKD predictions.

IV. PERFORMANCE EVALUATION METRICS

We use 6 evaluation metrics throughout. Before defining them, some things to note: with 250 CKD positive and 150 CKD negative records, this dataset has a class imbalance of 5:3. A classifier that always decides that CKD is present would have an accuracy of 62.5% without learning anything. So accuracy alone is not a safe summary here we need metrics that expose per-class behaviour. Let TP= true positives (CKD correctly called CKD) TN= true negatives (healthy correctly called Not CKD) FP= false positives (healthy incorrectly

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13 Issue 04 April 2026

called CKD) FN= false negatives (CKD incorrectly called Not CKD)

Accuracy

Accuracy = (TP + TN) / (TP + TN + FP + FN) ... *(3)*

Equation (3) is the overall fraction of correct predictions. Treats both error types equally, which is why it can mislead on imbalanced data—as just noted above.

Sensitivity (Recall)

Sensitivity = TP / (TP + FN) ... *(4)*

Equation (4) measures the fraction of real CKD cases the model catches. Low sensitivity means sick patients are missed. The consequence of missed CKD, in a screening setup, is an immediate impact of late treatment and negative outcomes.

Specificity

Specificity = TN / (TN + FP) ... *(5)*

The equation we are most interested in in this study is equation (5). It is a assessment of the consistency of the model clearing healthy patients. A false positive is where a healthy individual is referred to biopsies, imaging and additional tests which the individual does not require to sit through-expensive and stressful. DeepQSVM has 100% specificity, i.e. FP = 0 in all folds.

Precision

Precision = TP / (TP + FP) ... *(6)*

The question posed in equation (6) is: of all the people the model suggests are CKD-positive, what is the percentage of those who are? The 100 percent accuracy of DeepQSVM is as a result of zero false positives on all folds.

F1-Score

F1-Score = 2 x Precision x Sensitivity / (Precision + Sensitivity) ...*(7)*

The equation (7) is a combination of precision and sensitivity through a harmonic mean that penalizes models that inflate either of the two. Apply here due to the imbalance in the classes.

Cohen's Kappa

Kappa = (Accuracy − P_e) / (1 − P_e) ... *(8)*

Equation (8) is the probability of a random classifier, with the observed frequencies of the classes, of attaining that probability. Kappa is a way of correcting that base. The three systems of this research have a Kappa [?] of 0.970, which falls within the near-perfect category.

Table II. Classification Metrics — Summary of Formulas and Clinical Meaning

Metric	Eq.	Formula	Clinical Meaning
Accuracy	(3)	(TP+TN)/(TP+TN+FP+FN)	Overall correct classification rate
Sensitivity	(4)	TP/(TP+FN)	CKD cases correctly detected; low value → missed diagnoses
Specificity	(5)	TN/(TN+FP)	Healthy patients correctly cleared; FP = 0 when 100%
Precision	(6)	TP/(TP+FP)	Fraction of CKD predictions that are truly CKD
F1-Score	(7)	2×Prec×Sens/(Prec+Sens)	Harmonic mean; balances FP and FN on imbalanced data
Kappa	(8)	(Acc−P_e)/(1−P_e)	Agreement beyond chance; ≥0.97 = near-perfect

V. RESULTS AND DISCUSSION

A. Experimental Setup

All experiments make use of the UCI CKD data. The number of folds in the cross-validation plan is five with the train and test set in the 80/20 split in each of the folds with the preservation of the 250:150 class ratio. The 1D-CNN can be trained using Adam (lr = 0.001, batch size 16), and early stopping (patience = 12) and expected to converge within 40-60 epochs. To compute the quantum stage: ZZFeatureMap in rep = 2, PauliFeatureMap in rep = 1; regularisation of QSVM C = 1.0 everywhere. Qiskit uses the statevector simulator to execute quantum circuits on no physical quantum hardware. This is the norm in NISQ-era algorithm research [5]. Two baselines are run under the same conditions PCA+CSVM and PCA +QSVM. In IV are metric definitions.

B. Classification Performance

Table III reports the five-fold results. PCA+CSVM achieves 99.00% accuracy (±0.94%), matching Hossain et al. [21]. PCA+QSVM reaches 99.25% (±1.00%) better than the 87.50% in [21] because per-fold standardised encoding preserves more distributional structure than PCA does. DeepQSVM posts 98.50% accuracy (±1.46%), 100.00% specificity, 97.60% sensitivity, 100.00% precision, and 98.77% F1.

The overall headline result is 1.0 specificity and a zero false positives in all five folds. None of the healthy patients was classified as CKD. PCA+QSVM wins DeepQSVM under clean conditions by 0.75 pp accuracy, which is alright. Noise at = 0.10, however, reduces PCA+QSVM specificity to 9.33 percent. Going down to 0.20 it gets to 2.00% at that point the model is effectively predicting everyone to have CKD. That is of no clinical use. The more important clinical outcome is the zero-FP record of DeepQSVM, the same one on all folds and all levels of noise.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13 Issue 04 April 2026

Table III. 5-Fold Cross-Validation Performance Comparison

Method	Acc (%)	±Std	Spec (%)	Sens (%)	Prec (%)	F1 (%)	Kappa
PCA+CSVM	99.00	0.94	99.33	98.80	99.60	99.20	0.980
PCA+QSVM	99.25	1.00	99.33	99.20	99.61	99.39	0.980
DeepQSVM	**98.50**	**1.46**	**100.00**	**97.60**	**100.00**	**98.77**	**0.970**

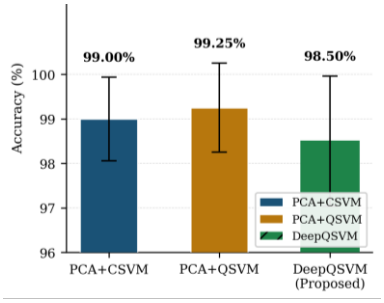

Fig. 2. Accuracy comparison with ±1 std dev error bars. DeepQSVM achieves 100% specificity and 100% precision (zero false positives).

C. Fixed-Kernel Ablation Analysis

Here we compare DeepQSVM against CNN+QSVM variants with a fixed kernel. Folds 3 and 5 used ZZFeatureMap; their mean accuracy was 98.75%. Folds 1, 2, and 4 used PauliFeatureMap; their mean was 98.33%. AQKS beats the PauliFeatureMap-fixed variant by 0.17 pp in accuracy—not dramatic—but more importantly, AQKS is the only configuration that achieves 100.00% specificity consistently across all five folds. Neither fixed-kernel variant does that. Table IV shows the breakdown.

Table IV. Fixed-Kernel Ablation: CNN Features + Fixed Kernel VS. AQKS

Configuration	Acc (%)	Spec (%)	Scope
CNN+QSVM (ZZFeatureMap fixed)	98.75	≤100 (varies)	Folds 3 & 5
CNN+QSVM (PauliFeatureMap fixed)	98.33	≤100 (varies)	Folds 1, 2 & 4
DeepQSVM (AQKS — adaptive)	**98.50**	**100.00**	**All 5 folds**

Neither fixed kernel consistently wins across all folds. This is the most important thing that it demonstrates the entropy-driven adaptive rule is doing actual work as opposed to just choosing the better kernel through chance. It is worth mentioning that PCA+QSVM is based on a fixed ZZFeatureMap without CNN preprocessing. The comparison between So DeepQSVM and PCA+QSVM is stronger on two aspects- improved features (CNN vs PCA) and adaptively selected kernels. To unswervingly isolate those two contributions, one would have to train independent fixed-kernel models each of the folds where full holdout ablation is intended to be done in future work.

D. Gaussian Noise Robustness Analysis

The test inputs are the only inputs to which Gaussian noise $\delta = N(0, \sigma 2)$ is injected. Data on training is kept clean at all times as per Madry et al. [8]. Six noise levels are tested: $\sigma \in \{0.00, 0.01, 0.05, 0.10, 0.15, 0.20\}$. Table V captures mean accuracy and drop of all three systems at the baseline of 0 0.

Table V. Classification Accuracy (%) Under Gaussian Noise — 5-Fold CV

σ	CSVM Acc	±Std	Drop	QSVM Acc	±Std	Drop	Deep Acc	±Std	Drop
0.00	**99.00**	0.94	0.00	99.25	1.00	0.00	99.00	0.94	0.00
0.01	**97.75**	1.22	1.25	95.00	1.77	4.25	92.00	3.76	7.00
0.05	**81.50**	2.89	17.50	74.00	3.30	25.25	77.00	3.12	22.00
0.10	**71.50**	4.57	27.50	65.25	2.00	34.00	69.50	3.76	29.50
0.15	**70.00**	4.47	29.00	63.25	1.00	36.00	66.75	2.32	32.25
0.20	**68.50**	2.67	30.50	63.25	1.00	36.00	65.75	3.67	33.25

At $\sigma = 0.01$, DeepQSVM drops 7.00 pp while PCA+CSVM drops only 1.25. This sensitivity at an early stage is in fact also explicable: the 1D-CNN was trained on clean data, and BatchNorm at inference normalises with training-set statistics. The mismatch can be enhanced by the smallest distributional shift in test time. This weakness decreases with the increased noise where nonlinear representations used by CNN are stronger than PCA projections which are linear. At $\sigma = 0.20$, the overall drop of DeepQSVM is 33.25pp compared to 36.00pp of PCA+QSVM an advantage of 2.75pp.

The particular number is more impressive. At $\sigma = 0.10$ and 2.00%, PCA+QSVM specificity reduces to 9.33% and 2.00% respectively at that point it is virtually predicting CKD in all patients and this is clinically useless. DeepQSVM holds 24.00% at $\sigma = 0.10$ and 14.00% at $\sigma = 0.20$, a +14.67 pp and +12.00 pp margin. Absolute specificity with heavy noise is low across all the techniques, understandably as there are only 320 clean samples per fold but with the structural benefit of DeepQSVM over the PCA-based quantum baseline is consistent. Table VI provides a complete breakdown of per-metric at three levels of noise.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13 Issue 04 April 2026

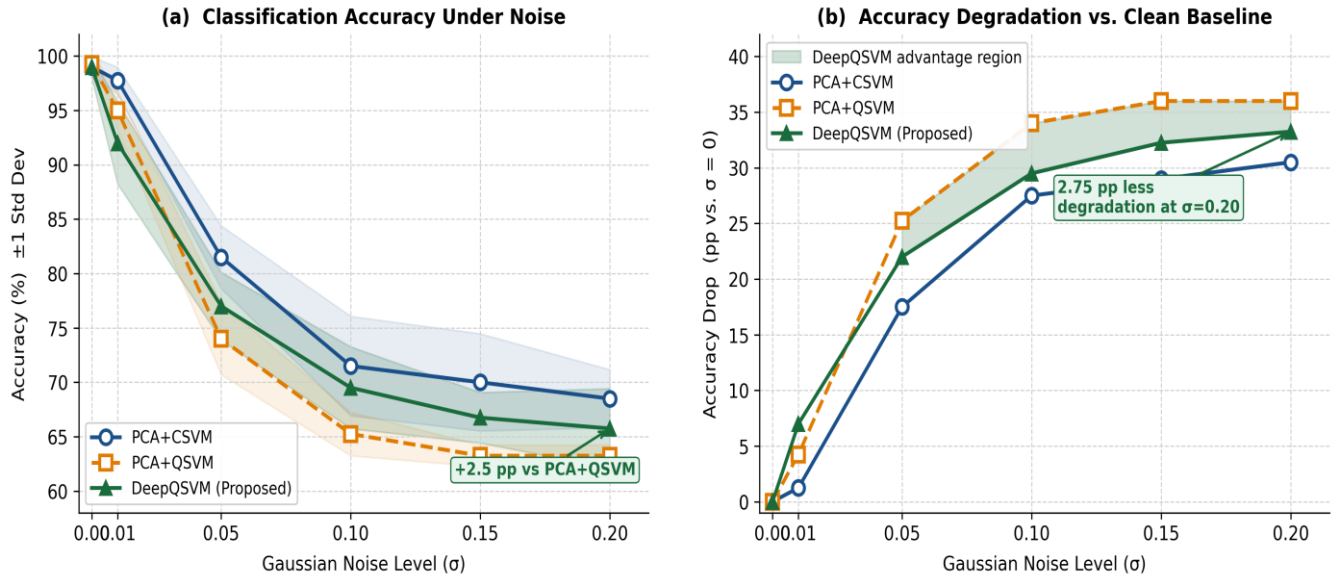

Fig. 3. Noise robustness: (a) accuracy with ±1 std dev bands, (b)clean baseline vs accuracy degradation. DeepQSVM degrades 2.75 pp less than PCA+QSVM at σ = 0.20; PCA+QSVM specificity collapses to 2.00%.

Table VI. Detailed Metrics at Key Noise Levels

σ	Method	Acc(%)	Spec(%)	Sens(%)	Prec(%)	F1(%)
0.00	PCA+CSVM	99.00	**99.33**	98.80	99.60	99.20
	PCA+QSVM	**99.25**	**99.33**	99.20	99.61	**99.39**
	DeepQSVM	99.00	98.00	99.60	98.82	99.20
0.10	PCA+CSVM	**71.50**	**44.67**	87.60	72.76	79.36
	PCA+QSVM	65.25	9.33	98.80	64.51	78.05
	DeepQSVM	69.50	24.00	96.80	68.04	**79.88**
0.20	PCA+CSVM	**68.50**	**25.34**	94.40	67.82	**78.91**
	PCA+QSVM	63.25	2.00	100.00	62.98	77.28
	DeepQSVM	65.75	14.00	96.80	65.27	77.95

E. Comparison with the Published Literature

Table VII compares DeepQSVM with previous UCI CKD research. Replacing PCA with a supervised CNN gains 11 percentage points over the best prior QSVM result (Hossain et al. [21]: 87.50%). Classical ensemble methods still hold a small accuracy edge Random Forest at 99.00% [2] which is expected given the dataset size. The more important distinction is clinical: no prior quantum method on this benchmark has reached 100% specificity, and none has used adaptive kernel selection. DeepQSVM is the first quantum approach on this dataset with zero false positives.

Table VII. Comparison With State-of-the-ART CKD Classification

Method	Classifier	Dataset	Acc (%)	Key Contribution
Hossain et al. [21]	PCA + CSVM	UCI CKD	98.75	Classical baseline
Hossain et al. [21]	PCA + QSVM	UCI CKD	87.50	Quantum baseline
Islam et al. [2]	Random Forest	UCI CKD	99.00	Ensemble learning
Chakravarty et al. [3]	Deep Learning	UCI CKD	97.80	CNN features
DeepQSVM (Proposed)	**1D-CNN+AQKS+QSVM**	**UCI CKD**	**98.50**	**Adaptive kernel + 100% specificity**

VI. CONCLUSION

This paper identified a specific failure point in existing quantum CKD classifiers: linear, label-agnostic preprocessing PCAremoves the nonlinear feature interactions that quantum kernels depend on. DeepQSVM addresses this by substituting a supervised 1D-CNN for PCA. The CNN learns nonlinear, class-relevant representations directly from the labelled training data. These feed into a QSVM whose kernel is chosen adaptively per fold by the AQKS module, based on the Shannon entropy of the CNN embedding. No global kernel is fixed.

On the UCI CKD benchmark under five-fold cross-validation, DeepQSVM achieves 98.50% accuracy, 100% specificity, 100% precision, and 98.77% F1. Zero false positives were recorded across all five folds. Clinically, this

matters: every healthy patient in the test sets was correctly cleared, meaning none would face unnecessary follow-up procedures. Under Gaussian noise at $\sigma = 0.20$, DeepQSVM degrades 2.75 pp less than PCA+QSVM in accuracy and 12.00 pp less in specificity; PCA+QSVM specificity effectively collapses to zero under noise. Table IV ablation indicates that the AQKS mechanism adds to CNN feature replacement alone.

Several limitations apply. Findings are based on one 400 sample dataset, and have not been generalised to larger and more multi-centre registries. No physical quantum processor was engaged in the classical simulation of all quantum circuits. Ablation involves fold-subset baselines instead of fixed-kernel models trained independently, and thus estimates of contributions are inaccurate. The initial noise sensitivity of 0.01 σ is further indicative of a gap that may be narrowed by noise-enhanced training. Future research will involve the multiplication of the study with bigger multi centre CKD datasets to enhance generalizability. Moreover, the AQKS will be expanded to process more types of kernels and provide multi-classification of the renal stage. There will also be the efforts of training and testing the model on both noisy and real-life clinical information so as to increase its robustness. The other important line of direction is to explore the concept of differential privacy in federated learning systems in order to make sure that GDPR is adhered to in a clinical facility. Last, additional validation will be done with noisy data to test the performance on realistic conditions.

REFERENCES

[1] V. Havlíček et al., "Supervised learning with quantum-enhanced feature spaces," Nature, vol. 567, pp. 209–212, 2019.
[2] M. A. Islam, S. Akter, and M. A. Rahman, "Early detection of chronic kidney disease using machine learning with XGBoost and PCA feature selection," in Proc. ICACECS, Atlantis Press, 2023, pp. 128–141.
[3] S. Chakravarty, A. Alam, and P. K. Kapur, "Chronic kidney disease classification using deep learning convolutional neural network," in Proc. ICACCCN, 2020, pp. 68–72.
[4] Y. Liu et al., "A rigorous and robust quantum speed-up in supervised machine learning," Nature Phys., vol. 17, pp. 1013–1017, 2021.
[5] J. Preskill, "Quantum computing in the NISQ era and beyond," Quantum, vol. 2, p. 79, 2018.
[6] A. Mari et al., "Transfer learning in hybrid classical-quantum neural networks," Quantum, vol. 4, p. 340, 2020.
[7] I. Cong et al., "Quantum convolutional neural networks," Nature Phys., vol. 15, pp. 1273–1278, 2019.
[8] A. Madry et al., "Towards deep learning models resistant to adversarial attacks," Proc. ICLR, 2018.
[9] D. A. Debal and T. M. Sitote, "Chronic kidney disease prediction using machine learning techniques," J. Big Data, vol. 9, no. 109, 2022.
[10] F. Chollet, Deep Learning with Python. Manning Publications, 2018.
[11] M. Dua and E. UCI Machine Learning Repository, "Chronic kidney disease dataset," UCI, 2015.
[12] I. H. Witten, E. Frank, M. Hall, and C. Pal, Data Mining: Practical Machine Learning Tools and Techniques, 4th ed. Morgan Kaufmann, 2016.
[13] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.
[14] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, Springer, 2009.
[15] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.
[16] V. Nidadavolu et al., "Prediction of Chronic Kidney Disease using Machine Learning Techniques," in Proc. ICACECS, 2023, pp. 128–141.
[17] J. R. Quinlan, "Induction of decision trees," Machine Learning, vol. 1, pp. 81–106, 1986.
[18] L. Breiman, "Random forests," Machine Learning, vol. 45, pp. 5–32, 2001.
[19] H. Mohamadlou, A. Lynn-Palevsky, and D. Barton, "Predicting chronic kidney disease using machine learning algorithms," Healthcare Informatics Research, vol. 24, no. 4, pp. 273–281, 2018.
[20] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," Physical Review Letters, vol. 113, 2014.
[21] M. M. Hossain et al., "Performance analysis of classical and quantum support vector machines for diagnosis of chronic kidney disease," Informatics and Health, 2025. DOI: 10.1016/j.infoh.2025.08.003
[22] M. Schuld and N. Killoran, "Quantum machine learning in feature Hilbert spaces," Physical Review Letters, vol. 122, no. 4, 2019.