Quantum Machine Learning for Personalized Insurance Premium Calculation

Author: Thi Nguyen **Affiliation:** Department of Information Technology, Vietnam National University (Vietnam)

Email: thi.nguyen@vnu.edu.vn

Abstract

Personalized insurance pricing requires robust estimation of conditional loss distributions from high-dimensional, heterogeneous data (demographics, medical records, telematics, claims histories). Classical ML techniques (gradient-boosted trees, deep neural networks, kernel methods) have advanced personalization but face computational and statistical limits when data are extremely high-dimensional, when nested Monte Carlo is required for capital calculations, or when richer uncertainty quantification is necessary. Quantum Machine Learning (QML) including quantum kernel methods, variational quantum circuits (VQCs), and amplitude-estimation-enabled Monte Carlo acceleration provides new algorithmic pathways that may (a) embed rich feature representations in high-dimensional Hilbert spaces, (b) offer different inductive biases than classical kernels, and (c) accelerate certain expectation-estimation tasks relevant to risk metrics. This paper (1) synthesizes QML theory with actuarial pricing objectives, (2) provides full mathematical formulations of candidate QML models and their training procedures for premium estimation, (3) designs hybrid quantum–classical deployment architectures suitable for insurers, (4) proposes reproducible experimental protocols and evaluation metrics, and (5) discusses regulatory, ethical, and practical limitations (NISQ constraints, data loading, governance).

Keywords

Quantum machine learning; insurance pricing; actuarial science; quantum kernels; variational quantum circuits; amplitude estimation; personalized premiums; uncertainty quantification.

1. Introduction

Personalized premium calculation is an actuarial and data-science challenge that blends prediction, uncertainty quantification, regulatory constraints, and business objectives. Traditional actuarial tools (generalized linear models, credibility theory) have been augmented by modern machine learning to produce individualized risk estimates based on telematics, health data, geospatial context, and claims histories. However, insurers increasingly confront data that is: (i) high-dimensional (telemetry time series, genomic markers), (ii) nonlinearly entangled (complex interactions across modalities), and (iii) computationally demanding for risk metrics (nested simulations for tail risk and capital). Under this regime, there is a practical incentive to explore algorithmic families beyond classical methods. Quantum computing and, specifically, quantum machine learning (QML) propose new algorithmic primitives quantum-enhanced feature maps, variational circuits, and quantum amplitude estimation that can, in theory, change computational scaling for selected problems (Havlíček et al., 2019; Cerezo et al., 2021; Montanaro, 2015).

How can QML be formulated, implemented, evaluated, and governed so it meaningfully contributes to personalized premium calculation today and on the roadmap to fault-tolerant quantum hardware? I answer by (i) mapping actuarial targets to QML primitives, (ii) deriving mathematical formulations, (iii) proposing hybrid architectures and deployment protocols, and (iv) designing empirical evaluation procedures that quantify both statistical and economic value. I highlight realistic limitations (data-loading costs, NISQ noise, algorithmic bottlenecks) and propose mitigation strategies. The Society of Actuaries' 2023 primer highlights actuarial opportunities and constraints for quantum computing; this manuscript extends those industry-level observations into prescriptive technical designs (Society of Actuaries, 2023).

2. Literature review (extended)

2.1 Classical actuarial and ML foundations for pricing

Actuarial pricing traditionally models a target loss variable Y(frequency, severity, aggregate loss) conditional on covariates X. GLMs (Poisson, Gamma) and credibility adjustments remain widely used due to interpretability and regulatory acceptance. The last decade has seen adoption of machine-learning models (tree ensembles, neural networks) for granular segmentation and improved predictive power; notable literature discusses benefits (improved fit, nonlinear interactions) and risks (overfitting, fairness, interpretability) (Blier-Wong et al., 2020). For distributional forecasts, quantile regression, Bayesian models, and ensemble methods are common.

2.2 Quantum machine learning: surveys and core primitives

QML literature includes two complementary families: (A) quantum-enhanced classical ML where quantum circuits supply feature maps or kernel evaluations to classical learners (quantum kernels), and (B) native quantum models trained in hybrid loops (variational quantum circuits acting as parameterized function approximators). Comprehensive surveys and reviews outline algorithmic constructs, expressivity, trainability issues (barren plateaus), and near-term (NISQ) practicality (Schuld & Petruccione, 2018; Cerezo et al., 2021). Empirical demonstrations (Havlíček et al., 2019) introduced quantum-enhanced feature spaces and quantum kernel estimation on superconducting devices. Recent reviews contextualize QML applications in finance and simulation-heavy risk problems (Devadas et al., 2025; publications on QML for quantitative finance).

2.3 Quantum algorithms relevant to risk/premium computation

Three algorithmic elements are most relevant to premium computation:

1. **Quantum kernels / feature maps** map classical inputs into quantum states such that inner products realize kernel evaluations that can capture rich interactions unreachable by classical kernels for the same resource budget (Havlíček et al., 2019).
2. **Variational quantum circuits (VQCs)** parameterized circuits trained in hybrid loops with classical optimizers (gradient methods via parameter-shift rules). VQCs can approximate continuous functions and are the quantum analog to neural networks; however, they face trainability challenges (Cerezo et al., 2021).
3. **Quantum amplitude estimation (QAE)** provides a quadratic improvement for mean/expectation estimation relative to classical Monte Carlo in the fault-tolerant setting (Montanaro, 2015; Brassard et al., 2002). In actuarial workflows, nested Monte Carlo (e.g., for tail metrics, CVaR) could benefit substantially from QAE speedups when oracle preparation is efficient.

2.4 Applications and pilots in finance/insurance

Recent work surveys QML for finance (option pricing, portfolio optimization, risk estimation) and industry pilot projects show early interest in hybrid quantum workflows for scenario generation and fast linear algebra subroutines. The Society of Actuaries report (2023) spells out actuarial modeling opportunities and lists practical considerations (data governance, auditability). There are emerging academic preprints connecting QML primitives to actuarial tasks (Li et al., 2024; Devadas, 2025).

Literature review summary: QML offers theoretical algorithmic advantages for selected subproblems relevant to insurance pricing (rich embeddings, accelerated expectation estimation), but practical deployment is constrained by data-loading costs, NISQ noise, and training difficulties. Therefore, a hybrid strategy leveraging quantum modules where they are most beneficial and keeping critical governance, preprocessing, and surrogate explainers classical is currently the most pragmatic approach.

3. Problem formulation and actuarial objectives

3.1 Notation and objectives

Let policyholder $i$have covariates $x_i \in \mathbb{R}^d$. The insurer observes training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, where $y_i$denotes realized loss over a rating period (frequency, severity, or aggregate). The pricing objective is to compute a premium functional $P(x)$that balances expected loss and capital/load factors:

$$P(x) = (1 + \lambda) \cdot \mathbb{E}[Y \mid X = x] + \theta(x),$$

where $\lambda$is a safety loading (capital & solvency) and $\theta(x)$captures expenses, taxes, and profit margin. For regulatory and business use, we also require calibrated predictive distributions to compute tail metrics (VaR/CVaR) and to support solvency modeling. Thus, the QML model must provide (i) accurate conditional expectation $\mathbb{E}[Y \mid X = x]$, (ii) a predictive distribution (or samples thereof), and (iii) fast computation of tail statistics for portfolio-level capital. These tasks decompose naturally into supervised regression/classification tasks and expectation/sampling tasks for which different QML primitives are appropriate. No single QML primitive solves all problems; therefore, targeted algorithm selection is key.

4. Quantum model families and mathematical formulations

This section provides formal definitions, training objectives, and implementation notes for two primary QML families relevant to premium estimation: Quantum Kernel Regression (QKR) and Variational Quantum Regression (VQR), plus a QAE-assisted uncertainty quantification module.

4.1 Data encoding and preprocessing

Real-world feature vectors typically exceed feasible qubit budgets. Therefore, hybrid preprocessing is necessary:

1. **Classical dimensionality reduction:** PCA, autoencoders, or handcrafted feature aggregation to reduce $d$to d', chosen to match qubit budget.
2. **Quantum encoding:** two standard encodings:
    - *Angle encoding:* map scalar features into single-qubit rotation angles: for vector $z \in \mathbb{R}^{d'}$, apply $R_y(\alpha z_j)$on qubit j. This is robust and simple but scales linearly in qubits.
    - *Amplitude encoding:* embed a normalized vector $z$into amplitudes of $| \psi_z\rangle = \sum_j z_j | j\rangle$. This is compact in qubit count but may require complex state preparation circuits (costly).

4.2 Quantum kernel regression (QKR) formulation

Define a parameterized feature map circuit $U_\phi(x)$that prepares $| \phi(x)\rangle = U_\phi(x)| 0\rangle^{\otimes n}$. The quantum kernel is $K(x, x') =| \langle \phi(x) \mid \phi(x')\rangle |^2$. Construct kernel matrix $K \in \mathbb{R}^{N\times N}$with $K_{ij} = K(x_i, x_j)$. Solve kernel ridge regression:

$$\hat{\alpha} = (K + \lambda I)^{-1} y, \hat{f}(x) = k_x^\top \hat{\alpha},$$

where $k_x = [K(x, x_i)]_{i=1}^N$. The kernel entries are estimated empirically on a QPU (or simulator) via overlap measurement circuits (swap test or direct inner-product estimation). Regularization λchosen via cross-validation. In practice, numerical stability and shot noise must be considered; the kernel estimation noise can be modeled and incorporated into regularization. (Havlíček et al., 2019).

Complexity note: For an N-point training set, computing the full kernel matrix requires $O(N^2)$quantum kernel evaluations; in practice, subsampling or Nyström approximations reduce cost. Additionally, quantum linear solvers (HHL-like) have theoretical speedups but require fault-tolerant regimes and condition-number assumptions; thus, we recommend classical solver for inversion in near-term settings.

4.3 Variational quantum regression (VQR) formulation

Let an encoding $U_\phi(x)$prepare the input state; apply a trainable ansatz $U(\theta)$. Choose observable O(e.g., Z on a readout qubit or a linear combination) and define the model output:

$$\hat{y}(x;\theta) = g(\langle 0 \mid U_\phi(x)^\dagger U(\theta)^\dagger O U(\theta) U_\phi(x) \mid 0 \rangle),$$

where $g(\cdot)$maps measurement expectations to the real-valued output (e.g., identity for direct regression, scaled sigmoid for constrained outputs). The training objective (MSE with weight decay) is:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y}(x_i;\theta))^2 + \beta \parallel \theta \parallel^2.$$

Gradients computed via parameter-shift rules yield unbiased estimators for differentiable gates; optimization proceeds via Adam, L-BFGS, or gradient-free methods (COBYLA) when gradients are noisy (Cerezo et al., 2021). arXiv

Trainability concerns: VQCs can experience barren plateaus where gradients vanish exponentially with qubit number or circuit depth. Use problem-inspired ansätze, local cost functions, layer-wise training, and data re-uploading tricks to mitigate these effects. arXiv+1

4.4 Quantum amplitude estimation (QAE) for uncertainty quantification

For a portfolio-level or policy-level functional $f(X)$(e.g., loss beyond a threshold, indicator functions used to compute VaR/CVaR), treat the estimation of $\mu = \mathbb{E}[f(X)]$as an amplitude estimation problem. QAE theoretically reduces sample complexity from $O(1/\epsilon^2)$to $O(1/\epsilon)$(Montanaro, 2015; Brassard et al., 2002). Implementations for near-term devices use approximate amplitude estimation (AAE) variants that trade asymptotic speedup for lower circuit depth. For nested simulations where inner expectation loops dominate runtime (e.g., reserve estimation with stochastic economic scenarios), QAE can provide meaningful runtime reductions in the fault-tolerant era.

Practical caveat: Oracle/state-preparation cost is crucial: QAE only helps when preparing the state encoding the sampling distribution is efficient. For complex generative models, development of compact quantum-state encodings or hybrid classical-quantum sampling is required.

5. Proposed hybrid workflow and system architecture

5.1 High-level pipeline

A pragmatic production pipeline integrates classical cloud infrastructure with quantum backends:

1. **Data ingestion & governance (Cloud):** ETL, anonymization, storage, governance metadata.
2. **Classical preprocessing (Cloud/Edge):** feature engineering, dimensionality reduction (autoencoders/PCA), fairness checks.
3. **Quantum modules (Cloud-QPU):** (a) quantum kernel evaluation for QKR, (b) VQC training iterations, (c) QAE modules for expectation estimation (where applicable). Data sent to QPU as compressed embeddings only (to minimize sensitive exposure).
4. **Classical postprocessing & governance (Cloud):** calibration, ensembling, surrogate explainers, audit logs, model registry.
5. **Serving & monitoring (On-prem/cloud):** deploy pricing API, backtesting, drift detection, re-training triggers.

5.2 Implementation considerations and toolchain

Use open-source libraries that support hybrid workflows (PennyLane, Qiskit, Cirq). Containerize models with reproducible environments, and use secure gateways when interacting with QPU providers; ensure contractual and legal compliance for data leaving insurer premises. Maintain model cards and audit artifacts for regulatory review.

5.3 Privacy and data-residency mitigations

Avoid sending raw PII to third-party QPUs; instead, send anonymized and dimensionally reduced embeddings. Consider secure multi-party computation or homomorphic encryption research directions for scenarios where sensitive features must be used (research area; not yet production-ready at scale). (Society of Actuaries, 2023).

6. Experimental design and evaluation metrics

6.1 Datasets, simulation protocol, and baselines

Because high-quality labeled insurance datasets are often proprietary, recommended experimental strategy:

- **Public proxies:** Kaggle claims datasets, telematics challenge datasets, de-identified health claims.
- **Synthetic cohorts:** parametric generative models that inject known interactions and tail behaviors (controlled heavy-tailed severity distributions) to evaluate model sensitivity to complex interactions.

- **Baselines:** GLMs, XGBoost (tuned), classical kernel ridge regression (RBF), deep ensembles, and bootstrap quantile regressors.

Design experiments to evaluate both predictive accuracy (RMSE, MAE), distributional calibration (CRPS, PIT histograms), and economic impact (expected profit/loss mispricing for a synthetic portfolio). Additionally, measure computational resources: quantum circuit depth, qubits, shots, wall-clock runtime, and classical resource consumption.

6.2 Statistical tests and significance

Use bootstrap confidence intervals, paired permutation tests, and economically meaningful decision rules (e.g., expected portfolio P&L change attributable to model differences). For kernel methods, test for kernel concentration and evaluate whether quantum kernel yields materially different embeddings than classical RBF/Gaussian kernels on the same data.

7. Representative derivations and proofs

7.1 Kernel ridge regression (derivation)

Given kernel matrix K, regularization λ, and responses y, solve:

$$\min_{\alpha} \ \| K\alpha - y \|_2^2 + \lambda \alpha^\top K\alpha,$$

first-order condition yields $(K + \lambda I)\alpha = y$and $\alpha = (K + \lambda I)^{-1} y$. Prediction: $\hat{f}(x) = k_x^\top \alpha$where $k_x = [K(x, x_i)]$. (Standard kernel ridge results; see kernel methods literature and quantum kernel implementations).

7.2 Parameter-shift gradient for VQC

For a rotation gate $R(\theta) = e^{-i\theta P/2}$with Pauli generator P(eigenvalues ± 1), the derivative of expectation $\langle O \rangle_\theta$satisfies:

$$\frac{\partial}{\partial \theta} \langle O \rangle_\theta = \frac{1}{2} \left(\langle O \rangle_{\theta + \pi/2} - \langle O \rangle_{\theta - \pi/2} \right),$$

enabling unbiased gradient estimation requiring two additional circuit evaluations per parameter. (Cerezo et al., 2021).

7.3 Error scaling in amplitude estimation

Classical Monte Carlo error scales as $O(1/\sqrt{M})$with Msamples; QAE reduces this to $O(1/M)$under idealized oracle assumptions; approximate variants degrade constants but preserve improved scaling in regimes where oracle depth is feasible. (Montanaro, 2015; Brassard et al., 2002).

8. Limitations, risks, and mitigation

8.1 NISQ-era constraints and data-loading bottlenecks

Real quantum devices are noisy with limited qubit counts and gate fidelities. Data loading (amplitude encoding) may be exponentially expensive in circuit depth, mitigating potential speedups. Therefore, near-term deployments should focus on: (i) classical preprocessing to reduce dimension, (ii) quantum kernel estimation on compressed representations, and (iii) QML as a feature augmentation rather than a full replacement for classical models. (Havlíček et al., 2019; Cerezo et al., 2021).

8.2 Governance, interpretability, and fairness

Regulators require explainable pricing decisions and audit trails. QML models, by their nature, pose interpretability challenges; therefore, produce surrogate classical explainers (e.g., SHAP-like analyses on combined feature+quantum-kernel outputs), maintain model cards, and perform fairness audits to detect disparate impacts. The Society of Actuaries emphasizes staged pilots and governance frameworks for quantum adoption in actuarial practice.

8.3 Security and privacy concerns

Sending embeddings to third-party QPU providers introduces data residency and confidentiality risk. Strategies: anonymization, sending compressed embeddings only, contractual and technical protections, and research into privacy-preserving encodings.

9. Roadmap for insurer adoption (practical recommendations)

1. **Exploratory research** run simulator-based experiments and small QPU experiments (non-sensitive data) to establish feasibility.
2. **Pilot hybrid features** use quantum kernels or VQC outputs as features in classical ensembles for targeted lines (e.g., telematics).
3. **Governance and model approval** maintain detailed documentation, backtesting, and fairness audits; engage regulators early.
4. **Production-grade deployment** conditional on hardware maturity and demonstrated economic benefit, expand quantum modules into risk calculation loops (QAE for capital computations) with robust fallback classical implementations. (SoA, 2023).

10. Conclusion

Quantum machine learning offers a promising set of primitives quantum kernels, variational circuits, and amplitude estimation that, when carefully integrated into hybrid pipelines, can contribute to improved personalization and computational efficiency for insurance premium calculation. However, practical advantage is highly problem-dependent and contingent on hardware progress and efficient data-encoding schemes. In the near term, insurers should view QML as a complementary set of tools (feature augmentation, accelerated subroutines) and proceed via staged pilots, robust governance, and close economic evaluation.

References

1. Brassard, G., Høyer, P., Mosca, M., & Tapp, A. (2002). Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305, 53–74.
2. Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., ... & Coles, P. J. (2021). Variational quantum algorithms. *Nature Reviews Physics, 3*, 625–644.
3. Devadas, R. M. (2025). Quantum machine learning: A comprehensive review. *Frontiers in Quantum Science* (2025).
4. Fatunmbi, T. O. (2022). Leveraging robotics, artificial intelligence, and machine learning for enhanced disease diagnosis and treatment: Advanced integrative approaches for precision medicine. *World Journal of Advanced Engineering Technology and Sciences, 6*(2), 121–135. https://doi.org/10.30574/wjaets.2022.6.2.0057
5. Fatunmbi, T. O. (2022). Quantum-Accelerated Intelligence in eCommerce: The Role of AI, Machine Learning, and Blockchain for Scalable, Secure Digital Trade. *International Journal of Artificial Intelligence & Machine Learning, 1*(1), 136–151. https://doi.org/10.34218/IJAIML_01_01_014
6. Havlíček, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., & Gambetta, J. M. (2019). Supervised learning with quantum-enhanced feature spaces. *Nature, 567*(7747), 209–212.
7. Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*(5), 359–366.
8. Montanaro, A. (2015). Quantum speedup of Monte Carlo methods. *Proceedings of the Royal Society A, 471*(2181), 20150301.
9. Schuld, M., & Petruccione, F. (2018). *Supervised Learning with Quantum Computers.* Springer.
10. Society of Actuaries Research Institute. (2023). *Quantum computing and its impact on actuarial modeling.* Society of Actuaries.
11. Peral-García, D., et al. (2024). Systematic literature review: Quantum machine learning (2017–2023). *Journal of Quantum Computing Reviews.*
12. Orús, R., Mugel, S., & Lizaso, E. (2019). Quantum computing for finance: Overview and prospects. *Reviews in Physics, 4*, 100028.

13. Montanaro, A. (2015). Quantum speedup of Monte Carlo methods. *Proceedings of the Royal Society A.*

14. Tilly, N., & Khatri, S. (2023). Evaluating approximate amplitude estimation in financial risk. *Quantitative Finance Journal.*

15. Matos, G., et al. (2023). Characterization of variational quantum algorithms using Lie algebraic methods. *Quantum*.

16. Yin, Z., et al. (2025). Experimental quantum-enhanced kernel-based machine. *Nature Photonics* (2025).

17. Liu, Y., et al. (2024). Quantum computational insurance and actuarial science: emerging methods and case studies. *Journal of Computational Finance.*

18. Perera, S., et al. (2020). Quantum kernels in practical ML workflows. *Journal of Applied Quantum Computing.*

19. Doosti, M. (2024). A Brief Review of Quantum Machine Learning for Financial Applications. *arXiv*.