SQA Undergraduate Research Projects 2026

The projects are intentionally designed for students who are new to quantum computing and quantum machine learning, and who will work primarily by executing, analysing, and extending provided Python and Jupyter Notebook code. Rather than focusing on software engineering or algorithm invention, the projects emphasise conceptual understanding, experimental reasoning, and disciplined interpretation of results produced by quantum simulation workflows.

Both projects operate within the application domain of computer networking and cybersecurity, but they approach the problem of understanding network behaviour from two complementary perspectives. One project addresses the problem of identifying abnormal behaviour in traffic streams through supervised classification, while the other addresses the problem of diagnosing likely root causes of congestion through probabilistic inference. Together, they expose students to two major paradigms of quantum computing: variational hybrid models and state based probabilistic reasoning.

Quantum-Based Anomaly Detection in Network Traffic Using Qiskit Variational Quantum Classifier

This project applies a Variational Quantum Classifier to the task of detecting anomalous network traffic patterns. The provided code generates a synthetic dataset that mimics simplified network telemetry collected over fixed time intervals. Each data record represents a one second snapshot of network activity and contains numerical features corresponding to packet rate, byte throughput, packet drop ratio, and a protocol identifier. These features are selected because they reflect common indicators used in operational network monitoring systems.

The synthetic dataset is constructed to include both normal traffic patterns and anomalous patterns. Normal samples reflect stable traffic behaviour, while anomalous samples are generated to exhibit characteristics such as sudden traffic bursts, abrupt changes in protocol usage, or persistently elevated packet drop rates. The dataset is balanced between the two classes to simplify interpretation of classifier behaviour and evaluation metrics.

Before any quantum processing occurs, the data undergoes classical preprocessing. Feature scaling is applied to ensure that all numerical inputs fall within bounded ranges suitable for use as rotation angles in quantum gates. This step is critical because quantum feature maps encode classical values as gate parameters, and unscaled or poorly scaled features can result in unstable circuit behaviour, poor optimisation performance, or meaningless quantum state representations. Students are expected to observe how scaling choices affect training convergence and classification outcomes.

Classical preprocessing also includes dataset splitting into training and testing subsets. Students should understand why separating data in this way is necessary to evaluate generalisation performance, and how leakage between training and testing data can lead to misleading results. Although the dataset is synthetic, the same principles apply as in real world machine learning workflows.

Quantum feature encoding is implemented using a ZZFeatureMap. This feature map applies parameterised single qubit rotations followed by entangling ZZ interactions between qubits. Through these operations, correlations between pairs of input features are embedded into the quantum state. The resulting state resides in a Hilbert space whose dimension grows exponentially with the number of qubits, allowing the representation of nonlinear relationships that may be difficult to capture with simple classical models.

Students should examine the structure of the feature map circuit and understand how the number of qubits relates directly to the number of input features. They should also observe how increasing the depth of the feature map affects circuit depth, execution time on simulators, and the expressive capacity of the model.

The classification model itself is implemented using a parameterised quantum circuit based on a RealAmplitudes ansatz. This ansatz consists of layers of rotation gates applied to each qubit, interleaved with entangling gates. Each rotation gate contains a tunable parameter that is adjusted during training. The ansatz determines the space of quantum states that the model can explore during optimisation.

Training is performed using a hybrid quantum classical optimisation loop. At each iteration, the quantum circuit is executed on a simulator to estimate expectation values, and a classical optimiser updates the circuit parameters to reduce a loss function derived from classification error. Students are expected to understand this alternating process and to recognise that the quantum circuit acts as a nonlinear model embedded within a classical optimisation routine.

During training, students will monitor loss values and observe convergence behaviour across iterations. They should compare runs using different random seeds or optimisation settings and comment on the stability and repeatability of results. The focus is on recognising qualitative patterns rather than maximising performance.

Model evaluation uses standard classification metrics such as accuracy and confusion matrices. Students must interpret false positives and false negatives in the context of network security, explaining what types of traffic patterns are incorrectly classified and why this may occur given the limited feature set and model capacity. They should also discuss the implications of misclassification in a practical monitoring context.

By the end of this project, students should be able to clearly explain the full pipeline from raw network data through classical preprocessing, quantum feature encoding, variational training, and final performance evaluation, using precise technical language.

At the circuit execution level, students should understand how measurement outcomes are converted into classical information used by the optimiser. The variational quantum circuit is executed multiple times per iteration, either through exact statevector evaluation or finite shot sampling, to estimate expectation values associated with the classifier output. These expectation values form the basis of the loss computation, linking quantum measurements directly to classical optimisation.

An important technical consideration is circuit depth and parameter count. As the number of qubits or ansatz layers increases, the number of trainable parameters grows, increasing expressiveness but also making optimisation more challenging. Students should observe how deeper circuits increase runtime and may introduce optimisation instability, even on simulators, and should be able to articulate this trade off clearly.

Students are also expected to examine how noise free simulation differs conceptually from execution on real quantum hardware. Although these projects use simulators, students should note that real devices introduce decoherence, gate errors, and measurement noise, which would further complicate training. This distinction is important when interpreting results and assessing claims about real world applicability.

Another key technical aspect is the role of the feature map in defining the geometry of the classification problem. The feature map determines how distances and similarities in classical feature space are represented in quantum state space. Students should reflect on how different encodings might emphasise or suppress certain feature relationships and how this impacts separability between normal and anomalous traffic.

Finally, students should be able to explain the end to end data flow in precise terms, from raw numerical inputs through scaled features, encoded quantum states, variational transformations, measurement, loss evaluation, and final classification decisions. Mastery of this pipeline description is a core learning outcome of the project.

At the data representation level, students should understand that the anomaly detection task is fundamentally a problem of separating overlapping distributions in a low dimensional feature space. Even in the synthetic dataset, normal and anomalous traffic are not perfectly separable, which means the classifier must learn a decision boundary that trades off false positives and false negatives. Students should be able to explain how this ambiguity appears in the confusion matrix and why perfect accuracy is neither expected nor realistic.

From a quantum modelling perspective, the feature map plays a central role in shaping the geometry of the classification problem. The ZZFeatureMap embeds classical feature vectors into quantum states using entangling operations that encode pairwise correlations. Students should understand that this embedding effectively transforms distances and angles in classical feature space into overlaps between quantum states. When two input samples are mapped to quantum states with high overlap, the classifier will find it difficult to distinguish them regardless of ansatz depth.

The variational ansatz defines the family of decision boundaries that the classifier can represent. A shallow ansatz restricts the model to relatively simple transformations of the encoded states, while a deeper ansatz increases expressive capacity at the cost of optimisation difficulty. Students should be able to explain why increasing circuit depth does not guarantee improved performance and may lead to unstable or inconsistent training behaviour even on simulators.

The hybrid training loop should be understood as a composition of two processes operating at different levels. The quantum circuit produces expectation values that depend nonlinearly on circuit parameters, while the classical optimiser treats these values as outputs of a black box function. Students should recognise that optimisation difficulties such as plateaus or slow convergence arise from the structure of this function rather than from coding errors.

Students should also understand the role of random initialisation. Different initial parameter values can lead to different optimisation trajectories and different local minima. When comparing runs, students should focus on qualitative trends across experiments rather than isolated numerical results.

Quantum-Based Diagnosis of Network Congestion Using Statevector Inference

This project addresses the problem of network congestion diagnosis by modelling it as a probabilistic reasoning task rather than a supervised learning task. Instead of learning from labelled examples, the project constructs a quantum circuit that explicitly represents possible network conditions and their relationships.

The network is abstracted into a finite set of binary conditions such as traffic spikes, high link utilisation, queue overflows, packet drops, and alert signals. Each condition is represented by a qubit, where the logical state of the qubit corresponds to the presence or absence of that condition. This representation allows all possible combinations of network conditions to be represented simultaneously within a quantum superposition.

Dependencies between network conditions are encoded using controlled quantum gates. These gates represent causal or correlational relationships, for example how increased utilisation may lead to queue buildup, or how packet drops may increase the likelihood of alerts. The structure of the circuit therefore defines a joint probability model over all possible network states.

Students should study the circuit structure carefully and understand how individual gates contribute to the overall dependency model. They should also understand how modifying or removing gates changes the implied relationships between conditions and therefore alters the resulting probability distribution.

The circuit is simulated using the statevector formalism. Unlike sampling based simulation, the statevector approach computes the exact quantum state, providing direct access to complex probability amplitudes for every basis state. The squared magnitude of each amplitude corresponds to the probability of the associated network condition combination.

Students will extract probabilities from the statevector and organise them into interpretable forms such as tables or plots. They will learn how to map between binary qubit states and semantic network conditions, and how to interpret probability mass distributed across many possible states.

Diagnosis is performed by conditioning the probability distribution on observed evidence. This involves selecting basis states consistent with observed symptoms and renormalising probabilities to compute conditional distributions over unobserved conditions. Students should understand this process as a direct implementation of conditional probability using quantum state representations.

By comparing multiple diagnostic scenarios, students will observe how adding or removing evidence redistributes probability mass and changes inferred root causes. They should be able to explain these shifts using both probabilistic reasoning and the structure of the underlying circuit.

This project does not involve training, optimisation, or data fitting. Instead, emphasis is placed on circuit construction, probability interpretation, and logical reasoning under uncertainty. Students must articulate how the simplifying assumptions of the model affect realism and generalisability.

From a mathematical perspective, the statevector represents a complete joint probability distribution over all binary network conditions encoded by the qubits. Each basis state corresponds to a unique combination of conditions, and the squared magnitude of its amplitude gives the probability of that configuration occurring within the model. Students should understand how this differs from marginal or conditional views commonly used in classical diagnostics.

Students should pay particular attention to the interpretation of low probability states. Even configurations with small probability mass can be informative when conditioned on evidence, and students should avoid focusing only on the single most probable outcome. Instead, they should reason about probability distributions and relative likelihoods across multiple competing explanations.

Another technical consideration is how evidence is applied during conditioning. Conditioning does not change the underlying circuit, but instead filters the probability distribution after simulation. Students should understand this as an inference step rather than a physical operation, and be able to explain the distinction clearly.

The structure of the quantum circuit implicitly encodes assumptions about causality and dependence. Students should be able to identify these assumptions by inspecting the circuit and discuss how changing or removing specific gates would alter diagnostic conclusions. This encourages careful thinking about model design and domain knowledge rather than blind execution.

As a final technical outcome, students should be able to describe the full diagnostic workflow, from selecting a set of observed symptoms, extracting conditional probabilities, comparing alternative root causes, and justifying conclusions using both numerical results and the structure of the circuit itself.

At a representational level, this project models the network as a discrete probabilistic system rather than a continuous or data driven one. Each qubit corresponds to a binary variable, and the full quantum state represents a joint distribution over all variables simultaneously. Students should understand that this representation is exhaustive, meaning every possible combination of network conditions is explicitly present in the model, even if its probability is small.

The quantum circuit encodes dependencies through gate structure rather than learned parameters. Controlled operations introduce correlations between qubits that correspond to assumed causal or conditional relationships in the network. Students should be able to identify which dependencies are hard coded into the circuit and explain how these assumptions influence inference outcomes.

Statevector simulation provides access to exact probabilities, but it also imposes exponential scaling with the number of qubits. Students should understand that this approach is feasible only for small systems and that the project intentionally uses a limited number of conditions to preserve interpretability. This constraint is a modelling choice rather than a limitation of the conceptual approach.

Conditional inference is performed entirely at the level of probability post processing. Students should understand that conditioning does not modify the quantum state physically, but instead restricts attention to a subset of basis states consistent with observed evidence. Renormalisation then produces conditional probabilities that can be interpreted using standard probabilistic reasoning.

Students should be encouraged to reason about probability mass rather than single outcomes. In many scenarios, multiple root causes may retain non negligible probability even after conditioning. Students should explain why this reflects genuine diagnostic uncertainty rather than model failure, and how such uncertainty would be handled in real operational contexts.

Finally, students should be able to compare this inference based approach with classification based approaches conceptually. While classification produces a single label, inference produces a distribution over explanations. Students should articulate the advantages and limitations of each approach and explain why both are relevant in network security and operations.

Work Practices and Outputs

For both projects, students are expected to run all notebooks locally using Anaconda and Jupyter, ensure results are reproducible, and maintain a clear experimental log. They should be able to explain what each major code section does, why parameters are chosen, and how outputs relate to underlying quantum and networking concepts.

Final outputs include fully runnable notebooks, generated figures and tables, and a written report describing the experimental setup, parameter settings, results, interpretation, limitations, and possible extensions. Assessment focuses on clarity of explanation, correctness of interpretation, and demonstrated conceptual understanding rather than software complexity or performance.