Trajectory Anomaly Detection Methods Documentation

Understanding Human Movement Patterns During Emergencies

📋 Table of Contents

📖 Overview - What We Built {#overview}

The Big Question: Can we automatically detect when someone's daily movement patterns become unusual during emergencies?

Our Solution: We built a smart system that reads people's descriptions of where they went and automatically detects unusual behavior patterns. Think of it as a behavioral detective that learns what's normal for each person and spots when something changes.

Key Innovation: Instead of assuming everyone behaves the same way, our system learns individual patterns and detects anomalies personalized to each person.

🧠 Natural Language Processing (NLP) Methods {#nlp-methods}

What is NLP?

Natural Language Processing is teaching computers to understand human language. When someone writes "I went to the store, then the hospital," we need the computer to extract meaningful information from this text.

Method 1: spaCy Named Entity Recognition (NER)

What it does: Identifies and extracts specific types of information from text.

Example:

- **Input:** "I visited the grocery store on Main Street"
- **Output:**
  - Location: "grocery store"
  - Street: "Main Street"

Why we use it: spaCy is really good at finding places, times, and organizations in text. It's like having a smart highlighter that automatically marks important information.

```python
# Simple example of how spaCy works
import spacy
nlp = spacy.load("en_core_web_sm")
text = "I went to Walmart and then to the hospital"
doc = nlp(text)
for entity in doc.ents:
    print(f"{entity.text} -> {entity.label_}")
# Output: Walmart -> ORG, hospital -> FACILITY
```

Method 2: BERT Transformers (Advanced NER)

What it does: Uses artificial intelligence to understand context and meaning in text, even when people write in different ways.

Example:

- **Input 1:** "I went grocery shopping"
- **Input 2:** "I picked up some food items"
- **Input 3:** "I bought groceries"
- **Output:** All three are recognized as the same activity (shopping for food)

Why we use it: BERT understands that different phrases can mean the same thing. It's like having a smart translator that knows "grocery shopping," "buying food," and "picking up groceries" all mean the same activity.

Method 3: Rule-Based Pattern Matching

What it does: Uses predefined rules to identify patterns in text.

Example Rules:

- If text contains "hospital," "clinic," or "doctor" → Medical activity
- If text contains "store," "shop," or "mall" → Shopping activity
- If text contains "home," "house," or "residence" → Home location

Why we use it: Sometimes simple rules catch things that complex AI might miss. It's like having a checklist of obvious patterns.

Method 4: Feature Reconciliation

What it does: Combines results from all three methods above to create the most accurate understanding.

Example:

- spaCy finds: "Store"
- BERT finds: "Shopping activity"
- Rules find: "Retail location"
- **Reconciliation result:** Shopping trip to retail store

Why this is powerful: Instead of relying on one method, we use three different approaches and combine their strengths. If one method misses something, the others catch it.

🤖 Transformer Models {#transformer-models}

What are Transformers?

Transformers are advanced AI models that understand context and relationships in text. Think of them as super-smart reading comprehension systems.

How We Used BERT (Bidirectional Encoder Representations from Transformers)

BERT's superpower: It reads text in both directions (left-to-right AND right-to-left) to understand context.

Example of why this matters:

- Sentence: "I went to the bank near the river"
- **Without context:** "bank" could mean financial institution or riverbank
- **With BERT:** It understands from context that this is probably a riverbank

Custom Training for Mobility Data

We fine-tuned BERT specifically for mobility and location data:

Training Examples:

- "I grabbed coffee" → Venue: Café, Purpose: Social/Dining
- "I went to get my medication" → Venue: Pharmacy, Purpose: Medical
- "I met friends for dinner" → Venue: Restaurant, Purpose: Social

Result: BERT learned to understand mobility-specific language and extract location, timing, and purpose information more accurately.

📊 Statistical Analysis Methods {#statistical-methods}

Shannon Entropy - Measuring Behavioral Diversity

What it measures: How varied or predictable someone's behavior is.

Simple explanation:

- **High entropy:** Person goes to many different places (unpredictable)
- **Low entropy:** Person goes to the same few places (predictable)

Formula in plain English:

- Look at all the places someone visits
- Calculate how evenly they spread their time across different places
- More variety = higher entropy = less predictable behavior

Example:

- **Person A:** Goes to 10 different places equally → High entropy
- **Person B:** Goes to work 80% of the time, 2 other places 10% each → Low entropy

Jaccard Similarity - Comparing People

What it measures: How similar two people's behavior patterns are.

Simple explanation:

- Look at Person A's set of activities and Person B's set of activities
- Calculate: (Common activities) ÷ (All unique activities between them)

- Result ranges from 0% (completely different) to 100% (identical)

Example:

- **Person A activities:** {Coffee shop, Office, Gym, Store}
- **Person B activities:** {Coffee shop, Office, Restaurant, Park}
- **Common:** {Coffee shop, Office} = 2 activities
- **Total unique:** {Coffee shop, Office, Gym, Store, Restaurant, Park} = 6 activities
- **Similarity:** 2/6 = 33%

Mann-Whitney U Test - Statistical Significance

What it tests: Whether differences between two groups are real or just random chance.

Simple explanation:

- Compares two groups (e.g., normal period vs emergency period)
- Doesn't assume data follows a normal distribution (more robust)
- Gives a p-value: if $p < 0.05$, the difference is probably real

Example:

- **Question:** Do people visit more places during emergencies?
- **Test:** Compare number of venues visited in normal vs emergency periods
- **Result:** $p = 0.03$ means there's only a 3% chance this difference is random

Cohen's d - Effect Size

What it measures: How big or meaningful a difference is, not just whether it exists.

Simple explanation:

- Small effect ($d = 0.2$): Noticeable to careful observers
- Medium effect ($d = 0.5$): Obvious to anyone looking
- Large effect ($d = 0.8$): Very obvious, major difference

Example:

- People visit 2.3 venues normally vs 2.8 venues during emergencies
- Cohen's $d = 0.4$ means this is a small-to-medium meaningful difference

---

🔍 Hidden Markov Models (HMM) Analysis {#hmm-analysis}

What is a Hidden Markov Model?

Simple explanation: HMMs discover hidden behavioral "modes" that drive observable actions.

Real-world analogy:

- You can see someone's actions (going to coffee shop, office, gym)
- But you can't directly see their mental state ("work mode," "relaxation mode," "crisis mode")
- HMM figures out these hidden states from the patterns of observable actions

How We Used HMMs

State Discovery - Finding Hidden Behavioral Modes

What it found:

- **Routine State:** Predictable work-home-essential services pattern
- **Social State:** More restaurants, entertainment, friend visits
- **Crisis State:** Medical facilities, emergency services, supply shopping
- **Exploration State:** New places, recreational venues, varied timing

Transition Analysis - How People Switch Between Modes

Example transitions:

- Normal → Crisis: Person switches from routine to emergency response
- Crisis → Routine: Person returns to normal after emergency passes
- Routine → Social: Weekend shift from work mode to social mode

Viterbi Algorithm - Finding Optimal Sequences

What it does: For any sequence of activities, finds the most likely sequence of hidden states.

Example:

- **Observed:** Home → Hospital → Pharmacy → Home
- **Most likely hidden states:** Routine → Crisis → Crisis → Routine
- **Interpretation:** Person started normal, entered crisis mode for medical issue, returned to normal

EM Training - Parameter Optimization

What it does: Automatically learns the probabilities that govern state transitions and activity choices.

Example learned probabilities:

- In "Crisis State": 40% chance of visiting medical facilities
- In "Routine State": 60% chance of going to work
- Transition from Crisis → Routine: 30% probability per day

⚙️ Core System Components {#core-components}

Component 1: SubTrajectoryExtractor

Purpose: Breaks down daily descriptions into individual trips.

How it works:

1. **Home detection:** Looks for keywords like "home," "house," "residence"
2. **Time gap analysis:** If there's more than 60 minutes between activities, it's probably a new trip
3. **Venue change analysis:** Different types of venues often indicate trip boundaries
4. **Return home rule:** Going back home almost always ends a trip

Example:

- **Input:** "I went to coffee, then work, then came home for lunch, then back to work, then to the gym, then home"
- **Output:**
  - Trip 1: Home → Coffee → Work → Home
  - Trip 2: Home → Work → Gym → Home

Component 2: UserPatternLearner

Purpose: Learns what's normal for each individual person.

What it learns:

- **Venue preferences:** What places does this person usually visit?
- **Timing patterns:** When do they usually go out? How long do trips take?
- **Sequence patterns:** What routes do they typically follow?
- **Trip characteristics:** How many trips per day? What purposes?

Example for User "Sarah":

- **Typical venues:** Coffee shop (60%), Office (40%), Gym (20%), Store (30%)
- **Typical timing:** Leaves home at 8:30 AM, returns by 6:00 PM
- **Common sequences:** Home→Coffee→Office, Office→Gym→Home
- **Average trip duration:** 45 minutes

Component 3: SubTrajectoryAnomalyDetector

Purpose: Compares each trip against that person's learned patterns to detect anomalies.

Five-dimensional analysis:

1. **Venue:** Is this an unusual place for this person?
2. **Temporal:** Is this an unusual time for this person?
3. **Sequence:** Is this an unusual route for this person?
4. **Duration:** Is this trip unusually long/short for this person?
5. **Purpose:** Is this an unusual reason for this person?

Component 4: DailyTrajectoryAggregator

Purpose: Combines individual trip anomalies to decide if a whole day is unusual.

Four aggregation methods:

1. **Counting:** If X% of trips are anomalous, day is anomalous
2. **Severity:** If any trip is extremely anomalous, day is anomalous
3. **Pattern:** If overall daily pattern is unusual, day is anomalous
4. **Weighted:** Important trips count more than others

🎯 Five-Dimensional Anomaly Detection {#anomaly-detection}

Dimension 1: Venue Anomaly

Question: "Does this person usually visit this type of place?"

Method: Calculate overlap between current trip venues and typical venues

- **High overlap (low anomaly):** Person goes to their usual coffee shop
- **Low overlap (high anomaly):** Person who never visits medical facilities goes to hospital

Scoring:

- Score = 1 - (overlap with typical venues)
- Range: 0 (completely normal) to 1 (completely unusual)

Dimension 2: Temporal Anomaly

Question: "Is this person traveling at an unusual time?"

Method: Z-score deviation from typical timing patterns

- Calculate person's average trip start time and standard deviation
- Measure how far current trip deviates from this average
- Normalize using 3-sigma rule (99.7% of normal data falls within 3 standard deviations)

Example:

- Sarah usually leaves at 8:30 AM ± 30 minutes
- Today she leaves at 2:00 AM
- Z-score = (2:00 - 8:30) / 0:30 = -13 (extremely unusual)
- Normalized score: 1.0 (maximum anomaly)

Dimension 3: Sequence Anomaly

Question: "Does this person usually visit places in this order?"

Method: Compare against learned sequence patterns

- Check if current sequence matches any previously seen sequences
- Give partial credit for partially matching sequences
- New sequences get high but not maximum scores (people do try new routes)

Example:

- **Learned sequences:** Home→Coffee→Office, Home→Store→Home
- **Current sequence:** Home→Hospital→Pharmacy
- **Result:** High anomaly score (new sequence type)

Dimension 4: Duration Anomaly

Question: "Is this trip taking unusually long or short?"

Method: Z-score deviation from typical trip duration for this person

- Calculate person's average trip duration and standard deviation

- Account for individual variations (some people are naturally fast/slow)

Example:

- Sarah's store trips usually take 30 ± 10 minutes
- Today's store trip takes 3 hours
- Z-score = (180 - 30) / 10 = 15 (extremely unusual)
- Normalized score: 1.0 (maximum anomaly)

Dimension 5: Purpose Anomaly

Question: "Is this an unusual reason for this person to travel?"

Method: Frequency-based scoring

- Rare purposes get higher anomaly scores
- Common purposes get lower anomaly scores
- Personalized to individual's typical activity patterns

Example:

- Sarah's purposes: Shopping (40%), Work (30%), Social (20%), Recreation (10%)
- Today's purpose: Medical emergency (0% historically)
- Result: High anomaly score for purpose

📈 Daily Aggregation Methods {#aggregation-methods}

Method 1: Counting Method

Logic: "If enough individual trips are weird, the whole day is weird."

How it works:

1. Count how many sub-trajectories are anomalous (score > threshold)
2. Calculate percentage of anomalous trips
3. If percentage exceeds threshold (e.g., 50%), classify day as anomalous

Example:

- Day has 4 trips
- 2 trips are anomalous (scores: 0.7, 0.8)

- 2 trips are normal (scores: 0.1, 0.2)
- Anomaly rate: 2/4 = 50%
- If threshold is 40%, day is classified as anomalous

Strengths: Simple, interpretable, catches frequent anomalies **Weaknesses:** Might miss days with one extreme anomaly

Method 2: Severity Method

Logic: "If any trip is extremely weird, the whole day is weird."

How it works:

1. Find the maximum anomaly score among all trips
2. If maximum score exceeds high threshold, classify day as anomalous

Example:

- Day has 4 trips with scores: 0.1, 0.2, 0.95, 0.1
- Maximum score: 0.95
- If threshold is 0.8, day is classified as anomalous

Strengths: Catches extreme anomalies, sensitive to major disruptions **Weaknesses:** Very conservative, might miss subtle pattern changes

Method 3: Pattern Method

Logic: "Does the overall daily pattern make sense?"

How it works:

1. Look at the sequence of all trips in a day
2. Compare against learned daily patterns
3. Consider total number of trips, timing distribution, venue diversity

Example:

- **Normal daily pattern:** 2-3 trips, mostly routine venues, consistent timing
- **Current day:** 7 trips, all new venues, scattered timing
- **Result:** High pattern anomaly even if individual trips seem OK

Strengths: Holistic view, catches overall routine disruption **Weaknesses:** Complex to tune, harder to interpret

Method 4: Weighted Method

Logic: "Some trips are more important than others."

How it works:

1. Assign weights to trips based on importance factors:
   - Duration (longer trips weighted more)
   - Venue rarity (unusual venues weighted more)
   - Purpose significance (medical, emergency weighted more)
2. Calculate weighted average of anomaly scores
3. Compare against threshold

Example:

- Trip 1: Score 0.2, Weight 1.0 (routine coffee)
- Trip 2: Score 0.8, Weight 3.0 (medical emergency)
- Weighted average: $(0.2\times1 + 0.8\times3) / (1+3) = 0.65$
- If threshold is 0.5, day is anomalous

Strengths: Considers importance, balanced approach **Weaknesses:** Weight assignment requires domain knowledge

Ensemble Voting

Final decision: All four methods vote, majority wins

Example:

- Counting method: Anomalous (2/4 trips weird)
- Severity method: Normal (no extremely high scores)
- Pattern method: Anomalous (overall disrupted routine)
- Weighted method: Anomalous (medical trip weighted heavily)
- **Final result:** 3/4 vote anomalous → Day classified as anomalous

---

🔧 Why These Methods Work Together {#why-it-works}

Complementary Strengths

1. **NLP Methods:** Handle different text styles and extract comprehensive information

2. **Statistical Methods:** Provide rigorous validation and measure significance
3. **HMM Analysis:** Discovers hidden patterns we might not have thought of
4. **Multi-dimensional Scoring:** Captures different aspects of anomalous behavior
5. **Ensemble Aggregation:** Combines different perspectives for robust decisions

Robustness Through Redundancy

- If one NLP method fails, others compensate
- If one aggregation method misses an anomaly, others catch it
- Statistical validation ensures findings aren't due to chance
- HMM provides unsupervised validation of discovered patterns

Personalization at Scale

- Every component learns individual patterns
- No assumptions about "normal" population behavior
- Adapts to individual differences in mobility patterns
- Scales to analyze many users simultaneously

Real-World Applicability

- Handles messy, real human language
- Works with incomplete or imperfect data
- Provides interpretable results
- Can be adapted to different domains (healthcare, security, urban planning)

🎯 Summary

We built a comprehensive system that:

1. **Understands human language** about mobility using multiple NLP approaches
2. **Learns individual patterns** rather than assuming everyone is the same
3. **Detects anomalies across five dimensions** for comprehensive analysis
4. **Combines multiple perspectives** through ensemble methods
5. **Validates findings statistically** to ensure they're not random chance
6. **Discovers hidden behavioral states** through unsupervised learning

The result is a robust, personalized, and scientifically rigorous approach to understanding how human mobility patterns change during crises. This methodology can be applied to healthcare monitoring, emergency planning, security applications, and urban planning.

Key Innovation: Moving from population-level assumptions to individual-level learning, revealing that crisis response is personal, not universal.