Marla Dewey-Sumner
EID: sumnermm
Statistical Machine Learning
October 22, 2025

Project Proposal: Predicting PQC Algorithm Performance

Problem Statement

Many organizations will soon need to move from today's encryption methods to post-quantum cryptography (PQC) to stay secure against future quantum computers. One big question is how much slower these new algorithms might be when running on real systems. This project will use basic machine learning and data analysis to predict how long PQC operations take (in milliseconds) under different conditions. The goal is to understand which algorithms are slower or faster and what factors, like message size, make the biggest difference.

Setup and Data

I will use two sources of data:

1. My own PQC test harness, built using the Open Quantum Safe (OQS) library, which measures how long it takes to run operations such as key generation, encapsulation, decapsulation, signing, and verification.

2. A public Kaggle dataset that includes PQC benchmark results for several algorithms (for example Kyber, FrodoKEM, and Dilithium), with message sizes, security levels, and timing information. Source: Tariq, R. (2024). PQC Algorithms Benchmark Data. Kaggle. https://www.kaggle.com/datasets/ranatariq09/pqc-algorithms-benchmark-data

Both datasets will be cleaned and combined so each row shows one measurement for one algorithm and one type of operation.

Key Assumptions

- The Kaggle dataset represents realistic PQC performance.
- The main factors affecting latency are algorithm type, operation type, and message size.
- Timing measurements are consistent enough to model with simple methods.

Methods

1. Data Cleaning and Setup: Make column names and units consistent, and merge both datasets into one table.
2. Exploratory Analysis: Use charts and summary statistics to see how latency changes with algorithm and message size.
3. Modeling: Try basic models like linear regression and random forest regression to predict latency.
4. Use quantile regression to estimate median and upper-percentile latency (e.g., p50 and p95), which helps capture normal and worst-case performance. Evaluate model accuracy using metrics like mean squared error (MSE) and RÇ scores.
5. Visualization: Plot actual vs. predicted latency and show which variables have the biggest effect on performance.

Expected Outcome

The project will produce a model that estimates both typical latency and upper latency ranges for different PQC algorithms. These predictions can help teams planning system migrations understand how PQC might affect response times and scalability. The final deliverables will include a Jupyter Notebook with the data, models, and charts, and a complete written project report describing the results, analysis, and conclusions.

Assistance Acknowledgment: This proposal was developed with drafting and editing support from ChatGPT (GPT-5) to improve clarity and structure.