University of Maribor

Faculty of Electrical Engineering and Computer Science

FERI

PATH OVERLAP DETECTION AND PROPERTY GRAPH CONSTRUCTION FROM TCX DATA FOR ADVANCED ANALYSIS

IEEE 23rd World Symposium on Applied Machine Intelligence and Informatics (SAMI 2025)

Authors:

Tilen Hliš (tilen.hlis@um.si), Iztok Fister (iztok.fister@proton.me), and Iztok Fister Jr. (iztok.fister1@um.si)

PRESENTATION AGENDA

1. Motivation and Purpose of the Article
2. TCX Files Reading
3. Property Graphs
4. Path overlap Detection Algorithm
5. Experimental Results
6. Discussion and Future Work
7. Conclusion

MOTIVATION

- **Cycling Performance Optimization:**
  Cyclists heavily rely on GPS data (speed, heart rate, distance) for training analysis and performance improvement.

- **Challenge of Data Complexity:**
  TCX files store large amounts of unstructured data (GPS, heart rate, cadence), making **advanced analysis difficult**.

- **Need for Deeper Insights:**
  Detecting overlapping paths in cycling sessions can reveal **behavioral patterns** and **performance trends**.

PURPOSE

- **Efficient Data Analysis:**
  Develop a method to convert TCX data into property graphs for advanced analysis.
- **Path Overlap Detection:**
  Detect overlapping cycling paths using KDTree-based algorithms with tolerance for GPS inaccuracies.
- **Performance Insights:**
  Enrich detected segments with performance metrics (speed, heart rate, power, cadence,...) for better analysis.
- **Future Expansion:**
  Enable integration with Numerical Association Rule Mining (NARM) for predictive behavior modeling.

TCX DATA STRUCTURE

What is a TCX File?

- **Training Center XML (TCX):**
  Developed to store data from GPS-enabled fitness devices.
- **Stores detailed workout data:**
  - **GPS Coordinates** (latitude, longitude, altitude).
  - **Performance Metrics** (speed, heart rate, cadence, power).
  - **Time-based session details.**

TCX File Hierarchy

- **Activity:**
  Overall session data (sport type, session metadata).
- **Lap:**
  Segments of the session (time, distance, heart rate).
- **Trackpoint:**
  Individual data points (timestamp, GPS, metrics).

TCX FILE


```xml
<TrainingCenterDatabase>
  <Activities>
    <Activity Sport="Biking">
      <Id>2023-10-19T10:38:47.000Z</Id>
      <Lap StartTime="2023-10-19T10:38:47.000Z">
        <TotalTimeSeconds>5050.412</TotalTimeSeconds>
        <DistanceMeters>29305.5</DistanceMeters>
        <MaximumSpeed>10.039999961853027</MaximumSpeed>
        <Calories>571</Calories>
        <AverageHeartRateBpm>
          <Value>121</Value>
        </AverageHeartRateBpm>
        <MaximumHeartRateBpm>
          <Value>160</Value>
        </MaximumHeartRateBpm>
        <Intensity>Active</Intensity>
        <TriggerMethod>Manual</TriggerMethod>
        <Track>
          <Trackpoint>
                <Time>2023-10-19T10:38:47.000Z</Time>
                <Position>
                  <LatitudeDegrees>46.3886827044189</LatitudeDegrees>
                  <LongitudeDegrees>15.728210052475333</LongitudeDegrees>
                </Position>
                <AltitudeMeters>237.60000610351562</AltitudeMeters>
                <DistanceMeters>0.0</DistanceMeters>
                <HeartRateBpm>
                  <Value>91</Value>
                </HeartRateBpm>
                <Extensions>
                  <ns3:TPX>
                    <ns3:Speed>0.0</ns3:Speed>
                  </ns3:TPX>
                </Extensions>
              </Trackpoint>
```

TCX FILES READING

TCXReader.jl Library (https://github.com/firefly-cpp/TCXReader.jl)

- **Custom-built Julia library** for efficient reading and processing of TCX files.
- Optimized for handling large datasets with **high performance**.
- Converts raw TCX data into structured formats for further analysis.

```
struct TCXTrackPoint
        time::DateTime
        latitude::Union{Nothing, Float64}
        longitude::Union{Nothing, Float64}
        altitude_meters::Union{Nothing, Float64}
        distance_meters::Union{Nothing, Float64}
        heart_rate_bpm::Union{Nothing, Int}
        cadence::Union{Nothing, Int}
        speed::Union{Nothing, Float64}
        watts::Union{Nothing, Int}
end
```

CONSTRUCTING THE PROPERTY GRAPH

Purpose of Property Graphs

- Provides a **structured representation** of TCX data for efficient storage and retrieval.
- Organizes **GPS track points** and related **performance metrics** for analysis.
- Facilitates the enrichment of detected overlapping segments with performance data (after detection using KDTree).

What is a Property Graph?

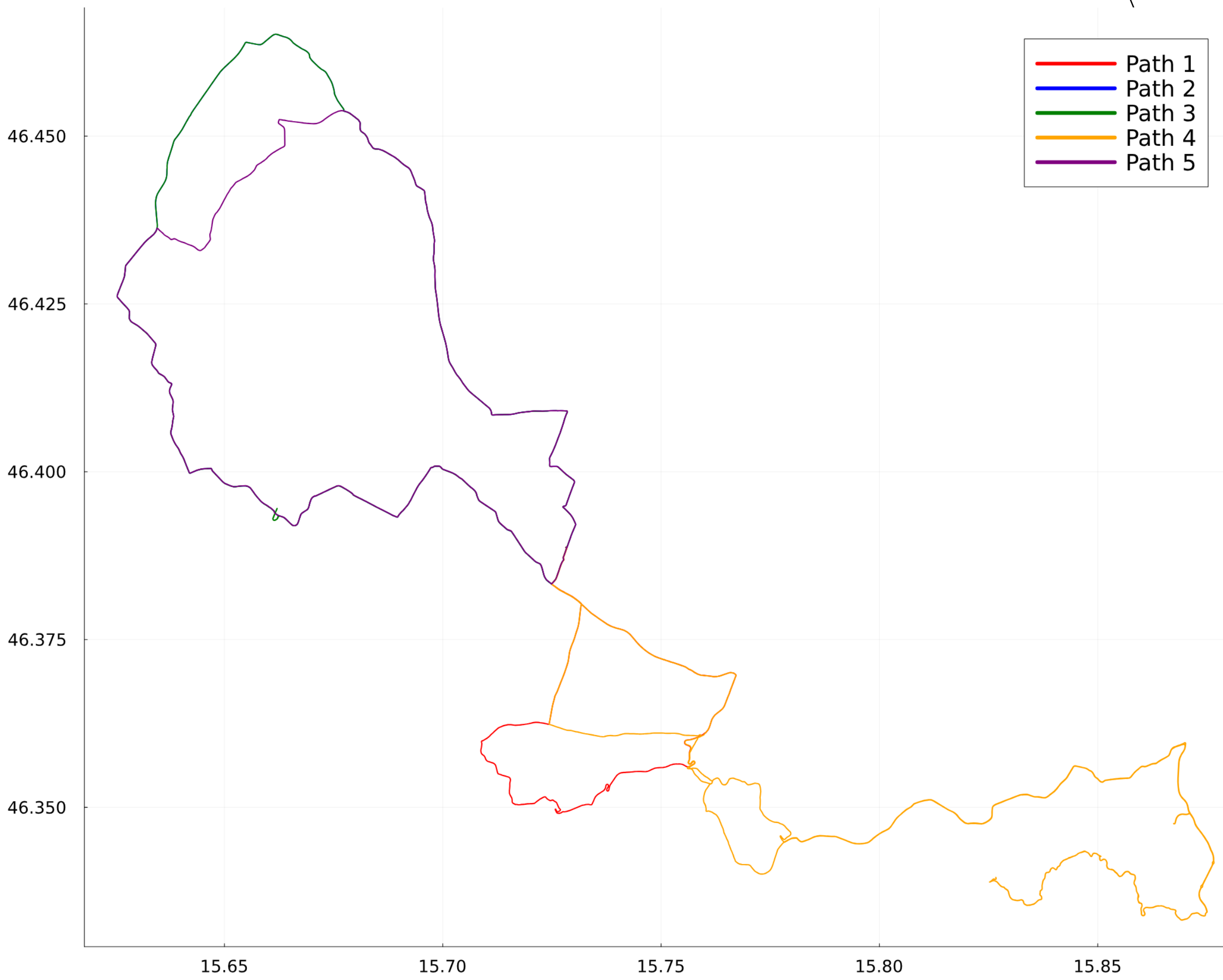
- A **Directed Multigraph** with nodes and edges storing rich data.
- **Nodes (Vertices):** Represent **GPS track points** from TCX files.
- **Edges (Arcs):** Connect consecutive points, forming the athlete's path.
- **Properties:** Store key-value pairs (e.g., speed, heart rate, cadence).

Formal Definition: $G=\langle N,A,K,V,\alpha,\kappa,\pi\rangle$

CONSTRUCTING THE PROPERTY GRAPH

Graph Construction Workflow

1. **Parse TCX Files:** using TCXReader.jl.
2. **Add Nodes:** Each GPS track point becomes a graph vertex.
3. **Connect Nodes with Edges:** Sequential points are linked to form a path.
4. **Attach Properties:** Key metrics (speed, heart rate, cadence,...) are stored as properties of nodes and edges.



PATH OVERLAP DETECTION

Motivation for Path Overlap Detection

- **Cyclists often repeat routes**, leading to overlapping segments in TCX data.
- **Detecting these overlaps** helps analyze how performance changes across sessions.
- Understanding repeated segments provides insights into **training behavior** and **physiological responses**.

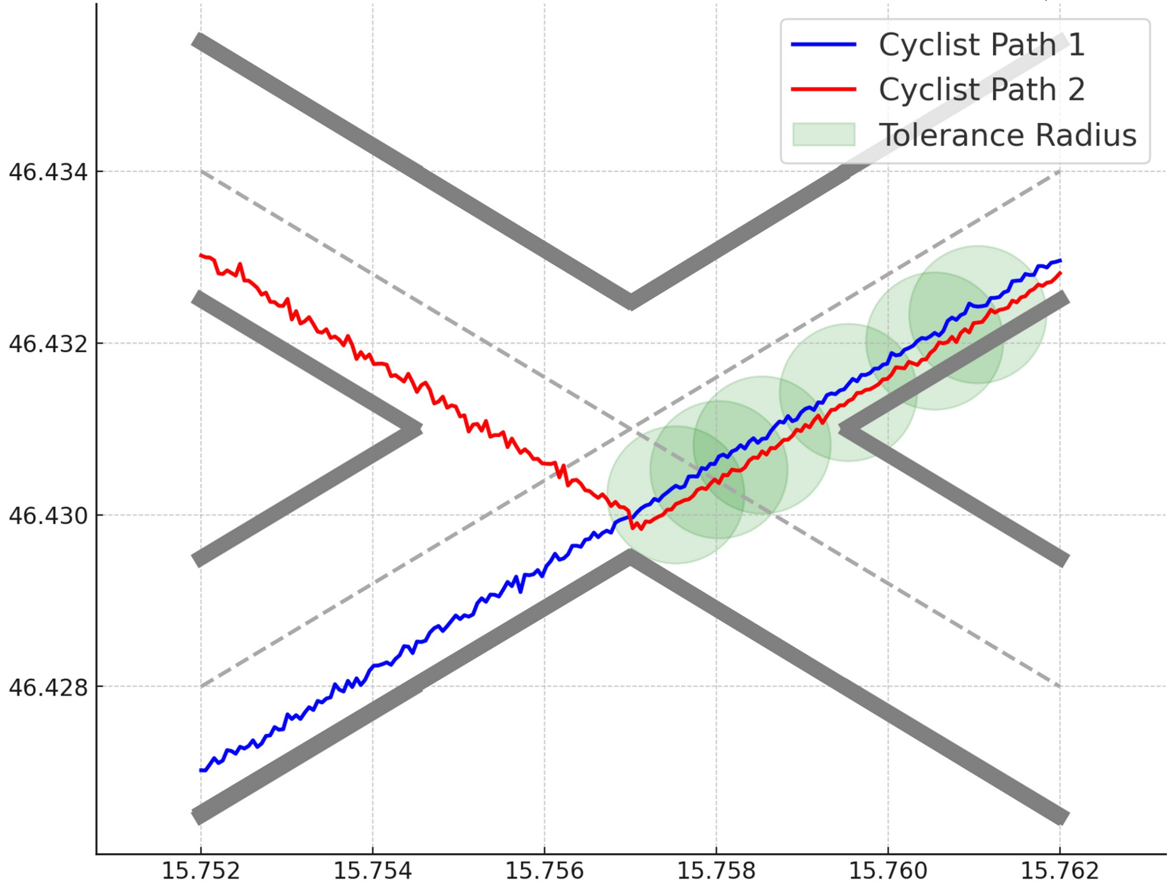
Challenges in Detection

- **GPS Inaccuracies:** Slight deviations in recorded data due to device errors.
- **Path Variations:** Small changes in route due to cyclist behavior or environmental factors.
- **Efficient Processing:** Large datasets require optimized algorithms for real-time analysis.

PATH OVERLAP DETECTION

Algorithm Overview:

1. **Parse GPS data** and build the **KDTree**.
2. **Iterate through paths** to check for nearby points in other sessions.
3. **Apply tolerance** to identify consistent overlaps.
4. **Aggregate overlapping segments.**



Advantages of the Method

- **Efficient Spatial Search** with KDTree for large datasets.
- **Accurate Detection** using a customizable tolerance to handle GPS noise.
- **Scalable Solution** for analyzing multiple sessions.

EXPERIMENTAL SETUP

Objective

- Evaluate the performance of the **Path Overlap Detection Algorithm**.
- Analyze how overlapping segments relate to **athlete performance**.

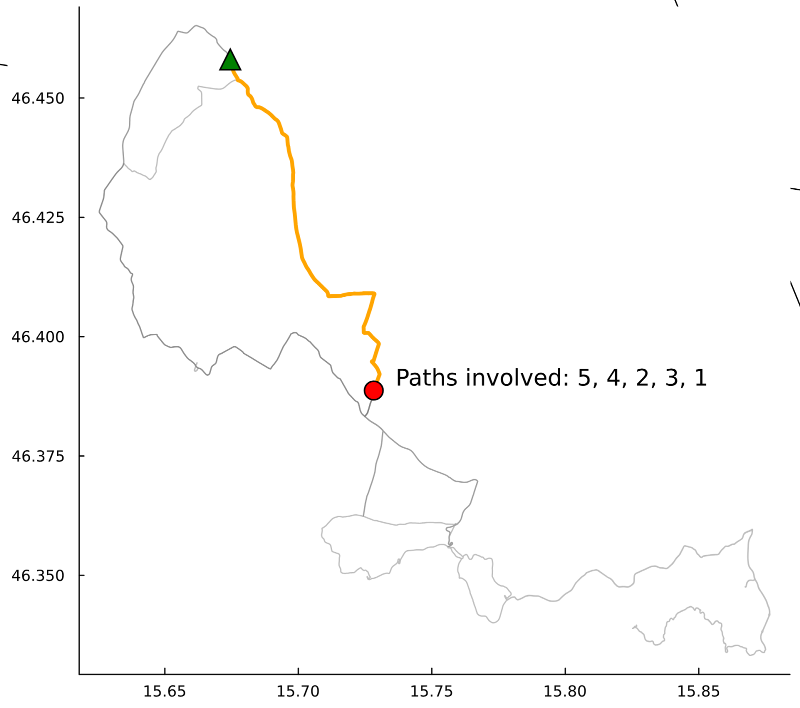
Dataset

- **Total Files:** 462 TCX files.
- **Selected for Analysis:** 5 files from the same cyclist.
- **Detected Overlapping Segments:** 39 overlapping segments.
- **Detailed Analysis:** Segment 30 (5 rides) and Segment 18 (3 rides).

RESULTS: SEGMENT 30 (5 RIDES)

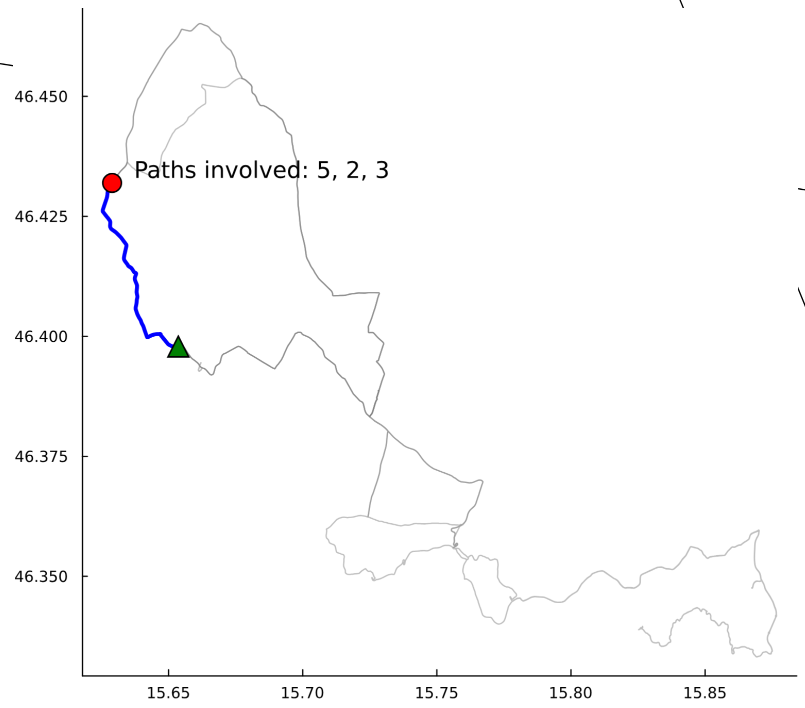
Observations

- **Higher variance** in speed and heart rate.
- **External factors** (e.g., wind) influenced performance (confirmed via Strava data).
- **Consistent cadence** indicates stable pedaling mechanics.



Metric	Min	Max	Mean	Median	Stdev
Speed (km/h)	22.5	25.2	24.16	24.8	1.19
Heart Rate (bpm)	143	165	151.6	147	9.48
Cadence (rpm)	79	86	83.4	85	2.88
Time (sec)	1,539	1,724	1,607.6	1,563	81.43

RESULTS: SEGMENT 18 (3 RIDES)



Observations

- **Consistent performance** across sessions.
- Low variance suggests **minimal external impact**.
- **Stable heart rate** and cadence throughout rides.

Metric	Min	Max	Mean	Median	Stdev
Speed (km/h)	21.5	22.3	21.93	22.0	0.40
Heart Rate (bpm)	162	166	163.67	163	2.08
Cadence (rpm)	75	81	78.33	79	3.06
Time (sec)	910	945	926.67	925	17.56

DISCUSSION

- **Effectiveness of the Algorithm:**
  The KDTree-based method effectively detected all overlapping segments with **zero false positives/negatives**.

- **Tolerance Factor:**
  Implementing a tolerance threshold allowed for **accurate detection** despite GPS inaccuracies.

- **Performance Insights:**
  Analysis of segments provided insights into how external factors (e.g., wind) and physical condition influenced performance.

- **Property Graph Utilization:**
  The property graph effectively organized TCX data, enabling easy retrieval of performance metrics for detected segments.

FUTURE WORK

1. **Integration with ARM:**
   Incorporate **Association Rule Mining (ARM)** for pattern discovery across multiple cyclists.
2. **Evolutionary Algorithms:**
   Integrate **Differential Evolution (DE)** and **Particle Swarm Optimization (PSO)** from NiaARM.jl for rule identification.
3. **Handling Larger Datasets:**
   Optimize the algorithm for large-scale, multi-user data analysis.
4. **External Factors Analysis:**
   Include environmental data (e.g., weather, terrain) to assess external impacts on performance.
5. **Expansion Beyond Sports Applications:**
   Adapt the algorithm for more **general use cases** beyond cycling and sports, enabling applications in areas like **transportation**, **logistics**, and **urban planning**.

THANK YOU FOR YOUR ATTENTION