QTagger: Quantum Machine Learning for Ransomware Tagging and Classification

Mentors: Dr. Simranjit Singh, Dr. Mohit Sajwan

Project Overview

Ransomware is one of the most pervasive and rapidly evolving cybersecurity threats today, with traditional machine learning models increasingly challenged by growing data size and complexity. This project proposes the use of Quantum Machine Learning (QML), with a focus on both purely quantum and hybrid classical-quantum models, to classify ransomware using large-scale malware datasets. The proposed project utilises frameworks like Qiskit, PennyLane, and TensorFlow Quantum, the project will explore techniques such as Quantum Support Vector Machines (QSVM), Variational Quantum Circuits (VQC), and Quantum Kernel estimation to build models that are efficient, scalable, and robust.

Interns will work on data preprocessing, feature extraction, model building, performance evaluation, and final reporting, while developing a strong foundation in scientific research practices.

Objectives

- Develop and benchmark quantum and hybrid models for ransomware classification.
- Compare performance with classical machine learning models.
- Evaluate the feasibility of quantum techniques on large, high-dimensional datasets.
- Train interns in scientific writing, quantum computing, and applied cybersecurity AI.

6-Week Plan

Week 1: Orientation & Literature Foundation

- Introduction to ransomware datasets and use cases.
- Onboarding with quantum programming tools (Qiskit, PennyLane, Cirq).
- Guided review of foundational research papers on QML and hybrid architectures.
- Intern deliverables: annotated summaries of two key papers.

Week 2: Dataset Preparation & Preprocessing

- Load and clean datasets such as EMBER, CIC-MalMem2022, or custom ransomware datasets.
- Feature extraction (e.g., API call traces, opcode sequences, binary entropy).
- Apply dimensionality reduction techniques (e.g., PCA/UMAP) for QML feasibility.
- Deliverables: Preprocessed dataset notebook + feature engineering report.

Week 3: Classical Baseline & Quantum Model Design

- Train and evaluate classical models (Random Forest, XGBoost, CNN).
- Design hybrid models: classical preprocessing + VQC or QSVM layer.
- Start coding basic quantum pipelines using simulators.
- Deliverables: Baseline performance summary; initial QML pipeline sketch.

Week 4: Model Implementation & Evaluation

- Implement and test full QML and hybrid models.
- Measure accuracy, F1, ROC-AUC, and compare to classical models.

- Identify scalability limits and propose enhancements.
- Deliverables: QML model codebase + preliminary results notebook.

Week 5: Optimization & Reporting

- Tune hyperparameters and perform validation experiments.
- Draft initial version of the research paper and implementation documentation.
- Visualize performance comparisons with classical benchmarks.
- Deliverables: Draft paper + optimized models.

Week 6: Final Review & Presentation

- Finalize paper, code, and documentation.
- Internal project presentation and peer review.
- Discuss future directions, possible conference submissions, and publication planning.

Mentorship & Guidance Plan

As the research mentor, We will provide a structured and interactive learning experience tailored to both technical depth and academic research readiness:

1. **Curated Learning Material**
    - A dedicated reading list will be shared including foundational papers and tutorials on quantum kernels, quantum SVMs, and hybrid neural architectures.
    - Additional resources will cover scientific writing and implementation documentation practices.
2. **Weekly Workshops**
    - We will conduct interactive sessions each week focusing on:
        - Effective scientific communication and paper writing.
        - Code documentation and reproducibility standards.
        - Interpreting experimental results for publication.
3. **Individual & Group Mentorship**
    - Each intern will have a 15–20-minute weekly check-in to address individual technical hurdles, receive focused feedback, and ensure alignment with project goals.
    - These stand-ups will also serve as a forum for cross-learning and collaboration among interns.
4. **Timely Feedback & Progress Tracking**
    - Written drafts and code contributions will be reviewed and commented upon within 24 hours to keep the research pace dynamic and responsive.
5. **Collaboration Tools & Reporting**
    - Interns will be guided on using GitHub for version control, documentation practices, and collaborative development.
    - Weekly deliverables and retrospectives will help in tracking and iterating over the project plan.