Towards Network Intrusion Detection via Quantum Machine Learning: A Reality Check

Conference Paper · February 2025

CITATIONS
0

READS
105

4 authors:

Vincenzo Spadari
University of Naples Federico II
6 PUBLICATIONS **6** CITATIONS
SEE PROFILE

Idio Guarino
University of Naples Federico II
16 PUBLICATIONS **86** CITATIONS
SEE PROFILE

Domenico Ciuonzo
University of Naples Federico II
165 PUBLICATIONS **6,490** CITATIONS
SEE PROFILE

Antonio Pescapè
University of Naples Federico II
327 PUBLICATIONS **15,840** CITATIONS
SEE PROFILE

Towards Network Intrusion Detection via Quantum Machine Learning: A Reality Check

Vincenzo Spadari*, Idio Guarino†, Domenico Ciuonzo*, Antonio Pescapè*
*University of Napoli Federico II, †University of Verona
{vincenzo.spadari, domenico.ciuonzo, pescape}@unina.it, idio.guarino@univr.it

***Abstract*—*Network security* is a theme facing a continuous transformation, due to the diversity of users and devices that populate the Internet. On the technology side, *quantum computing* represents a reality in progress, offering new solutions and applications. Among these, *Quantum Machine Learning* (QML) is a good candidate to be employed in network security, thanks to benefits like computation speed-up and efficient treatment of big volumes of data. In this paper we analyze the effectiveness of two classical QML approaches (named AMPE and ANGE) in *Attack Classification* (AC) and *Misuse Detection* (MD) scenarios, comparing with two DL approaches (named 1D-CNN and HYBRID). Two popular and publicly available IOT security-aware datasets, i.e., IOT-NIDD and EDGE-IIOT, are considered for experimental evaluation. Moreover, we further examine the algorithms by performing a *cross-evaluation*, to test robustness of such models in network contexts they were not explicitly trained for. The experimental campaign we conduct shows how QML can represent a valid choice for the deployment in IOT network intrusion detection systems.**

***Index Terms*—Deep Learning, Network Intrusion Detection System (NIDS), Network Traffic Analysis, Quantum Machine Learning.**

I. INTRODUCTION

Nowadays, *network security* is an increasingly felt necessity. The use of the Internet is extensively widespread due to the spawn of new users and new devices, and the scope of this demand is growing up. Computers, smartphones, and disparate kinds of systems contribute to the introduction of traffic into the web. Noticeably, over the past few years, *Internet-of-Things* (IOT), representing the massive access to the Internet of common usage devices (e.g., home appliances, TV sets, smartwatches) has assumed relevance, participating in the growth of the mole and the transformation of traffic [1]. In such a context, where traffic intensifies, also new network threats emerge [2, 3], and should be managed properly. Therefore, *network traffic analysis* becomes a fundamental tool. To cover network management requirements, traffic analysis includes some relevant tasks like (i) *traffic prediction* (TP), i.e., to comprehend in advance how traffic evolves, (ii) *traffic classification* (TC), i.e., to identify traffic by marking the app generating it or the kind of traffic (e.g., *malware attack, type of service*), and (iii) *anomaly detection* (AD), i.e., to spot anomalies by the knowledge of usual behavior.

Recently, *Artificial Intelligence* (AI) and, more in detail, *Machine Learning* (ML) and *Deep Learning* (DL) techniques have risen as the most effective to address attack classification task, tackling the actual limits of older methods, e.g., *port-based identification*, *deep packet inspection*, become ineffective due to the evolution of traffic characteristics. The primary helpfulness of ML/DL approaches stands in their ability to be instructed in automatically recognizing patterns in data. Despite its benefits and its suitability to the problem, DL training is *time-expensive*, especially considering the *amount of data* required.

At another technological latitude, *quantum computing* [4, 5], is representing a subject of innovation, by the revision of the classical computer science standards. Thanks to its principles, it is able to obtain results over the traditional computing capabilities. *Quantum Machine Learning* (QML) [6, 7, 8] is a well-known application in this research field, letting quantum circuits exploit ML processes, through *potentially more efficient training and inference* sessions. QML is poised to emerge as a strong competitor, with the potential to surpass certain limitations of classical ML by adapting effectively to each problem. By combining the classical handling of data collection and pre-processing to a quantum computation phase, QML represents an option in network traffic analysis, significantly in security problem solving, where research is always seeking new valid supports. Notwithstanding the several works [9, 10, 11, 12, 13] that handle network security questions by the employment of QML techniques, the assessment often *leverages outdated datasets*. Furthermore, *the heterogeneity of contexts to test the models*, at the best of our knowledge, is not explored.

Hence, **our main contribution** is: (i) *a comparison between different* QML *and* DL *algorithms* on a realistic IOT network security use case, including an investigation on the capabilities in *Attack Classification* (AC) and *Misuse Detection* (MD) scenarios, with the *project of classification algorithms* at bi-directional flow level accounting an *early inference constraint* (i.e., to run each prediction on the initial packets within each biflow) (ii) *a robustness analysis of* QML *algorithms* in

This work is partially supported by the "PNRR ICSC National Research Centre for High Performance Computing, Big Data and Quantum Computing (CN00000013)", under the NRRP MUR program funded by the NextGenerationEU. Also, this study is partially carried out within the "xInternet" project—funded by the Ministero dell'Università e della Ricerca—within the PRIN 2022 program (D.D.104–02/02/2022). This manuscript reflects only the authors' views and opinions and the Ministry cannot be considered responsible for them.

contexts different than the one they were trained in (viz. *cross-evaluation*), with a comparison with the DL counterparts.

The paper is organized as follows: Sec. II explores state-of-the-art ML/DL and QML applications in network traffic analysis, particularly aiming at IoT security contexts; Sec. III explains our QML-based methodology, followed by the experimental setup in Sec. IV; Sec. V offers the experimental findings, and Sec. VI ends up with a summary and future directions of research work.

II. Related work

The set of methods provided by traffic analysis sustains network management for many purposes, e.g., capacity planning, quality of service, and security. Thanks to their peculiarities, ML and DL rise as dominant tools, particularly in IoT security contexts [14, 15]. The study moves after both MD [16, 17] and AC [18, 19, 20, 21].

The need for quick response in handling cyber-threats move to investigate thoroughly traffic through *early traffic classification*, as reported in [22]. Additionally, network robustness could be assessed via *cross-evaluation*, performed by multiple works [23, 24, 25] demonstrating how, in a security background, ML/DL models show a strong dependency on the network they were trained in.

These works encounter practical concerns, e.g., training is resource intensive, and research is thus pushed to seek new methodological and technological horizons.

Quantum computing [4, 5] has demonstrated its potential in (i) offering a speed-up solving classical problems [26, 27], (ii) providing better performance results w.r.t. traditional instruments, and (iii) addressing unfeasible challenges. Therefore, QML [6, 7, 8] prevails as a quantum computing application with several examples, including network intrusion detection.

At the *state-of-the-art*, QML applied to network traffic analysis is primarily engaged in security scenarios. Among these, misuse detection is a well-explored topic [9, 10, 11]. Abreu et al. [12] compared three different QML models with their classical counterparts on both binary and multiclass scenarios. The authors then developed QuantumNetSec [13], a framework that, by the combination of several QML algorithms, is able to resolve the aforementioned classification tasks. Kumar and Swarnkar [28] leveraged a Quantum Support Vector Classifier to drive a multiclass task, feeding it with two IoT datasets for training and inference, and proving its superiority among several classical ML/DL models in terms of usual performance metrics (i.e., *precision*, *recall*, *F1-score*).

Positioning: most reviewed works, applying QML techniques in IoT security contexts, focus on data which is *generally deprecated* [9, 11, 16, 18, 23], and thus do not give a faithful reflection of the actual traffic profiles. Our study proposes to exploit datasets that were collected since 2021, offering *a realistic snapshot of the current* IoT *traffic characteristics*. To the best of our knowledge, this work is the first to apply *early traffic classification* with QML models in IoT contexts. Also, at the *state-of-the-art* there is *no evidence of the cross-evaluation*, highlighting the novelty of this study.

III. Methodology

Quantum computing fundamentals: *Quantum bit* (viz. qubit) is the basic information unit used by quantum computers. Qubits owe their power of expressivity to a property called *superposition*. As classical computers involve the well-known bits, which are exclusively 0 or 1, a qubit has faculty of existing in both states simultaneously. In detail, an elementary quantum state $|\psi\rangle$ has a vectorial representation in a Hilbert space $\mathcal{H}$ and depictable by the following formula:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

where $\alpha, \beta \in \mathbb{C}$ and $|0\rangle, |1\rangle \in \mathcal{H}$, $\mathbb{C}$ as the set of complex numbers. In this form, $|\alpha|^2$ (resp. $|\beta|^2$) is the probability of the qubit to stand in the basis state $|0\rangle$ (resp. $|1\rangle$).

Entanglement is another powerful capability of qubits. It represents the interaction between two (or more) qubits staying bound, regardless of time and distance. Features in a dataset encoded in a quantum state can see their mutual correlation being exploited this way. A *quantum circuit* is built by an assembly of *quantum gates*, analogous to classical logic gates but with qubits as input. They are represented as unitary matrices in $\mathcal{H}$, and return rotated (group of) qubits.

Quantum Neural Networks: QML integrates the quantum computing paradigm into traditional ML/DL models. By leveraging quantum circuits for data representation and computation, QML algorithms can potentially cover wider spaces in searching for solutions more efficiently.

Quantum Neural Networks (QNNs) [29] is a basic approach within the field of QML, well adapting to the requirements of a classification task. The core of a QNN is embodied by *two* components: (i) *a feature map* and (ii) *an ansatz*. The feature map is a circuit that yields data passage from a classical fashion to a quantum state. There are several ways to realize this step. We discuss the following:

Amplitude embedding [30], which is particularly suitable when feature vector (i.e., a data sample) has an ample size and a dimensional reduction pre-processing is not feasible. By this method, given a feature vector $\boldsymbol{x}$ of size N, it is represented in a quantum state $|x\rangle$ of $n = \lceil \log_2 N \rceil$ qubits as follows:

$$|x\rangle = \sum_{i=1}^{2^n} x_i |n_i\rangle \tag{2}$$

where x_i is the i-th feature, whereas n_i is the i-th quantum state according to the computational basis. Feature vector needs to be normalized ($\sum_i |x_i|^2 = 1$). The case where the feature vector size is not a power of 2 can be accommodated by adopting padding strategies.

Angle embedding [30, 31, 32], recommended whenever there is a chance to perform some preliminary steps on a feature vector in order to compress the information in a smaller object. Through angle embedding, for a feature vector $\boldsymbol{x}$ of size n, it returns a quantum state $|x\rangle$ of n qubits by applying

qubit rotations as

$$|x\rangle = \bigotimes_{i=1}^{n} (\sin(x_i)|0\rangle + \cos(x_i)|1\rangle) \quad (3)$$

where $\bigotimes$ denotes the tensor product among the n qubits, highlighting that each feature is encoded into one qubit *without interactions*. Preliminary data dimensionality reduction can be performed by a dense layer that is also subject to the training procedure. This is due to the necessity of having a one-to-one mapping between the size of the feature vector and the number of qubits (that could be expensive for large-dimensional input data). In simple terms, a cut down feature vector $\boldsymbol{x}_r = \boldsymbol{h}_e(\boldsymbol{x}; \boldsymbol{\theta}_e)$ is fed into the embedding.

Conversely, the ansatz is a parametrized ($\boldsymbol{\theta}_c$) quantum circuit that leads to the quantum results of the inference task (in this case, a classification problem), and thus, it is where the optimization process focuses. The ansatz operation can be described by the formula

$$|y\rangle = \boldsymbol{U}(\boldsymbol{\theta}_c)|x\rangle \quad (4)$$

where $\boldsymbol{U}(\boldsymbol{\theta}_c)$ is a (composite) unitary transformation applied to the (n-qubit) input state $|x\rangle$.

The ansatz can be unfolded as

$$\boldsymbol{U}(\boldsymbol{\theta}_c) = \boldsymbol{U}_1(\boldsymbol{\theta}_{c,1}) \circ \cdots \circ \boldsymbol{U}_L(\boldsymbol{\theta}_{c,L}) \quad (5)$$

in which L is the number of sequential layers within the circuit, $\boldsymbol{\theta}_{c,\ell}$, with $\ell = 1, \ldots, L$, denotes the set of trainable parameters of the ℓ-th quantum layer, and "$\circ$" is the composition operator. A single layer accounts for the participation of more quantum gates, hiring either (i) single qubits (e.g., rotation gates, Hadamard gate) or (ii) on two or more qubits (e.g., CNOT gates).

Information on the results is extracted from the quantum circuit via a *measurement process* on the wires to let the data get back into a classical fashion. Typically, the expectation value of the Pauli-Z operator is taken, operating on generic qubit $|y_i\rangle = \alpha_i|0\rangle + \beta_i|1\rangle$, corresponding to $m_i = |\alpha_i|^2 - |\beta_i|^2$. The output, embedded in the vector $\boldsymbol{m} \triangleq [m_1, \ldots m_n]$, can be processed by further dense layer as $\boldsymbol{o} = \boldsymbol{h}_p(\boldsymbol{m}; \boldsymbol{\theta}_p)$. Finally, it is used to feed a softmax activation to obtain the confidence vector $\boldsymbol{p} \in [0, 1]^C$ for the TC task.

Optimization of the model parameters (i.e., $(\boldsymbol{\theta}_c, \boldsymbol{\theta}_p)$ or $(\boldsymbol{\theta}_e, \boldsymbol{\theta}_c, \boldsymbol{\theta}_p)$) is conducted performing classical computing operations. In detail, the QNN is trained to minimize a loss $\mathcal{L}(\boldsymbol{p}, \boldsymbol{p}_g)$ representing the error explained by the soft-output vector $\boldsymbol{p}$ and the one-hot encoded ground truth $\boldsymbol{p}_g$ over multiple training samples.

Models evaluation workflow: In this work, we address two *early* classification tasks at the *bi-directional flow* (viz. biflow) level, based on the analysis of its initial packets: (*i*) *Misuse Detection* (MD), which involves distinguishing between benign and malicious network traffic and (*ii*) *Attack Classification* (AC), which also aims to identify the specific attack class.

Moreover, we inspect a *cross-evaluation* scenario in which, given two different datasets X and Y, a model is trained on the training set of X (i.e., benign and malicious traffic) and is evaluated on a combined test set that includes (i) benign traffic from the test set of X and (ii) malware traffic from the test set of Y. This analysis aims to assess the ability of the model to detect malware traffic in a network environment distinct from the one where the training data was collected.

For each biflow, we consider partial traffic information, since it is adequate data to drive our tasks [22]. We leverage the sequence of $N_f = 3$ transport-level header fields extracted from the first $N_P = 10$ packets. These fields include: (*i*) payload length (`PL`), (*ii*) packet direction (`DIR`), and (*iii*) TCP window-size (`TCPWIN`). For `DIR`, upstream and downstream directions are encoded as $+1$ and -1 values, respectively, while `TCPWIN` is set to 0 for UDP packets.

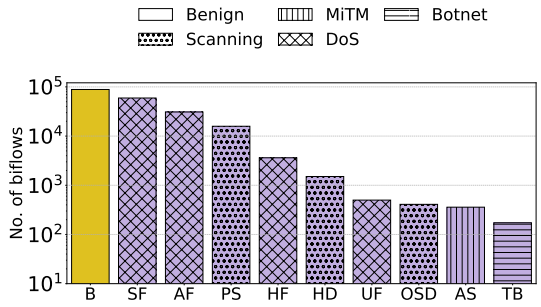
Algorithms: We compare two QML models in addressing the tasks above, also considering two DL models as a baseline. Both QML and DL models are described as follows:

1) QNN *with Amplitude Embedding* (AMPE): a quantum neural network that directly receives a vector of 32 features, which is encoded using an amplitude embedding feature map onto 5 qubits and an ansatz made of strongly entangling layers ($L = 3$);
2) QNN *with Angle Embedding* (ANGE): hybrid classical-quantum neural network with an initial dense layer for feature extraction, followed by a 5-qubit circuit using angle embedding as the feature map and the same ansatz as for ANGE;
3) *1-D Convolutional Neural Network* (1D-CNN): composed by two convolutional layers (with 32 and 64 filters, respectively, kernel size of 3 and ReLU as activation function), each pursued by a max pooling operation of size 2. The convolutional block is followed by a flattening step, a dense layer with 128 units and ReLU activation, and a dropout layer with a rate of 0.5.
4) *Hybrid Convolutional/Recurrent Neural Network* (HYBRID): consists of two convolutional layers with 32 and 64 filters, respectively, using a $(4, 2)$ kernel and ReLU activation, each followed by batch normalization. After reshaping, a Long Short-Term Memory layer (100 units) and a dense layer (100 units and ReLU activation) are added, interleaved by dropout layers with rates of 0.2 and 0.4, respectively.

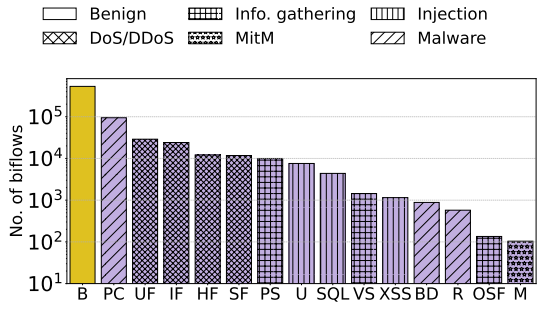
All the classifiers end up with a (softmax-activated) dense layer whose output dimension equals the number of classes C. Before feeding the classifiers, we performed some preprocessing steps. First, we scale each header field using a *per-feature MinMax* scaler. Furthermore, due to implementation constraints of the QML algorithms, each biflow was reshaped into an array of size $(N_f \cdot N_P)$ before feeding it to the models.

IV. EXPERIMENTAL SETUP

Evaluation datasets: We use two popular IOT datasets, IOT-NIDD and EDGE-IIOT. Fig. 1 shows their class composition and sample distribution.



(a) IoT-NIDD



(b) Edge-IIoT

Fig. 1. Biflow distribution across classes on (a) IoT-NIDD and (b) Edge-IIoT. Benign and Attack classes are shown in yellow and violet, respectively. Hatches represent the attack category. Values on y-axis are in *log*-scale.

IoT-NIDD was generated using two smart home devices, i.e., a smart speaker and a Wi-Fi camera, and they captured it through a wireless connected node [33]. They discerned among 10 classes (Fig. 1a), i.e., *benign* traffic (B), and various IoT malware attacks, including *Host Discovery* (HD), *Port Scanning* (PS), *OS/Version Detection* (OSD), *ARP Spoofing* (AS), *SYN Flooding* (SF), *Telnet Bruteforce* (TB), *UDP Flooding* (UF), *ACK Flooding* (AF), and *HTTP Flooding* (HF). These are distinguished by four categories, i.e., *Scanning*, *MiTM*, *DoS*, and *Mirai Botnet* (viz. *Botnet*).

Edge-IIoT was collected instrumenting a testbed with several different IoT/-IIoT (Industrial IoT) devices, e.g., temperature sensors, humidity sensors, water-level sensors [34]. Traffic from diverse categories of attacks was considered, for a total of 15 classes (Fig. 1b): (*i*) *DoS/DDoS attacks*, including *TCP SYN flood DDoS* (SF), *UDP flood DDoS* (UF), *HTTP flood DDoS* (HF), *ICMP flood DDoS* (IF); (*ii*) *Information gathering*, including *Port scanning* (PS), *OS fingerprinting* (OSF), *Vulnerability scanning* (VS); (*iii*) *Main-in-the-middle attacks* (M), including *ARP spoofing* (AS), *DNS spoofing* (DS); (*iv*) *Injection attacks*, including *XSS*, *SQL*, *Uploading* (U); (*v*) *Malware attacks*, including *Backdoor* (BD), *Password cracking* (PC), *Ransomware* (R). Due to class imbalance, Edge-IIoT has been undersampled, using only 1% of samples from classes with over 100K biflows.

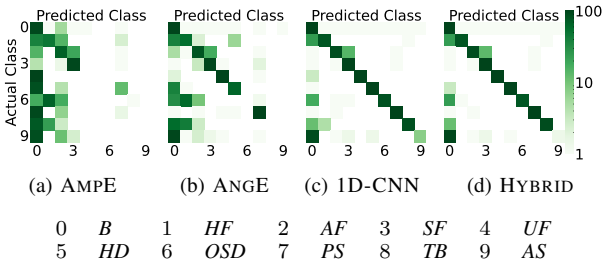
Implementation details: We used Python's Keras and TensorFlow libraries for building DL models. We also utilized PennyLane to create and simulate QML models. The experiments relied on PennyLane's default qubit simulator (viz. `default.qubit`), which acts in ideal conditions, i.e., without quantum noise.

Table I: Average performance of AmpE, AngE, 1D-CNN, and Hybrid on IoT-NIDD and Edge-IIoT (AC case). Results are presented as $mean \pm std.dev.$ over 5 repetitions. For each metric, the best DL and QML approaches are highlighted in green and blue, respectively.

	IoT-NIDD		Edge-IIoT	
	Accuracy [%]	**F1-score [%]**	**Accuracy [%]**	**F1-score [%]**
AmpE	79.82 ± 0.70	30.13 ± 0.64	95.17 ± 0.47	53.84 ± 1.77
AngE	92.46 ± 0.22	58.80 ± 0.55	96.85 ± 0.18	63.34 ± 4.63
1D-CNN	96.83 ± 0.16	85.00 ± 1.01	97.14 ± 0.01	73.22 ± 3.03
Hybrid	97.01 ± 0.04	84.27 ± 0.31	97.14 ± 0.00	74.03 ± 0.48

Evaluation setup: Datasets were split according to a $4:1$ ratio for training and testing to evaluate the classifiers. The training set was further partitioned with a $9:1$ ratio for the creation of a validation set. The training was executed within 50 epochs—with a batch size of 32—for minimizing the categorical cross-entropy loss function and using the Adam optimizer—with a learning rate of 0.01, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-7}$. To prevent overfitting, the early-stopping technique was applied by monitoring validation accuracy. To tackle TC fluctuations issue, led by the stochasticity of training, we trained (and tested) all the models *five* times by pinning the seed.

V. Experimental Evaluation

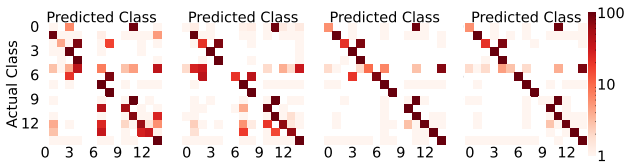


(a) AmpE (b) AngE (c) 1D-CNN (d) Hybrid

0 *B* 1 *HF* 2 *AF* 3 *SF* 4 *UF*
5 *HD* 6 *OSD* 7 *PS* 8 *TB* 9 *AS*

Fig. 2. Confusion matrices of (a) AmpE, (b) AngE, (c) 1D-CNN, (d) Hybrid w.r.t. IoT-NIDD (AC case). Values are reported as $mean$ over 5 repetitions.

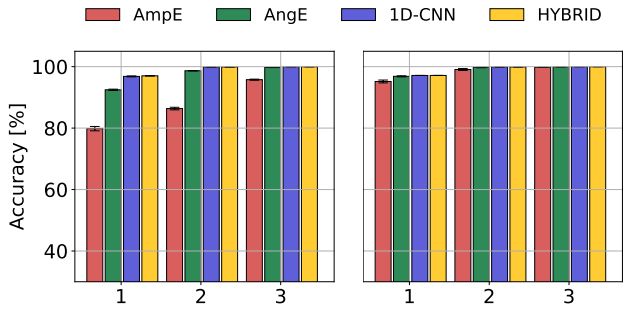
Attack Classification: Our first experimental insight pushes on the ability of the models to return reliable results when inferring biflows among multiple classes. Results are shown in Tab. I. Regarding Accuracy, AngE and AmpE under-perform, w.r.t. DL models, with respective 4.55% and 17.19% distance from the performance of the best algorithm (i.e., Hybrid). This gap appears deeper considering F1-score, as AmpE offers 30.13% against 85.00% returned by the best model (i.e., 1D-CNN). The performance disparity between DL and QML models narrows by involving Edge-IIoT. Factually, the distance between the best QML algorithm in terms of



(a) AMPE (b) ANGE (c) 1D-CNN (d) HYBRID

0	*BD*	1	*HF*	2	*IF*	3	*SF*	4	*UF*
5	*M*	6	*OSF*	7	*PC*	8	*PS*	9	*R*
10	*SQL*	11	*U*	12	*VS*	13	*XSS*	14	*B*

Fig. 3. Confusion matrices of (a) AMPE, (b) ANGE, (c) 1D-CNN, (d) HYBRID, w.r.t. EDGE-IIOT dataset (AC case). Values are reported as *mean* over 5 repetitions.



(a) IOT-NIDD (b) EDGE-IIOT

Fig. 4. Top-K Accuracy scores for (a) IOT-NIDD, and (b) EDGE-IIOT datasets. Values are reported as $avg. \pm std.dev.$ over 5 runs.

Accuracy (i.e., ANGE) and the best DL (i.e., HYBRID) is 0.29%. This becomes higher regarding F1-score, touching a 10.69% difference.

IOT-NIDD AC task is heavily affected by B class super-abundance. As shown in Fig. 2, diverse classes are confounded with B. Particularly, AMPE (Fig. 2a) and ANGE (Fig. 2b) encounter some difficulties in inferring correctly the less represented classes (e.g., TB, AS, OSD). Notwithstanding the higher number of classes, EDGE-IIOT is overall accurately classified by the models (Fig. 3). Some noticeable confounding patterns are: (i) R predicted as U, (ii) OSF predicted as B, and (iii) BD predicted as U.

As considering the Top-K Accuracy[1], in IOT-NIDD case (Fig. 4a), AMPE reports a hop from 79.82% with $K = 1$ to 95.76% with $K = 3$. Conversely, ANGE presents a certain improvement from 92.46% with $K = 1$ to 99.72% with $K = 3$. EDGE-IIOT (Fig. 4b) either unveils an increase along K value, with $\sim 99\%$ value with $K = 3$.

Takeaways: *In* IOT-NIDD *case,* QML *models do not over-perform w.r.t.* DL *models. Regarding the former,* ANGE *overcomes* AMPE *for each metric considered. The performance gap between* QML *and* DL *narrows by considering* EDGE-IIOT *dataset.*

Misuse Detection: By post-processing outcomes, we can deduce the ability of the classifier to discriminate between benign and malicious traffic. A summary of Accuracy and F1-score is reported in Tab. II. All the classifiers clearly exhibit an increase in performance, w.r.t. AC case, for both the datasets involved. ANGE reaches 98.89% (resp. 99.86%) by classifying IOT-NIDD (resp. EDGE-IIOT) considering either Accuracy and F1-score, minimally far from the best (i.e., 1D-CNN and HYBRID, with +0.08% gap). AMPE performs worse than ANGE, achieving $\sim -4\%$ (resp. $\sim -1\%$) of both metrics considering IOT-NIDD (resp. EDGE-IIOT) dataset.

Takeaways: ANGE *achieves similar performance w.r.t.* DL *models by both Accuracy and F1-score, while* AMPE *encounters a certain gap.*

[1]It is the ratio of correct predictions to the total, considering a classification correct if the correct inference is among the top K class scores.

Table II: Average performance of AMPE, ANGE, 1D-CNN, and HYBRID on IOT-NIDD and EDGE-IIOT (MD case). Results are presented as $mean \pm std.dev.$ over 5 runs. For each metric, the best DL and QML approaches are highlighted in green and blue, respectively.

	IOT-NIDD		EDGE-IIOT	
	Accuracy [%]	**F1-score [%]**	**Accuracy [%]**	**F1-score [%]**
AMPE	93.42 ± 0.13	93.31 ± 0.13	98.88 ± 0.45	98.59 ± 0.56
ANGE	97.18 ± 0.27	97.14 ± 0.27	99.89 ± 0.12	99.86 ± 0.15
1D-CNN	99.05 ± 0.05	99.04 ± 0.05	99.97 ± 0.00	99.96 ± 0.00
HYBRID	99.05 ± 0.03	99.03 ± 0.03	99.97 ± 0.00	99.96 ± 0.00

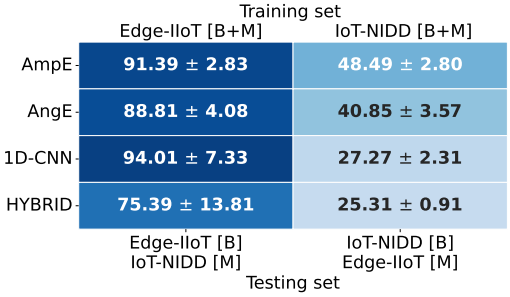


Fig. 5. F1-score for *cross evaluation*. Results are shown as $avg. \pm std.dev.$ over 5 runs.

Cross-evaluation: As a further investigation, we conduct a *cross-evaluation* analysis. Fig. 5 shows the results in terms of macro-averaged F1-score so that the imbalance between classes is considered. On the one hand, fixing EDGE-IIOT as a training dataset, AMPE achieves performance close to that of the best model (i.e., 1D-CNN), with a gap of $\approx -2.62\%$. In contrast, ANGE shows a larger gap of $\approx -5.20\%$ from the best. On the other hand, when considering IOT-NIDD as the training dataset, QML algorithms overcome the DL ones, with AMPE performing the best (i.e., 48.49% of F1-score), followed by ANGE (i.e., 40.85% of F1-score). These results suggest that QML methods are more effective in managing class imbalance.

Takeaways: *Using* EDGE-IIOT *as the training set,* DL *models (viz.* 1D-CNN*) perform best, followed closely by* AMPE*. In contrast, when trained on* IOT-NIDD*,* QML *methods surpass*

DL *approaches, with* AMPE *achieving the highest F1-score, indicating that* QML *methods are better at managing class imbalance.*

VI. CONCLUSIONS AND FUTURE PERSPECTIVES

This paper exposes numerous findings about QML in network traffic analysis, notably in IOT security scene. As the best of our knowledge, this work is the first to exploit *cross-evaluation* within QML to a network traffic environment. In addition, we leverage QML for *early classification*.

Regarding AC with the application of IOT-NIDD and EDGE-IIOT, QML models offer lower performance, than DL baselines, achieving 63.34% vs. 74.03% (resp. 58.80% vs. 85.00%) best F1-score for IOT-NIDD (resp. EDGE-IIOT). The performance gap is reduced when considering MD task, whereas QML perform similarly to DL counterparts ($> 93\%$ and $> 98\%$ in each metric for IOT-NIDD and EDGE-IIOT, respectively). Moreover, cross-evaluation enlightens the faculty of QML algorithms in handling unbalanced data. Particularly, for IOT-NIDD there is a dominance of AMPE over the tested options, with 48.49% vs. 27.27% in F1-score w.r.t. the best DL approach (i.e., 1D-CNN). The results raise the need to further explore QML in IOT contexts.

Future directions for this work lead to the study of different QML models, in order to provide a more profound survey of the availability of solutions. In addition, other security datasets should be leveraged. Moreover, another field of research is Explainable AI (XAI) [35], which may give insights in the dependence between the inputs and the outcomes of this kind of algorithms. In conclusion, a practical concern, that could offer developments for this work, is represented by the execution within a real quantum hardware environment, that would uncover some questions about quantum technology, e.g., device noise, qubit decoherence.

REFERENCES

[1] S. Sinha, "State of IoT 2024: Number of connected IoT devices growing 13% to 18.8 billion globally," 2024.

[2] A. Qamar, A. Karim, and V. Chang, "Mobile malware attacks: Review, taxonomy & future directions," *Future Generation Computer Systems*, vol. 97, pp. 887–909, 2019.

[3] Q.-D. Ngo, H.-T. Nguyen, V.-H. Le, and D.-H. Nguyen, "A survey of IoT malware and detection methods based on static features," *ICT express*, vol. 6, no. 4, pp. 280–286, 2020.

[4] R. Rietsche, C. Dremel, S. Bosch, L. Steinacker, M. Meckel, and J.-M. Leimeister, "Quantum computing," *Electronic Markets*, vol. 32, no. 4, pp. 2525–2536, 2022.

[5] S. S. Gill, A. Kumar, H. Singh, M. Singh, K. Kaur, M. Usman, and R. Buyya, "Quantum computing: A taxonomy, systematic review and future directions," *Wiley Online Library, Software: Practice and Experience*, vol. 52, no. 1, pp. 66–114, 2022.

[6] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Taylor & Francis Contemporary Physics*, vol. 56, no. 2, pp. 172–185, 2015.

[7] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.

[8] J. D. Martín-Guerrero and L. Lamata, "Quantum machine learning: A tutorial," *Neurocomputing*, vol. 470, pp. 457–461, 2022.

[9] A. Gouveia and M. Correia, "Towards quantum-enhanced machine learning for network intrusion detection," in *IEEE NCA'20*.

[10] P. Bhattacharya, A. Kumari, S. Tanwar, I. Budhiraja, S. Patel, and J. J. Rodrigues, "Quant-jack: Quantum machine learning to detect cryptojacking attacks in IIoT networks," in *IEEE ICC Workshop'24*.

[11] A. Kukliansky, M. Orescanin, C. Bollmann, and T. Huffmire, "Network anomaly detection using quantum neural networks on noisy quantum computers," *IEEE Trans. on Quantum Engineering*, 2024.

[12] D. Abreu, C. E. Rothenberg, and A. Abelém, "QML-IDS: Quantum machine learning intrusion detection system," in *IEEE ISCC'24*.

[13] D. Abreu, D. Moura, C. Rothenberg, and A. Abelém, "QuantumNetSec: Quantum machine learning for network security," *Authorea Preprints*, 2024.

[14] I. Rakine, K. El Guemmat, S. Ouahabi, I. Atouf, and M. Talea, "IoT intrusion detection: A review of ML and DL-based approaches," in *IEEE IRASET'24*.

[15] N. Niknami and J. Wu, "Advanced ML/DL-based intrusion detection systems for software-defined networks," in *Network Security Empowered by Artificial Intelligence*. Springer, 2024, pp. 121–146.

[16] L. Li, Y. Yu, S. Bai, Y. Hou, and X. Chen, "An effective two-step intrusion detection approach based on binary classification and k-NN," *IEEE Access*, vol. 6, pp. 12 060–12 073, 2017.

[17] N. Elsakaan and K. Amroun, "A comparative study of machine learning binary classification methods for botnet detection," in *Springer ACS'21*.

[18] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *EAI BICT'16*.

[19] Y. Lai, J. Zhang, and Z. Liu, "Industrial anomaly detection and attack classification method based on convolutional neural network," *Security and Communication Networks*, no. 1, p. 8124254, 2019.

[20] Y. Yang, J. Cheng, Z. Liu, H. Li, and G. Xu, "A multi-classification detection model for imbalanced data in NIDS based on reconstruction and feature matching," *Journal of Cloud Computing*, vol. 13, no. 1, p. 31, 2024.

[21] M. Arsalan, M. Mubeen, M. Bilal, and S. F. Abbasi, "1D-CNN-IDS: 1D CNN-based intrusion detection system for IIoT," in *IEEE ICAC'24*.

[22] I. Guarino, G. Bovenzi, D. Di Monda, G. Aceto, D. Ciuonzo, and A. Pescapé, "On the use of machine learning approaches for the early classification in network intrusion detection," in *IEEE M&N'22*.

[23] S. Al-Riyami, A. Lisitsa, and F. Coenen, "Cross-datasets evaluation of machine learning models for intrusion detection systems," in *ICICT'21*.

[24] C. Guida, A. Nascita, A. Montieri, and A. Pescapé, "Cross-evaluation of deep learning-based network intrusion detection systems," in *IEEE FiCloud'23*.

[25] M. Cantone, C. Marrocco, and A. Bria, "On the cross-dataset generalization of machine learning for network intrusion detection," *arXiv*, 2024.

[26] R. LaPierre and R. LaPierre, "Shor algorithm," *Introduction to Quantum Computing*, pp. 177–192, 2021.

[27] G.-L. Long, "Grover algorithm with zero theoretical failure rate," *Physical Review A*, vol. 64, no. 2, p. 022307, 2001.

[28] R. Kumar and M. Swarnkar, "QuIDS: A quantum support vector machine-based intrusion detection system for IoT networks," *Journal of Network and Computer Applications*, vol. 234, p. 104072, 2025.

[29] W. Li, Z.-d. Lu, and D.-L. Deng, "Quantum neural network classifiers: A tutorial," *SciPost Physics Lecture Notes*, p. 061, 2022.

[30] T. Hur, L. Kim, and D. K. Park, "Quantum convolutional neural network for classical data classification," *Quantum Machine Intelligence*, vol. 4, no. 1, p. 3, 2022.

[31] A. Senokosov, A. Sedykh, A. Sagingalieva, B. Kyriacou, and A. Melnikov, "Quantum machine learning for image classification," *Machine Learning: Science and Technology*, vol. 5, no. 1, p. 015040, 2024.

[32] M. Hdaib, S. Rajasegarar, and L. Pan, "Quantum deep learning-based anomaly detection for enhanced network security," *Springer, Quantum Machine Intelligence*, vol. 6, no. 1, p. 26, 2024.

[33] H. Kang, D. H. Ahn, G. M. Lee, J. D. Yoo, K. H. Park, and H. K. Kim, "IoT network intrusion dataset," 2019.

[34] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40 281–40 306, 2022.

[35] A. Nascita, G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapé, "A survey on explainable artificial intelligence for internet traffic classification and prediction, and intrusion detection," *IEEE Communications Surveys & Tutorials*, 2024.