Quantum Machine Learning Course: Week 8 Project - Space Debris Risk Challenge

Overview

By 2030, the rapid expansion of low Earth orbit (LEO) satellites is expected to significantly increase the risk of collisions with debris from defunct spacecraft, threatening critical infrastructure for global communications, navigation, climate monitoring, and scientific research. The International Space Traffic Management Agency (ISTMA) is proactively developing advanced tools to assess and mitigate these risks, protecting over 30,000 satellites from millions of debris fragments, ranging from tiny fragments to large derelict objects. The Space Debris Risk Dataset (SDRD) provides complex orbital and telemetry data, including measurements like relative velocity, debris size, orbital inclination, and environmental factors, with noise from sensor inaccuracies and inconsistent formats. Your task is to predict the collision risk level (0: Low Risk, 1: Moderate Risk, 2: High Risk) for debris encounters using 40 high-dimensional features, comparing classical Support Vector Machines (SVMs) with Quantum Support Vector Machines (QSVMs) through hybrid quantum-classical approaches. Quantum machine learning may uncover intricate patterns in this noisy data, offering insights into effective risk assessment for safer space operations. You must preprocess the data, engineer features, perform feature selection, and submit your solution as described below by July 6, 2025, 8:00 PM IST.

Objective

Develop models to accurately predict the collision risk level for test set debris encounters, comparing the performance of classical and quantum SVM approaches. You must split the training data to create a validation set, preprocess the noisy data, engineer features, perform feature selection, and compare the two approaches.

Approach

Preprocess the dataset to handle noise, missing values, outliers, and inconsistent units. Engineer new features to improve model performance and perform feature selection to reduce dimensionality while retaining predictive power. Develop a classical SVM and a hybrid QSVM (using a quantum kernel paired with a classical classifier) for multi-class classification. Use the provided starter notebook for data loading and score calculation, and implement your own models to evaluate performance on your validation set using balanced accuracy and macro F1-score, comparing the classical and quantum approaches.

Dataset

The Space Debris Risk Dataset (SDRD) includes:

- **Training**: 5,000 samples (40 features + `risk_level`)
- **Test**: 300 samples (40 features, no labels)
- **Features**: 40 numerical features representing realistic LEO orbital and telemetry data, constrained to practical ranges:
  - **Orbital Parameters**:
    - `semi_major_axis_km`: 6,700–7,800 km
    - `eccentricity`: 0–0.05
    - `inclination_deg`: 0–120°
    - `orbital_period_min`: 90–120 min
    - `apogee_km`: 500–2,000 km
    - `perigee_km`: 400–1,800 km
    - `right_ascension_deg`: 0–360°
    - `argument_perigee_deg`: 0–360°
  - **Debris Properties**:
    - `debris_size_cm`: 1–100 cm
    - `debris_mass_kg`: 0.01–10 kg
    - `radar_cross_section_m2`: 0.0001–1 m^2
    - `debris_age_years`: 0–50 years
    - `material_density_kgm3`: 1,000–10,000 kg/m^3
  - **Relative Dynamics**:
    - `relative_velocity_kms`: 0–15 km/s
    - `miss_distance_km`: 0.01–100 km
    - `closest_approach_time_hr`: 0–24 hr
    - `relative_angular_momentum`: 1e6–1e9 $kg \cdot m^2/s$
    - `collision_prob_raw`: 0–0.001
  - **Satellite State**:
    - `sat_altitude_km`: 400–2,000 km
    - `sat_velocity_kms`: 7–8 km/s
    - `sat_mass_kg`: 100–5,000 kg
    - `sat_orbital_energy_j`: 1e9–1e12 J
    - `sat_antenna_gain_db`: 10–50 dB
    - `sat_power_w`: 500–10,000 W
  - **Environmental Factors**:
    - `solar_flux_index`: 70–250 (F10.7 index)
    - `geomagnetic_index`: 0–9 (Kp index)
    - `atmospheric_drag_coeff`: 0.1–2.5
    - `space_weather_index`: 0–100

        - `ionospheric_delay_ms`: 0–10 ms
    - **Sensor Data**:
        - `range_error_m`: 0–50 m
        - `velocity_error_ms`: 0–10 m/s
        - `azimuth_deg`: 0–360°
        - `elevation_deg`: -90–90°
        - `sensor_noise_db`: 0–30 dB
        - `tracking_accuracy_m`: 0–100 m
    - **Derived Metrics**:
        - `orbital_energy_j`: 1e9–1e12 J
        - `relative_inclination_deg`: 0–60°
        - `debris_kinetic_energy_j`: 1e6–1e10 J
        - `sat_maneuverability_score`: 0–1
        - `debris_trajectory_uncertainty_m`: 0–500 m
        - `collision_time_window_min`: 0–60 min
- **Labels**: 3 classes:
    - 0: Low Risk
    - 1: Moderate Risk
    - 2: High Risk
- **Access**: Provided as `.csv` files (`train.csv`, `test.csv`) and a starter Google Colab notebook in a Google Drive folder.

Instructions

Your Google Colab notebook must include:

- Code for preprocessing, feature engineering, feature selection, model training (classical SVM and QSVM), evaluation on your validation set, and test set predictions, building upon the starter notebook's data loading and score calculation functions.
- Proper Markdown cells explaining your approach, including preprocessing, feature engineering, feature selection, model training, evaluation results (with visualizations like confusion matrices or correlation plots), and a comparison of classical and quantum approaches (e.g., circuit depth, qubit count, runtime, kernel matrix properties, and performance differences).
- A `predictions.csv` file with three columns: `sample_id` (integer index 0 to 299), `label_classical` (predicted risk level: 0, 1, or 2), and `label_quantum` (predicted risk level: 0, 1, or 2).
- Reported validation scores (balanced accuracy and macro F1-score) in a Markdown cell, using the provided score calculation functions.

Evaluation Scores: Your submission will be evaluated based on:

- **Model Performance**: Assessed on the withheld test set using:

  - Balanced accuracy: (Sensitivity_class0 + Sensitivity_class1 + Sensitivity_class2) / 3, where sensitivity is the recall for each class.
  - Macro F1-score: Average of F1-scores across the three classes, where F1-score per class is 2 × (precision × recall) / (precision + recall).
- **Comparison of Approaches**: Assessed on the novelty and depth of your comparison between classical and quantum approaches, including validation performance, circuit depth, qubit count, runtime, kernel matrix properties, and impact of feature engineering/selection.
- **Code Quality**: Assessed on the clarity and modularity of your code.
- **Documentation**: Assessed on the clarity of Markdown cells and optional 3-minute video explaining your approach.

Submission Format

Submit via a Google Form (link shared on Friday, July 4, 2025) by July 6, 2025, 8:00 PM IST:

- A link to your Google Colab notebook with **viewing access** enabled.
- A `predictions.csv` file (embedded in the notebook or linked) with columns: `sample_id`, `label_classical`, `label_quantum`.
- An optional 3-minute video (Google Drive link) explaining your approach. Submissions with similar code or outputs to others will result in disqualification. Do not use external datasets or pre-trained models; plagiarism leads to disqualification.

Simulator Usage

Use simulators (Qiskit Aer or PennyLane's lightning.qubit) for QSVM development before attempting hardware, as Qiskit's free plan limits runtime to 10 minutes. Perform feature selection to manage quantum circuit complexity, ensuring efficient use of qubits for the high-dimensional dataset.

Starter Colab Notebook

The starter notebook (`Starter_QML_Week8.ipynb`), provided in the Google Drive folder, contains code for loading train.csv and test.csv, functions to calculate balanced accuracy and macro F1-score, a template for generating predictions.csv, and sample Markdown cells. You are responsible for preprocessing the data to handle noise, missing values, outliers, and unit inconsistencies, as well as implementing your own classical and quantum SVM models.