Data Exploration & Classification Strategy Analysis

BMED 712 Project - Critical Design Decisions

QUESTION 1: Should we do more data exploration before ML?

SHORT ANSWER: YES, ABSOLUTELY! 🎯

You're catching a critical oversight. Jumping straight to ML without understanding the data is a recipe for:

- ❌ Missing obvious patterns that explain results
- ❌ Not understanding WHY the model works (or fails)
- ❌ Poor feature engineering (for classical ML)
- ❌ Weak discussion section in your report

ESSENTIAL DATA EXPLORATION TO DO NOW

1. SIGNAL VISUALIZATION & QUALITY CHECK ⭐ CRITICAL

What to look for:

> Goal: Understand what gait patterns LOOK like for each pathology

Specific analyses:

A. Compare Representative Trials

```python
# For each pathology, plot:
# - Healthy trial (baseline)
# - Neuro trial (what's different?)
# - Ortho trial (what's different?)

# For EACH of the 4 sensors (HE, LB, LF, RF)
# Look at acceleration patterns during walking
```

Expected insights:

- **Healthy**: Smooth, regular, symmetric patterns

- **Neuro (CIPN)**: Irregular, variable stride-to-stride, shuffling
- **Ortho (ACL)**: Asymmetric (left vs right), compensatory movements

B. Check for Data Quality Issues

```
python

Issues to look for:
□ Missing values (NaN, inf)
□ Sensor failures (flat lines)
□ Outliers (extreme values)
□ Sampling rate inconsistencies
□ Trial length variations
```

Why this matters:

- One bad sensor can ruin a trial
- Need to decide: filter out bad trials or impute?
- Affects train/test split strategy

---

2. GAIT CYCLE ANALYSIS ⭐⭐ VERY IMPORTANT

What to look for:

```
Goal: Understand the temporal structure of walking
```

Key metrics:

- **Stride time**: Time between consecutive heel strikes (same foot)
- **Cadence**: Steps per minute
- **Stride length**: Distance covered per stride
- **Symmetry**: Left vs right comparison

Expected findings:

Pathology	Stride Time	Cadence	Symmetry
Healthy	~1.0s	~110 steps/min	High (symmetric)
Neuro	Variable	Slower (~90)	Medium
Ortho	Longer	Slower (~95)	**Low (asymmetric)**

Why this matters:

- These are CLINICALLY MEANINGFUL features
- Doctors use these to assess patients
- Can validate your ML model ("Does it learn what doctors know?")

---

3. FREQUENCY DOMAIN ANALYSIS ⭐ IMPORTANT

What to look for:

Goal: Identify dominant frequencies in gait

Analysis:

```python
# For each sensor, compute FFT (Fast Fourier Transform)
# Look for:
# - Dominant frequency (should be ~1-2 Hz for walking)
# - Frequency spread (tight = regular, wide = irregular)
# - Harmonic structure
```

Expected insights:

- **Healthy**: Sharp peak at gait frequency (~1.5 Hz)
- **Neuro**: Broader, less defined peak (irregular)
- **Ortho**: May have split peaks (asymmetric gait)

Why this matters:

- Frequency features are powerful for classical ML
- Can reveal subtle differences not visible in time domain

- May explain why certain sensors are important

---

4. SENSOR CORRELATION ANALYSIS ⭐ IMPORTANT

What to look for:

Goal: Understand how sensors relate to each other

Analysis:

```python
# Compute correlation between:
# - HE vs LB (how coupled is head to trunk?)
# - LF vs RF (bilateral symmetry)
# - Vertical vs horizontal accelerations
```

Expected insights:

- **Healthy**: High LF-RF correlation (symmetric)
- **Ortho**: Low LF-RF correlation (asymmetric)
- **Neuro**: Variable correlations (inconsistent)

Why this matters:

- Explains sensor ablation results
- Guides feature engineering
- Reveals redundancy (maybe don't need all 37 channels?)

---

5. STATISTICAL COMPARISON ⭐ ⭐ VERY IMPORTANT

What to look for:

Goal: Prove pathologies are statistically different

Analysis:

```python
```

```
# For each metric (stride time, cadence, RMS acceleration):
# 1. Compute mean ± std for each pathology
# 2. Run ANOVA or Kruskal-Wallis test
# 3. Post-hoc tests (which pairs are different?)
```

Example output:

```
Stride Time:
  Healthy: 1.05 ± 0.12s
  Neuro:   1.28 ± 0.24s **
  Ortho:   1.18 ± 0.18s *

  ANOVA: p < 0.001 (significantly different)
  Post-hoc: Healthy ≠ Neuro (p<0.001)
            Healthy ≠ Ortho (p<0.05)
            Neuro ≈ Ortho (p=0.12)
```

Why this matters:

- **This is your JUSTIFICATION for ML!**
- Shows that discrimination IS possible
- Identifies hardest classification pairs
- Informs model interpretation

RECOMMENDED EXPLORATION WORKFLOW

Phase 1: Visual Inspection (1-2 days)

Day 1-2: Create comprehensive visualization notebook
□ Plot 3 example trials per pathology (all 4 sensors)
□ Zoom into one gait cycle
□ Check for data quality issues
□ Document observations

Phase 2: Gait Metrics (2-3 days)

Day 3-4: Extract and compare gait parameters
□ Detect heel strikes (gait events)
□ Compute stride times, cadence

□ Statistical tests (ANOVA)
□ Create comparison plots

Phase 3: Frequency Analysis (1-2 days)

Day 5-6: Frequency domain exploration
□ FFT for each sensor
□ Power spectral density plots
□ Dominant frequency extraction
□ Compare across pathologies

Phase 4: Advanced Analysis (1 day - optional)

Day 7: Additional insights
□ Sensor correlations
□ Bilateral symmetry analysis
□ Trial duration distributions
□ Subject-level variability

Total: 5-7 days BEFORE starting ML

WHY THIS MATTERS FOR YOUR PROJECT

1. Better Features (Classical ML)

- Know which features discriminate → better RF/XGBoost
- Stride time, cadence, RMS are gold standard

2. Model Validation

- "Our model learned stride variability predicts Neuro" ← GOOD
- "Our model works but we don't know why" ← BAD

3. Error Analysis

- If model confuses Neuro/Ortho → you already know they're similar in stride time
- Can explain failures with domain knowledge

4. Publication Quality

- Reviewers EXPECT exploratory analysis
- "Understanding your data" is #1 rule of ML

5. Hypothesis Generation

- "We expect head sensor (HE) to distinguish Neuro due to postural instability"
- Test hypothesis with sensor ablation

WHAT YOU CAN SKIP (FOR NOW)

❌ **Subject demographics** - Already in paper, not critical for initial model ❌ **Multi-modal fusion** - Stick to IMU data first ❌ **Advanced preprocessing** - Processed data is already good ❌ **Cross-dataset validation** - Only one dataset available

---

==========

QUESTION 2: 3-Class vs 8-Class Classification?

SHORT ANSWER: Do BOTH, start with 3-class 🎯

OPTION A: 3-Class (Recommended Primary)

Classification Task:

> Healthy (HS) vs Neuro (CIPN) vs Ortho (ACL)

Advantages: ✅

1. **Aligns with clinical workflow**
   - First decision: Is patient healthy or impaired?
   - Second decision: Is impairment neurological or orthopedic?
2. **Sufficient sample size**
   - Healthy: 360 trials ✓
   - Neuro: 784 trials ✓

- Ortho: 212 trials ✓ (smallest, but acceptable)

3. **Clear clinical meaning**
    - Different treatment pathways
    - Different specialists (neurologist vs orthopedist)
4. **Matches reference paper**
    - Your paper used similar groupings
    - Can compare results directly
5. **Robust evaluation**
    - Enough data for proper train/val/test split
    - Leave-one-subject-out feasible

Disadvantages: ❌

1. **Loses subtype information**
    - Are all neuro conditions the same? Unknown.
    - Are all ortho conditions the same? Unknown.
2. **May be "too easy"**
    - If very different, might not test robustness
3. **Doesn't show full potential**
    - "Multi-pathology" name implies more granularity

OPTION B: 8-Class (Recommended Secondary)

Classification Task:

Based on dataset, likely:
1. Healthy (HS)
2. Parkinson's Disease (PD)
3. Cerebellar Ataxia (CA)
4. CIPN
5. Stroke (CVA)
6. Hip Osteoarthritis (HOA)
7. Knee Osteoarthritis (KOA)
8. ACL injury

Advantages: ✅

1. **True multi-pathology analysis**
    - Shows model can discriminate fine-grained patterns
    - Aligns with "robust gait phenotyping"
2. **More clinically valuable**
    - Specific diagnosis → specific treatment
    - E.g., PD meds vs physical therapy for CVA
3. **Tests true robustness**
    - Within-neuro: Can model distinguish PD from CIPN?
    - Within-ortho: Can model distinguish HOA from KOA?
4. **Better sensor ablation insights**
    - E.g., "RF critical for ACL (unilateral) but not HOA (bilateral)"
5. **Publication impact**
    - More impressive result
    - Closer to real-world deployment

Disadvantages: ❌

1. **Class imbalance gets WORSE**

> If Neuro (784) splits into:
> - PD: 300 trials
> - CIPN: 350 trials
> - CVA: 100 trials
> - CA: 34 trials ← TOO SMALL!

2. **Need MORE data per class**
    - Rule of thumb: 100+ samples per class
    - May not have this for rare subtypes
3. **Training more difficult**
    - Harder optimization
    - More class weights to tune
    - Longer training time
4. **Evaluation more complex**

- 8×8 confusion matrix (vs 3×3)
- Harder to interpret
- Need per-class metrics for 8 classes

5. **May not have ground truth**
    - Need to check: Are subtypes labeled in metadata?

MY RECOMMENDATION: HIERARCHICAL APPROACH 🎯

Do BOTH in sequence:

Stage 1: 3-Class (Week 1-3)

Primary analysis:
□ Healthy vs Neuro vs Ortho
□ All ML models (RF → CNN → Multi-Stream)
□ Full sensor ablation (15 combinations)
□ Robust evaluation (LOSO-CV)
□ Complete analysis

Result: Solid baseline, guaranteed to work

Stage 2: 8-Class (Week 3-4, if time permits)

Extended analysis:
□ Check if subtypes are labeled
□ If yes: Train best model from Stage 1
□ Focus on within-category confusion
□ Limited sensor ablation (top 3 configs)
□ Compare to 3-class results

Result: "Bonus" analysis, shows full potential

CLASSIFICATION STRATEGY COMPARISON

Aspect	3-Class	8-Class
Difficulty	Medium	Hard

Aspect	3-Class	8-Class
Clinical Value	High	Very High
Sample Size	Adequate (212-784)	Borderline (34-350?)
Training Time	Faster	Slower
Interpretability	Easier	Harder
Risk	Low (guaranteed results)	Medium (may not converge)
Publication	Good	Better (if works)
Project Scope	Perfect for 4 weeks	Tight for 4 weeks

CRITICAL: CHECK YOUR DATA FIRST!

Before deciding, you MUST verify:

```python
# Load metadata and check subtype labels
import json

pathologies = {}
for trial_file in all_metadata_files:
    with open(trial_file) as f:
        meta = json.load(f)
        pathology = meta.get('pathology')  # Main class
        subtype = meta.get('subtype')      # Subclass?

        if pathology not in pathologies:
            pathologies[pathology] = {}
        if subtype:
            pathologies[pathology][subtype] = \
                pathologies[pathology].get(subtype, 0) + 1

print("Class distribution:")
for path, subtypes in pathologies.items():
    print(f"\n{path}:")
    for sub, count in subtypes.items():
        print(f"  {sub}: {count}")
```

If output shows:

```
Healthy:
  HS: 360

Neuro:
  PD: 143
  CVA: 49

Ortho:
  HOA: 44
  ... (other ortho conditions)
```

Then 8-class is feasible!

But if output shows:

```
Healthy:
  HS: 360

Neuro:
  CIPN: 784  (no subtypes)

Ortho:
  ACL: 212  (no subtypes)
```

Then stick with 3-class (or find subtypes elsewhere)

FINAL RECOMMENDATIONS

Primary Goal (Required):

✓ 3-Class classification (Healthy vs Neuro vs Ortho)
✓ Complete exploratory data analysis
✓ Multiple ML models with ablation
✓ Robust evaluation + error analysis

Stretch Goal (If time permits):

✓ 8-Class classification (all subtypes)
✓ Compare 3-class vs 8-class performance

✓ Hierarchical classification analysis
✓ Per-subtype sensor importance

Project Timeline Adjustment:

Original plan:

- Week 1: RF baseline
- Week 2: CNN
- Week 3: Multi-Stream + ablation
- Week 4: Report

NEW RECOMMENDED PLAN:

- **Week 1: Exploratory Data Analysis** ← NEW!
- Week 2: RF + XGBoost (3-class)
- Week 3: CNN + Multi-Stream (3-class) + ablation
- Week 4: 8-class (if feasible) + Report

DELIVERABLES CHECKLIST

Must Have (3-Class):

□ Comprehensive EDA notebook with visualizations
□ Statistical tests showing pathologies differ
□ 3+ ML models trained and compared
□ Sensor ablation results (15 configs)
□ Confusion matrices + per-class metrics
□ Error analysis with clinical interpretation

Nice to Have (8-Class):

□ Subtype distribution analysis
□ 8-class classification results
□ Within-category confusion analysis
□ Hierarchical classification comparison
□ Subtype-specific sensor importance

WHAT I'LL CREATE FOR YOU NEXT

Based on your questions, you need:

1. 📊 **Complete EDA Notebook**
    - Signal visualization
    - Gait cycle analysis
    - Statistical tests
    - Quality checks
2. 🔍 **Data Structure Verification Script**
    - Check for subtype labels
    - Count samples per class
    - Recommend 3-class or 8-class
3. 📈 **Updated Project Timeline**
    - Include EDA phase
    - Adjust ML timeline
    - Milestones and checkpoints

Which one should I create first? Or all three? 🚀

SUMMARY

Question 1: More exploration? → YES! 5-7 days of EDA is essential

Question 2: 3-class or 8-class? → Start with 3-class (guaranteed), add 8-class if feasible

Next steps:

1. Verify subtype labels exist
2. Complete EDA (Week 1)
3. Train 3-class models (Week 2-3)
4. Attempt 8-class if time (Week 4)

This approach gives you:
✓ Solid understanding of data
✓ Guaranteed 3-class results
✓ Potential for impressive 8-class
✓ Complete, publishable analysis

Ready to start? Let me know what to build first! 🎯