QuantumShield: Securing IoT Against Botnets with Quantum Computing

Duration: 6 Weeks
Project Mentor: Dr. Mohit Sajwan

1. Introduction & Motivation

The rapid expansion of Internet of Things (IoT) devices has revolutionized communication and automation across domains, but it has also introduced numerous security challenges. One of the most critical threats is botnet-based cyberattacks that exploit IoT vulnerabilities to create massive, coordinated attacks. QuantumShield aims to explore the integration of quantum computing with intelligent feature selection and machine learning for effective botnet detection in IoT networks. This project leverages quantum principles and nature-inspired algorithms to build a scalable, efficient, and future-proof cybersecurity solution.

2. Goals and Objectives

- Conduct a comprehensive background study on botnet attacks, IoT network architecture, and quantum computing.
- Apply nature-inspired algorithms for feature selection to enhance learning models.
- Develop a quantum-enhanced or hybrid classical-quantum model for botnet detection.
- Evaluate performance metrics such as accuracy, precision, recall, and F1-score.
- Explore future directions for improving scalability, robustness, and real-world deployment.

3. Weekly Breakdown

Week 1: Background Study & Onboarding

- Introduction to IoT networks, security vulnerabilities, and botnet attacks.
- Basics of quantum computing (qubits, gates, superposition, entanglement).
- Tools: Python, Scikit-learn, Qiskit/PennyLane, Jupyter, Discord.
- Readings and video lectures shared via Google Drive/Notion.

Deliverables: Summary notes, setup environment, GitHub repo access.

Week 2: Dataset Analysis & Preprocessing

- Load and explore IoT datasets (e.g., Bot-IoT, TON_IoT).
- Perform data cleaning, normalization, and visualizations.

Deliverables: EDA report and cleaned dataset.

Week 3: Feature Selection using Nature-Inspired Algorithms

- Study and apply Genetic Algorithms, Particle Swarm Optimization, etc.
- Select relevant features to optimize model performance.

Deliverables: Feature selection scripts and comparison metrics.

Week 4: Quantum/Hybrid Model Design & Implementation

- Implement quantum ML models (e.g., QSVM, VQC) using Qiskit/PennyLane.
- Compare with classical models like Random Forest, XGBoost.

Deliverables: Quantum model prototype and baseline comparison.

Week 5: Model Evaluation & Optimization

- Hyperparameter tuning, cross-validation, confusion matrix, ROC.
- Record precision, recall, F1-score, and accuracy.

Deliverables: Performance comparison and result visualizations.

Week 6: Final Report & Presentation

- Finalize documentation, prepare presentation slides.
- Discuss extensions like real-time deployment and quantum adversarial learning.

Deliverables: Final report, GitHub codebase, and presentation.

4. Guidance & Communication

- Daily or bi-weekly updates via Discord.
- Weekly review meetings.
- Interns grouped into Data, Feature, and Modeling teams.
- Continuous mentoring and code reviews.

5. Tools & Resources

- Python, Scikit-learn, Pandas, Seaborn
- Qiskit or PennyLane for quantum ML
- GitHub, Discord, Notion/Google Drive

6. Future Work & Research Direction

- Explore adversarial robustness using quantum techniques.
- Real-time deployment on edge devices.
- Academic paper submission in QML or Cybersecurity forums.
- Collaboration with quantum research labs.