Task Breakdown for Group Members

Smart Bus Stop Analyzer Project

COMPX523 - Assignment 3

Overview

This document outlines the specific tasks and responsibilities for each of the 7 group members. Each member has been assigned tasks that match approximately 14-15% of the total workload.

Member 1: Algorithm Design & Core Implementation

Estimated Workload: 15%

Primary Responsibilities:

- Design the overall system architecture
- Implement the main `SmartBusStopAnalyzer` class
- Integrate CapyMOA algorithms (HalfSpaceTrees, AdaptiveRandomForest)
- Develop the core processing pipeline

Specific Tasks:

1. Create the main Python module structure
2. Implement `process_instance()` method
3. Set up CapyMOA model initialization
4. Write algorithm documentation
5. Code review for other members' contributions

Deliverables:

- `smart_bus_stop_analyzer.py` (core module)
- Algorithm design documentation
- Integration tests

Member 2: Data Processing & Feature Engineering

Estimated Workload: 14%

Primary Responsibilities:

- Load and preprocess the CSV datasets
- Design and implement feature extraction
- Handle missing data and outliers
- Create temporal features

Specific Tasks:

1. Implement data loading functions
2. Create `extract_temporal_features()` method
3. Develop `create_feature_vector()` method
4. Design feature normalization strategy
5. Write data preprocessing documentation

Deliverables:

- Data preprocessing pipeline
- Feature engineering module
- Data quality report

Member 3: Anomaly Detection Module

Estimated Workload: 14%

Primary Responsibilities:

- Implement anomaly detection using HalfSpaceTrees
- Design anomaly scoring mechanism
- Create anomaly visualization
- Evaluate detection performance

Specific Tasks:

1. Implement `detect_anomaly()` method
2. Tune anomaly threshold parameters
3. Create anomaly history tracking
4. Design anomaly alert system
5. Generate anomaly detection metrics

Deliverables:

- Anomaly detection module
- Performance evaluation report
- Anomaly visualization plots

Member 4: Prediction Module & Optimization

Estimated Workload: 14%

Primary Responsibilities:

- Implement passenger flow prediction
- Optimize model performance
- Create multi-horizon forecasting
- Evaluate prediction accuracy

Specific Tasks:

1. Implement `predict_passenger_flow()` method
2. Design multi-horizon prediction strategy
3. Optimize AdaptiveRandomForest parameters
4. Implement prediction error tracking
5. Create performance benchmarks

Deliverables:

- Prediction module
- Performance optimization report
- Prediction accuracy analysis

Member 5: Visualization & Dashboard

Estimated Workload: 15%

Primary Responsibilities:

- Create all visualizations in Jupyter notebook
- Design real-time dashboard simulation
- Implement plotting functions
- Create presentation graphics

Specific Tasks:

1. Implement all matplotlib/seaborn visualizations
2. Create `create_dashboard()` function
3. Design interactive plots
4. Generate figures for report
5. Create video demo of dashboard

Deliverables:

- Complete Jupyter notebook with visualizations
- Dashboard simulation
- Video demo (1-2 minutes)

- Presentation graphics

Member 6: Network Analysis & Multi-Stop Integration

Estimated Workload: 14%

Primary Responsibilities:

- Implement `MultiStopNetworkAnalyzer` class
- Design correlation analysis between stops
- Detect network-wide patterns
- Create network visualization

Specific Tasks:

1. Implement multi-stop processing logic
2. Design correlation matrix updates
3. Create `detect_network_anomaly()` method
4. Analyze stop-to-stop relationships
5. Visualize network patterns

Deliverables:

- Network analysis module
- Correlation analysis report
- Network pattern visualizations

Member 7: Documentation & Presentation Lead

Estimated Workload: 14%

Primary Responsibilities:

- Write the final report
- Create presentation slides
- Coordinate video recording
- Ensure code documentation
- Lead presentation preparation

Specific Tasks:

1. Write all sections of the report
2. Create presentation slides
3. Coordinate with all members for content
4. Review and edit all documentation
5. Prepare presentation speech

Deliverables:

- Final report (LaTeX/PDF)
- Presentation slides
- Presentation script
- Documentation review

Collaboration Guidelines

Communication

- Daily stand-up meetings (15 minutes)
- Shared Git repository with branch protection
- Slack channel for quick questions
- Weekly progress reviews

Code Standards

- PEP 8 compliance for all Python code
- Docstrings for all functions

- Type hints where applicable
- Unit tests for critical functions

Timeline

Week 1:

- Members 1-2: Core implementation and data processing
- Members 3-4: Algorithm modules
- Members 5-7: Planning and initial documentation

Week 2:

- Members 1-4: Integration and testing
- Member 5: Visualization development
- Members 6-7: Network analysis and documentation

Week 3:

- All members: Integration testing
- Members 5-7: Final visualizations and report
- Presentation preparation

Git Workflow

```
main
 ├── feature/core-algorithm (Member 1)
 ├── feature/data-processing (Member 2)
 ├── feature/anomaly-detection (Member 3)
 ├── feature/prediction (Member 4)
 ├── feature/visualization (Member 5)
 ├── feature/network-analysis (Member 6)
 └── feature/documentation (Member 7)
```

Quality Assurance

- Code review required for all PRs
- Minimum 80% test coverage
- Performance benchmarks must pass
- Documentation must be complete

Risk Management

Potential Risks & Mitigation

1. **CapyMOA compatibility issues**
   - Mitigation: Early testing, fallback implementations
2. **Large dataset processing**
   - Mitigation: Implement chunking, optimize memory usage
3. **Integration challenges**
   - Mitigation: Clear interfaces, regular integration tests

Backup Assignments

If any member cannot complete their tasks:

- Member 1 backs up Member 4
- Member 2 backs up Member 3
- Member 5 backs up Member 6
- Member 7 coordinates reassignments

Success Criteria

Technical

- ✓ All algorithms implemented using CapyMOA
- ✓ Processing time < 5ms per instance

- ✓ Memory usage < 100MB
- ✓ Anomaly detection F1-score > 0.85
- ✓ Prediction MAE < 5 passengers

Documentation

- ✓ Complete code documentation
- ✓ Comprehensive report (4-6 pages)
- ✓ Professional presentation
- ✓ Working demo video

Collaboration

- ✓ All members contribute equally
- ✓ Code reviews completed
- ✓ On-time delivery
- ✓ Successful presentation