QUANTERA TECH

Research Proposal

Research Topic: Detection of Malware Using Quantum Convolutional Neural Network

Abstract:

Modern cybersecurity faces unprecedented challenges from sophisticated malware attacks targeting critical infrastructure systems, necessitating revolutionary detection methodologies that transcend the limitations of classical approaches. Traditional signature-based detection methods prove inadequate against zero-day threats, while classical machine learning approaches struggle with computational complexity and real-time adaptability to evolving attack patterns. This research proposes an advanced malware detection framework leveraging Quantum Convolutional Neural Networks (QCNNs) through a novel multi-encoding distributed architecture specifically designed to address current quantum hardware constraints while maximizing quantum computational advantages. The methodology employs a comprehensive six-stage pipeline integrating quantum data encoding strategies including hybrid angle encoding with section-specific Portable Executable (PE) binary analysis using distributed 8-qubit QCNNs. Each malware binary is systematically decomposed into critical PE sections (.text, .data, .rdata, .rsrc, .reloc), converted to 8×8 grayscale images, and processed through specialized quantum circuits employing parameterized gates with entanglement patterns for enhanced feature extraction. The distributed quantum processing outputs are then integrated through classical ensemble methods including XGBoost and Random Forest for final classification, with this hybrid classical-quantum integration serving as an optional enhancement to the core quantum framework. Experimental validation will be conducted on comprehensive malware datasets including BODMAS and PEMachineLearning repositories, along with additional specialized datasets, to ensure robust evaluation across diverse attack vectors. The proposed framework aims to significantly outperform classical approaches while maintaining practical deployment feasibility on NISQ-era quantum devices. Expected contributions include establishing systematic benchmarking standards for quantum cybersecurity applications, developing scalable quantum-classical hybrid integration strategies, and creating open-source frameworks for quantum-enhanced malware detection. This research addresses critical gaps in current quantum machine learning applications for cybersecurity, providing both theoretical advances in distributed quantum processing and practical solutions for real-world malware detection challenges in an increasingly connected digital infrastructure.

Keywords: quantum machine learning, quantum convolutional neural networks, malware detection, quantum computing, cybersecurity, distributed computing, hybrid quantum-classical systems, NISQ devices

Literature Review:

Quertier et al. (2023) addresses the challenges of limited qubit availability and information loss in quantum-based malware detection. The authors note that traditional QCNNs, constrained to eight or fewer qubits, have historically underperformed compared to classical models, particularly when tasked with analyzing high-dimensional malware binary images. To overcome this, the researchers introduce a distributed QCNN framework that decomposes each malware binary into five critical Portable Executable (PE) sections (.text, .data, .rdata, .rsrc, .reloc), converting each into an 8×8 grayscale image. Each section is then processed by a dedicated 8-qubit QCNN employing parameterized entangling gates for convolution and pooling layers for dimensionality reduction, with data encoded via angle embedding. The outputs—section scores or a value of -1 for missing sections—are then integrated through an XGBoost classifier for final malware or benign classification. The dataset is sourced from BODMAS and PEMachineLearning repositories, comprising tens of thousands of labeled PE files, and is split into subsets for QCNN training, scoring function training, and final testing. Results show that while individual QCNNs on sections like .rdata achieve moderate F1-scores (up to 0.78), the hybrid approach with XGBoost significantly outperforms both single-QCNN and classical baselines, reaching

83% accuracy and a 0.83 F1-score—representing a 20% improvement over monolithic QCNN models. The findings highlight the particular significance of the .rdata and .rsrc sections for malware detection, the utility of treating missing sections as informative features, and the synergy between quantum and classical methods for robust classification. The architecture is also scalable, allowing new sections to be added without retraining existing QCNNs. The authors recommend future research on incorporating additional PE sections, optimizing QCNN architectures for specific section types, and exploring weighted scoring strategies to further enhance detection performance. This work demonstrates that distributed quantum processing of binary sections, combined with classical ensemble learning, can substantially improve malware detection accuracy while remaining feasible for near-term quantum hardware.

1. Link:- https://arxiv.org/pdf/2312.12161

Akash et al.(2022) highlighted the growing concern of malware threats targeting smart grid devices, particularly due to the increased connectivity and seamless firmware updates in modern power systems[1]. One notable study proposes a cloud-based, device-specific malware detection system that leverages a Quantum Convolutional Neural Network (QCNN) integrated with Deep Transfer Learning (DTL) to address the limitations of classical machine learning approaches in this domain. The primary challenge tackled by this research is the inadequacy of conventional CNNs to extract deep, discriminative features from malware image files, especially under the computational constraints of current quantum hardware, which limits the number of available qubits and thus the input image size. To overcome this, the authors design a QCNN architecture that encodes malware binaries—collected from both benign firmware and manipulated Conti ransomware—into grayscale images suitable for quantum processing, with further dimensionality reduction to fit within a seven-qubit limit. The QCNN employs quantum circuits, including RY gates for data encoding and controlled-rotation gates for convolution, inspired by classical Sobel filters, to enhance feature extraction. The DTL component utilizes a pre-trained ResNet50V2 model, retraining only the final layers on device-specific data to adapt to new smart grid devices without losing generalization. The system is deployed on the IBM Watson Studio cloud platform, utilizing IBM Quantum processors for the quantum computations. Experimental results demonstrate that the QCNN-based approach achieves higher detection accuracy and faster convergence compared to classical CNNs, validating the effectiveness of quantum feature extraction in malware discrimination. However, the study also acknowledges limitations such as the need for aggressive image downscaling due to qubit constraints, focus on a narrow range of malware types, and reliance on specific quantum hardware, with future work suggested to address broader malware scenarios, improve noise resilience, and expand device applicability

2. Link:- https://par.nsf.gov/servlets/purl/10566964

Dataset:

BODMAS Dataset:

Link: https://ieeexplore.ieee.org/document/9474321

PE malware Learning Dataset:

Link: https://arxiv.org/abs/1804.04637

PE malware Machine Learning Learning Dataset:

[Warning!!] Don't download the files which contains inside this link

Link: https://practicalsecurityanalytics.com/pe-malware-machine-learning-dataset/

Malware Analysis Dataset [Kaggle]:

Link: https://www.kaggle.com/datasets/ang3loliveira/malware-analysis-datasets-top1000-pe-imports

More info on Malware dataset:

Link: https://hal.science/hal-03881198/document

Methodology:

Major Stages are:

1. Data Collection
2. Data Preprocessing
3. Encoding
4. QCNN Classification
5. Hybrid Classical-Quantum Integration (Optional)
6. Performance Evaluation

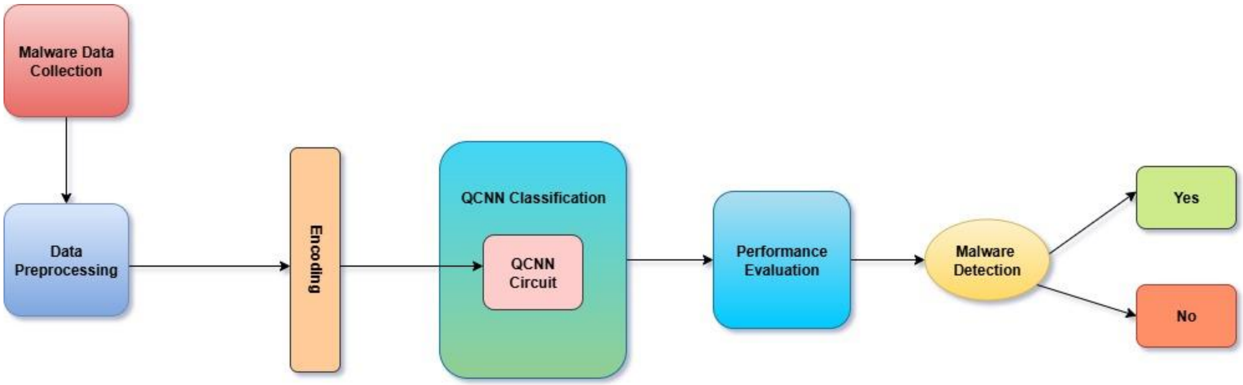

Figure: Malware Detection Work Flow

Explanation of the Stages-

1. Dataset Collection:

- BODMAS Dataset: 57,293 malicious PE files for comprehensive malware representation
- PEMachineLearning Dataset: 201,549 binary files ensuring diverse malware coverage
- Synthetic Variants: Generated samples for robustness testing

2. Data Preprocessing

- PE Section Extraction: Utilizing LIEF library for precise section identification (.text, .data, .rdata, .rsrc, .reloc)
- Optimized Image Conversion: Transform each 8×8 section into grayscale images with enhanced visualization techniques
- Intelligent Dimensionality Reduction: Apply PCA to reduce 64 features to 8 while preserving critical information
- Missing Section Handling: Systematic approach using -1 scoring for absent sections to maintain informational value

3. Encoding:

There different types of encoding strategy-

Amplitude Encoding (AE):

Maps classical data to quantum amplitudes: $|\phi(x)\rangle = (1/||x||)\ \Sigma_i\ x_i|i\rangle$

Advantage: Exponential data compression (2^n classical bits → n qubits)

Challenge: Circuit depth O(poly(N)) for state preparation

Qubit Encoding:

Individual data mapping: $|\phi(x_i)\rangle = \cos(x_i/2)|0\rangle + \sin(x_i/2)|1\rangle$

Advantage: Constant circuit depth, direct classical-quantum correspondence

Implementation: RY rotation gates with parameter scaling

Dense Qubit Encoding:

Dual-parameter encoding: $|\phi(x_j)\rangle$ = e^(-i$x_{j2}\sigma_y$/2) e^(-i$x_{j1}\sigma_x$/2)$|0\rangle$

Advantage: 2× data density per qubit compared to standard qubit encoding

Challenge: Increased complexity in parameter optimization

Hybrid Direct Encoding (HDE):

Block-based amplitude encoding: Multiple independent amplitude-encoded blocks

Parallelization advantage: Reduced circuit depth while maintaining data capacity

Optimal block size determination through empirical optimization

Hybrid Angle Encoding (HAE):

Normalized angle-based encoding resolving amplitude normalization issues

Implementation: Systematic angle assignment preventing information bias

Advantage: Consistent performance across varied data distributions

4. QCNN Classification:

Quantum Architecture Design:

Multi-QCNN Framework: Deploy five separate 8-qubit QCNNs, each specialized for specific PE sections

Circuit Optimization: Three-layer architecture with alternating convolutional and pooling layers

Adaptive Training: Utilize SPSB optimization for efficient parameter convergence

Quantum Circuit Components:

Convolutional Layers:

Parameterized two-qubit gates with translational invariance

Gate operations: RY(θ), RX(θ), RZ(θ) rotations with controlled interactions

Entanglement patterns: Nearest-neighbor and all-to-all connectivity options

Pooling Layers:

Quantum dimensionality reduction through partial trace operations

Controlled operations: CRZ(θ_1), CRX(θ_2) for adaptive pooling

Information preservation: Optimal qubit selection for tracing

Quantum Advantage Mechanisms:

Parallel State Evaluation: Leverage quantum superposition for simultaneous multi-state processing

Entanglement-Based Features: Exploit quantum correlations for enhanced pattern recognition

Resource-Efficient Design: Maximize information extraction within NISQ constraints

Distributed QCNN Implementation (Optional):

Five Independent QCNNs: Each optimized for specific PE section characteristics

- .text section QCNN: Optimized for executable code pattern recognition
- .data section QCNN: Specialized for data structure analysis
- .rdata section QCNN: Enhanced for read-only pattern detection
- .rsrc section QCNN: Advanced resource-based malware identification
- .reloc section QCNN: Relocation pattern analysis for obfuscation detection

Hybrid Classical-Quantum Integration:

(Optional Stage)

Advanced Score Aggregation Framework:

Missing Section Intelligence:

Systematic Absent Section Encoding: -1 scoring for missing PE sections

Information Preservation: Absence patterns as discriminative features

Adaptive Weight Assignment: Dynamic section importance based on malware type

Classical Ensemble Integration:

XGBoost Implementation: Gradient boosting for optimal section score fusion

Random Forest Alternative: Ensemble decision trees for robustness comparison

Neural Network Integration: Deep learning layers for complex score relationships

Adaptive Weighting Strategies:

Section Importance Analysis: Statistical significance testing for PE sections

Dynamic Weight Adjustment: Real-time adaptation based on malware evolution

Uncertainty Quantification: Confidence intervals for classification decisions

5. Performance Evaluation:

Multi-Metric Performance Assessment:

Accuracy Metrics: Standard classification accuracy, precision, recall, F1-score

Quantum-Specific Metrics: Circuit depth efficiency, parameter utilization, entanglement generation

Adversarial Robustness: Performance against obfuscated and adversarial malware samples

Computational Efficiency: Training time, inference speed, quantum resource utilization

Comparative Analysis Framework:

Classical Baselines: XGBoost comparisons

Quantum Alternatives: Single QCNN, hierarchical quantum classifiers

Hybrid Approaches: Various quantum-classical integration strategies

Hardware Compatibility: NISQ device constraints and error rate analysis

Statistical Validation:

Cross-Validation: K-fold validation with stratified sampling

Bootstrap Analysis: Confidence interval estimation for performance metrics

Significance Testing: Statistical hypothesis testing for quantum advantage claims

Primary Objectives:

1. Achieve Superior Detection Performance: Target accuracy improvement over classical approaches through optimized quantum feature extraction

2. Develop Distributed QCNN Architecture: Create a scalable framework that overcomes current qubit limitations while maintaining quantum advantages

3. Implement Distributed Quantum Processing: Design section-specific QCNN networks for PE binary analysis with intelligent score aggregation

4. Implement Hybrid Integration: Design practical quantum-classical fusion strategies for real-world deployment (Optional)

5. Investigate quantum advantages in adversarial malware scenarios

6. Develop scalable architectures supporting extended malware analysis

8. Optimize quantum circuit architectures for NISQ hardware constraints