The 39th Annual AAAI Conference on Artificial Intelligence

FEBRUARY 25 – MARCH 4, 2025 | PHILADELPHIA, PENNSYLVANIA, USA

Quantum Machine Learning Techniques for Network Intrusion in Software-Defined Networks

Joan Lo[1], **José A. Lázaro**[1], Josep R. Casas[1], Javier Ruiz-Hidalgo[1], Àlex Solé[1], Samael Sarmiento[3], Adolfo Lerín[4], Ricardo Martínez[2], Josep M. Fàbrega[2]

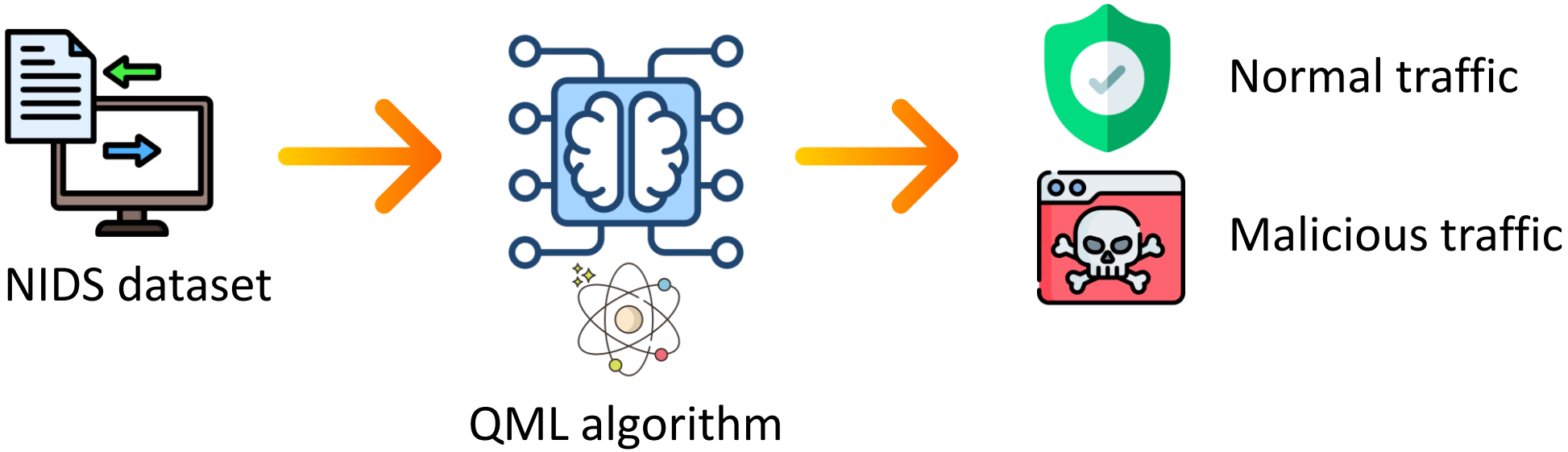
[1]Universitat Politècnica de Catalunya – UPC, Barcelona, Spain
[2]Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Castelldefels, Spain
[3]LuxQuanta Technologies S.L., Castelldefels, Spain
[4]CognitIAs, Madrid, Spain

6GEWOC

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

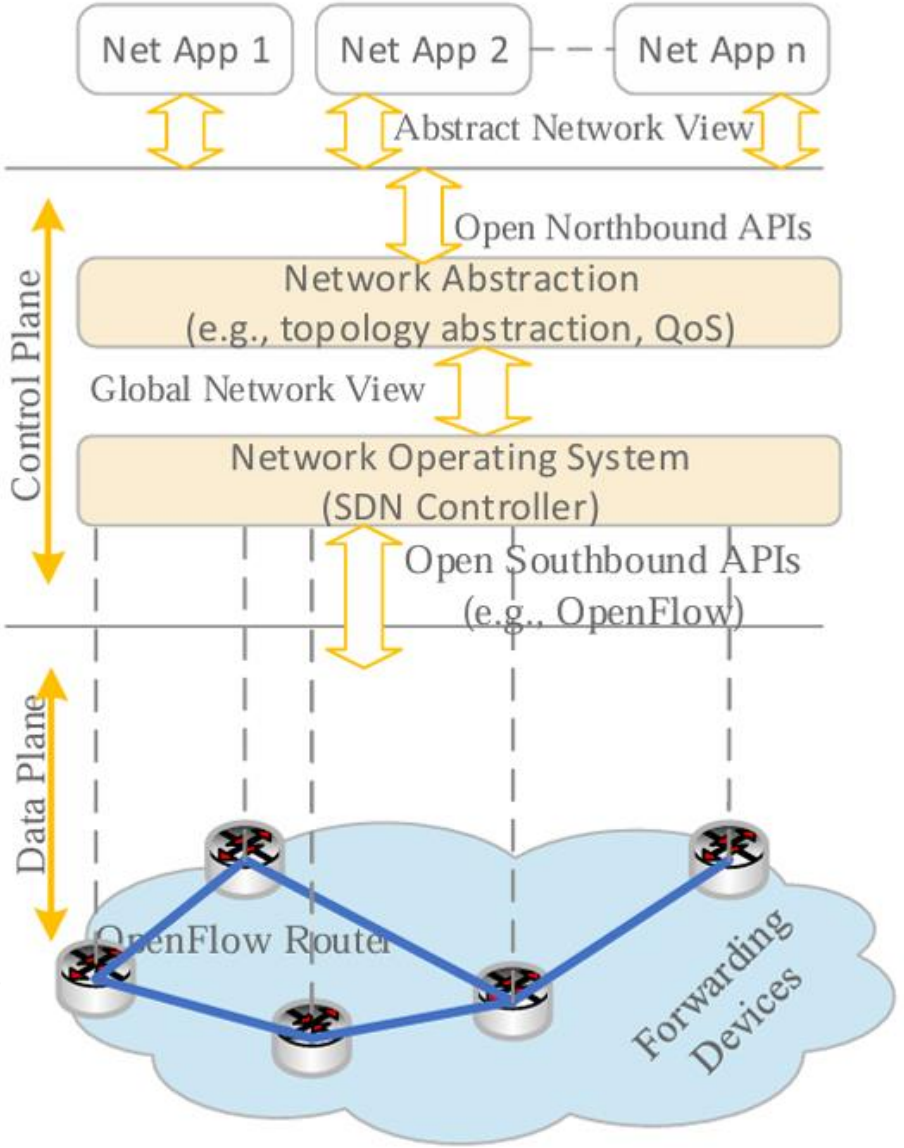
Introduction

- This work explores the application of **Quantum Machine Learning (QML)** algorithms for **Network Intrusion Detection Systems (NIDS)** in **Software-Defined Networks (SDN)**, comparing their performance with **classical machine learning methods**
- The objective is to classify network attacks from a NIDS dataset



Software Defined Networking (SDN)

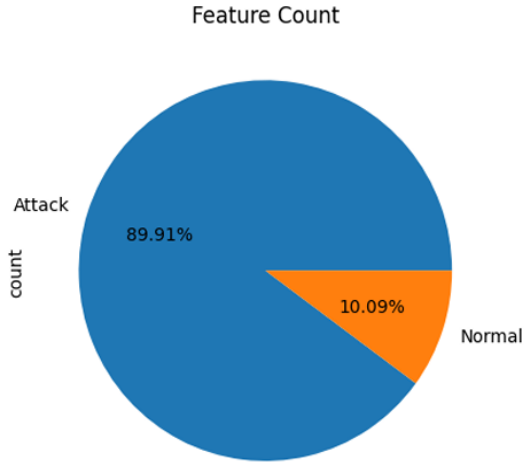
- Network traffic volumes and patterns increase every day → necessity of a **new SDN paradigm** (started to gain recognition in 2012)
- **OpenFlow** protocol paper[1] was published in 2008
- Classical Networks have **decentralized behavior**
- Software-Defined Networks have **centralized behavior**

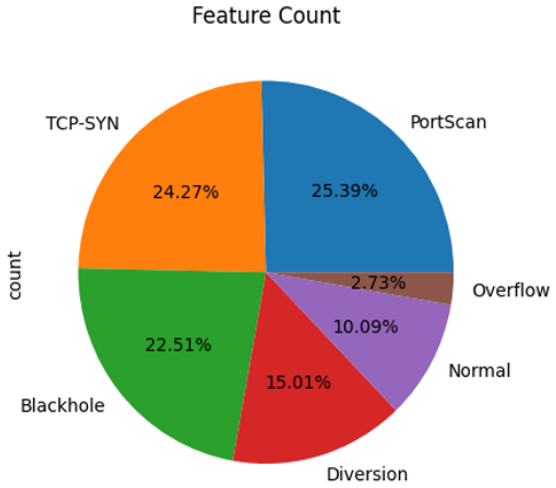


1: Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. "OpenFlow: enabling innovation in campus networks". SIGCOMM Comput. Commun. Rev. 38, 2 (April 2008), 69–74.
2: Image ref: Hakiri, Akram & Gokhale, et. al. (2014). "Software-Defined Networking: Challenges and research opportunities for Future".

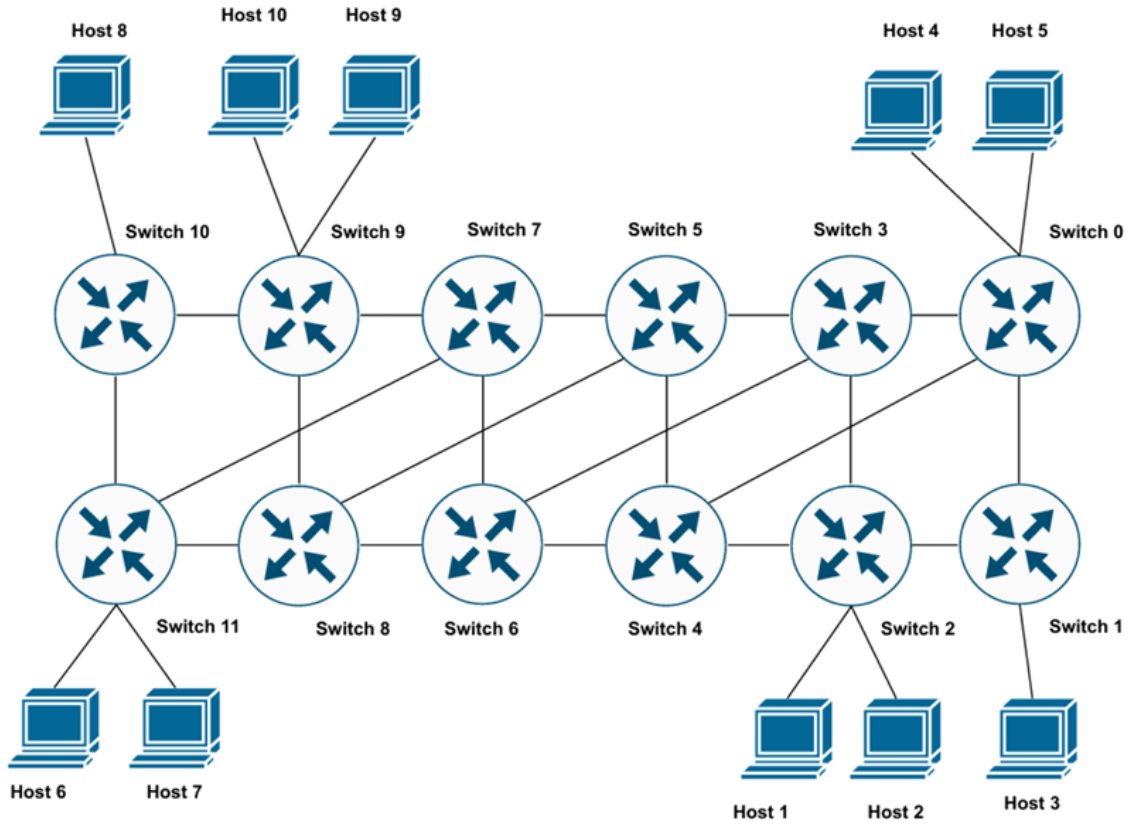
Network Intrusion Detection Dataset (UNR-IDD)

- University of Nevada - Reno Intrusion Detection Dataset (**UNR-IDD**).
- **37412 samples** of **34 features** (as Packets Rx/Tx Dropped, flow entries, etc.[1]).
- **Class Labels:** Normal (10%), Attack (90%): TCP-SYN, PortScan, Overflow, Diversion and Blackhole).



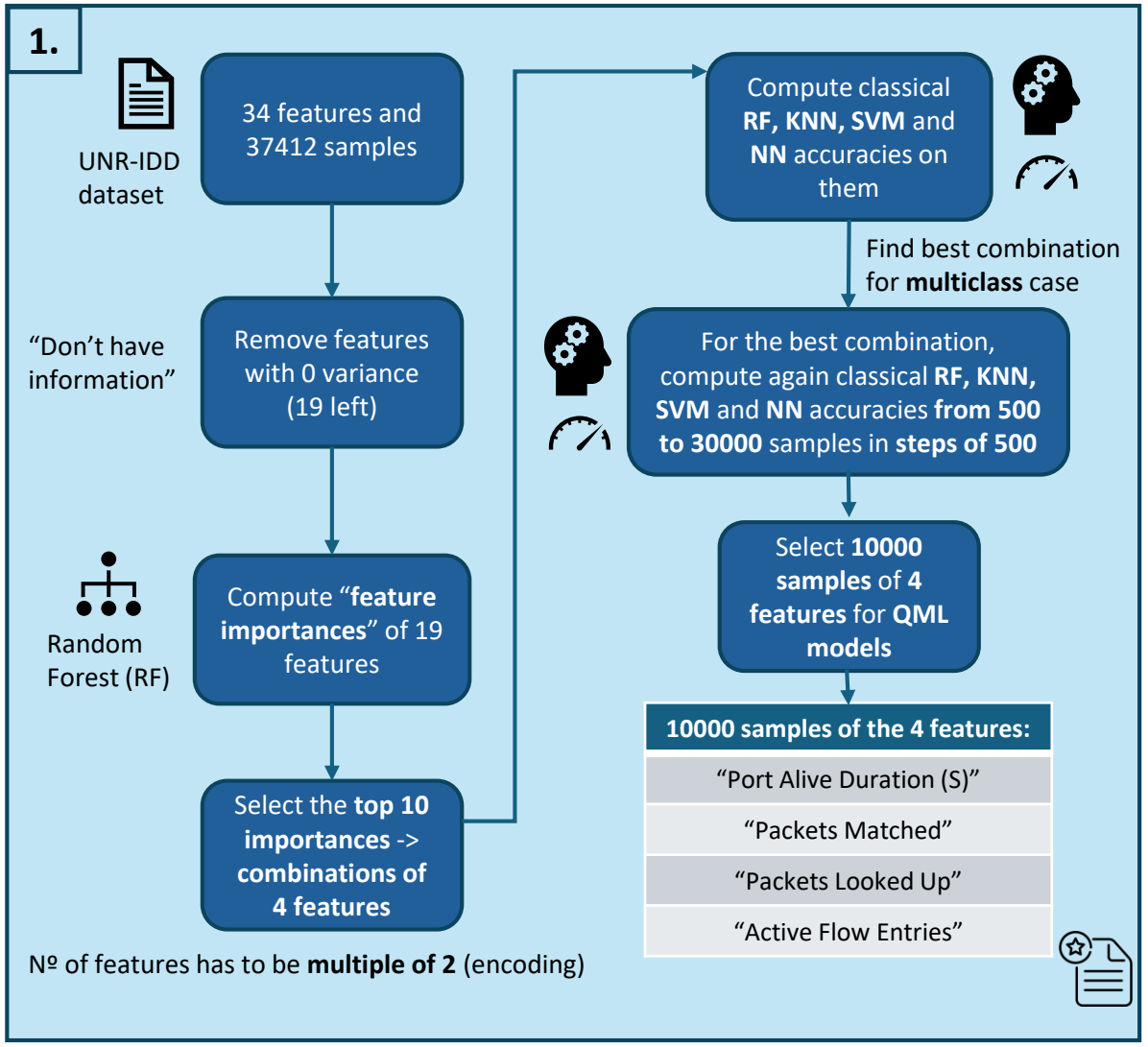


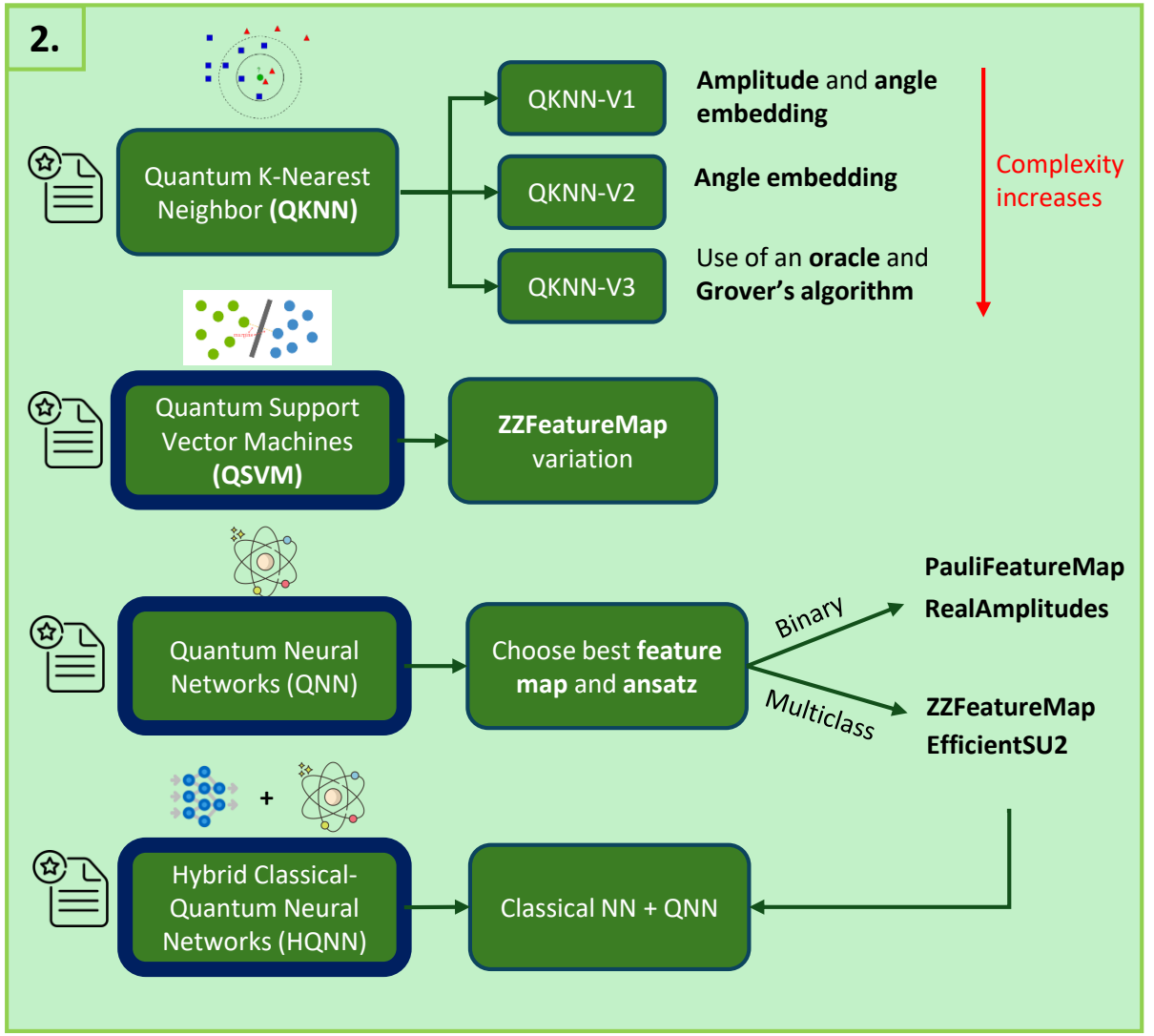
UNR-IDD SDN topology



1: Image ref: T. Das, O. A. Hamdan, R. M. Shukla, S. Sengupta and E. Arslan, "UNR-IDD: Intrusion Detection Dataset using Network Port Statistics".

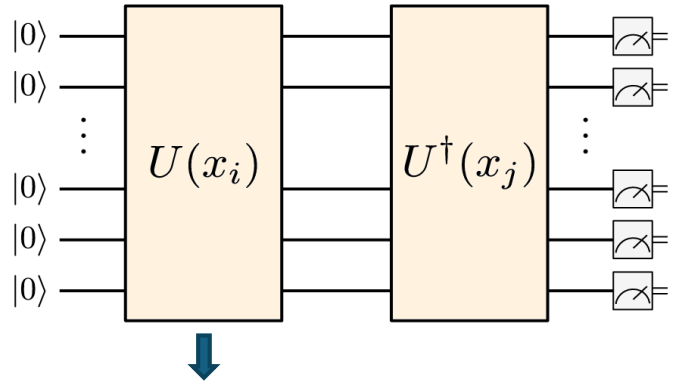
Methodology steps: Reduce & Train

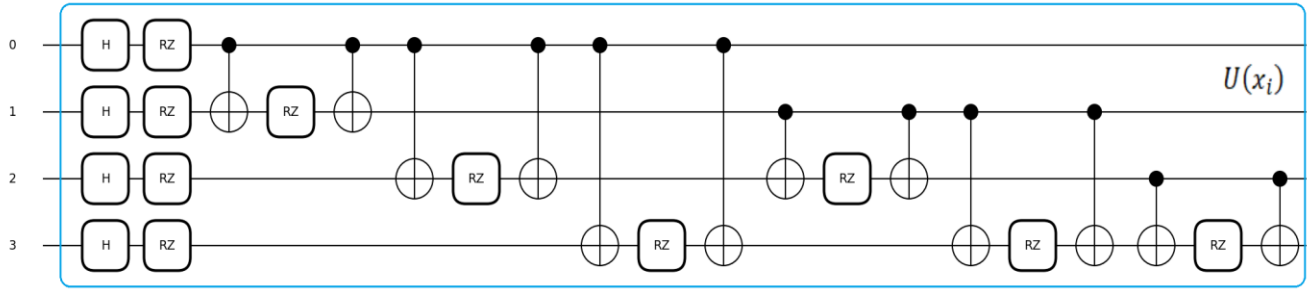


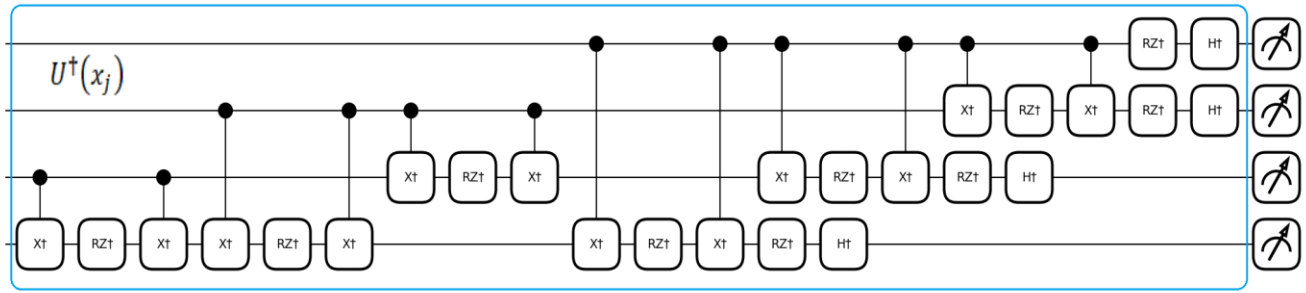


QML Algorithms for NIDS: QSVM

- Maximize objective function: $L(\alpha) = \sum_{j=1}^{M} y_j \alpha_j - \frac{1}{2} \sum_{j,k=1}^{M} \alpha_j K_{jk} \alpha_k$
- Quantum kernel trick: $K(x_i, x_j) = |\langle \phi(x_i) | \phi(x_j) \rangle|^2$; $|\phi(x)\rangle = U(x)|0^n\rangle$
- Customized version of ZZFeatureMap[1]
- $G = \begin{bmatrix} K(x_1, x_1) & \dots & K(x_1, x_M) \\ \vdots & \ddots & \vdots \\ K(x_M, x_1) & \cdots & K(x_M, x_M) \end{bmatrix}$







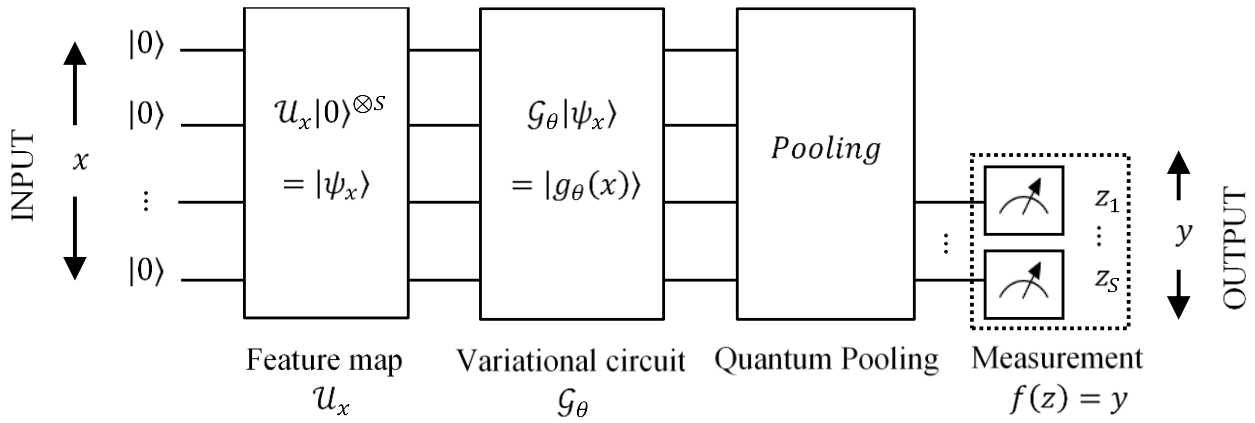
Run classical SVM with precomputed K:

- **Train:** compute G with both $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as train samples
- **Test:** compute G with $\boldsymbol{x}_i$ as test samples and $\boldsymbol{x}_j$ as train samples

1: Qiskit Contributors. (2024). Qiskit Documentation. https://docs.quantum.ibm.com/api/qiskit

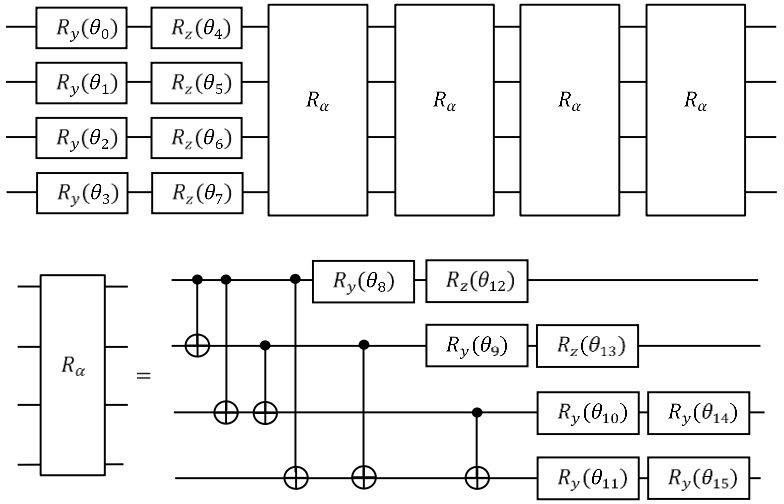
QML Algorithms for NIDS: QNN

QNN architecture scheme

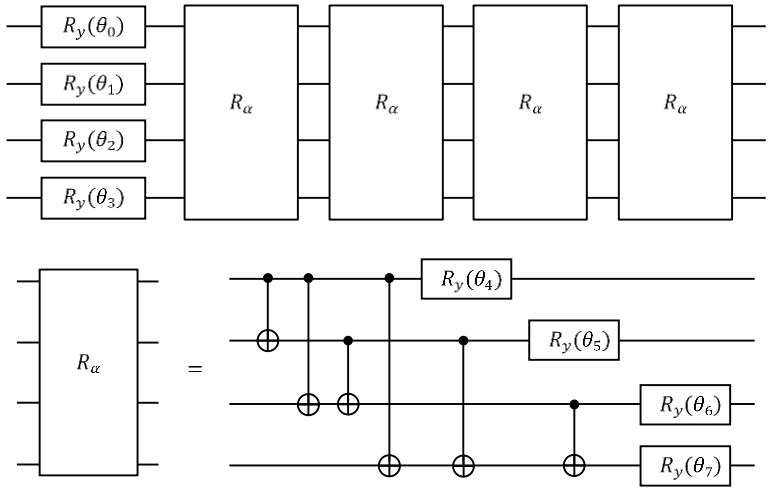


- **Binary** classification (best):
  - PauliFeatureMap[1] ($\mathcal{U}_x$) + RealAmplitudes[1] ($\mathcal{G}_\theta$)
- **Multiclass** classification (best):
  - ZZFeatureMap[1] ($\mathcal{U}_x$) + EfficientSU2[1] ($\mathcal{G}_\theta$)
- **COBYLA** optimizer for 600 iterations
- **Binary L2** and **normal cross-entropy** loss functions
- **Parameter-shift rule** for backpropagation

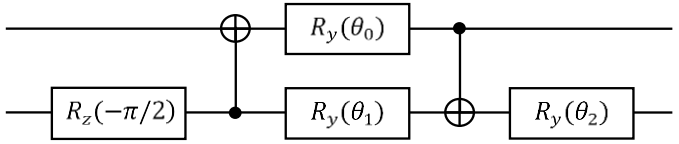
EfficientSU2 variational circuit $\mathcal{G}_\theta$



RealAmplitudes variational circuit $\mathcal{G}_\theta$



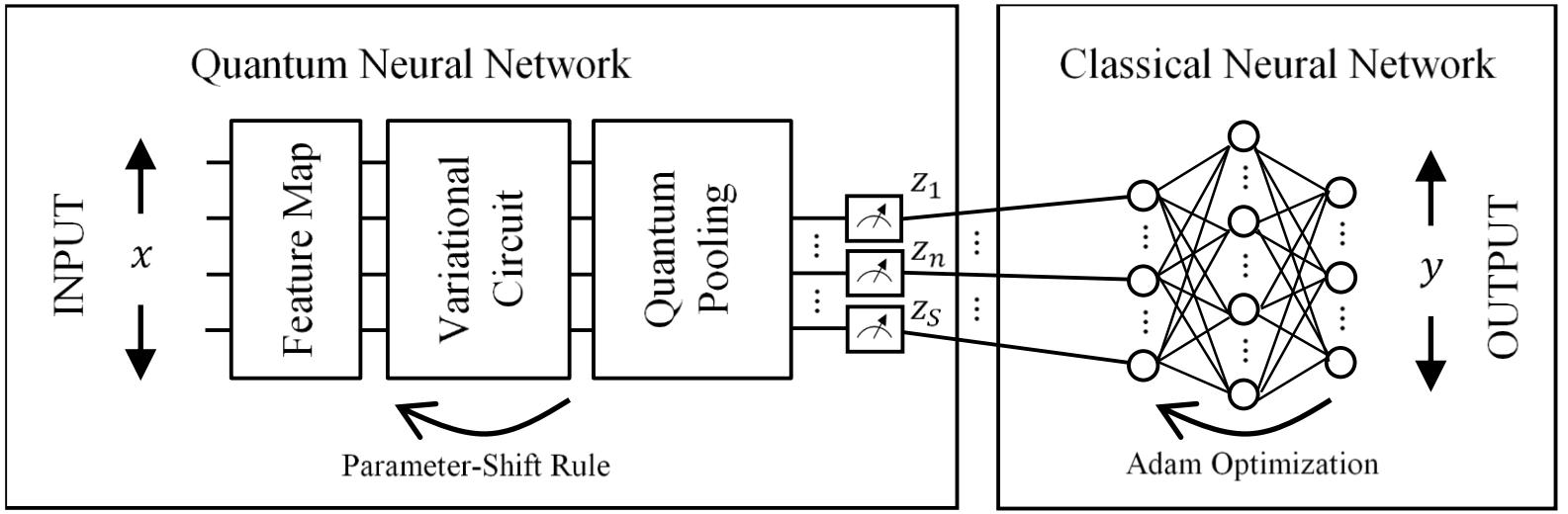
Quantum Pooling layer



1: Qiskit Contributors. (2024). Qiskit Documentation. https://docs.quantum.ibm.com/api/qiskit

QML Algorithms for NIDS: HQNN

HQNN architecture scheme

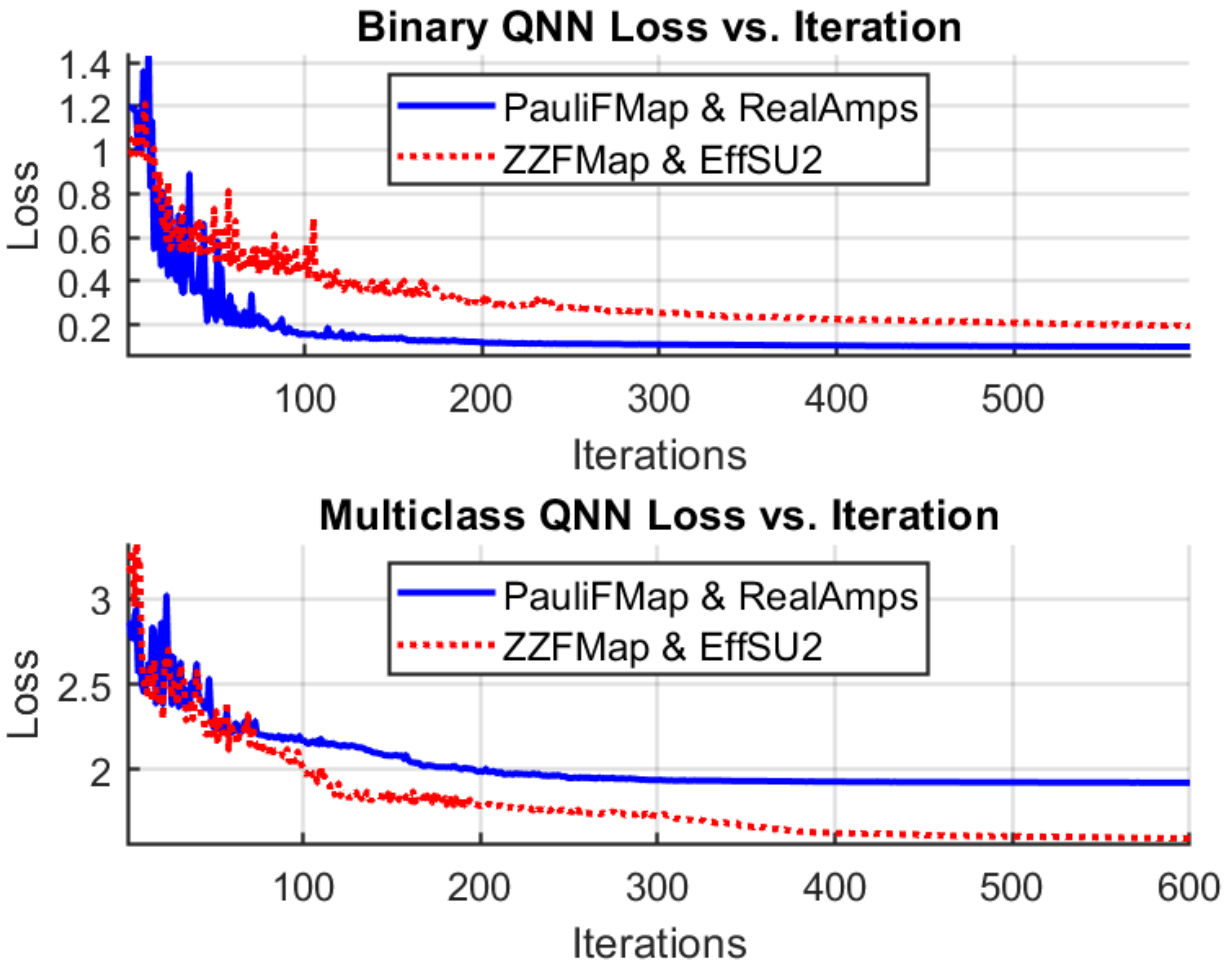


- Reuse of QNN feature map and variational circuit best combinations
- **Dropout** and **batch normalization**
- **Binary** and **normal cross-entropy** loss functions
- **50 epochs** of training

Results: QNN Training Losses

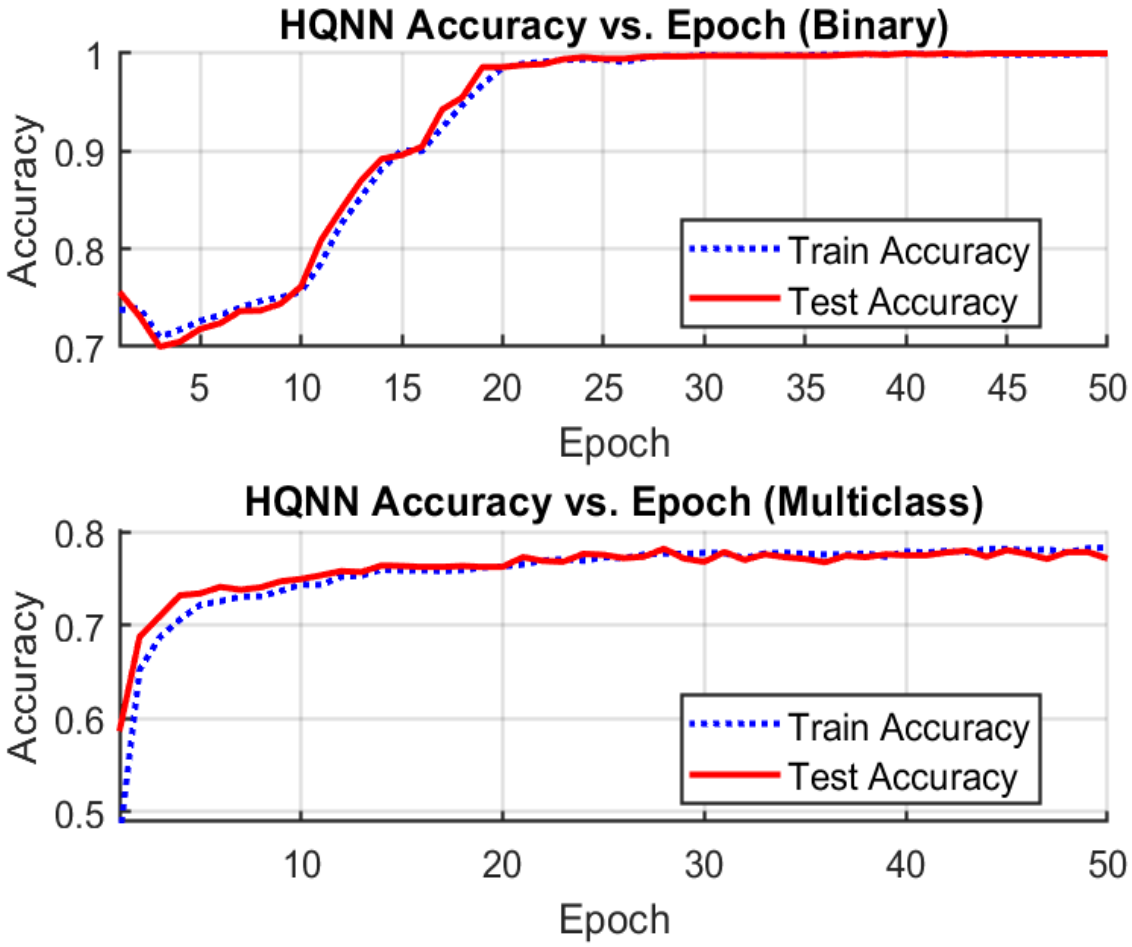
- The combination of **ZZFeatureMap** and **EfficientSU2** shows greater difficulty in reducing the loss compared to the **PauliFeatureMap** and **RealAmplitudes** combination for the binary case

These best circuit combinations are then utilized for HQNN in binary and multiclass, respectively



Results: HQNN Training

- Binary implements **ZZFeatureMap** and **EfficientSU2**
- Multiclass implements **PauliFeatureMap** and **RealAmplitudes**
- During training process, train and test accuracies are compared
- In binary case, the accuracy starts to stabilize around epoch 20
- In multiclass case, the accuracy has a steep increment until around epoch 5, to then increase slowly



Results: Overall Classical and Quantum models

Classical Models	Classification Type	Accuracy	$\overline{F1}$ score
RF	Binary	100%	100%
	Multiclass	87.35%	87.71%
KNN	Binary	100%	100%
	Multiclass	81.45%	89.35%
SVM	Binary	100%	100%
	Multiclass	**69.65%**	67.73%
NN	Binary	99.50%	98.64%
	Multiclass	**77.10%**	73.25%

Quantum Models	Classification Type	Accuracy
QKNN-V1 (Ang.)	Binary	100%
	Multiclass	46.80%
QKNN-V1 (Amp.)	Binary	90.05%
	Multiclass	19.80%
QKNN-V2	Binary	100%
	Multiclass	56.90%
QKNN-V3*	Binary	85%
	Multiclass	55%
QSVM	Binary	100%
	Multiclass	**72.40%**
QNN	Binary	98.90%
	Multiclass	63.55%
HQNN	Binary	100%
	Multiclass	**78.24%**

*200 samples

Conclusions & Future Research

Considering a **reduced number of 4 features** (“Port Alive Duration (S)”, “Packets Matched”, “Packets Looked Up” and “Active Flow Entries”), this **benchmarking** of quantum and classical ML algorithms shows that QML provides better accuracy for:

- **QSVM** reaches 72.40% ahead of the 69.65% achieved by classical SVM
- **Hybrid solution HQNN** reaches 78.24%, surpassing the 77.10% achieved with classical NN

Future Research is required in order to:

- Analyze accuracy of both CML and QML with **higher number of features**
- Test in **real quantum computer** for checking computational performance in a more realistic scenario
- **Further explore** other QML algorithms and model architectures

Thank you!

Get in Touch with Us!

Joan Lo Anguera

José Antonio Lázaro

UPC – Universitat Politècnica de Catalunya - BarcelonaTech

✉ jose.antonio.lazaro@upc.edu

✉ joan.anguera@upc.edu

https://6G-ewoc.eu

6G-ewoc-project

Spanish MICIU founded, TRAINER-B (PID2020-118011GB-C22).

MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES

AGENCIA ESTATAL DE INVESTIGACIÓN

6G EWOC

Co-funded by the European Union 6G SNS

The 6G-EWOC project has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101139182.