Reimplementation and Evaluation of Hybrid Quantum Deep Learning and Variational Quantum Classifier-Based Models for Botnet DGA Attack Detection

Mohammad Alshurbaji
School of Computing
Clemson University
Clemson, South Carolina, 29631
Email: malshur@clemson.edu

Fateme Mazdarani
School of Computing
Clemson University
Central, South Carolina, 29630
Email: fmazdar@clemson.edu

***Abstract*—This study revisits and extends prior work on leveraging quantum machine learning for cybersecurity domain generation algorithm (DGA) detection by reimplementing and evaluating both classical and quantum approaches. We replicate the Variational Quantum Classifier (VQC) using updated Qiskit frameworks and assess its performance across various combinations of quantum feature maps, variational circuits, and optimizers. Additionally, we attempt to reproduce a hybrid quantum-classical deep learning model but encounter significant compatibility issues due to software deprecations. Our experimental results confirm that, with careful design, VQC models can approach or match prior benchmarks in some cases. However, the challenges encountered highlight the limitations of current quantum tooling for real-world cybersecurity applications. This work underscores both the promise and the practical barriers of integrating quantum computing into threat detection systems and provides a foundation for future research in scalable and robust quantum-enhanced cybersecurity solutions.**

1. Introduction

Botnets represent one of the most persistent and dangerous threats in cybersecurity today. A botnet is a network of compromised devices - often referred to as "zombie computers" - that are controlled remotely by an attacker, typically without the owners' knowledge. These networks can be mobilized for various malicious purposes, such as launching distributed denial-of-service (DDoS) attacks, sending spam emails, stealing sensitive information, and disrupting services at scale. Notable real-world examples include the infamous Mirai botnet, which crippled large sections of the internet in 2016, and the Emotet botnet, notorious for its modular, evolving attacks on financial institutions.

One critical method used by botnet operators to maintain communication with their command and control (C&C) servers is the Domain Generation Algorithm (DGA). Instead of relying on a single static domain that can easily be blacklisted or taken down by defenders, DGAs allow bots to generate a large number of domain names dynamically, daily or even hourly. This dynamic nature makes it extremely challenging for security professionals to predict and block all possible malicious domains in advance. For instance, a bot infected with a DGA malware might attempt to contact hundreds or thousands of pseudo-randomly generated domains each day, waiting for the attacker to activate one legitimate domain among them.

Given the increasing sophistication of botnet architectures and DGA techniques, developing effective detection methods has become a major area of research. Traditional machine learning approaches have shown strong results in identifying statistical patterns in domain names to distinguish between benign and malicious domains. However, as the complexity and evasiveness of attacks grow, there is an emerging interest in exploring whether quantum machine learning can offer enhanced detection capabilities by leveraging the properties of quantum computing, such as superposition and entanglement, to better capture complex patterns in data.

In the original work titled "Hybrid Quantum Deep Learning and Variational Quantum Classifier-Based Model for Botnet DGA Attack Detection," quantum machine learning approaches - including a hybrid quantum-classical deep learning model and a Variational Quantum Classifier (VQC) - were proposed and evaluated. The study demonstrated that these models could achieve competitive, and in some cases superior, performance compared to classical methods, especially when leveraging specific quantum circuit designs and optimization strategies.

Building upon this foundation, our project aimed to review, reproduce, and extend the findings of the original study. We successfully implemented classical machine learning models on a subset of the dataset, recreated the VQC model from scratch, and evaluated its performance. Furthermore, while attempting to re-implement the hybrid model, we encountered practical challenges that reveal the current limitations of applying quantum approaches in real-world cybersecurity problems.

This reproduction study not only validates key insights from the original work but also provides a deeper understanding of the obstacles researchers face when bridging the gap between quantum theory and practical deployment. It also reaffirms the importance of continuous innovation

in botnet detection strategies, especially as cybercriminals continue to adapt and evolve their techniques.

May 1, 2025

2. Methods

2.1. Datasets and features

For this study, we utilized a subset of the dataset employed in the original work, which was specifically curated for botnet Domain Generation Algorithm (DGA) detection. The complete dataset consists of over 1.8 million domain names, comprising 1,000,000 legitimate domains sourced from the Alexa Top 1M list and 803,333 malicious domains generated by ten distinct DGA-based botnet families, including Conficker, Cryptolocker, Matsnu, Pushdo, and Zeus, among others.

TABLE 1. DATASETS OF DGA AND LEGITIMATE DOMAIN NAMES

Dataset	Samples	Sample Domain Name
Alexa	1,000,000	wikipedia.org
Conficker	100,000	hkcoaxcnjf.net
Cryptolocker	100,000	tgrmkncpkhaj.biz
Goz	1,667	atgxucqshdurghqdjyxti.ru
Matsnu	100,000	scoreadmireluckapplyfitcouple.com
New_Goz	1,666	1jwwz47ue0sakssvy4e1jx8z03.org
Pushdo	100,000	nafwupwi.ru
Ramdo	100,000	wsqwecgygoaakesq.org
Rovnix	100,000	theirtheandaloneinto.com
Tinba	100,000	ghlwtuutpkwm.com
Zeus	100,000	dd12071h5p3isqezvxr16k41j1.net

To accommodate computational resource constraints and to maintain focus on the feasibility of classical and quantum experiments, we selected a representative subset of this dataset for our experiments. The subset preserved the diversity of the original data, ensuring that both benign and DGA-generated domains were sufficiently included to retain the core challenges of the classification task.

Feature engineering in our study adhered to the methodology outlined in the original work. Each domain name was transformed into a structured feature vector comprising seven numerical attributes, designed to encapsulate the statistical and structural patterns characteristic of DGA-generated domains. These features included measurements such as character length, entropy-based randomness, relative entropy comparisons against legitimate and malicious domain profiles, information radius across multiple botnet families, and tree-based predictive scores, as well as reputation metrics derived from n-gram analyses. This comprehensive feature set aimed to capture both the inherent randomness of algorithmically generated domains and their subtle deviations from typical benign structures, providing a robust foundation for classification tasks. Working with a representative subset of the full dataset allowed us to ensure computational feasibility, particularly for quantum simulations, while maintaining the integrity and diversity necessary for meaningful performance evaluations.

2.2. Classical ML Methods

At the beginning of any machine learning project, it is often beneficial to first explore classical methods. These methods serve as a benchmark, providing a baseline of performance before delving into more complex models. Classical algorithms are well-established, widely understood, and computationally efficient, making them ideal for initial evaluations. In our work, we utilized five different classical machine learning algorithms for Botnet DGA attack detection: Logistic Regression, Random Forest, Naive Bayes, Extra Tree, and an Ensemble method. These models exhibited promising results, with accuracies ranging from 82.69% to 95.19%. Specifically, Random Forest achieved the highest accuracy of 95.19%, followed closely by Extra Tree (94.54%) and Ensemble (94.55%). Despite these satisfactory results, it is clear that as the complexity of the task increases, the need for more advanced methods, such as quantum computing, arises. These classical methods serve as an important starting point, but for further improvements and handling more sophisticated scenarios, quantum-enhanced models are required.

TABLE 2. CLASSICAL MACHINE LEARNING MODELS PERFORMANCE

Model	Accuracy
Logistic Regression	0.895460
Random Forest	0.951909
Naive Bayes	0.826862
Extra Tree	0.949399
Ensemble	0.945498

2.3. VQC algorithm-based model

The first quantum ML-based classification algorithm implemented in this research is the Variational Quantum Classifier algorithm (Fig. 1). VQC is a supervised learning approach that relies on a parameterized quantum circuit to classify input data. The model operates by encoding classical data into quantum states through feature mapping, applying a variational form parameterized by tunable gates, and measuring observables to predict class labels.

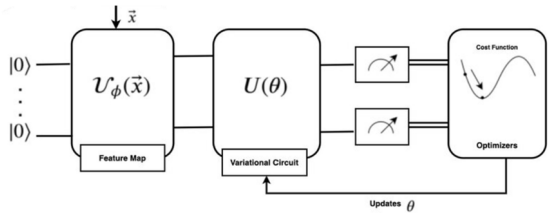

Figure 1. Illustration of the VQC model

The attractiveness of VQC stems from its potential to efficiently capture complex data patterns using fewer resources than purely classical models, particularly in high-dimensional feature spaces. Moreover, VQC models are highly flexible, allowing researchers to customize the feature

mapping circuits, variational ansatz, and optimization strategies based on the problem characteristics. This flexibility makes VQC a strong candidate for benchmarking in early-stage quantum machine learning studies.

As illustrated in Table 3, the original study examined the combinations of 12 types of optimizer algorithms, four types of feature maps, and four types of variational form circuits to identify a combination that produces high accuracy for botnet DGA detection.

TABLE 3. COMBINATIONS OF VQC ALGORITHM'S COMPONENTS

Feature Maps	Variational Forms	Optimizers
RawFeatureVector	RealAmplitudes	AQGD, CG, COBYLA
ZFeatureMap	EfficientSU2	GSLS, L$_B FGS_B$, NELDER_MEAD
ZZFeatureMap	TwoLocal	NFT, P$_B FGS$, POWELL
PauliFeatureMap	Excitation Preserving	SLSQP, SPSA, TNC

2.3.1. Quantum feature-mapping. Feature mapping is a critical step in quantum machine learning, where classical data is encoded into quantum states. In our VQC implementation, we evaluated different types of feature maps, all of which We implemented using Qiskit:

- **RawFeatureVector**:
  Embeds input data directly into quantum amplitudes.

- **ZFeatureMap**:
  Applies Pauli-Z rotations based on input features.

- **ZZFeatureMap**:
  Incorporates two-qubit entangling operations to capture feature correlations.

Each feature map transforms the input feature vector into a quantum state in different ways, potentially affecting the expressibility and learnability of the resulting model.

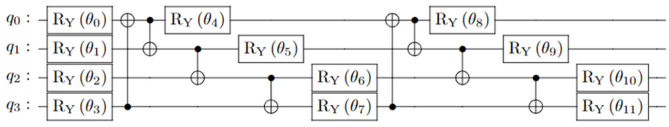

Figure 2. Illustration of four qubits RealAmplitudes circuit.The rotation y (Ry) gate is a single-qubit rotation through angle θ (radians) around the y-axis.

2.3.2. Variational circuits. Variational circuits, or ansatz, are parameterized quantum circuits used to map input quantum states to output states optimized for a specific task. The original work explored several ansatz structures:

- **RealAmplitudes**:
  Consists of alternating Y rotations and CNOT entanglements. RealAmplitudes prepares quantum states comprising only real amplitudes, whereas the complex part is always 0.

- **TwoLocal**:
  Alternates layers of single-qubit rotations and two-qubit entangling gates.

- **EfficientSU2**:
  Comprises layers of single-qubit operations from SU(2) group - 2×2 unitary matrices with determinant one (such as Pauli rotation gates)- interleaved with CNOT gates to generate entanglement.

- **ExcitationPreserving**:
  Maintains the number of qubit excitations across layers, suitable for some physical systems.

The choice of variational circuit significantly impacts the learning ability and optimization landscape of the VQC.

2.3.3. Optimizers. In VQC, optimization algorithms are essential for adjusting the variational circuit parameters to minimize the loss function and improve model performance. The original work, employed twelve optimizers available in Qiskit's library at that time, namely: Analytic Quantum Gradient Descent (AQGD), Conjugate Gradient (CG), Constrained Optimization by Linear Approximation (COBYLA), Gaussian-Smoothed Line Search (GSLS), Limited-Memory Broyden–Fletcher–Goldfarb–Shanno with bounds (L_BFGS_B), Nelder–Mead, Nakanishi–Fujii–Todo (NFT), P_BFGS, Powell's Method, Sequential Least-Squares Programming (SLSQP), Simultaneous Perturbation Stochastic Approximation (SPSA), and Truncated Newton Conjugate-Gradient (TNC), as outlined in Table 3.

In adapting the original study, we faced limitations in reproducing the full set of optimization algorithms, as the original library is no longer maintained- -this issue will be discussed in the following section. As a result, we focused on the COBYLA optimizer, one of the top-performing methods in the initial work. COBYLA, a gradient-free algorithm, is well-suited for variational quantum circuits where gradients are noisy or hard to compute. To complement this, we also incorporated the ADAM optimizer, a widely used method in classical machine learning. Together, these choices provided a practical and reliable optimization strategy aligned with the original study's outcomes and compatible with latest frameworks.

2.3.4. Implementation Challenges. Rebuilding the VQC model required extensive adaptations due to:

- Deprecation of Qiskit Aqua and removal of built-in VQC classes.
- Changes in simulator backends and noise model APIs.
- Updates to parameter binding mechanisms.

The removal of the Qiskit Aqua module led to the elimination of built-in VQC classes and high-level abstractions that previously streamlined quantum machine learning

workflows. These updates collectively demanded a low-level, modular reimplementation of the VQC model, significantly increasing the engineering effort compared to earlier versions of Qiskit-based quantum ML development.

2.3.5. VQC-Based Model Experiments. Following the reimplementation, we conducted a series of experiments using the dataset described in Section 2.1. Specifically, we performed experiments by combining three feature maps, three variational forms, and two optimizer algorithms. All experiments were executed on Qiskit's state vector simulator to provide a noise-free environment and maintain computational efficiency. To manage resource constraints, we limited our study to simulations rather than running circuits on real quantum hardware. The goal of these experiments was to systematically investigate how different combinations of local optimizers, feature maps, and variational forms influence the classification accuracy achieved by the VQC models.

2.4. Hybrid Quantum Deep Learning Model

The original study proposed a hybrid quantum-classical deep learning model(Fig. 3). The architecture integrates classical neural network layers with a quantum component using a Parameterized Quantum Circuit (PQC) framework. The classical portion was implemented using Keras Sequential models, consisting of fully connected (dense) layers with ReLU activations, a dropout layer for regularization, and a final dense layer with a sigmoid activation function for binary classification. A quantum layer, implemented via the PennyLane library, was inserted between the classical layers. This layer employs an embedding circuit to encode classical input data into quantum states, followed by a trainable ansatz.

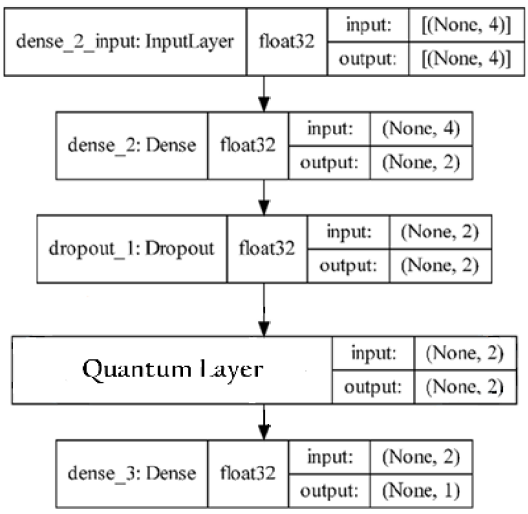

Figure 3. Hybrid quantum-classical deep learning model

The experiments were done with six quantum circuit configurations (two embedding methods-Angle Embedding and IQP Embedding- and three ansatz types: Basic Entangler Layers, Random Layers, and Strongly Entangling Layers). Simulations were carried out using the qiskit.aer backend via the PennyLane-Qiskit plugin, enabling noise-aware execution. To mimic realistic quantum conditions, noise models from eight IBM quantum devices were applied.

In attempting to replicate the proposed hybrid model, we faced several implementation challenges due to significant updates in the software ecosystem. First, compatibility issues between TensorFlow and the latest versions of PennyLane caused integration failures during model construction and training. Second, several features of the Qiskit-PennyLane plugin used in the original implementation have since been deprecated or modified, making it difficult to reproduce the same simulation workflow. Subsequently, our attempts to attach device-specific noise models to the PennyLane simulation backend were also unsuccessful.

3. Results and Discussion

We evaluated our implementation of the Variational Quantum Classifier using combinations of three feature maps-`RawFeatureVector`, `ZFeatureMap`, and `ZZFeatureMap`-and three variational forms: `RealAmplitudes`, `TwoLocal`, and `EfficientSU2`. Each configuration was tested using two optimizers: COBYLA and ADAM. The classification accuracy results are presented in Table 4.

Across all tested circuits, the ADAM optimizer underperformed relative to COBYLA, with most accuracies clustering around or below 53%. This outcome highlights the challenges of gradient-based optimization in quantum models and emphasizes the need for careful optimizer selection in VQC training.

TABLE 4. VQC CLASSIFICATION ACCURACY WITH DIFFERENT FEATURE MAPS, VARIATIONAL FORMS, AND OPTIMIZERS

Feature Map	Variational Form	COBYLA	ADAM
RawFeatureVector	RealAmplitudes	62.2%	—
	TwoLocal	—	—
	EfficientSU2	—	—
ZFeatureMap	RealAmplitudes	63.5%	50%
	TwoLocal	65.9%	53%
	EfficientSU2	68.0%	51%
ZZFeatureMap	RealAmplitudes	53.0%	49%
	TwoLocal	68.7%	48%
	EfficientSU2	68.7%	50%

3.1. Comparison with Original Study

Table 5 presents a side-by-side comparison of the classification accuracy achieved by our implementation of the Variational Quantum Classifier (VQC) using the COBYLA optimizer and the corresponding results reported in the original study.

TABLE 5. COMPARISON OF VQC ACCURACY BETWEEN OUR IMPLEMENTATION AND THE ORIGINAL STUDY USING COBYLA OPTIMIZER

Feature Map	Ansatz	Original Study	Ours
RawFeatureVec	RealAmplitudes	84.4%	62.2%
	TwoLocal	84.4%	—
	EfficientSU2	84.4%	—
ZFeatureMap	RealAmplitudes	76.8%	63.5%
	TwoLocal	74.6%	65.9%
	EfficientSU2	74.4%	68.0%
ZZFeatureMap	RealAmplitudes	53.2%	53.0%
	TwoLocal	68.4%	68.7%
	EfficientSU2	68.3%	68.7%

For the `RawFeatureVector` feature map combined with the `RealAmplitudes` ansatz, our implementation attained an accuracy of 62.2%, substantially lower than the 84.4% reported in the original study. No valid results were obtained for the other two ansätze with this feature map in our setting, as the components were incompatible in our implementation.

Likewise, the results obtained using the `ZFeatureMap` were lower than anticipated, which may be attributed to the feature map's sensitivity to input encoding and circuit configuration. Furthermore, discrepancies in the execution environment, including the quantum simulator or changes in the component's behavior across Qiskit versions, might have influenced the final performance.

In contrast, for the `ZZFeatureMap`, our implementation closely matched the original results. In particular, both the `TwoLocal` and `EfficientSU2` ansätze achieved an accuracy of 68.7%, slightly surpassing the original accuracies of 68.4% and 68.3%, respectively. The `RealAmplitudes` configuration yielded nearly identical performance to the reference (53.0% vs. 53.2%).

Overall, while discrepancies exist-particularly for the `RawFeatureVector`-the results for `ZFeatureMap` and `ZZFeatureMap` demonstrate good consistency with the original study, validating the correctness and effectiveness of our reimplementation under the updated software stack.

4. Conclusion

In this study, we reviewed and partially reproduced the hybrid quantum deep learning and Variational Quantum Classifier (VQC) model for botnet DGA attack detection. By re-implementing the VQC from scratch using updated versions of Qiskit, we validated several findings of the original work, particularly those related to circuit design and optimizer performance. Our results show that the `ZZFeatureMap`, when combined with expressive ansätze and the COBYLA optimizer, yields classification accuracy consistent with or exceeding original benchmarks.

Despite resource and software limitations, our experiments confirm the potential of VQC-based models in cybersecurity applications, especially when circuit configurations are carefully chosen. However, challenges in reproducing the hybrid quantum-classical model-primarily due to software deprecations and integration issues-underscore the evolving nature of quantum development environments and the practical barriers to widespread adoption.

Overall, our findings reinforce the promise of quantum-enhanced learning models in threat detection while emphasizing the need for continued toolchain stability to enable scalable and reproducible research in quantum machine learning.

5. Future Work

Future efforts should focus on extending this study in several key directions. First, a broader exploration of optimizers-including gradient-based and hybrid methods-could offer deeper insight into convergence behavior and performance trade-offs in VQC training. Additionally, evaluating models on real quantum hardware, rather than simulators, would allow for the assessment of noise resilience and practical deployment feasibility.

Another direction involves revisiting the hybrid quantum-classical model using updated frameworks.

Finally, expanding the feature set or employing quantum feature selection techniques could improve the model's capacity to generalize across diverse botnet behaviors. As quantum resources mature, exploring multi-class classification tasks or time-evolving threat landscapes may further demonstrate the applicability of quantum machine learning in real-world cybersecurity contexts.

Acknowledgments

The authors would like to thank Hatma Suryotrisongko and Yasuo Musashi, authors of the original study titled "Hybrid Quantum Deep Learning and Variational Quantum Classifier-Based Model for Botnet DGA Attack Detection", whose work served as the foundation for our implementation and analysis.

We also extend our appreciation to Dr. Rong Ge for her guidance and support throughout the Quantum Computing course. This paper was developed as the final project for the course, and her insights were invaluable to our learning process.

References

[1] H. Suryotrisongko and Y. Musashi, *Hybrid Quantum Deep Learning and Variational Quantum Classifier-Based Model for Botnet DGA Attack Detection*, in Proceedings of the 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), pp. 112–118, 2021.

[2] H. Abraham, A. AduOffei, R. Agarwal, *et al.*, Qiskit: An Open-source Framework for Quantum Computing, [Online]. Available: https://qiskit.org

[3] V. Bergholm, J. Izaac, M. Schuld, C. Granade, and N. Killoran, *PennyLane: Automatic Differentiation of Hybrid Quantum-Classical Computations*, arXiv preprint arXiv:1811.04968, 2018.

[4] M. Schuld and F. Petruccione, *Machine Learning with Quantum Computers*, Springer, 2nd ed., 2021.

[5] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, *Quantum Circuit Learning*, Phys. Rev. A, vol. 98, no. 3, p. 032309, 2018.

[6] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, *Cost-Function Dependent Barren Plateaus in Shallow Parameterized Quantum Circuits*, Nature Communications, vol. 12, no. 1, pp. 1–12, 2021.

[7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.