Hardware Assisted and
Blockchain Empowered Security Lab

QUANTUM TECHNOLOGIES FOR ANOMALY DETECTION IN FUTURE NETWORKS

Christian Esposito
University of Salerno
esposito@unisa.it

Conference ...

Introduction

Intrusion Detection Systems (**IDS**) play a critical role in ensuring security.

Anomaly detection has proven to be an effective approach for intrusion detection, establishing a baseline of normal system.

ML has emerged as a powerful tool for both, enabling systems to automatically classify network traffic as normal or malicious.

Introduction

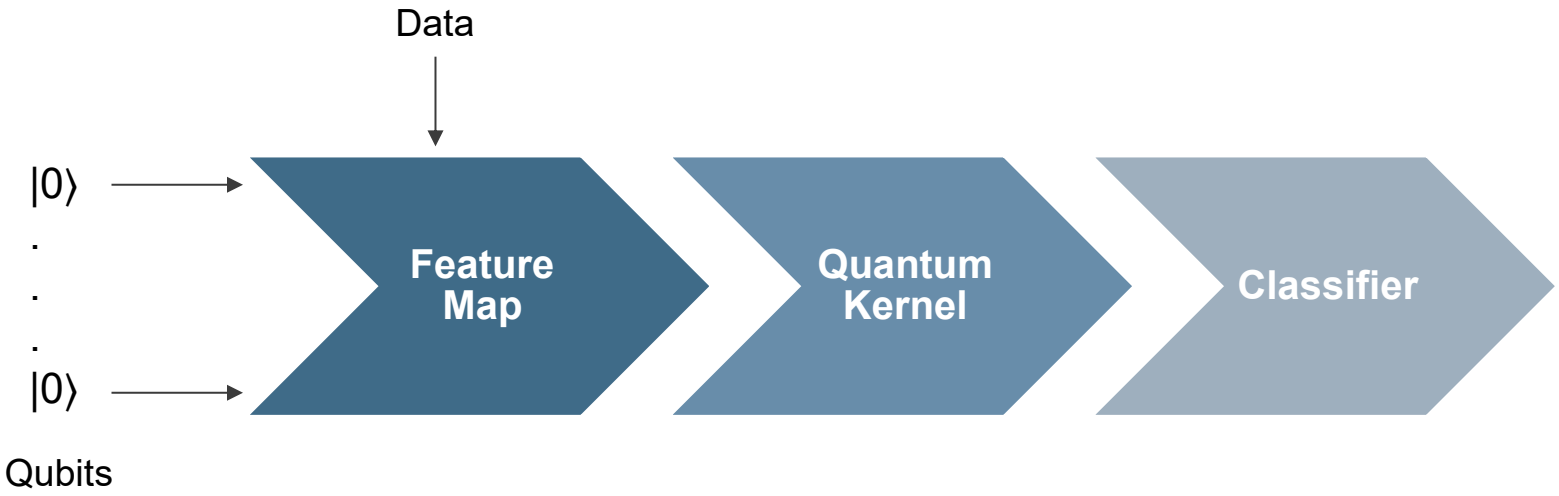
Quantum Machine Learning (**QML**) has the potential to outperform classical methods.

QML can handle complex, high-dimensional data more efficiently, leading to **faster training times** .

Quantum Support Vector Machines (**QSVM**), Variational Quantum Classifiers (**VQC**), and **hybrid** quantum-classical **neural networks**

QSVM

Similarly to SVM, the **kernel trick** is used to map data into a higher-dimensional space after transforming the data into a **quantum feature space**.



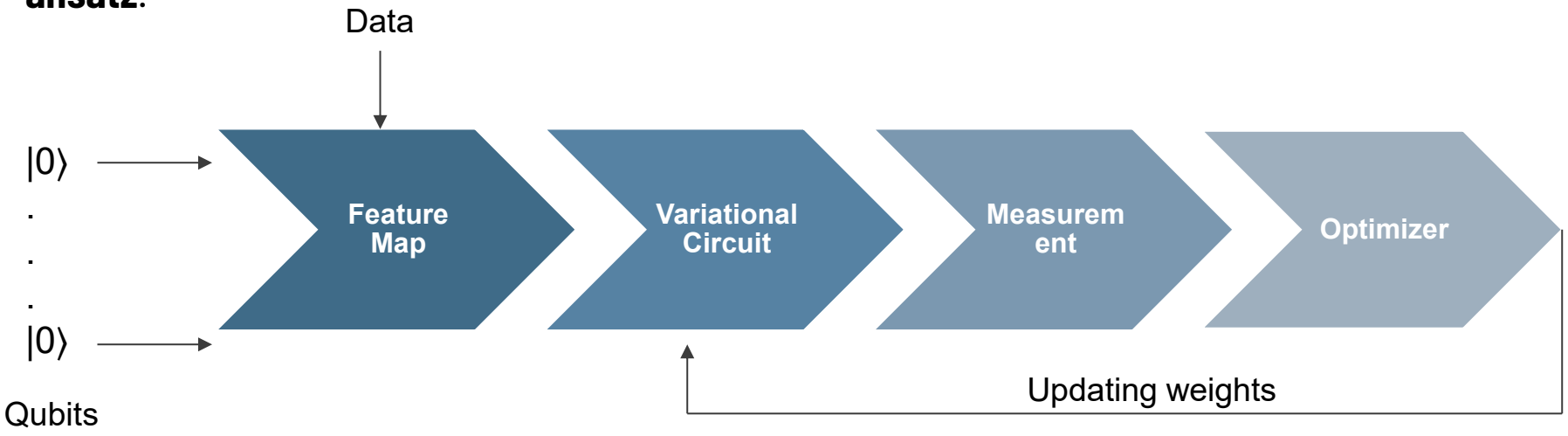
Pegasos-QSVC

Inspired by the Pegasos algorithm (**Primal Estimated sub-GrAdient SOlver for SVM**), it employs a **stochastic gradient descent** approach to solve the primal optimization problem of SVMs.

Pegasos updates model parameters using only a small subset of the data, thereby **reducing computational costs** and ensuring training complexity is independent of the training set size.

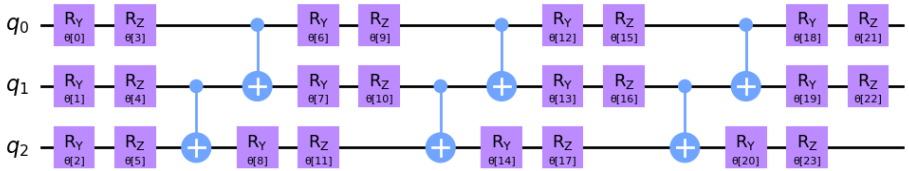
VQC

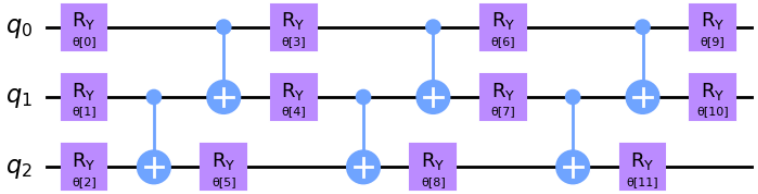
It employs a parameterized quantum circuit trained with **classical optimization methods** to perform classification tasks.
As a type of variational quantum algorithm (VQA), the VQC depends heavily on the choice of **ansatz**.

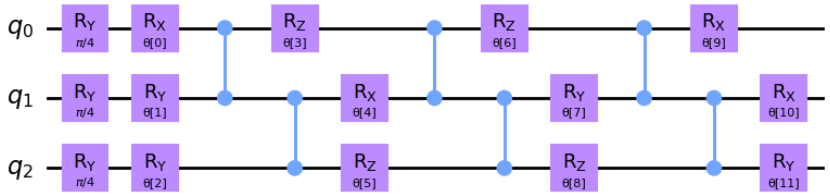


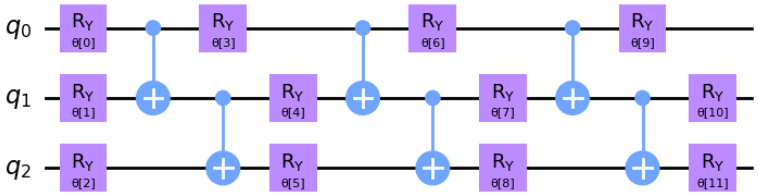
Ansatz

Two-Local, Pauli Two-Design, Real Amplitudes, and EfficientSU2, each designed with distinct features and trade-offs suited to various quantum computing problems.









Introduction

ML's **reliance on quality datasets** can limit its scalability and applicability.

A promising advancements in ML for intrusion detection is the application of **Generative Adversarial Networks** (GANs).

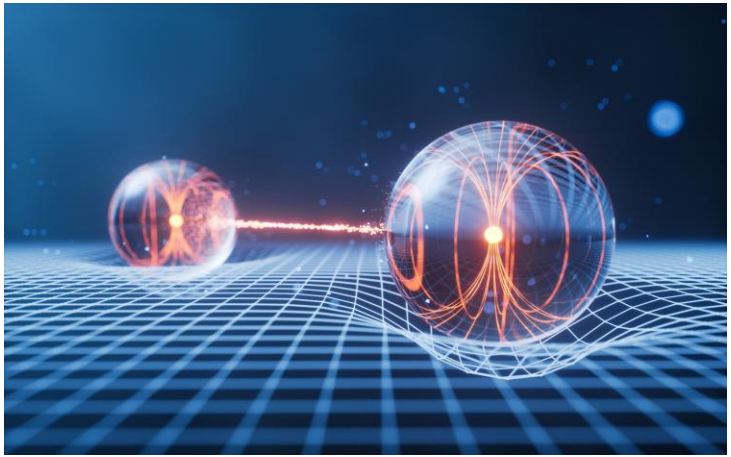
GANs can generate **synthetic network traffic** to augment training datasets and address class imbalance issues.

Introduction

Classical ML models suffer from limitations, such as managing **high-dimensional spaces**.

Quantum ML has the potential to overcome the scalability and performance limitations of classical ML techniques.

QML leverages entanglement and superposition to efficiently **capture complex patterns.**

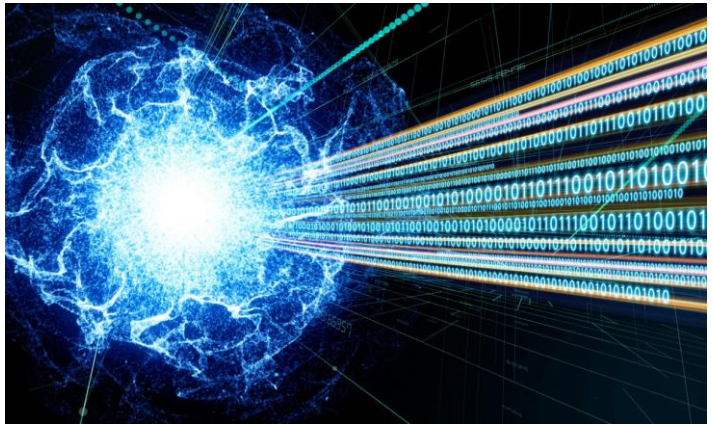


Quantum GAN

Two primary configurations: full quantum and hybrid.

A typical hybrid QGAN consists of two components:

- **Quantum Generator** (G): A parameterized quantum circuit (PQC) that generates quantum states representing data samples.
- **Classical Discriminator** (D): A classical neural network that evaluates the similarity between generated samples and real data.



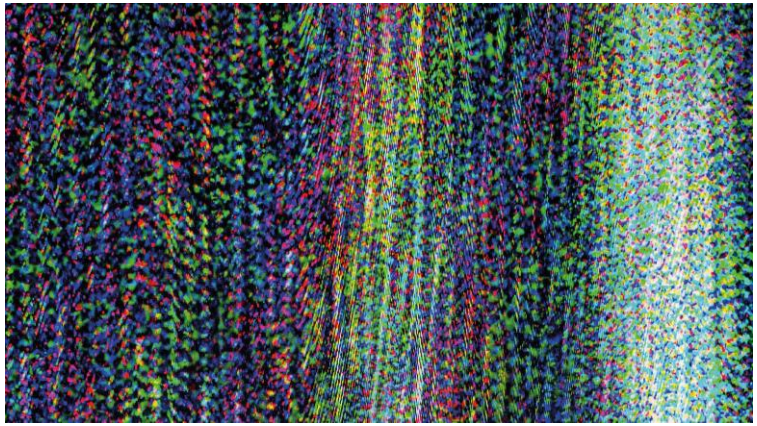
The generator aims to learn the underlying data distribution, while the discriminator attempts to distinguish between real and generated samples. The **goal** is to **iteratively refine both components** through adversarial training.

Quantum Noise

The presence of **noise** in current quantum hardware remains a major challenge

Evaluations that **ignore these aspects** may falsify the true capabilities of QML models.

While **testing on real quantum machines** is valuable for validation, it is often resource-intensive and costly.

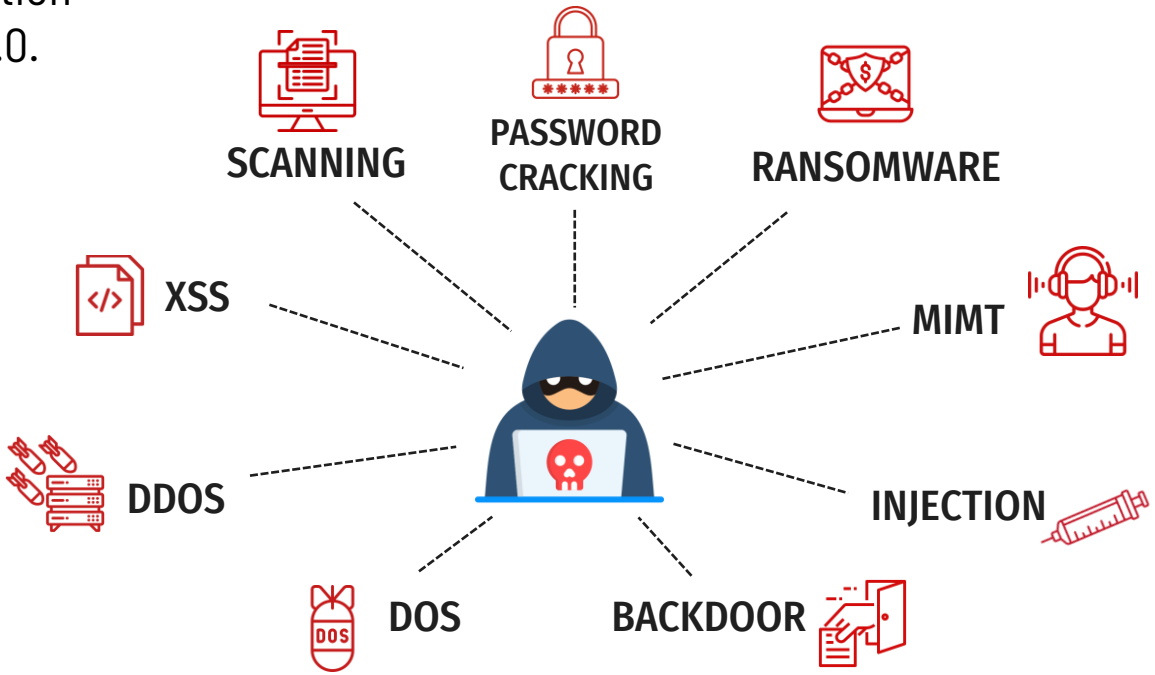


Objective

- **Improve performance** and **efficiency** of Intrusion detection model using QML.
- Use **Noisy quantum simulators** to emulate real hardware imperfections, enabling the testing of QML architectures under more realistic conditions.

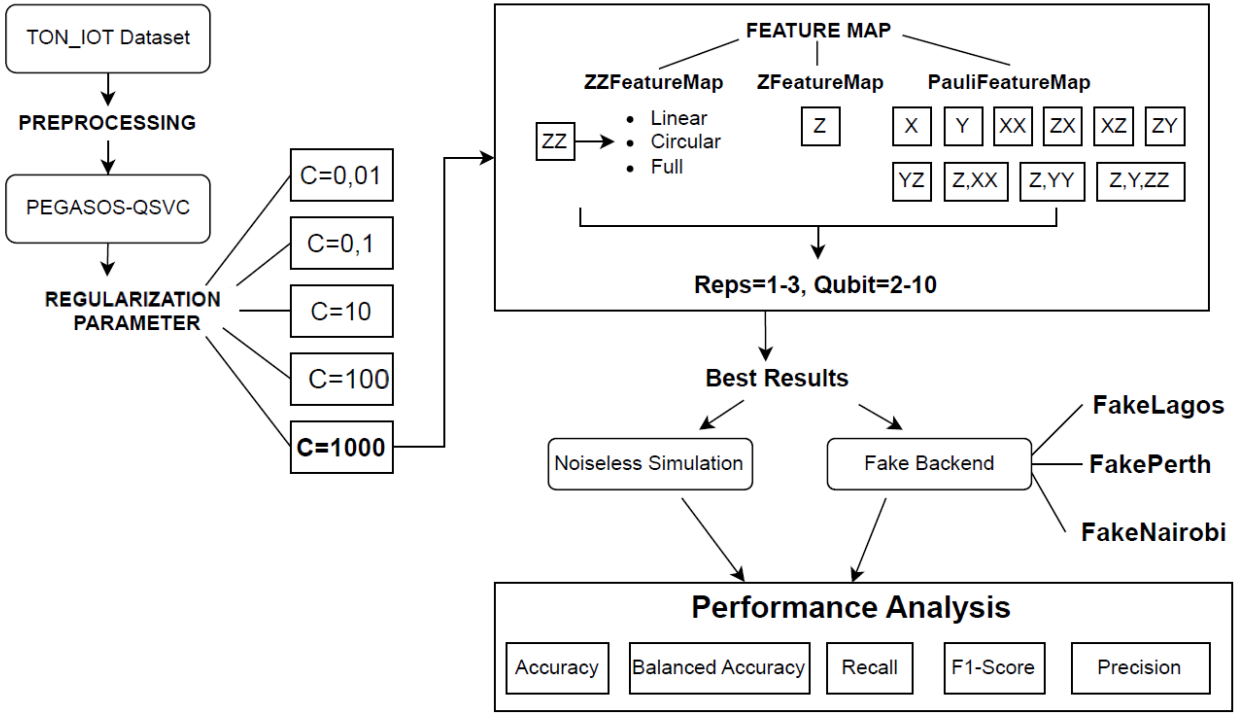
Dataset & Preprocessing

- **TON_IOT** dataset, a next-generation dataset collection for Industry 4.0.
- **PCA** with different numbers of principal components.
- **Min-Max scaling** in range [0, 1] .



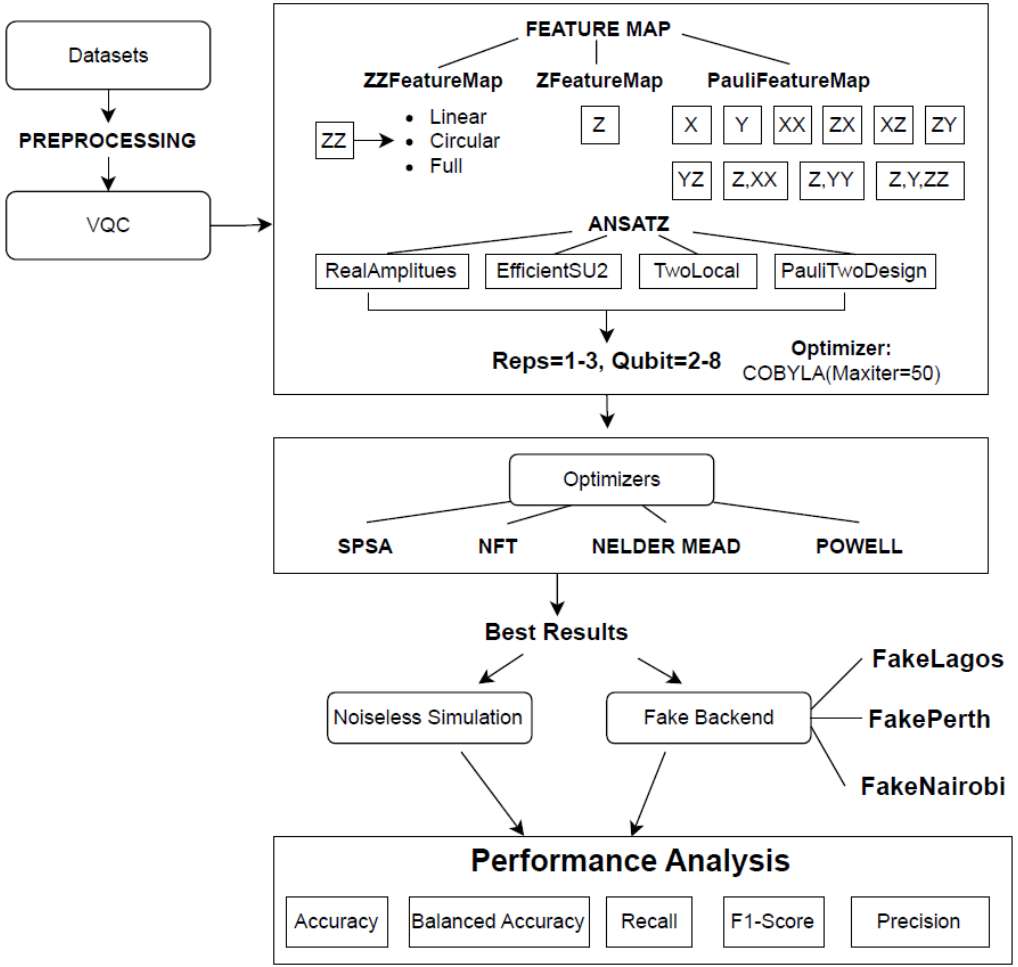
Methodology

Pegasos-QSVC



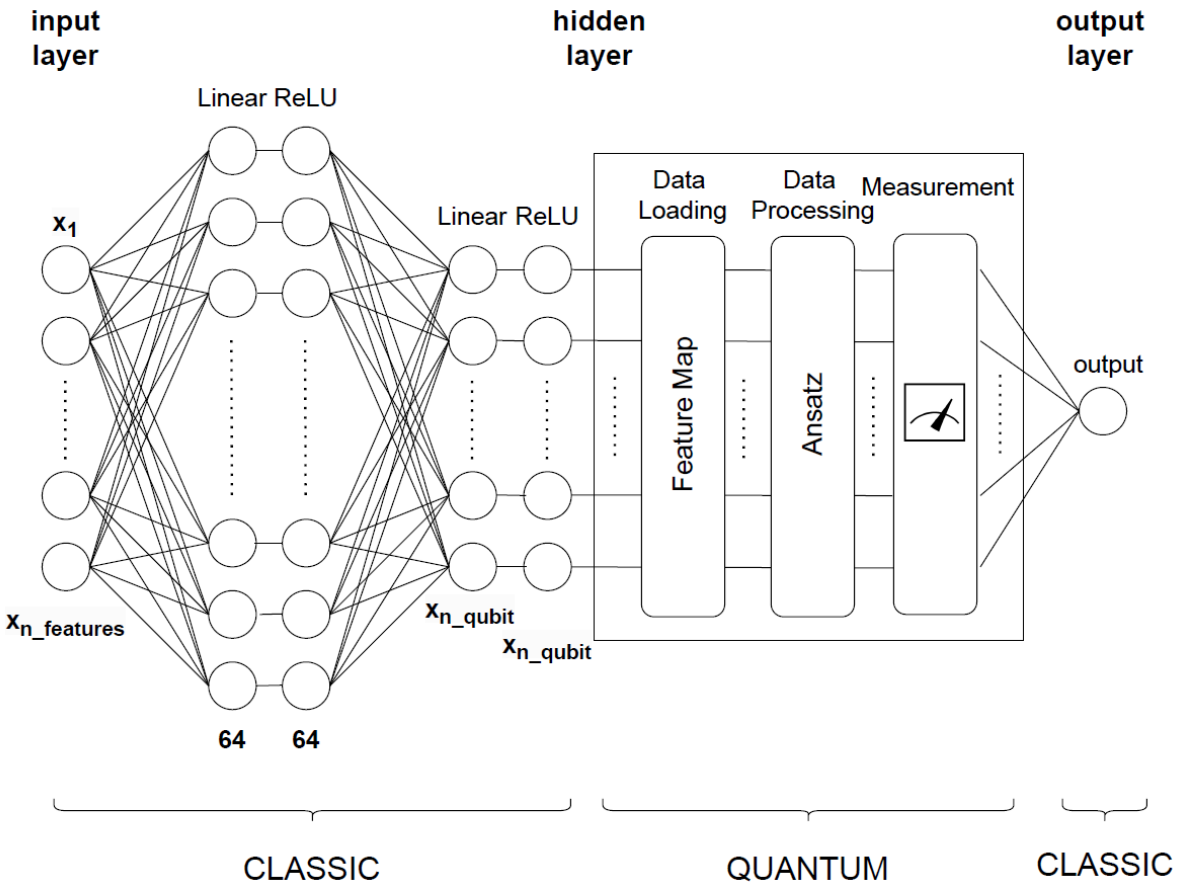
Methodology

VQC



Methodology

Hybrid quantum-classical neural network



Noisy computation

Key Performance Metrics

- **T1 (Relaxation Time):** Time a qubit stays excited before decaying; limited by environmental interactions.
- **T2 (Dephasing Time):** Duration a qubit remains coherent; shorter than T1 due to additional phase noise.
- **Readout Error:** Probability of measuring the wrong qubit state, even after perfect operations.
- **Gate Errors**:
  - rz: Minimal error (virtually implemented)
  - sx, x: Affected by control noise
  - cx (CNOT): Highest error rate, critical for entanglement

Noisy simulation

Noisy simulators provide a more practical alternative for evaluating several configurations.

The selected quantum circuits were tested under noisy conditions using **fake backends** provided by **IBM**, which are designed to **mimic the behavior of IBM Quantum systems** and are built using system **snapshots**.

These snapshots contain information about the simulated quantum device, such as the coupling map, which describes the **physical connections between qubits**, and the **qubit properties.**

Optimization and Results

Pegasos-QSVC - Regularization parameter C

A **higher C** reduces penalties for misclassified points, resulting in wider margins and better generalization but increasing the risk of bias or **underfitting**.

Conversely, a **lower C** forces the algorithm to fit misclassified points more closely, leading to narrower margins and a risk of **overfitting** while capturing more complex patterns.

Pegasos QSVC	Accuracy	Precision	Recall	F1-Score
C= 1000	92,25%	92,77%	97,18%	94,92%
C= 100	91,61%	92,02%	97,18%	94,53%
C= 10	79,16%	89,48%	81,65%	85,39%
C= 1	74,56%	74,56%	100%	85,42%
C= 0.1	74,56%	74,56%	100%	85,42%
C= 0.01	74,56%	74,56%	100%	85,42%

Optimization and Results

Pegasos-QSVC – Fine-tuning

2 to 10 **qubits**, 14 **feature maps**, and 1 to 3 **repetitions** for each -> 378 experiments

The tests were conducted initially on a noiseless simulator and subsequently on a noisy simulator to optimize the number of tests

Best FeatureMap	Accuracy	Precision	Recall	F1 Score	Balanced Accuracy
q=6, ZZFeatureMap (Linear), r=1	94,12%	96,10%	96,01%	96,05%	92,29%
q=6, PauliFeatureMap [ZX], r=1	94,16%	96,04%	96,13%	96,09%	92,26%
q=4, PauliFeatureMap [Z,YY], r=1	93,17%	93,91%	97,15%	95,50%	89,34%
q=7, ZFeatureMap, r=2	95,44%	96,93%	96,96%	96,94%	93,98%
q=5, PauliFeatureMap [XX], r=2	94,14%	96,22%	95,91%	96,06%	92,43%
q=6, PauliFeatureMap [Z,XX], r=2	95,61%	97,11%	96,99%	97,05%	94,27%
q=4, PauliFeatureMap [Z,Y,ZZ], r=2	94,57%	95,89%	96,86%	96,37%	92,35%
q=5, ZZFeatureMap (Linear), r=3	94,35%	95,65%	96,83%	96,24%	91,96%

Optimization and Results

Pegasos-QSVC – Fine-tuning

- Qubit 4-7
- Reps increase stability
- ZZFeatureMap Linear, Pauli feature map, need of entanglement

Best FeatureMap	Accuracy	Precision	Recall	F1 Score	Balanced Accuracy
q=6, ZZFeatureMap (Linear), r=1	94,12%	96,10%	96,01%	96,05%	92,29%
q=6, PauliFeatureMap [ZX], r=1	94,16%	96,04%	96,13%	96,09%	92,26%
q=4, PauliFeatureMap [Z,YY], r=1	93,17%	93,91%	97,15%	95,50%	89,34%
q=7, ZFeatureMap, r=2	95,44%	96,93%	96,96%	96,94%	93,98%
q=5, PauliFeatureMap [XX], r=2	94,14%	96,22%	95,91%	96,06%	92,43%
q=6, PauliFeatureMap [Z,XX], r=2	95,61%	97,11%	96,99%	97,05%	94,27%
q=4, PauliFeatureMap [Z,Y,ZZ], r=2	94,57%	95,89%	96,86%	96,37%	92,35%
q=5, ZZFeatureMap (Linear), r=3	94,35%	95,65%	96,83%	96,24%	91,96%

Optimization and Results

Pegasos-QSVC – Noisy simulation

Same qubits architecture with different error level.

FakeLagos -> good single-gate error probability but high two-qubit gate error and readout error.

FakeNairobi, FakePerth -> higher gate error levels but lower readout error and two-qubit gate error.

Configuration	Fake Nairobi	Fake Lagos	Fake Perth
6 QUBIT, PauliFeatureMap [ZX], reps=1	Acc: 93,03% Precision: 95,42% Recall: 97,17% F1-Score: 93,71% Acc B: 89,05%	Acc: 88,10% Precision: 93,31% Recall: 90,62% F1-Score: 91,95% Acc B: 85,79%	Acc: 88,10% Precision: 93,31% Recall: 90,62% F1-Score: 91,95% Acc B: 85,79%
4 QUBIT, PauliFeatureMap [Z,YY], reps=1	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%	Acc: 91,07% Precision: 91,44% Recall: 97,12% F1-Score: 94,19% Acc B: 85,23%	Acc: 91,54% Precision: 91,99% Recall: 97,15% F1-Score: 94,50% Acc B: 86,18%
5 QUBIT, ZFeatureMap, reps=1	Acc: 93,05% Precision: 95,43% Recall: 97,18% F1-Score: 93,73% Acc B: 89,07%	Acc: 93,05% Precision: 95,43% Recall: 97,18% F1-Score: 93,73% Acc B: 89,07%	Acc: 93,05% Precision: 95,43% Recall: 97,18% F1-Score: 93,73% Acc B: 89,07%
7 QUBIT, ZFeatureMap, reps=2	Acc: 82,37% Precision: 92,28% Recall: 83,33% F1-Score: 87,58% Acc B: 81,45%	Acc: 90,43% Precision: 90,23% Recall: 97,75% F1-Score: 93,84% Acc B: 83,37%	Acc: 82,37% Precision: 92,28% Recall: 83,33% F1-Score: 87,58% Acc B: 81,45%
5 QUBIT, PauliFeatureMap [XX], reps=2	Acc: 91,28% Precision: 91,69% Recall: 97,15% F1-Score: 94,34% Acc B: 85,67%	Acc: 89,84% Precision: 96,23% Recall: 89,89% F1-Score: 92,96% Acc B: 89,79%	Acc: 82,42% Precision: 91,02% Recall: 84,79% F1-Score: 87.80% Acc B: 80,14%
6 QUBIT, PauliFeatureMap [Z,XX], reps=2	Acc: 74,56% Precision: 74,56% Recall: 100% F1-Score: 85,42% Acc B: 50,00%	Acc: 74,56% Precision: 74,56% Recall: 100% F1-Score: 85,42% Acc B: 50,00%	Acc: 74,56% Precision: 74,56% Recall: 100% F1-Score: 85,42% Acc B: 50,00%
5 QUBIT, ZZFeatureMap (Linear), reps=3	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%	Acc: 92,11% Precision: 94,83% Recall: 97,08% F1-Score: 92,68% Acc B: 87,31%	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%
4 QUBIT, PauliFeatureMap [Z,Y,ZZ], reps=3	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%

Optimization and Results

Pegasos-QSVC – Noisy simulation

One of the best configuration (**PauliFeatureMap[Z, XX]** using 2 repetitions and 6 qubits) becomes the worst when noise is applied.

ZFeatureMap shows good performance even in the presence of noise.

Configuration	Fake Nairobi	Fake Lagos	Fake Perth
6 QUBIT, PauliFeatureMap [ZX], reps=1	Acc: 93,03% Precision: 95,42% Recall: 97,17% F1-Score: 93,71% Acc B: 89,05%	Acc: 88,10% Precision: 93,31% Recall: 90,62% F1-Score: 91,95% Acc B: 85,79%	Acc: 88,10% Precision: 93,31% Recall: 90,62% F1-Score: 91,95% Acc B: 85,79%
4 QUBIT, PauliFeatureMap [Z,YY], reps=1	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%	Acc: 91,07% Precision: 91,44% Recall: 97,12% F1-Score: 94,19% Acc B: 85,23%	Acc: 91,54% Precision: 91,99% Recall: 97,15% F1-Score: 94,50% Acc B: 86,18%
5 QUBIT, ZFeatureMap, reps=1	Acc: 93,05% Precision: 95,43% Recall: 97,18% F1-Score: 93,73% Acc B: 89,07%	Acc: 93,05% Precision: 95,43% Recall: 97,18% F1-Score: 93,73% Acc B: 89,07%	Acc: 93,05% Precision: 95,43% Recall: 97,18% F1-Score: 93,73% Acc B: 89,07%
7 QUBIT, ZFeatureMap, reps=2	Acc: 82,37% Precision: 92,28% Recall: 83,33% F1-Score: 87,58% Acc B: 81,45%	Acc: 90,43% Precision: 90,23% Recall: 97,75% F1-Score: 93,84% Acc B: 83,37%	Acc: 82,37% Precision: 92,28% Recall: 83,33% F1-Score: 87,58% Acc B: 81,45%
5 QUBIT, PauliFeatureMap [XX], reps=2	Acc: 91,28% Precision: 91,69% Recall: 97,15% F1-Score: 94,34% Acc B: 85,67%	Acc: 89,84% Precision: 96,23% Recall: 89,89% F1-Score: 92,96% Acc B: 89,79%	Acc: 82,42% Precision: 91,02% Recall: 84,79% F1-Score: 87.80% Acc B: 80,14%
6 QUBIT, PauliFeatureMap [Z,XX], reps=2	Acc: 74,56% Precision: 74,56% Recall: 100% F1-Score: 85,42% Acc B: 50,00%	Acc: 74,56% Precision: 74,56% Recall: 100% F1-Score: 85,42% Acc B: 50,00%	Acc: 74,56% Precision: 74,56% Recall: 100% F1-Score: 85,42% Acc B: 50,00%
5 QUBIT, ZZFeatureMap (Linear), reps=3	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%	Acc: 92,11% Precision: 94,83% Recall: 97,08% F1-Score: 92,68% Acc B: 87,31%	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%
4 QUBIT, PauliFeatureMap [Z,Y,ZZ], reps=3	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%

Optimization and Results

Pegasos-QSVC – Noisy simulation

FakeLagos is more sensitive to feature maps with a higher number of CX gates.

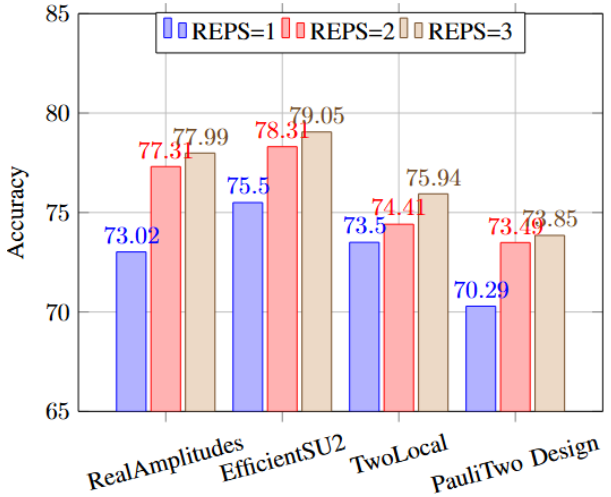
It performs well when using the ZFeatureMap, which does not involve entanglement between qubits.

Configuration	Fake Nairobi	Fake Lagos	Fake Perth
6 QUBIT, PauliFeatureMap [ZX], reps=1	Acc: 93,03% Precision: 95,42% Recall: 97,17% F1-Score: 93,71% Acc B: 89,05%	Acc: 88,10% Precision: 93,31% Recall: 90,62% F1-Score: 91,95% Acc B: 85,79%	Acc: 88,10% Precision: 93,31% Recall: 90,62% F1-Score: 91,95% Acc B: 85,79%
4 QUBIT, PauliFeatureMap [Z,YY], reps=1	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%	Acc: 91,07% Precision: 91,44% Recall: 97,12% F1-Score: 94,19% Acc B: 85,23%	Acc: 91,54% Precision: 91,99% Recall: 97,15% F1-Score: 94,50% Acc B: 86,18%
5 QUBIT, ZFeatureMap, reps=1	Acc: 93,05% Precision: 95,43% Recall: 97,18% F1-Score: 93,73% Acc B: 89,07%	Acc: 93,05% Precision: 95,43% Recall: 97,18% F1-Score: 93,73% Acc B: 89,07%	Acc: 93,05% Precision: 95,43% Recall: 97,18% F1-Score: 93,73% Acc B: 89,07%
7 QUBIT, ZFeatureMap, reps=2	Acc: 82,37% Precision: 92,28% Recall: 83,33% F1-Score: 87,58% Acc B: 81,45%	Acc: 90,43% Precision: 90,23% Recall: 97,75% F1-Score: 93,84% Acc B: 83,37%	Acc: 82,37% Precision: 92,28% Recall: 83,33% F1-Score: 87,58% Acc B: 81,45%
5 QUBIT, PauliFeatureMap [XX], reps=2	Acc: 91,28% Precision: 91,69% Recall: 97,15% F1-Score: 94,34% Acc B: 85,67%	Acc: 89,84% Precision: 96,23% Recall: 89,89% F1-Score: 92,96% Acc B: 89,79%	Acc: 82,42% Precision: 91,02% Recall: 84,79% F1-Score: 87.80% Acc B: 80,14%
6 QUBIT, PauliFeatureMap [Z,XX], reps=2	Acc: 74,56% Precision: 74,56% Recall: 100% F1-Score: 85,42% Acc B: 50,00%	Acc: 74,56% Precision: 74,56% Recall: 100% F1-Score: 85,42% Acc B: 50,00%	Acc: 74,56% Precision: 74,56% Recall: 100% F1-Score: 85,42% Acc B: 50,00%
5 QUBIT, ZZFeatureMap (Linear), reps=3	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%	Acc: 92,11% Precision: 94,83% Recall: 97,08% F1-Score: 92,68% Acc B: 87,31%	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%
4 QUBIT, PauliFeatureMap [Z,Y,ZZ], reps=3	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%

Optimization and Results

VQC – Ansatz and FeatureMap

- 4 **ansatzes**, 1-3 **reps**
- ZZFeatureMap and EfficientSU2 best stability in performance
- PauliFeatureMap good performance but unstable



Ansatz	ZZFeatureMap			PauliFeatureMap								
	Linear	Full	Circular	Z	Y	XZ	ZX	YZ	ZY	Z,YY	Z, XX	Z,Y,ZZ
RealAmplitudes	85,04	92,48	84,19	88,75	91,54	82,18	75,40	89,62	89,74	86,13	78,90	83,93
EfficientSU2	90,57	90,81	90,81	88,44	91,82	82,28	83,91	90,55	86,32	93,17	89,13	90,26
TwoLocal	86,03	84,78	88,23	81,45	91,54	84,10	80,86	84,03	84,62	86,20	75,99	81,76
PauliTwoDesign	87,24	82,99	87,36	81,99	83,32	86,08	76,84	79,14	79,77	84,64	70,65	77,65

Optimization and Results

VQC – Optimizers and best configurations

- **Qubits** 3-5
- **COBYLA** generally achieves the highest accuracies and is the most stable.
- **SPSA** and **NFT** also perform well, but are less stable
- **NELDER MEAD** and **POWELL** achieve competitive results only in specific cases.

Best configurations	COBYLA	SPSA	NFT	NELDER MEAD	POWELL
q=3, ZZ (Linear), EfficientSU2, r=2	91,23%	89,10%	90,10%	85,18%	87,43%
q=3, Pauli [Y], RealAmplitudes, r=2	91,54%	91,47%	91,54%	88,42%	83,06%
q=5, Pauli [Z, YY], EfficientSU2, r=2	91,11%	92,79%	90,17%	81,73%	86,29%
q=3, ZZ (Circular), EfficientSU2, r=2	90,81%	92,13%	94,18%	94,21%	89,84%
q=3, Pauli [Z,Y,ZZ], EfficientSU2, r=3	90,26%	93,00%	93,60%	92,98%	93,57%
q=4, ZZ (Full), RealAmplitudes, r=3	92,48%	81,43%	83,25%	85,99%	82,37%

Optimization and Results

VQC – Noisy simulation

- The configuration 3 QUBIT, PauliFeatureMap [Y], RealAmplitudes(reps=2), SPSA consistently delivers **the best overall results** with accuracy above 91\% and F1-Score up to 94.19\%.
- In contrast, the configuration 5 QUBIT, PauliFeatureMap [Z, YY], EfficientSU2 (REPS=2), SPSA shows significant **performance drops**, especially on Fake Lagos and Fake Perth.
- FakeNairobi and FakePerth perform better than FakeLagos with circuits that require **more entanglement**, such as the ZZFeatureMap, and with EfficientSU2, which are more complex compared to Real Amplitudes.

Optimization and Results

Hybrid – Fine-tuning

- **Simpler circuits** contributes to optimizing the performance
- Qubit 3-5

Configuration	Noiseless simulation
3 QUBIT, PauliFeatureMap [”Z”, ”YY”], RealAmplitudes (reps=3)	Acc: 83,7% Precision: 90,3% Recall: 68,1% F1-Score: 71,6%
3 QUBIT, PauliFeatureMap [”YZ”], EfficientSU2 (reps=2)	Acc: 93,1% Precision: 92,3% Recall: 89,1% F1-Score: 90,5%
3 QUBIT, PauliFeatureMap [”Z”, ”YY”], EfficientSU2 (reps=2)	Acc: 93,1% Precision: 92,3% Recall: 89,1% F1-Score: 90,5%
3 QUBIT, PauliFeatureMap [”ZY”], TwoLocal (reps=3)	Acc: 74,6% Precision: 37,3% Recall: 50,0% F1-Score: 42.7%
5 QUBIT, PauliFeatureMap [”Y”], RealAmplitudes (reps=3)	Acc: 93,0% Precision: 92,2% Recall: 89,1% F1-Score: 90,5%
3 QUBIT, PauliFeatureMap [”ZY”], EfficientSU2 (reps=2)	Acc: 93.1% Precision: 92.2% Recall: 89.1% F1-Score: 90.5%
5 QUBIT, PauliFeatureMap [”ZY”], EfficientSU2 (reps=1)	Acc: 95,4% Precision: 96,9% Recall: 96,9% F1-Score: 96,9%
5 QUBIT, PauliFeatureMap[”Z”, ”YY”], EfficientSU2 (reps=2)	Acc: 88,5% Precision: 88,5% Recall: 88,1% F1-Score: 83,1%
6 QUBIT, ZZFeatureMap(Full), EfficientSU2 (reps=2)	Acc: 74,6% Precision: 37,3% Recall: 50,0% F1-Score: 42.7%

Optimization and Results

Hybrid – Noisy simulation

- Results are highly dependent on the backend used.
- Drop in performance with FakeLagos, due to greater sensitivity to errors in two-qubit gates.
- FakePerth maintains a consistent level of performance.

Configuration	Fake Nairobi	Fake Lagos	Fake Perth
5 QUBIT, PauliFeatureMap [”ZY”], EfficientSU2 (reps=1)	Acc: 90.0% Prec.: 89.9% Rec.: 83.4% F1-Sc.: 85.9%	Acc: 86.8% Prec.: 87.2% Rec.:77.0% F1-Sc.: 80.3%	Acc: 94.2% Prec.: 93.1% Rec.: 91.7% F1-Sc.: 92.4%

Result Analysis

- Pegasos-QSVC stands out for achieving **better performance** than VQC.
- VQC delivers strong results, but their performance is **highly sensitive** to parameters, requiring systematic testing.
- The hybrid neural network performs well, showcasing the benefits of **combining classical and quantum paradigms** for robust and efficient models.

Noiseless

Configuration	Performance
PegasosQSVC - 6 QUBIT, PauliFeatureMap [Z,XX], reps=2)	Accuracy: 95,61% Precision: 97,11% Recall: 96,99% F1 Score: 97,05%
VQC - 3 QUBIT, ZZFeatureMap(Circular), EfficientSU2(REPS=3), NELDER_MEAD	Accuracy: 94,21% Precision: 96,46% Recall: 95,75% F1 Score: 96,10%
Hybrid Quantum-Classical Neural Network - PauliFeatureMap [Y], 5 QUBIT, reps=1, ADAM	Accuracy: 95.44% Precision: 96.96% Recall: 96.93% F1 Score: 96.94%

Noisy

Configuration	Performance
PegasosQSVC - 5 QUBIT, ZFeatureMap, (reps=1)	Accuracy: 93,05% Precision: 95,43% Recall: 97,18% F1 Score: 93,73%
VQC - 3 QUBIT, PauliFeatureMap [Y], RealAmplitudes(reps=2), COBYLA	Accuracy: 91,57% Precision: 91,99% Recall: 97,15% F1 Score: 94,50%
Hybrid Quantum-Classical Neural Network - PauliFeatureMap [Y], 5 QUBIT, reps=1, ADAM	Accuracy: 94.2% Precision: 93.1% Recall: 91.7% F1-Score: 92.4%

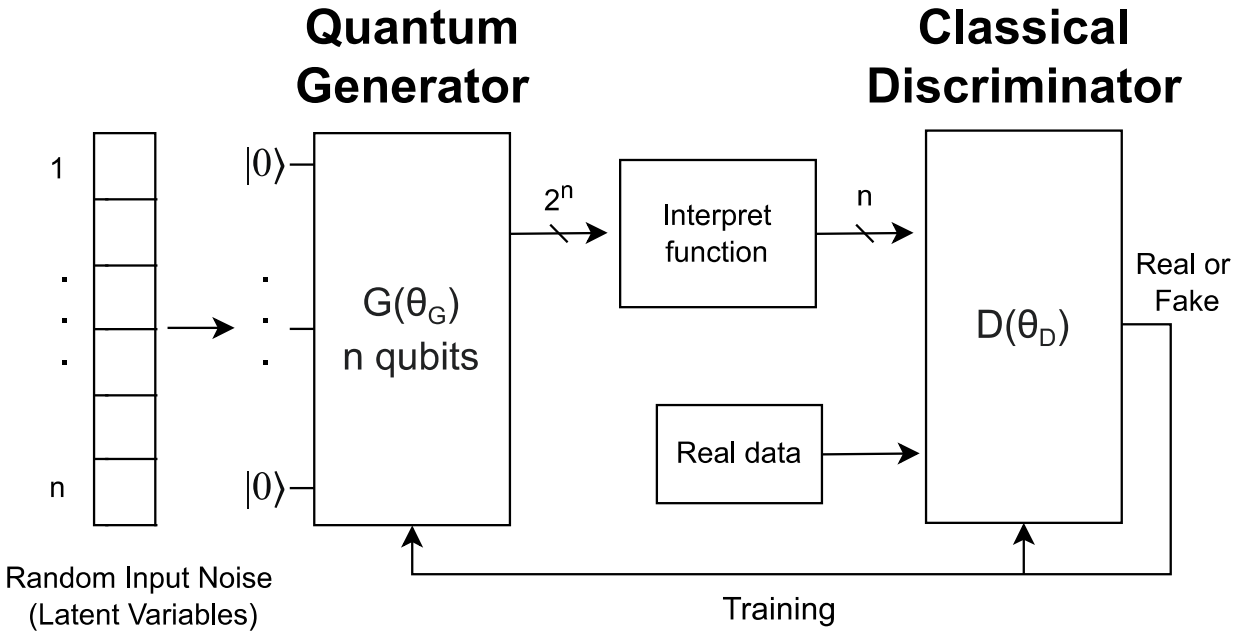
Result Analysis

QML models are heavily **influenced by the noise characteristics** of the simulated quantum backends.

Once the real quantum machine for training is selected, it is useful to assess the machine's resistance to two-qubit gate errors in order to decide the circuit structure to execute.

Machines more sensitive to these errors may require circuits with less entanglement, while machines less prone to errors can handle more complex circuits with greater entanglement.

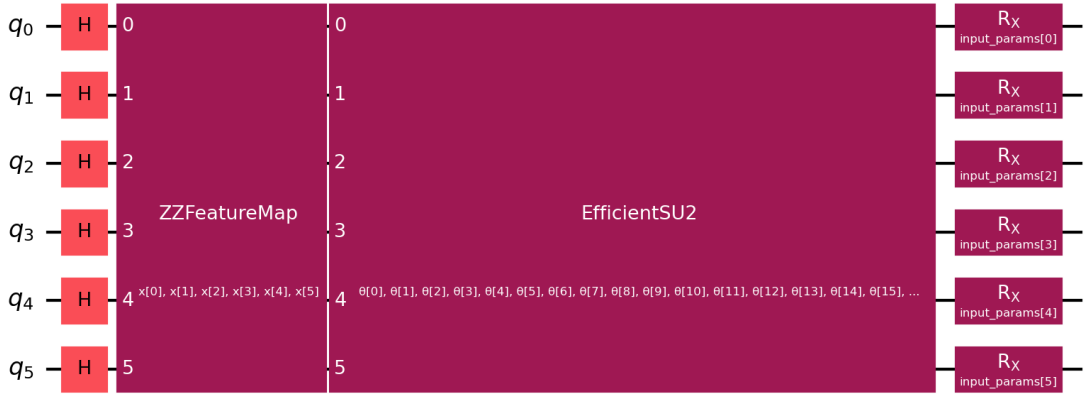
QGAN Configuration



Generator

Each qubit in the circuit represents a feature of the dataset.

- Hadamard gates to all qubits
- Feature map
- Ansatz
- Rotation gates to inject noise



Interpret function

The interpret function transforms the generator's output, which corresponds to the probability of observing each **combination of qubit states**, with a **cardinality of 2^n**, **into** a lower-dimensional representation of **cardinality n**.

For each qubit i, the interpret function **extracts its marginal probability** by summing over all measurement outcomes where qubit i is in state 1.

Interpret function

Let P(x) represent the probability of observing a particular n-qubit state $x = (x_1, x_2, \ldots, x_n)$, where $x_i \in \{0, 1\}$. Then, the marginal probability p_i for qubit i is computed as:

$$p_i = \sum_{x : x_i = 1} P(x),$$

The resulting vector is $(p_1, p_2, \ldots, p_n)$, each component p_i lies in the range [0, 1], making it directly comparable to the features in the real dataset.

This dimensionality reduction simplifies the discriminator's input without sacrificing the ability to distinguish between real and generated data.

Discriminator

The discriminator is a classical feedforward neural network.

Dense hidden layers with **LeakyReLU functions**, to introduce non-linearity while mitigating vanishing gradients.

Evaluation metrics such as **loss values**, **accuracy**, and the **F1 score** monitor performance, while the **quality of generated data** is validated by comparing its statistical properties with real data.

The experiments are replicable, and the source code is available on https://github.com/francocirill/qgan

Configurations tested

#qubits: 3-6

Ansatz: EfficientSU2, Real Amplitude

Reps: 3-10

Discriminator Hidden Layer **Size**: 16-128

Learning rate: 0.001-0.01

Epochs: 80

Generator Reps Impact

A clear trend can be observed where **increasing** the number of **generator repetitions** generally leads to **slightly improved performance** metrics, though the magnitude of improvement tends to decrease as the repetitions increase, **stabilizing after 8-10 reps**.

However, while the gains in performance are incremental, the **computational time** required to train the model **increases substantially**.

Learning Rate Sensitivity

The **learning rates** for both the generator and discriminator **play a crucial role** in determining the stability and convergence of the QGAN.

While most configurations use balanced learning rates (e.g., 0.01 for both), experiments with significantly smaller or unbalanced learning rates (e.g., 0.001/0.005 or 0.003/0.008) show mixed results.

These settings sometimes lead to minor performance drops, highlighting **the importance of carefully tuning the learning rates**.

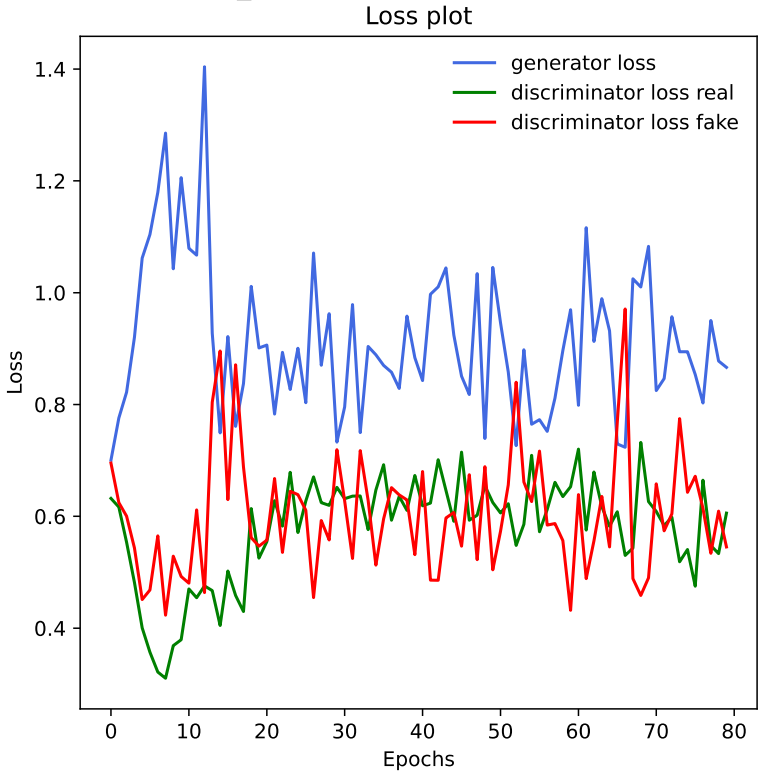
Generator params

EfficientSU2 demonstrates slightly better performance across most configurations, particularly when combined with 6 principal components.

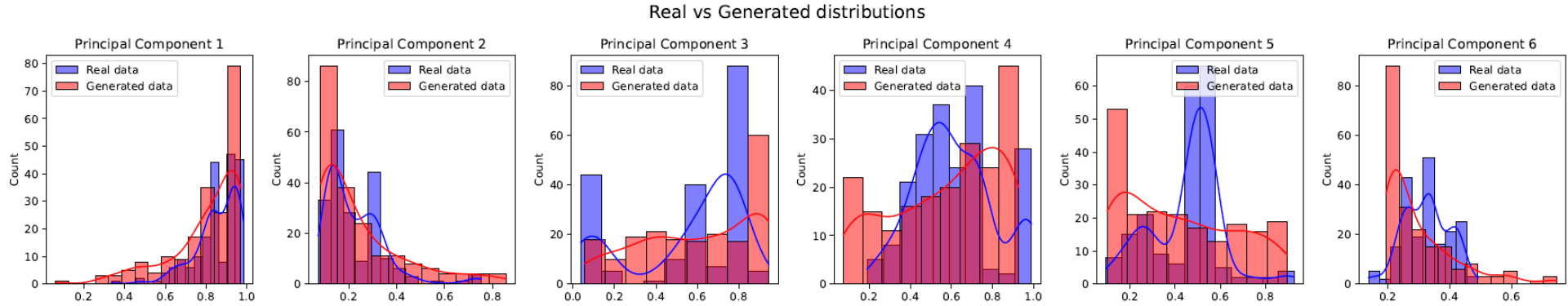
Reducing the **number of features** generally leads to a slight decrease in performance, indicating that retaining more features provides the model with richer information to generate better outputs.

Loss plot

The highest-performing configuration combines **6 PCA features**, the **EfficientSU2** ansatz, **9 generator repetitions**, and **128** discriminator **neurons**. This setup achieves an **accuracy of 0.937** and an **F1 score of 0.9384.**

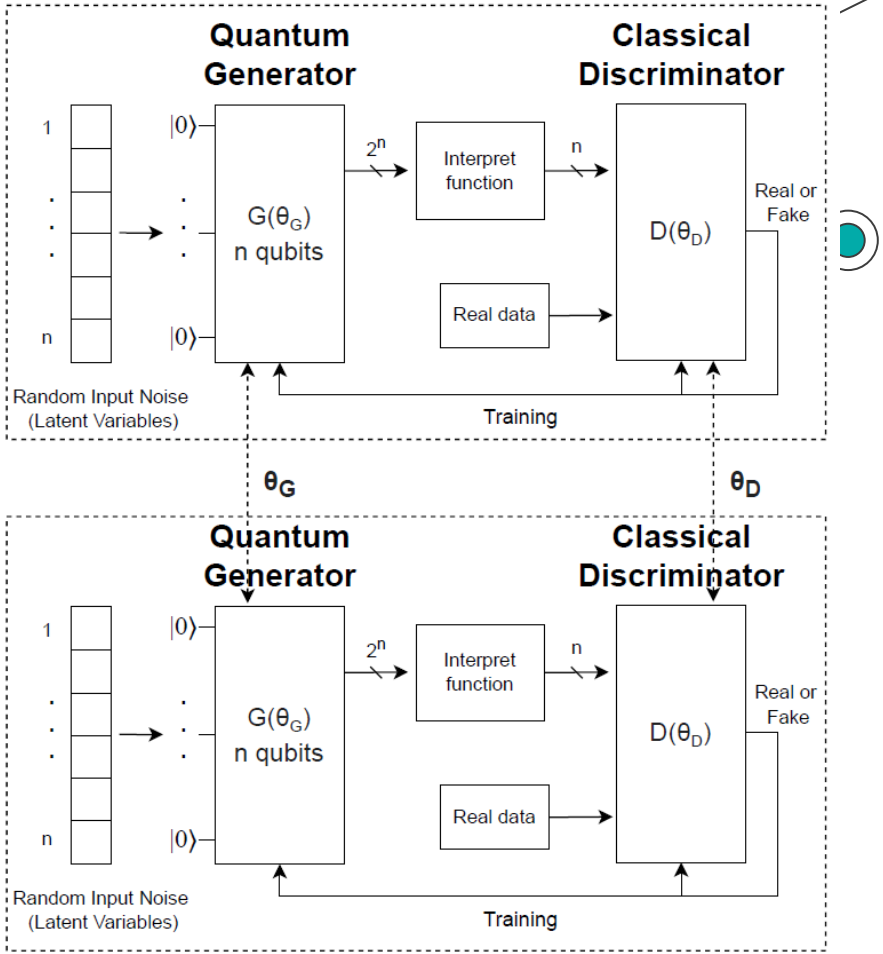


Distribution plot



Federated Approach

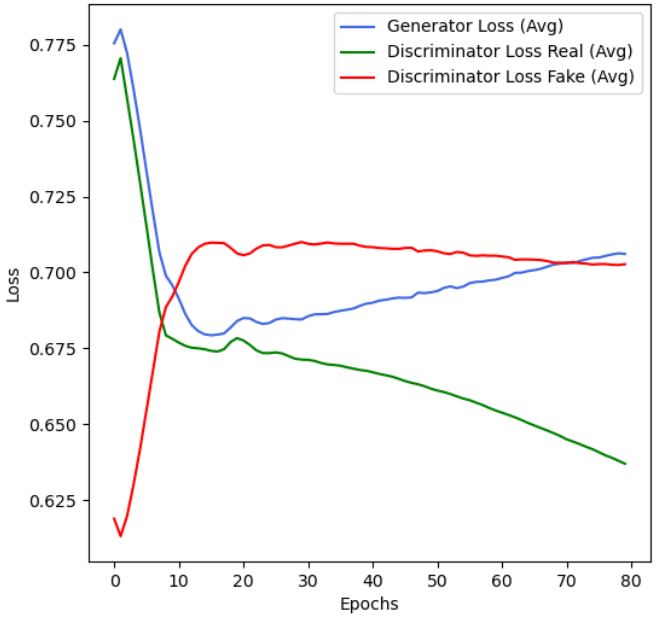
- Generator and discriminator weights are aggregated separately
- Each quantum generator operates with a unique seed
- FedAvg



Loss plot

The highest-performing configuration among **4 nodes** combines **6 PCA features**, the **EfficientSU2** ansatz, **3 generator repetitions**, and **128** discriminator **neurons**.

This setup achieves an **accuracy of 0.9125** and an **F1 score of 0.9034.**



Noisy Simulation

Evaluated using **IBM's FakeBackend** to compare performance degradation and convergence time.

This backend simulates quantum devices based on system snapshots.

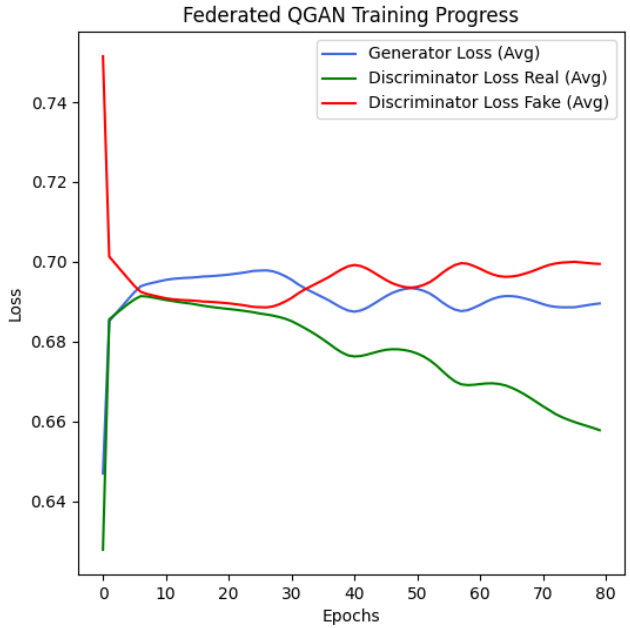
Simulations were performed with the **Qiskit-Aer package** using the Sampler primitive.

While noisy simulations yield different results, starting with noise-free simulations provides a useful baseline before **assessing the impact of noise**.

Loss plot

Evaluated on the **FakeNairobi backend**, a noisy quantum simulator with seven qubits.

To address problems related to the generator loss, the **discriminator's learning rate** was reduced, achieving an optimal balance between the two models.

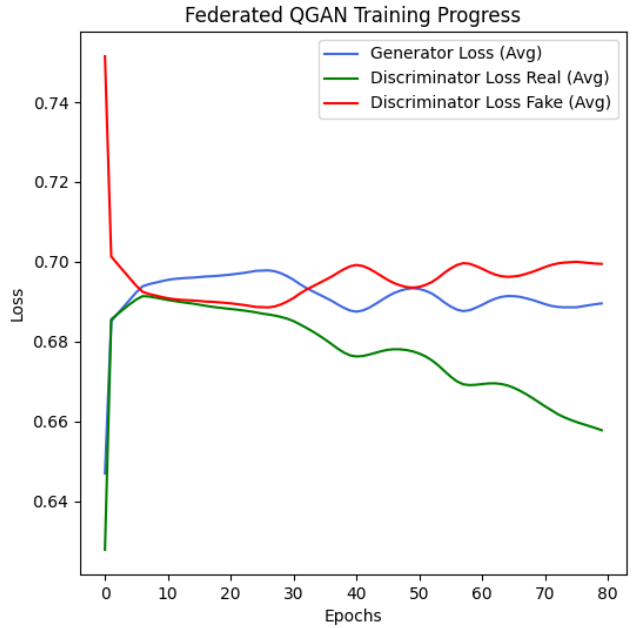


Loss plot

Final model reached an **accuracy** of **0.8738** and an **F1-score** of **0.8797.**

Importance of balancing the generator and discriminator's capabilities.

The **learning rate** proves to be a key tunable parameter in this regard.

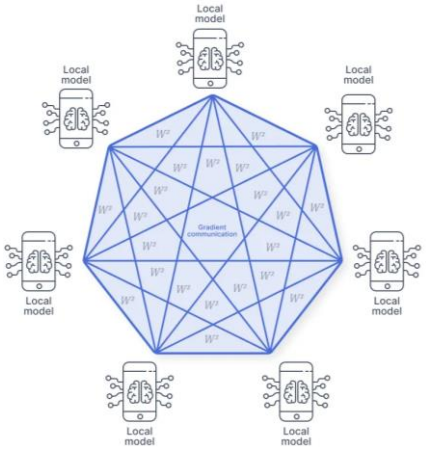


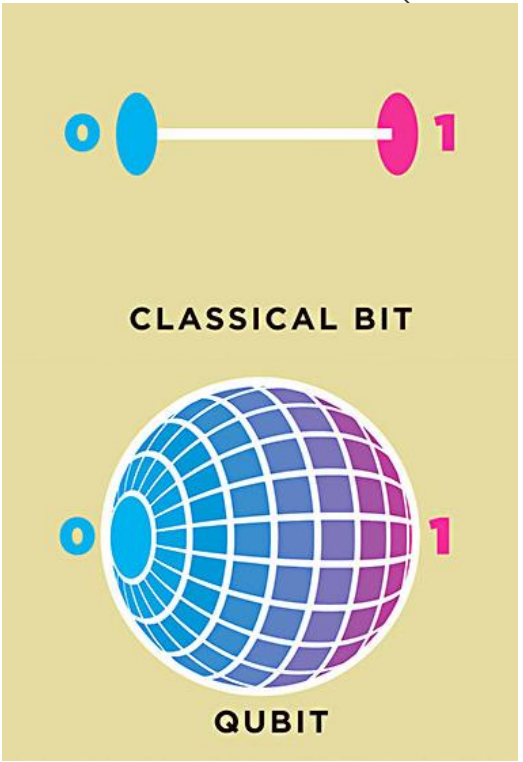
Authentication

Increasing necessity for **authentication mechanisms** of quantum devices.

This is essential for the integration of these machines into **larger quantum networks**.

In **Quantum Federated Learning** it is essential for all collaborating devices within the learning algorithm to be authenticated



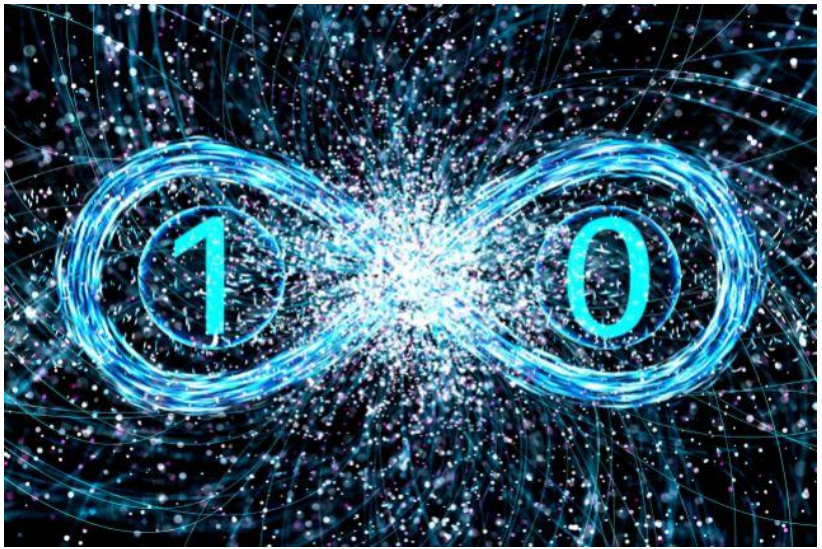


Authentication

Leveraging **quantum principles** for device authentication presents a compelling avenue.

Quantum properties such as superposition, entanglement provide inherent advantages over classical methods.

For instance, quantum-based authentication schemes can exploit the **no-cloning theorem**.



Authentication

Traditional methods rely on **cryptographic key management**, a process that has long been vulnerable to various risks and challenges.

One of the primary **vulnerabilities** lies in the potential **exposure** of cryptographic **keys.**

Authentication

Stored keys are always **susceptible** to **interception** or **theft** by malicious actors

The **distribution** of keys presents logistical **challenges**, particularly in large-scale systems or distributed networks

Ensuring the **secure transfer** and storage of keys can be **complex** and error-prone.

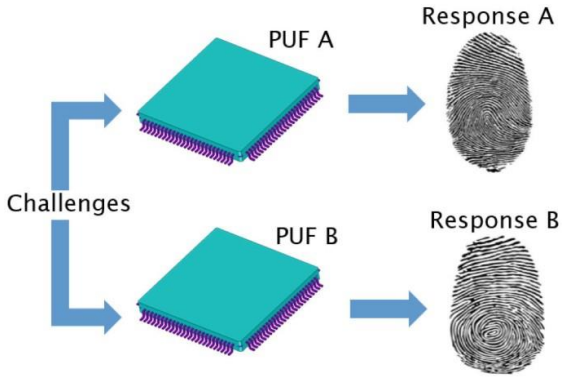
Physical Unclonable Functions

Physical Unclonable Functions (PUFs) have acquired attention for their ability to provide device authentication without relying on stored secrets.

PUFs exploit the **inherent physical variations** introduced by the manufacturing process to generate **unique identifiers** for each device.

PUFs are based on the **Challenge-Response** paradigm.

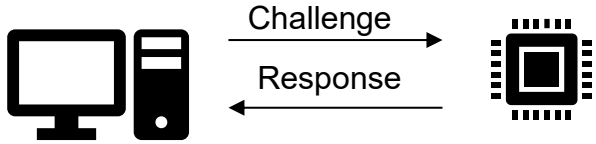
An example is the **SRAM-PUF** which exploit the SRAM cells imperfections for device authentication.



Physical Unclonable Functions

In a classical scenario, the process typically involves **two main phases**.

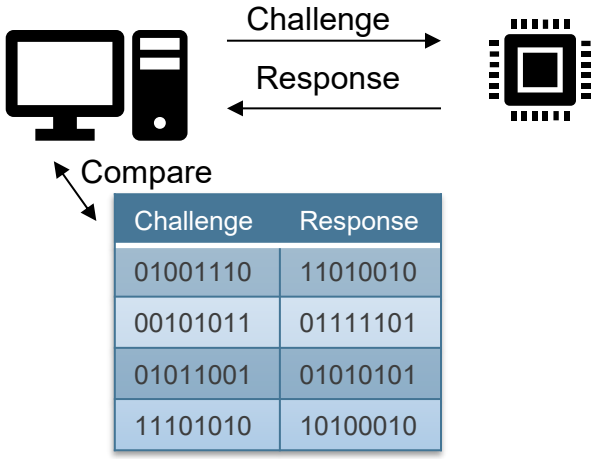
- **Enrollment Phase** involves capturing the inherent nanoscale imperfections within the circuitry.



Challenge	Response
01001110	11010010
00101011	01111101
01011001	01010101
11101010	10100010

Physical Unclonable Functions

- **Authentication Phase**: the PUF is utilized to verify the identity of a device.



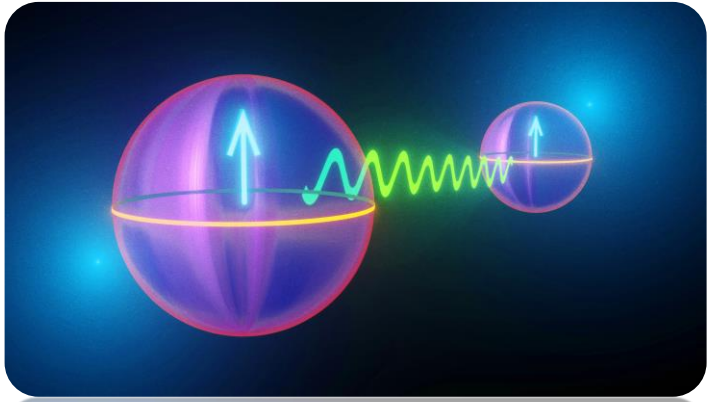
Challenge	Response
01001110	11010010
00101011	01111101
01011001	01010101
11101010	10100010

However, recent studies have revealed that some conventional PUF implementations suffer from security vulnerabilities, susceptibility to cloning, and vulnerability to machine learning-based attacks, casting doubts on their reliability.

Quantum Physical Unclonable Functions

Quantum PUFs (QPUFs) exploit the inherent errors in quantum devices to generate unique and unpredictable responses, offering heightened security compared to classical PUFs.

QPUFs offer distinct advantages rooted in the **no-cloning theorem**, ensuring that an arbitrary quantum state cannot be perfectly replicated without altering the original state.



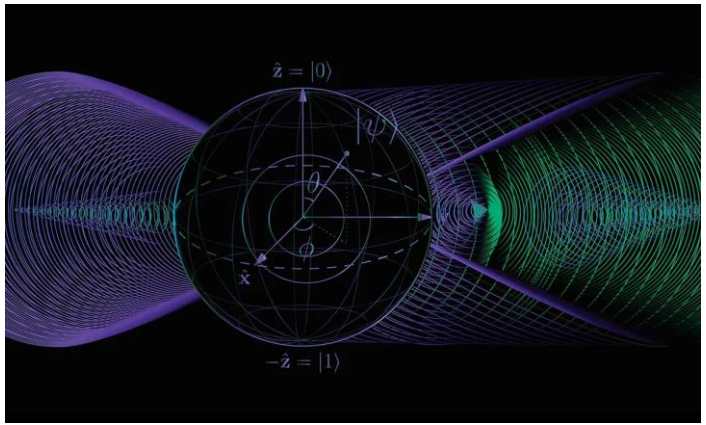
Quantum Physical Unclonable Functions

Today's Quantum computers are susceptible to **various types of quantum errors**, which stem from diverse sources such as manufacturing imperfections, control errors, environmental interactions.

There are several errors:

1. **Gate error** introduces a probability of error in logical operations.

2. **Decoherence** occurs as qubits interact with the environment, leading to state loss.



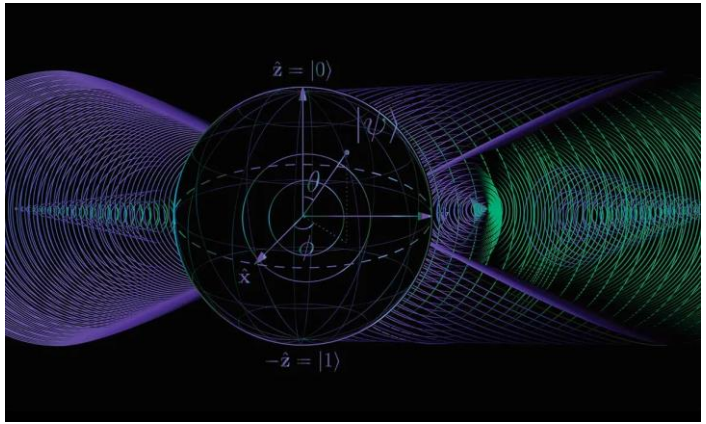
Quantum Physical Unclonable Functions

3. **Readout errors** may result from imperfections in readout circuitry, causing bit-flips.

4. **Single-qubit errors** arise from errors in single-qubit gates, such as the Hadamard gate or rotation gates.

5. **Two-qubit errors** stem from errors in two qubit gates, like the CNOT gate.

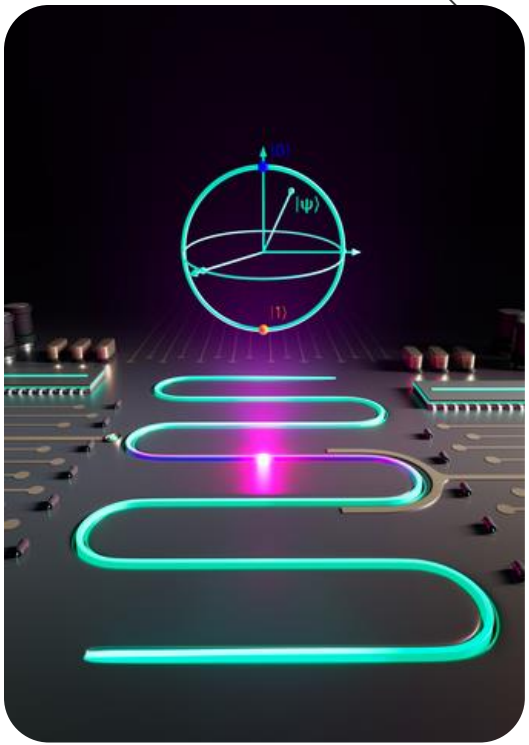
6. **Crosstalk** occurs when parallel gate operations on different qubits impact each other's performance.



Quantum Physical Unclonable Functions

The **rates of these errors** vary among qubits and hardware, providing a **unique signature** for identification.

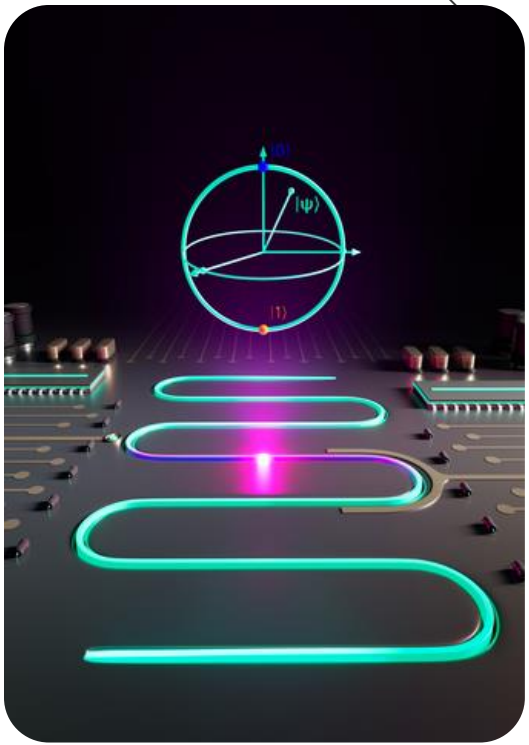
All these errors are inherently leveraged by QPUFs to generate unique responses for each device.



Quantum Physical Unclonable Functions

However, QPUFs also pose significant technical **challenges**, including the need for precise control over quantum states.

Their practical realization and integration into real-world applications require **further exploration** and technological advancements.



Objective

Our method aims to address the limitations of current technology in crafting an efficient QPUF system.

Challenges

Designing an effective QPUF circuit for authentication mechanisms is challenging due to the **complexity of managing quantum states**.

The **goal** is also to create a simple system that does **not** require **additional hardware**, such as quantum channels or quantum memory.

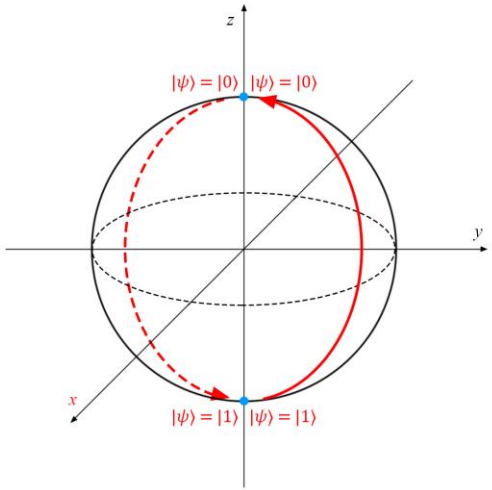
Objective

- **Quantum circuit** to function as a QPUF, evaluating instability, randomness, and uniqueness metrics to assess its efficacy.

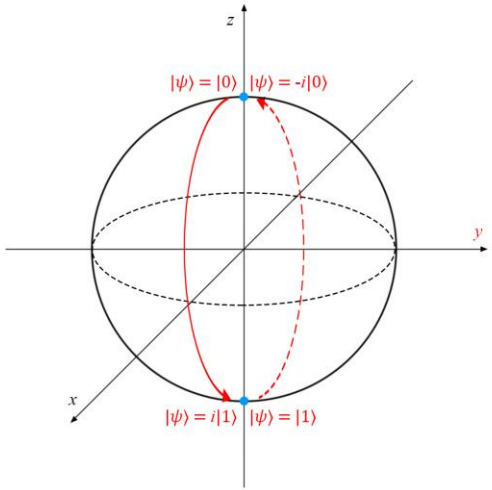
- **Authentication scheme** that leverages the challenge-response paradigm.

Proposed QPUF

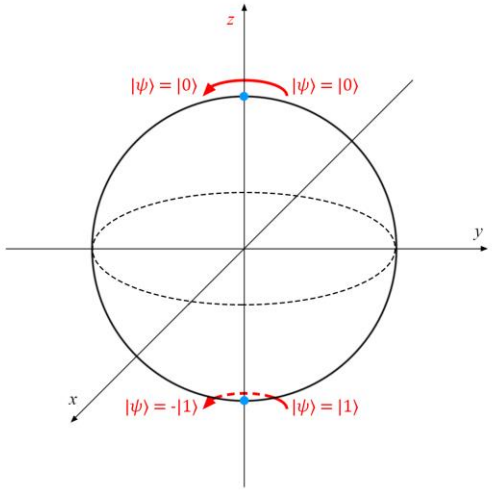
The proposed QPUF circuit leverages the inherent biasing of qubits towards the **1** or **0** state to generate the response. This biasing can arise from gate errors, including those associated with **X-Y-Z rotation** gates, or from **readout** errors.



Pauli X

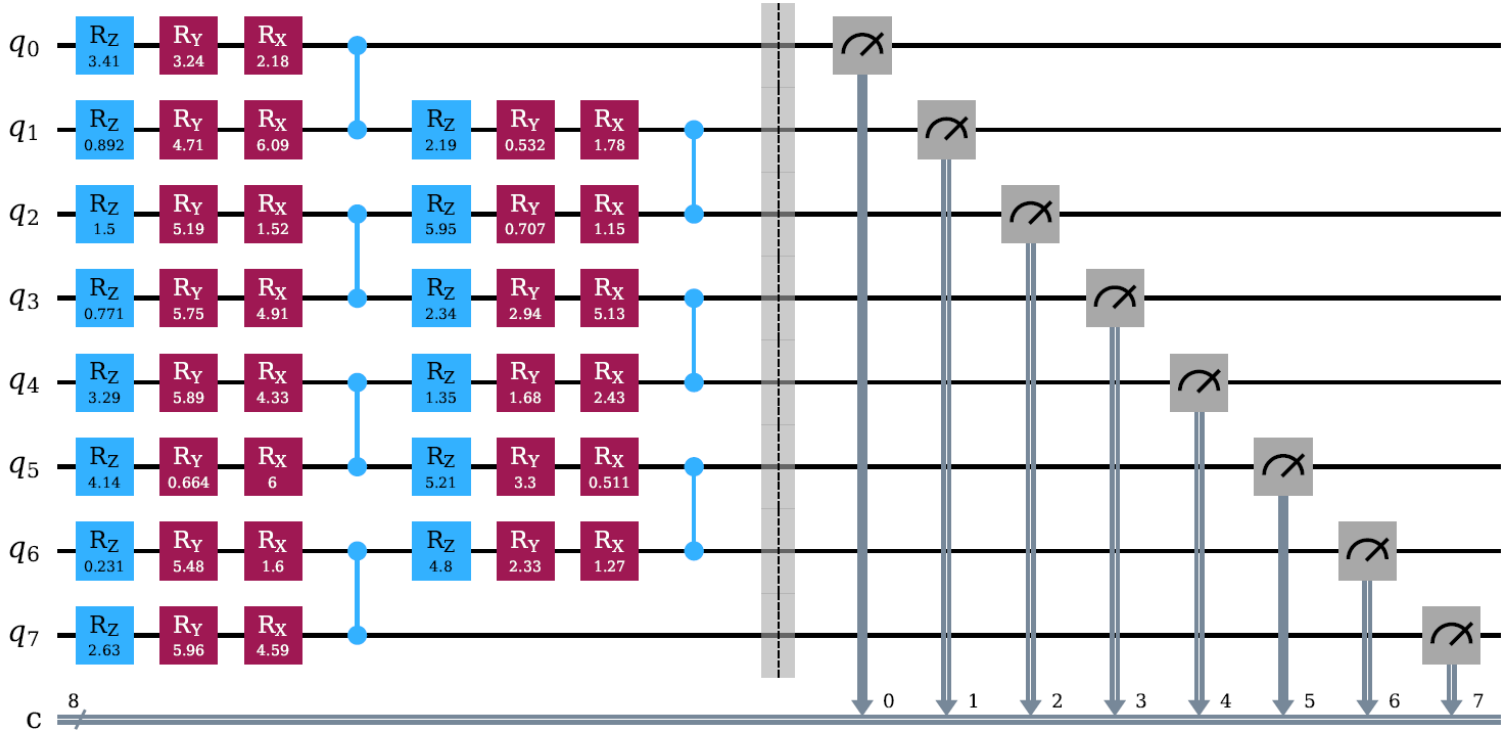


Pauli Y



Pauli Z

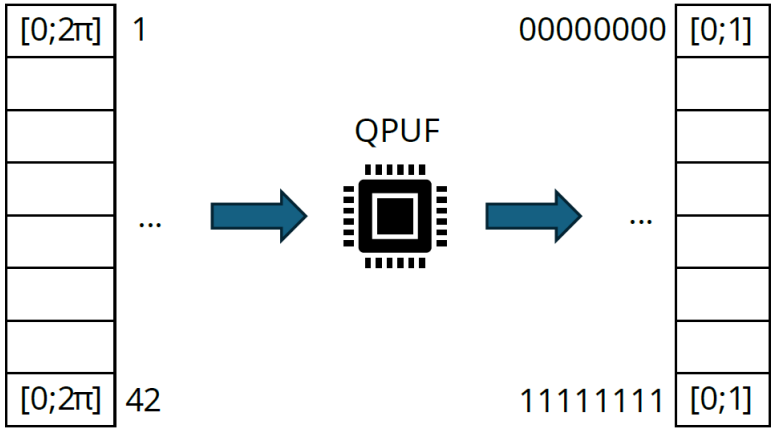
Proposed QPUF



Proposed QPUF

Challenges appear as arrays including all parameters for every individual gate within a single instance.

Each respective **response** constitutes the outcome of measuring all qubits following the execution of the circuit a specific number of times.



Proposed QPUF

The resultant circuit addresses issues delineated in literature, namely the requirement for **entangled qubits** facilitated by the utilization of CZ gates, and the necessity for a **heightened challenge space** achieved through the parameterization of X-Y-Z rotation gates.

Challenges and responses of the mechanisms are not of a quantum nature, implying that there is **no requirement for a quantum communication channel**, nor does the verifier need to possess **quantum memory** to store challenges and responses.

QPUF metrics

Instability refers to variations and unpredictability in the QPUF responses of a single device. If it is too high, can compromise the reliability and security of QPUF-based systems.

In order to evaluate such Instability, we can compare QPUF responses by means of a Normalized Absolute Probabilistic Distance (NAPD) over every response pair.

$$\mathrm{NAPD}\left(r_n^k, r_m^h\right) = \frac{1}{2}\sum_{q=1}^{Q} |r_{n,q}^k - r_{m,q}^h|$$

$r^k_{n,q}$ ->
- *q*-th combination of qubit results on $Q=2^{nqubits}$
- *n*-th challenge
- *k*-th device

QPUF metrics

Instability for the *k*-th device can be estimated as follows:

$$\text{Instability}\,(k) = \frac{1}{N}\sum_{n=1}^{N}\frac{2}{D\,(D-1)}\sum_{i=1}^{D}\sum_{j=i+1}^{D}\text{NAPD}(r_{n,i}^{k}, r_{n,j}^{k})$$

$r^{k}_{n,i}$ -> *i*-th output of D executions
n-th challenge out of N
k-th device

For each challenge and for each distinct pair of executions on the same challenge, NAPD is computed.

Ideally, Instability in QPUFs should be minimized, indicating consistent responses over time.

QPUF metrics

Randomness refers to the stochastic nature of responses, ensuring that each response is maximally distinct from another, even when presented with similar challenges.

The Randomness for the *k*-th device can be estimated as follows:

$$\text{Randomness}\,(k) = \frac{2}{N\,(N-1)} \sum_{n=1}^{N} \sum_{m=n+1}^{N} \text{NAPD}(r_n^k, r_m^k)$$

r^k_n -> *n*-th challenge out of N
k-th device

For each distinct pair of challenges, NAPD is computed.

Ideal Randomness values are ones as close to 1 as possible.

QPUF metrics

Uniqueness: denote that each generated output from every single device is distinct from outputs of all other devices, giving the same input.

The Uniqueness can be estimated as follows:

$$\text{Uniqueness} = \frac{2}{K\,(K-1)} \sum_{k=1}^{K} \sum_{h=k+1}^{K} \frac{1}{N} \sum_{n=1}^{N} \text{NAPD}(r_n^k, r_n^h)$$

r^k_n -> *n*-th challenge out of N
k-th device out of K

For each distinct pair of devices and for each challenge, NAPD is computed.

Ideal Uniqueness values are ones as close to 1 as possible.

Experimental results

Real quantum hardware provided by **IBM**, namely **ibm_brisbane**, **ibm_kyoto**, and **ibm_osaka**, has been employed, leveraging the **Qiskit** SDK.

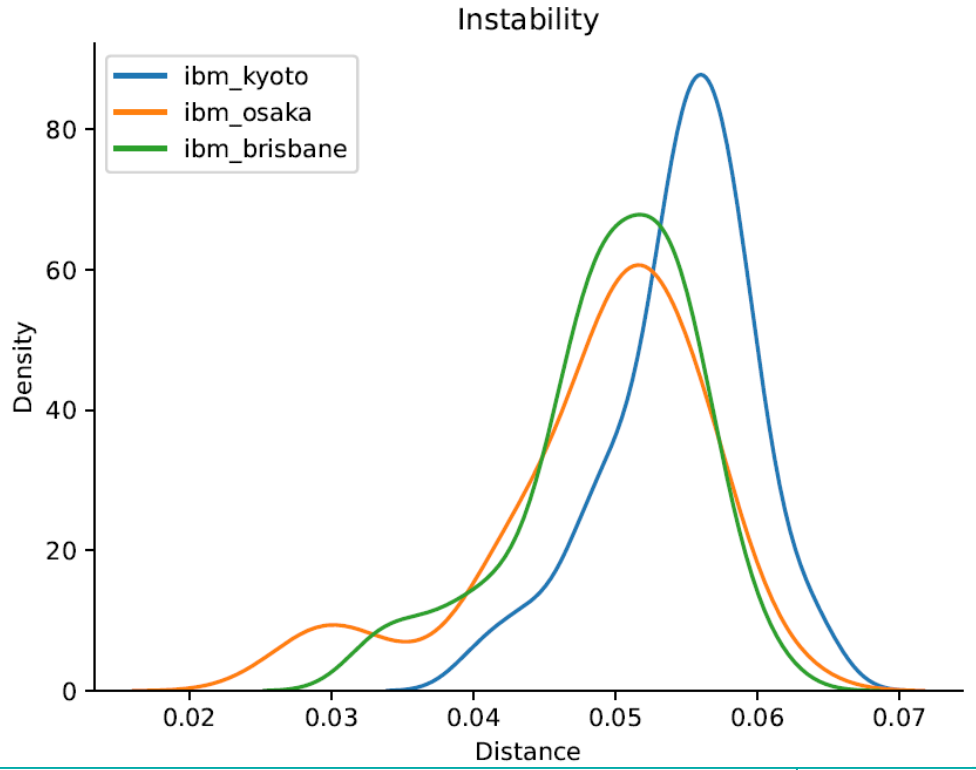
In general, the QPUF circuit described has been employed with a qubit count of **8** and a measurement **shot** count of **20,000** per circuit.

Experimental results

Specifically, concerning **Instability** testing, a set of **10** challenges (N) and **5** executions each (D) were utilized.

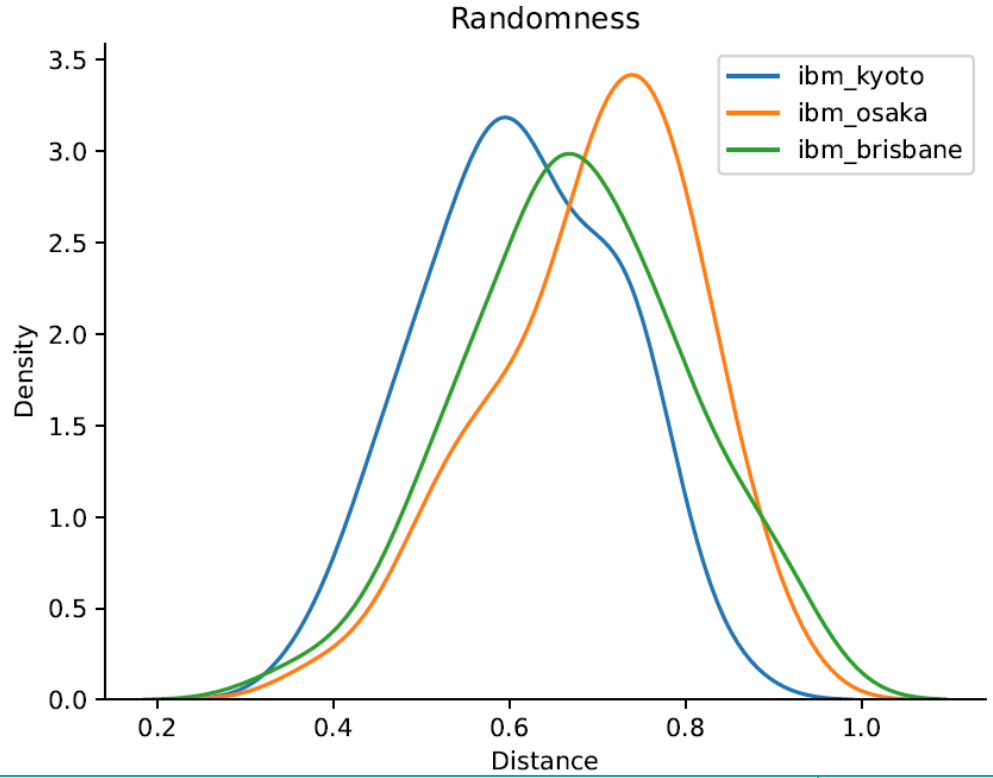
It has been observed that increasing the number of shots or the number of executions leads to a decrease in Instability values.

Therefore, without limitations on the utilization of quantum computers, the results can certainly be improved.



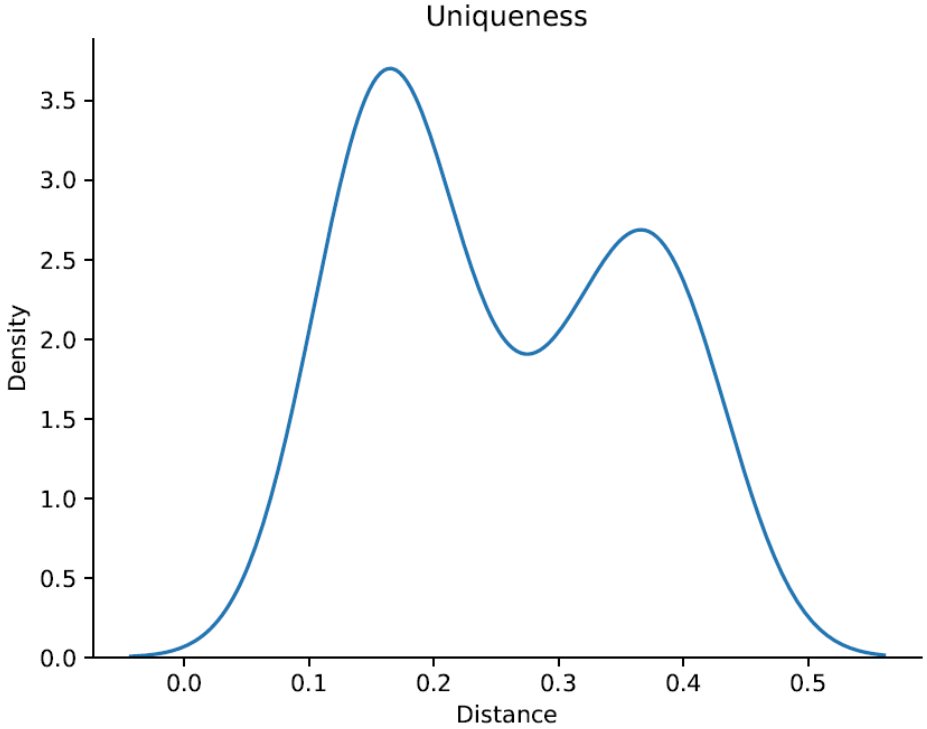
Experimental results

Concerning **Randomness** testing, the parameter N is set to **10**.



Experimental results

Concerning **Uniqueness** testing, the parameter N is set to **10** and K is **3** for the three IBM devices used.



The experiments are replicable, and the source code is available on https://github.com/francocirill/QPUF

Analysis

It is essential to consider that the derived Instability **values** serve as an **upper bound**, given the continuous improvement observed by increasing the number of executions of the quantum circuit.

The achieved results aimed at **obtaining acceptable values** rather than aiming for the best possible outcomes, primarily due to limitations in utilizing the quantum computing service.

Analysis

Randomness and Uniqueness values are not bad, especially considering that achieving the ideal value of 1 poses a significant challenge.

This is because it would require each response to have a probability of 1 for only one combination of qubits and 0 for all others.

Overall, the metric graphs indicate low Instability, good Randomness and a satisfactory Uniqueness. However, the Uniqueness of devices **can be** further **enhanced by** employing the **authentication scheme** described subsequently.

Device authentication scheme

Like a classical PUF authentication method, the QPUF authentication scheme proposed is composed of two different operations.

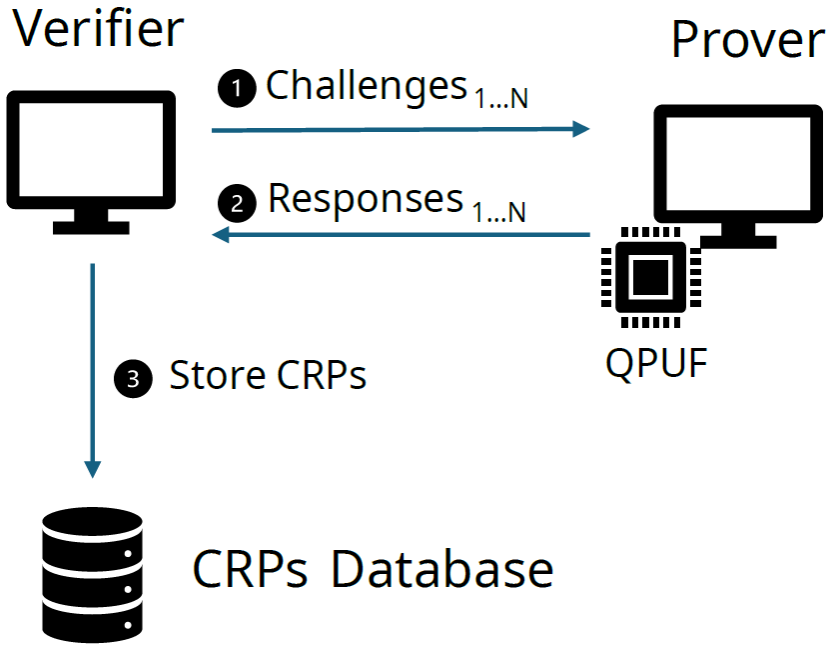
An **enrollment phase**, where QPUF information is stored by the verifier, and an **authentication phase**, where the prover prove his identity using its QPUF.

Device authentication scheme

Enrollment

During the **enrollment phase**, the verifier randomly generates a number N of challenges and evaluates the responses using the QPUF of the device in a secure environment, ensuring that information cannot be captured and remains safe.

The verifier store the resulting CRPs in a table.

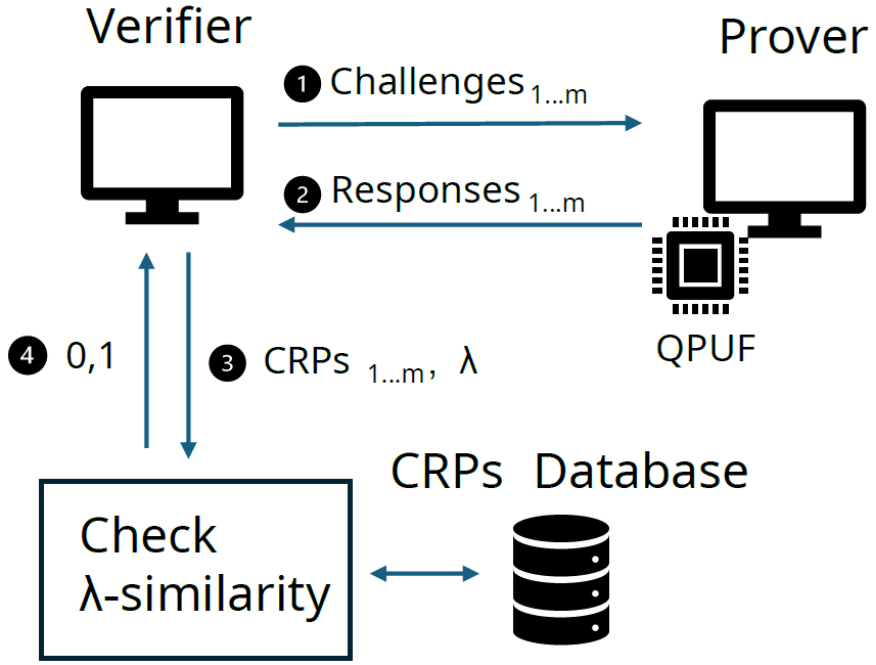


Device authentication scheme

Authentication

The verifier transmits m challenges, typically selected randomly.

These challenges are conveyed via a public classical channel to the prover, which elicits the responses to the received challenges exploiting the QPUF mechanism and send them to the Verifier.

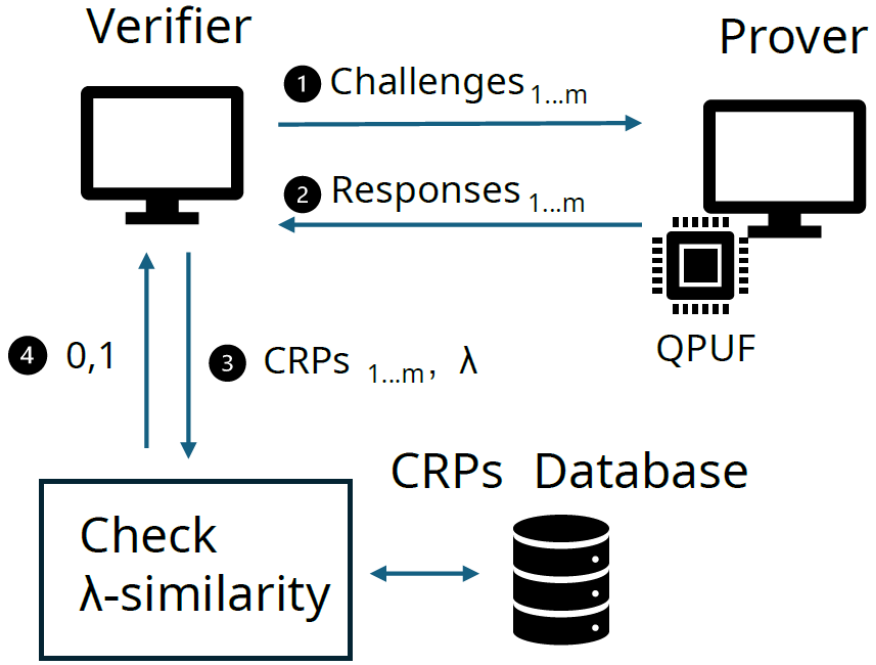


Device authentication scheme

Authentication

Finally, the verifier checks whether all the received responses are similar to those stored in the local database.

Upon affirmation, the prover is authenticated; otherwise, no information regarding the similarity of the responses is disclosed, and the process halts.



Device authentication scheme

Similarity

Two responses are considered similar if their NAPD does not exceed a certain threshold parameter, **λ**, that can be determined through instability analysis.

In our case, based on experimental results, **λ** is set to **0.07**, corresponding to **7%**.

This implies that two responses are considered similar only if their distance does not exceed 7%, as this was the maximum instability observed between pairs of responses for the same challenge and device.

Authentication scheme analysis

It is essential to recall that the parameter **λ can be** further **lowered** by conducting a more detailed instability analysis, thus performing **multiple shots** of circuit executions.

The security of the scheme can be enhanced by **increasing** the **number of challenges** required for the authentication, as only a response indicating whether the authentication was successful or not is returned.

Therefore, no information about the correctness of each individual response is provided; an attacker would need to guess all the responses to find out whether they are all correct.

Thank you!

Christian Esposito

QUANTUM TECHNOLOGIES FOR ANOMALY DETECTION IN FUTURE NETWORKS

Contact: esposito@unisa.it

NGI SARGASSO

Hardware Assisted and Blockchain Empowered Security Lab