# Deep Classification of Frequently-Changing Activities from GPS Trajectories

Emre Eftelioglu
efteli@amazon.com
Amazon
Bellevue, Washington, USA

Gil Wolff
wolffg@amazon.com
Amazon
Bellevue, Washington, USA

Sai Krishna Tejaswi Nimmagadda
snnimmag@amazon.com
Amazon
Hyderabad, India

Vishal Kumar
vishku@amazon.com
Amazon
Bellevue, Washington, USA

Amber Roy Chowdhury
amberch@amazon.com
Amazon Inc
Bellevue, Washington, USA

## ABSTRACT

Classifying trip modalities, i.e. driving, walking, etc., from GPS trajectories is one of the fundamental tasks for urban mobility analytics. It can be used for efficient route planning, human activity recognition, and public transportation design where understanding the time and location of transitioning to different modalities may provide additional insights. Informally, given a GPS trajectory consisting of temporally ordered GPS locations, trip modality/activity classification aims to assign trip modes to each GPS point. It is a challenging task due to the associated noise with the GPS data, the lack of knowledge about the underlying road network as well as the driving traffic conditions which may affect the trip behavior (e.g. driving slower than walking speed at rush hour traffic). Despite its widespread applications, the existing methods are either dependent on multi-sensor data (such as GPS, IMU, Camera, etc.) or use heuristic-based filtering to classify modalities of the trajectory datasets. Moreover, they consider limited number of transitions per trip making them inadequate for more frequent activity changes. In this paper, we propose a novel deep neural network architecture, Frequent Activity Classification Network *FACNet*, leveraging a bi-directional LSTM network and a custom Attention module to infer modality of GPS points in a trajectory with frequent modality changes. Our supervised learning approach depends only on the GPS trace without any additional inputs, making it applicable to a wide variety of modality related problems. Experiments confirm the superiority of our method compared to the related work as well as heuristic approaches. Finally, we provide access to a set of anonymized GPS trajectories that is made available to the broader research community to provide opportunities to further improve the existing research on the topic.


*IWCTS '22, November 1, 2022, Seattle, WA, USA*

ACM ISBN 978-1-4503-9539-7/22/11…$15.00
https://doi.org/10.1145/3557991.3567784

## CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Information systems** → **Geographic information systems**.

## KEYWORDS

Geospatial Data, GPS trajectories, Neural Networks, Urban Mobility

**ACM Reference Format:**
Emre Eftelioglu, Gil Wolff, Sai Krishna Tejaswi Nimmagadda, Vishal Kumar, and Amber Roy Chowdhury. 2022. Deep Classification of Frequently-Changing Activities from GPS Trajectories. In *The 15th ACM SIGSPATIAL International Workshop on Computational Transportation Science (IWCTS '22) (IWCTS '22), November 1, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3557991.3567784

## 1 INTRODUCTION

Given a GPS trajectory which consists of temporally ordered GPS points, trip modality classification aims to assign trip modes at the granularity of each GPS point. For the sake of simplicity, in this paper, two major trip modes were considered: walking and driving. Trip mode classification is important for applications such as human activity recognition, public transportation design, as well as efficient route planning [12]. Understanding these activities per trip can lead to better planning for urban infrastructure, e.g. planning parking locations, understanding real average speeds on roads, etc. as well as better understanding of the underlying causes of frequent modality changes, e.g. changing buses between stations, etc. In last mile logistics, trip mode classification is important to determine how much time is spent for driving related activities and package delivery related activities, i.e. from parking location to the customer's doorstep. Since such examples involve more frequent activity transitions, the detection of such cases become harder. Yet, accurately classifying high frequency modality changes become critical.

### 1.1 Challenges

Trip modality detection is a challenging task due to a variety of reasons. First, GPS data often have a variety of quality issues. The ubiquity of smartphones makes them a primary GPS data source. Yet smartphone sensors are embedded System on Chip (SoC) devices with limited signal reception, leading to accuracy degradation

in urban canyons, and near or inside buildings. More accurate GPS sensors are larger and consume more energy, making them unsuitable as smartphone sensors [13].

Second, GPS data is usually sampled at a low temporal resolution, bringing an ambiguity. For example, if GPS points are sampled at 5s intervals and the last known speed is 20 m/s, this may introduce a 100-meter radius blind-zone where the location of the driver is not known. There is always an option to have more frequent sampling, but higher sampling frequency causes increased storage and bandwidth costs, reduced battery life and varying noise especially with different brands/models of smartphones with different efficiency and sensor quality.

Third, GPS points are usually inadequate to successfully understand the human behavior when there is no additional data that can be coupled with it. In our work, we assume that we lack any additional sensor data, e.g. Inertial Measurement Unit (IMU). IMU is very beneficial for trip modality detection [16]. In fact, smart watches and fitness trackers use these sensors to determine walking activities often with high accuracy. However, these may bring an additional data transfer and storage cost as well as additional battery impact, which we try to avoid. It is worthwhile to note that the proposed approach in this paper is flexible enough to be extended for additional features that can be generated with IMU sensor data which can further improve the performance.

Finally, simple rule based approaches [19] may not be adequate since the traffic conditions vary by the underlying urbanization as well as traffic rules. For example, in downtown areas, at rush hour, the traffic often slows down enough for rule-based approaches to mistakenly infer walking. Similarly, the signal attenuation inside buildings often cause GPS points to bounce around, causing naïve rule-based approaches to mistakenly infer driving due to high speed calculated from between GPS points. Moreover, such rule based approaches typically due not enforce physical knowledge, and may lead to nonsensical output, such as driving for 1 second. Therefore, spatial and temporal dependencies should be taken into account.

## 1.2 Related Work

Related work on activity classification from GPS trajectories are usually focused on detecting bus/bicycle/walk types of activities by not only using the GPS points, but also with the addition of IMU sensor readings [6, 16]. Moreover, most existing work requires high sampling rates for GPS points (usually 1 point per second - pps) to be able to couple these with the IMU readings. More importantly, the existing approaches have to deal with a very limited number of transitions throughout the entire trajectory. For example, a trajectory starts with a walk to the bus stop, then the trip mode continues with bus mode, next walking and at the end of the day the sequence repeat in reverse. In our work, we focus on high frequency of transitions between different modalities. For example, in the in last mile delivery, these transitions may be more than 20 per hour. The comparison between these is shown in the Figure 1. Left side of Figure 1 illustrates traditional activity classification where the right side represents the high frequency activity classification problem we are focused on. In summary, even minor misclassifications would become much more detrimental due to these high frequency of transitions.


**Figure 1: Illustration of frequent number of transitions for an activity classification task. Left side represents traditional work seen in the research community. Right side represent the high frequency activity classification problem we focus on. Increased number of transitions have negative impact on the performance of all models.**

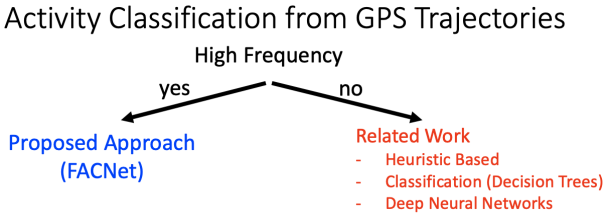



**Figure 2: Taxonomy of the related work**

As shown in Figure 2, despite the differences in their objective, there is a wide variety of research on activity classification from GPS trajectories [17, 21]. The work in this area generates trip-related features (e.g. speed, acceleration, etc) for each GPS point and uses them in a machine learning model to predict the modalities. Similarly, some work in this domain uses features that are generated by a deep neural network architecture to avoid the use of manually-chosen features [1, 2, 4]. Yet, these do not consider the temporal dependencies between each GPS point causing the models to have abrupt, i.e. physically impossible transitions (impossible de/acceleration rates, etc.). Finally, there are approaches which use LSTM framework to allow the model to learn the temporal dependencies between individual GPS points [7]. The approaches in this area provide state-of-the-art results for activity classification from GPS traces.

Some representative work [5] in this area uses embeddings over the point-based speed features. However, in [5], a simple addition of embedding vectors is performed before forwarding them to LSTM layer potentially limiting the performance of the model due to bottleneck in information flow. In [2], the authors proposed a semi-supervised method that use (de)convolutional auto-encoders to improve the model performance. Although the results are promising, the approach is not directly comparable to ours, since the proposed model focuses on detection of segments which are longer than a pre-defined threshold (20 minutes), instead of individual point-level classifications, which is the focus of our approach.

## 1.3 Contributions

Our contributions are twofold:

***First***, we propose a novel deep learning architecture, Frequently Changing Activity Classification Network *FACNet*, to infer the activity modalities from GPS trajectories. *FACNet* architecture is inspired from the large body of Natural Language Processing (NLP) research. When each GPS trajectory is considered as a paragraph and each sub-trajectory sequence is considered as a sentence, and

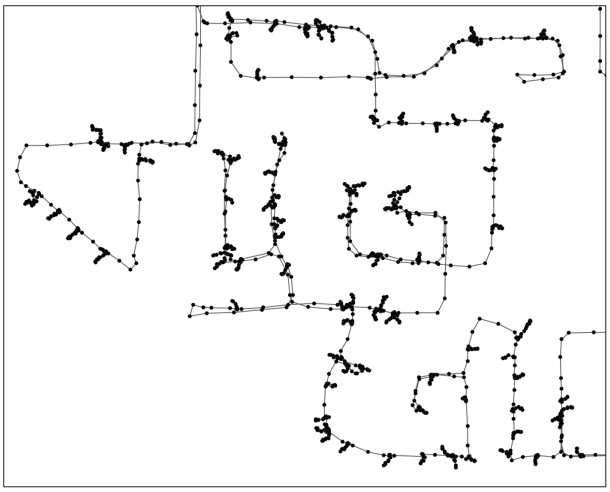


**Figure 3: An example GPS trajectory that was used in tests conducted for this paper. The entire trip is $> 6$ hours long which is represented with 4213 GPS points. Example shown in the figure is a subset with 2 hours of the trip with 1400 GPS points. A basemap is omitted for privacy.**

each GPS point is considered a word, our problem becomes analogous to a Part of Speech (PoS) tagging problem [11, 15]. Using this idea, we developed an architecture that uses an encoder, a dual layer bi-directional LSTM and a decoder module. We also developed an attention module that is convolved with the LSTM output to provide better focus for the classifications temporally nearby. We generate features that represent the motion, the geometric shape, and distances to incorporate all behavioral, road dependent and noise related signals collected by the GPS data. These features are fed to an encoder module to identify the intrinsic relationship between each other.

Even though there are efforts in the mobility classification field which use deep learning architectures and similar feature generation techniques, our proposed approach is unique in that (i) we use an attention module to improve the model performance, (ii) we blend different types of features, and (iii) we particularly focus on the high frequency activity transitions. Our experiments show that the proposed *FACNet* method, compared to an heuristic and a deep learning based method, outperforms them substantially. In addition, we propose two supplementary metrics (i.e. teleportation distance, distance to vehicle line) that can be used to evaluate any activity classification technique without labeled ground-truth data.

***Second***, as part of our commitment to the research in this topic, we provide access to a sample of the GPS trajectory data that we used in our experiments [3]. We believe that such data will open new opportunities in the field of the GPS Activity Classification. Researchers in academia will have access and be able to use it to evaluate the performance of their existing research.

### 1.4 Scope and Outline

This paper proposes an activity classification model which classifies individual GPS points in a trajectory. By framing the problem as a point-classification problem, our framework both subdivides the full trajectory into contiguous-activity sub-trajectories, called "activity segments", and assigns an activity label for each activity segment. This distinguishes our framework from other methods that accept sub-trajectories as input, which is a simpler problem.

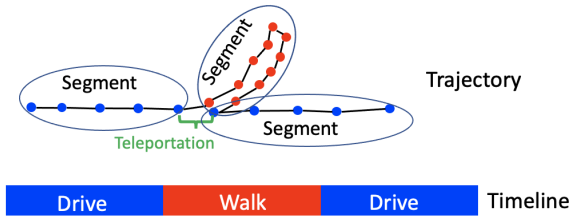



**Figure 4: An illustration of GPS trajectory, Segment, Activity Timeline as well as teleportation gap (best in color).**

The proposed approach currently considers two modalities, i.e. driving and not-driving (e.g. walking). Other modalities are out of the scope of this specific paper, although the architecture may easily be extended for other modalities too.

Many additional inputs can improve the model, i.e. underlying road network, IMU data, building outlines, vehicle OBD sensors, etc. However, we assume that the availability of all these are limited. Therefore, the goal is to achieve the maximum possible accuracy by only using the GPS trajectory, which makes the model usable in a wide variety of applications.

The rest of the paper is structured as follows: Section 2 introduces the basic concepts followed by the formal problem definition. Next, Section 3 explains the proposed approach, i.e. *FACNet*. The evaluations are done in Section 4. Finally Section 5 concludes the paper and provides an overview of our future directions.

## 2 BASIC CONCEPTS AND PROBLEM STATEMENT

In this section, we will describe the basic concepts that the proposed approach built on followed by the formal problem definition.

### 2.1 Basic Concepts

Definition 1. ***Trajectory:*** *A trajectory* $\mathbf{T}$ *is a sequence of temporally ordered points* $p_{1,2,\ldots,m}$*, where each* $p_i \in \mathbf{T}$ *represents a location (longitude, latitude) on the surface of Earth and a recording timestamp* $t$*. Since a trajectory has a temporal component, for each point* $p_i$*,* $p_{i+1}$*,* $t_{i+1} > t_i$ *where* $t$ *is the recording timestamp of the GPS point. The set of all trajectories is denoted with* $\mathcal{T}$*.*

In this paper, we use GPS trajectories with points that are sampled with 5 second intervals. However, feature generation can be extended to use other sampling rates as well. An example GPS trajectory is shown in Figure 3. The example is shown without a basemap to eliminate any privacy concerns. In this example there are 4213 GPS points with 5 second sampling rate corresponding to 6 hours of trip time and over 100 modality transitions.

Definition 2. ***Segment:*** *An activity segment* $\mathbf{S} \subset \mathbf{T}$ *is a sub-trajectory where all GPS point share the same modality. We will use*

*the terms segment and activity segments interchangeably. The three ellipses in Figure 4 illustrate* 3 *segments that constitute the trajectory.*

A GPS Trajectory **T** is a representation of a continuous phenomena (e.g. a trip) with discrete representation of GPS points. Thus, activity classification is the estimation of the underlying real world activity transitions over the discrete time intervals of GPS points. Due to this discrete sampling, the classifications also happen over these GPS points instead of arbitrary times.

Definition 3. ***Transition Point:*** *A transition point $p_t$ is the first or last point of a segment. If $p_t$ is the first point of a segment then $p_{t-1}$ and $p_t$ differ in modality. If $p_t$ is the last point of a segment then $p_t$ and $p_{t+1}$ differ in modality.*

## 2.2 Problem Statement

The Classification of Frequently-Changing Activities problem is formally defined as follows:

**Given:**

(1) A set of GPS Trajectories $\mathcal{T}$ where each trajectory $\mathbf{T} \in \mathcal{T}$ is a representation of movement with discrete set of GPS points $p \in \mathbf{T}$.
(2) A set of annotated timelines which correspond to each GPS trajectory in the training data.

**Classify:**

Each GPS point in a GPS trajectory as "Driving" or "Walking".

**Objective:** Accurately classify each GPS point in each GPS trajectory.

**Constraints:**

(1) Continuity of activity segments, i.e. not breaking them into multiple partitions.
(2) Handling high frequency of modality changes.
(3) Creating a generic model with a possibility to handle additional inputs, i.e. more classes, additional sensor data, change of sampling rates.

**Example:** Given the GPS trajectory in Figure 3, Figure 5 shows a zoomed in area from the proposed model output. In this specific example, all of the 26 walking segments were successfully detected. The point-wise accuracy is 98%, meaning that 98% of GPS points were classified correctly.

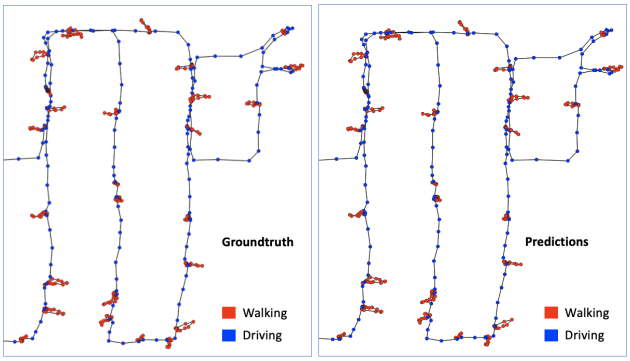
# 3 PROPOSED APPROACH

In this section, first, we will describe the features generated for each GPS point for a GPS trajectory and describe how we prepare our datasets for training and inference. Next, we will explain how the groundtruth data collection was done. Finally, we will describe the overall architecture of our proposed *FACNet* framework in detail.

## 3.1 Feature Generation

Despite their simple form of just longitude, latitude and timestamp, GPS trajectories provide many rich features to identify the activity characteristic.

**Timelag:** In our model, we add the timelag between GPS points as a feature. The sampling rate does not provide much information since we have constant sampling rate of 5 seconds. However, our exploratory analysis showed that the mobile phone operating system sometimes sends the same location repeatedly when the signal is lost or the battery is depleted. To eliminate the effect of such cases, we drop the duplicate records, causing a varying time lag between GPS points when that happens. Therefore, we use this as an input feature to the model.




(a) Groundtruth Activity Classes (b) Proposed Model Output

**Figure 5: Given the input GPS trajectory in Figure 3, a zoomed in subset of the prediction output. To make a visual comparison, left side shows the hand annotated groundtruth and right shows the predictions (best in color).**

**Motion Features:** We derive all motion based features using the distances between each consecutive GPS point. The Haversine distance calculates the distance between two GPS points on a sphere using the latitude and longitudes. We use Haversine distance as a feature, which provides a simple yet scalable enough approximation to the actual distances. The Haversine distance can be calculated as follows:

$$distlag = 2rsin^{-1}\left(\sqrt{sin^2\left(\frac{\Phi_2-\Phi_1}{2}\right) + cos(\Phi_1)cos(\Phi_2)sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right)$$

where $r$ (Earth radius for a perfect sphere) is assumed to be 6371000 meters, $\Phi$ and $\lambda$ are the radians of longitudes and latitudes respectively.

Once these distances are generated, we use the *timelag*s to calculate, pairwise speeds as $speed_i = distlag_i/timelag_i$, acceleration/deceleration as $acc_i = (speed_i - speed_{i-1})/timelag$, jerk with $jerk_i = (acc_i - acc_{i-1})/timelag$.

*Bearing:* Despite the common usage of bearing as-is in the research community, but compass bearing is a cyclic value, i.e. $\theta = 359°$ is close to $\theta = 1°$. Thus, instead of using compass bearings, we represent the bearing with two features. The combination of these two features provide a representation of similarity in directions. The bearing related features can be calculated as follows:

$x = sin(\lambda_2 - \lambda_1) \times cos(\Phi_2)$
$y = cos(\Phi_1) \times sin(\Phi_2) - sin(\Phi_1) \times cos(\Phi_2) \times cos(\lambda_2 - \lambda_1))$
$\theta = (\deg(arctan(y/x)) + 360) \bmod 360$
Hence, $bearing_{cos} = cos(\theta)$, $bearing_{sin} = sin(\theta)$.

We also add an additional bearing related feature, namely bearing change, which is in between $0 - 180$ to indicate the turns. Finally, we developed a *turn* (i.e. a quantized bearing) feature to indicate left and right turns. This feature is generated to distinguish to help the model learn from the waits for left turns from faster right turns that doesn't often require waiting.

**Geometric Features:** Our third type of feature is geometry related. Geometric indicators can help the model distinguish the shape of the trajectory—driving correlates to straight lines while walking correlates to a jittery pattern. We use straightness index defined as cord distance between the first and last points divided by the path distance. We also use Frèchet distance. These features are generated for $N = 3$ and $N = 5$ points surrounding each GPS point. Thus we generate 4 geometric features in total.

Overall, we generate 18 features and associate these with each GPS point.

## 3.2 Ground-truth Data Annotations

We recruited a small group of volunteer delivery drivers that gave us permission to film them during delivery using dashcams. The SD cards were collected at the end of each day, uploaded, faces were blurred, and the videos were sent for manual annotation. Two annotators annotated each video at 0.5 FPS (one frame per two seconds). When two annotators did not arrive at sufficiently similar annotations, a third annotator would act as an arbiter. Our dataset covers multiple drivers and multiple cities.

Even with such a rigorous process, there are still some faulty annotations. We investigated the examples where the annotation did not match our model's prediction, and in some cases we found sufficient evidence to conclude that the annotation was wrong—we removed these cases from the dataset.

Algorithm 1 shows how groundtruth activities are assigned to each GPS point. Since these timelines are in a different sampling rate than the GPS points, we iteratively check each GPS point to determine their corresponding activity. In step 2 in Algorithm 1, for the transition points, we check if a large gap is introduced by flipping an assigned activity. For such cases, we reversed back the activity to eliminate "teleportation" problems in training. The teleportation phenomenon is defined in Definition 7 which is the distance of the last and first transition points between two consecutive driving events.

**Algorithm 1** Algorithm to Assign Groundtruth Annotations to GPS Points

**Input:**
1) A GPS trajectory $\mathbf{T} \in \mathcal{T}$,
2) An annotated timeline $\mathbf{A} = \{(t_{start}, t_{end}, a)\}$ of activities with start and end timestamps, and
3) An ambiguity threshold $\rho$

**Output:**
Trajectory $\mathbf{T} \in \mathcal{T}$ with assigned activities.

**Algorithm:**
**Step 1: Assign Activities to Each GPS Point**
**for each** $(t_{start}, t_{end}, a) \in \mathbf{A}$ **do**
  **for each** GPS Point $p \in \mathbf{T}$ **do**
    **if** $t(p) \geq t_{start}$ and $t(p) < t_{end}$ **then** activity$(p) \leftarrow a$
**Step 2: Correct Ambiguous Annotations**
**for each** $p_i, i = 1, \ldots, N$ **do**
  **if** activity$(p_{i-1})$ = Driving and activity$(p_i)$ = Walking **then**
    last-driving-index = $i - 1$
  **if** activity$(p_{i-1})$ = Walking and activity$(p_i)$ = Driving **then**
    next-driving-index = $i$
    teleport-distance = $dist(p_{\text{last-driving-index}}, p_{\text{next-driving-index}})$
    **if** teleport-distance $\geq \rho$ **then**
      $p_j$ = Driving, $\forall$ last-driving-index $< j <$ next-driving-index

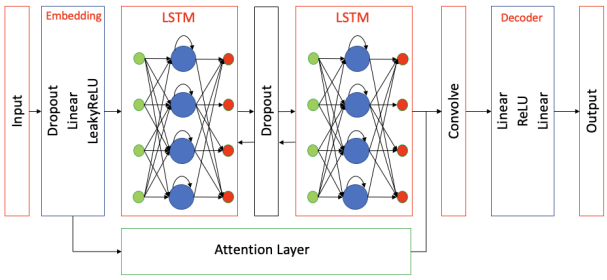



**Figure 6: FACNet Architecture.**

## 3.3 Model Architecture

To capture the intrinsic spatial and temporal relationships between the GPS points, we developed a dual-layer bi-directional LSTM model. Our proposed architecture takes a pre-defined window of GPS points, i.e. sub-trajectory (6 minutes, 72 GPS points), with its 18 features which are min-max scaled to reduce the effect of outliers. These windowed features are fed to an embedding layer, which converts the continuous variables into a discrete valued vector. The output of the embedding layer is used as an input to a bi-directional LSTM module as well as an attention module which is used as a sliding window and convolved with the output of the LSTM. Finally, the convolved output from the previous layer is used in a decoder layer to create the outputs. Figure 6 shows an illustration of the *FACNet* architecture.

**Intuition:** Recurrent Neural Networks are known to be successful for learning tasks which use sequential datasets as input. However, these also suffer from vanishing/exploding gradient problems. To eliminate these, they are supported by memory gates for input and forget. These allow better control over the long-range dependencies. This type of RNNs are known as Long Short Term Memory (LSTM) networks [20]. By stacking two layers of LSTM modules we provide better extraction of more intrinsic relationships and using a bi-directional LSTM decision allows the model see the past and future at the same time to predict the classification of the current point, which is feasible since we use the network for offline classification of already collected trajectories. In addition, the attention module [14, 18] provides better performance by capturing the importance of different embedded features within the time window. Finally, the decoder module generates the outputs of the classification with its two classes.

## 3.4 Model Training

In the model training, we used the hand-annotated ground-truth data that we collected as described in 3.2. We used more than 800 hours of annotated trajectory data. Each trajectory is split to 72 GPS point sub-trajectories corresponding to 6 minutes.

The first few points of a sub-trajectory do not have past context for the LSTM to utilize, which could make the LSTM predictions less accurate at the start (similarly at the end) of the sub-trajectory. To mitigate this, we stagger the sub-trajectories, so they overlap. The first and last 2 minutes are used for context while the middle 2 minutes are used for scoring the model. Each point of the original trajectory will appear in the middle section of *exactly one* sub-trajectory, so no points from the original trajectory are discarded.

Since our dataset was collected in North America, to help it generalize to countries with left hand traffic, we randomly flip the longitude ($Longitude \rightarrow -Longitude$) of 50% of the sub-trajectories.

While choosing the parameters for the model architecture, we used the attention window length as 4 and embedding size as 12. We used a dual-layer bi-directional LSTM architecture with 36 LSTM units. We did multiple experiments with these hyper-parameters and the ones reported are the best performing.

At the training stage, we used Learning Rate as 0.005, optimizer as *AdamW* [9]. In addition, we also used cosine annealing learning rate scheduling [8] to improve the performance of training.

Finally, the model was trained for 100 epochs and the training data was split by $0.75 - 0.25$ training/validation set.

## 4 EXPERIMENTAL EVALUATION

The objective of the experiments was twofold: To evaluate the performance of *FACNet* with different settings as well as against different metrics, and to compare its performance with a baseline approach, i.e. a heuristic based approach. We picked a related work [10] for comparison and evaluate its performance as well. To achieve these goals, the following questions were asked (1) What is the effect of different modules/LSTM cells over the model performance? (2) What is the effect of the urban canyons versus residential neighborhoods on the accuracy? While answering these questions, we used multiple metrics, i.e. GPS point based accuracy, timeline accuracy, gap/teleportation metric as well as point-to-line distance metric. Next, we will describe the evaluation metrics in more detail followed by the experimental results.

### 4.1 Metrics

The activity classification from GPS points problem can be evaluated from multiple metrics.

Definition 4. ***Point-based Accuracy:*** *This is the metric used in model training. It compares the ground-truth with the predicted class of each GPS point and computes the proportion correctly classified. Thus, it becomes*

$$Accuracy_{point} = \frac{|p^{true}_{Walking}|+|p^{true}_{Driving}|}{|\mathbf{T}|}$$

In Figure 4, the GPS trajectory has 20 GPS points with 10 driving and 10 walking points. Accuracy would be calculated for these GPS points regardless of their class. A single point error will have the same impact on point-based accuracy, regardless of whether the error is located near an activity transition, or in the middle of an activity segment, but these two cases might have different impact on downstream applications that use classified GPS traces. For this reason, we also devised a second metric, called segment accuracy, defined below.

Definition 5. ***Segment (Same Modality Group) Accuracy:*** *Segment accuracy represents how many of the segments as defined in Def. 2, i.e. the groups of GPS points with the same modality, are correctly detected by the model. This specific metric makes more sense in terms of high frequency of transitions between different modalities. Due to minor differences, this metric may be misleading if used with absolute differences between sets. To eliminate such issues, we add an additional parameter, i.e. $\tau$, that represent a tolerance (in terms of seconds) for the overlapping number of GPS points. Thus, the segment accuracy can be defined as*

$$Accuracy_{seg} = \frac{|seg^{true}_{OnFoot}|+|seg^{true}_{Drive}|}{|seg|}$$

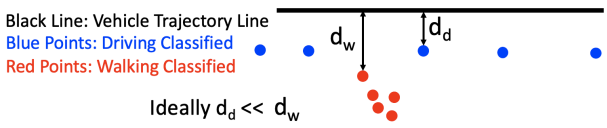



**Figure 7: Illustration of how distance to vehicle line metric is calculated.**

In Figure 4, the GPS trajectory has 3 segments, i.e. 2 driving segment and 1 walking segment. A perfect segment accuracy would capture all 3 despite slight shifts which would be in a reasonable error margin, i.e. $\tau$.

Definition 6. ***Temporal Accuracy:*** *Temporal accuracy aims to understand the over/under estimation of timelines of activities. It considers the fraction of time that was correctly predicted compared to the entire duration of a trajectory. Thus, the temporal accuracy can be defined as* $Accuracy_{temp} = \frac{duration^{true}_{OnFoot}+duration^{true}_{Drive}}{duration_{\mathbf{T}}}$

**Large Scale Analysis Metrics:** Despite the efforts to collect ground-truth datasets in scale, it is hard to continuously collect such data since it is labor-intensive and requires high effort. Therefore, we developed two complementary metrics to mimic the performance for a real world scenario. These metrics are not bounded ($0 \leq metric \leq \infty$). Yet, they provide a sense of performance when used together with the existing techniques especially in comparative analysis.

Definition 7. ***Teleportation Distance:*** *Teleportation metric, e.g. gap distance, is the distance between the last transition point of a driving event to the first transition point of the next driving event. Logically, since the vehicles cannot drive by themselves, the end of a driving event should also be the start of the next driving event. We use this insight to calculate the average/minimum/maximum teleportation distances for each GPS trajectory. This metric cannot be used alone. Suppose all GPS points are classified as driving, teleportation distance will be* 0*. Therefore, we complement this metric with an additional metric, i.e. distance to vehicle line.*

Figure 4 shows an example teleportation event in green. The gap between the last and first point of the two driving events, represents the aforementioned metric. It is worth noting that due to the sampling rate, e.g. 5 seconds, as well as the imperfection of the GPS sensors, we do not expect the teleportation distances to be 0m as in an ideal scenario. However, empirically we saw that, especially for comparative analysis purposes, the teleportation metric is a good indicator of the model performance.

Definition 8. ***Distance to Vehicle Line:*** *Assuming an on-board GPS sensor is available on a vehicle and the GPS data to be classified is collected from a smartphone, we expect these two data to move together for driving events, and diverge for walking events. Using this idea, we developed another metric, i.e. distance to vehicle line, which calculates the perpendicular distance of each GPS point to the line that is generated by the trajectory collected from the vehicle. Next, we*

| LSTM Units | LSTM Layers | Accuracy | Loss |
|---|---|---|---|
| 12 | 3 | 0.93 | 0.15 |
| 12 | 2 | 0.91 | 0.18 |
| 18 | 3 | 0.94 | 0.165 |
| 18 | 2 | 0.94 | 0.155 |
| 24 | 3 | 0.95 | 0.12 |
| 24 | 2 | 0.95 | 0.127 |
| 28 | 3 | 0.95 | 0.115 |
| 28 | 2 | 0.95 | 0.105 |
| 32 | 3 | 0.95 | 0.105 |
| **32** | **2** | **0.96** | **0.099** |
| 64 | 3 | 0.94 | 0.116 |
| 64 | 2 | 0.95 | 0.114 |

**Table 1: Experiments with the varying LSTM cells/layers.**

*group these to collect the highest distance. For a good classifier, we expect this number to be low for driving classified points and larger for walking classified points.*

An illustration of the proposed distance to vehicle line metric is shown in Figure 7. Assuming the black line is a Vehicle GPS trajectory that was converted to a line, the driving GPS points should be closer to the line and walking GPS points should be farther away. It is important to note that due to the noise associated with the GPS data, we expect to have occasional divergences. However, if multiple classification approaches are compared side by side, we expect to have a better distinction using this metric.

The metrics introduced in Definition 7 and Definition 8 allow us to compare models over large scale datasets without ground truth annotations.

## 4.2 Experimental Dataset

For the experiments, we chose 50 GPS trajectories with more than $> 100$ transitions each from driving to walking and driving to walking. The trajectories were selected to cover areas with different characteristics, such as downtown, rural, residential, etc. All trajectories are $\geq 5$ hours in length and all were sampled with 5 second intervals.

## 4.3 Self Evaluation

*4.3.1 Effect of number of LSTM cells on the model performance:* In this experiment, we increased/decreased the number of layers/cells for LSTM architecture to find the best parameters in an empirical way. Table 1 shows that the model performs best when the number of LSTM cells are 32 and there are 2 layers. The accuracy number presented here indicates point pairwise accuracy which is at 96% with the proposed parameters.

*4.3.2 Effect of Attention Module:* To understand the performance with the usage of the Attention module, we trained the model with and without the attention module and compared the results. Table 2 shows that the performance improves by the usage of the attention module. Even though the change is not drastic, improved performance may mean less confusion especially for the classification of transition points.

*4.3.3 Distribution of Accuracies:* To understand the performance of the model, we calculated the distribution of the point-wise, segment

| Attention Module | LSTM Units | LSTM Layers | Accuracy | Loss |
|---|---|---|---|---|
| **With** | **32** | **2** | **0.96** | **0.099** |
| Without | 32 | 2 | 0.95 | 0.107 |

**Table 2: Experiments with/out Attention Layer.**

| Accuracy | Number of trajectories | | |
|---|---|---|---|
| | point-wise acc. | segment acc.($\tau = 10$) | temporal acc. |
| 1 | - | 8 | - |
| 0.99 | - | 18 | - |
| 0.98 | 5 | 9 | 4 |
| 0.97 | 13 | 9 | 12 |
| 0.96 | 10 | 2 | 9 |
| 0.95 | 7 | 2 | 7 |
| 0.94 | 10 | 2 | 11 |
| 0.93 | 3 | - | 5 |
| 0.92 | 2 | - | 2 |

**Table 3: Point-based Accuracies for the test trajectories.**

| Method | Min Acc. | Max. Acc. | Mean Acc. | Median Acc. |
|---|---|---|---|---|
| **FACNet** | **0.92** | **0.98** | **0.96** | **0.96** |
| Related Work [10] | 0.82 | 0.94 | 0.88 | 0.88 |
| Heuristic (3mph) | 0.69 | 0.92 | 0.84 | 0.84 |
| Heuristic (5mph) | 0.81 | 0.94 | 0.88 | 0.88 |
| Heuristic (7mph) | 0.77 | 0.91 | 0.85 | 0.86 |

**Table 4: Point-wise Accuracy comparison.**

($\tau = 5s$) and temporal accuracies over the 50 GPS trajectories. Table 3 shows the results. From all these metrics perspective, the model performed with 96% median accuracy. This shows that the model consistently performed well and stably across different types of geographies. As a side note, the total number of stops across all these 50 trajectories is 5726 and the total duration is 380 hours.

## 4.4 Comparative Evaluation over Accuracy

We conducted a comparative analysis with a heuristic as well as a state-of-the-art [10] model. The related work in [10] uses a stacked Convolutional LSTM architecture to discover intrinsic relationships in the data. It uses 3 sets of inputs, i.e. 4 features (velocity, acceleration, bearing change, jerk) generated from the GPS data, deep features created using these, and finally optional weather related information. Since we did not have the weather related data available for our training dataset, we omitted it from model training and evaluation. Authors in the [10] claim this specific input increases the performance by $1-3\%$. In addition, since the model code was not openly available, we replicated the architecture (Section 3 and Figure 2 in [10]). Finally, to have a fair comparison, we used the same training and test datasets that are used in our proposed approach for the training of the related work [10].

For heuristic approach, we used a relatively simple approach which calculates point pairwise speeds, followed by 3, 5, 7 mph thresholds to classify GPS points.

Table 4 shows that the proposed approach performs much better than the other approaches in terms of the minimum, maximum, mean and median point-wise accuracies. Despite the highly complex architecture of the related work, the performance was not

| Method | Min Acc. | Max. Acc. | Mean Acc. | Median Acc. |
|---|---|---|---|---|
| **FACNet** | **0.92** | **0.98** | **0.96** | **0.96** |
| Related Work [10] | 0.81 | 0.94 | 0.88 | 0.88 |
| Heuristic (3mph) | 0.68 | 0.92 | 0.83 | 0.84 |
| Heuristic (5mph) | 0.81 | 0.93 | 0.87 | 0.88 |
| Heuristic (7mph) | 0.78 | 0.91 | 0.86 | 0.86 |

**Table 5: Temporal Accuracy comparison.**

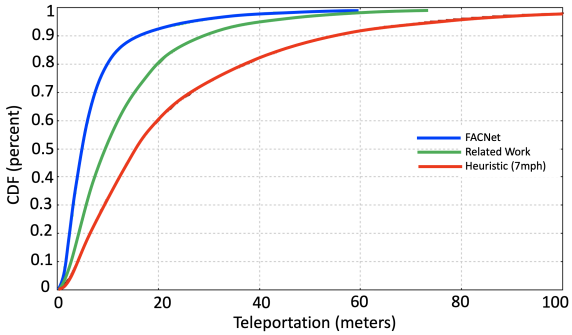



**Figure 8: Experiments with the teleportation metric for a large scale set of GPS trajectories. The related work is [10] with the additional improvements described in [5].**

significantly better than the heuristic approaches. This may be due to the fact that the features generated for the related work were similar and movement oriented features, i.e. speed, acceleration, jerk. Therefore, in some cases, these failed to capture the context of the movement.

In our final comparative analysis, we evaluated the temporal accuracies for each model. In Table 5, it can be seen that the temporal accuracy performance is very similar to the point-wise accuracies in Table 4. This is expected since the sampling rate of the GPS points cause these numbers to be very similar apart from minor perturbations. However, overall both tables show that the proposed approach significantly outperforms all other methods.

Segment accuracy (for walking) comparisons are omitted since with a sufficiently low speed limit, the heuristic approach can detect all walking patterns. Without the use of the other methods, the segment accuracy may not be a good indicator for comparative analysis.

## 4.5 Comparative Evaluation over Teleportation Distance

While using groundtruth data is highly accurate to understand the model performance, it is hard to scale. Therefore, for a large scale comparison, we decided to evaluate the model performance using the newly proposed teleportation distance metric. For this experiment, we used 5000 GPS trajectories. Each of these is collected for an entire day ($> 6$ *hours*) from smartphone GPS sensors. Figure 8 shows the cumulative distribution function (CDF) of the gap distances. The plot indicates that $> 90\%$ of the walking classifications have $< 20m$ teleportation gap. Our proposed approach significantly outperforms both the related work and the heuristics approach on the teleportation gap metric.

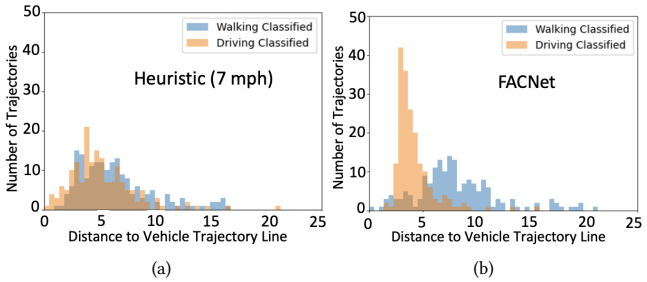



**Figure 9: Comparison of the distributions of the distances to the vehicle lines. The left side shows the distribution with an heuristic approach whereas the right side shows the metric from the FACNet approach.**

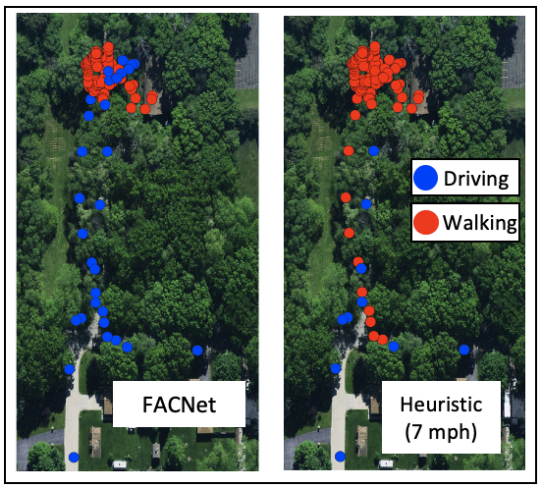



**Figure 10: Model performance in a cul-de-sac when a vehicle moves slowly.**

## 4.6 Comparative Evaluation over Distance to Vehicle Line

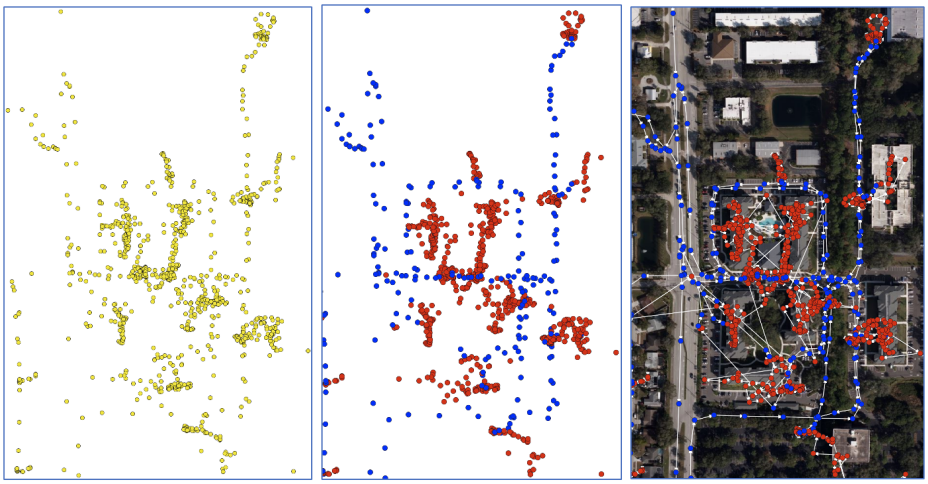
For the next experiment, we collected 386 GPS trajectories from 65 vehicles, and coupled these with the GPS trajectories collected from smartphone sensors, to obtain trajectory (vehicle and smartphone) pairs spanning an entire day. Table 6 shows the results. Ideally, when a smartphone GPS trajectory is classified, we expect a difference of average vehicle line distances between walking and driving classified points. However, it should be noted that when a vehicle stops, the person who starts walking will be closer to the vehicle GPS trajectory line before walking away. Therefore, instead of considering the distances in meters, comparison with relative average distances of the classes makes more sense. In this example, we see that the differences are substantial, i.e. 2.3× vs. 1.3×.

We also calculated these using a per trajectory pair (vehicle and smartphone) basis. Figure 9 shows how these distances are distributed. It can be seen from the comparative plots that the FACNet provides a better distinction than a heuristic approach.

| Method | Location | # of Drivers | # of Trajectories | Walking class Avg. Distance to Vehicle Line | Drive class Avg. Distance to Vehicle Line | Difference |
|---|---|---|---|---|---|---|
| FACNet | Seattle, WA | 30 | 191 | 9.25 m | 4 m | 2.3x |
| Heuristic (7mph) | | | | 5.97 m | 4.69 m | 1.3x |
| FACNet | Twin Cities, MN | 35 | 195 | 7.3 m | 3.2 m | 2.3x |
| Heuristic (7mph) | | | | 5.3 m | 3.7 m | 1.4x |

**Table 6: Comparison between the distances to the vehicle lines.**


**Figure 11: Model performance in a residential neighborhood when there are houses side-by-side.**



(a) (b) (c)

**Figure 12: From left to right, input GPS points, the output from FACNet and finally the output with additional contextual information, i.e. connected GPS points that are overlaid on satellite image.**

## 4.7 Qualitative Examples

Finally, we also visually inspected the outputs to understand the model performance with context (i.e. underlying roads, buildings, etc.). Figure 11 shows a residential neighborhood. In this example, the Driving classified GPS points are represented with Blue and Non-Driving are depicted with Red. GPS points are connected with white lines to indicate their temporal order. Overall, the example shows that the model perfectly captured the 6 stops in a residential neighborhood without merging any together. With a heuristic approach, these would either split into more stops (with lower threshold) or merged together (higher threshold) causing a loss of the contextual information about these stops.

Figure 10 shows another example in a cul-de-sac. As can be seen, FACNet output on the left can capture the slow-down on the short road segment after turn whereas the heuristic method classifies such slow-downs as walking.

We also want to emphasize the importance of any additional data that can be used in the future to improve the model. Figure 12 shows this over an example. Figure 12(a), the input GPS points are depicted with yellow. It is evident that without contextual information (e.g. road network, building outlines, etc.), the output is hard to interpret. Figure 12(b) shows the output from FACNet for these data. Despite the linearity of Driving (blue) GPS points that may overlay with the roads, without additional context, it is still hard to extract any valuable insights. Finally, Figure 12(c) shows the output with a

satellite image as a basemap as well as connects the GPS points with white lines. Using this contextual information, it can be seen that the model performed pretty well even under heavy noise. Also, the Figure shows how, in the future, additional contextual information can be used to further refine the proposed approach.

## 5 CONCLUSION AND FUTURE WORK

Classifying human activities from GPS trajectories is needed for use cases where additional sensor input may not be available. GPS data, collected from smartphone sensors, includes noise which is exacerbated at lower sampling rates, making the classification problem a difficult one and an active research topic. However, existing approaches either use additional sensor inputs besides GPS locations, make restrictive assumptions or cannot handle the situations involving frequent transitions between different activity classes. In this paper, we tackled the case where transitions between modalities are frequent. We proposed a deep neural network approach, *FACNet*, which predicts frequently switching trip modalities with high accuracy. Our proposed approach uses only Longitude/Latitude/timestamp data, but is flexible enough to be extended to include additional inputs. Our experiments show that the proposed approach outperforms the existing state-of-the-art as well as a heuristic approach. To evaluate performance on large-scale data, we introduce two new metrics, Teleportation Distance and Distance to Vehicle Line, which do not require hand-annotated ground truth.

In the future, we plan to extend *FACNet* from two different perspectives. First, we plan to enrich the input features with additional data. Such data will not be limited to additional smartphone data, which has also been used in past research, but include additional geospatial context such as the building outlines, underlying road network, etc. Second, we plan to work on inferring finer sub-classes of driving and walking. For example, walking can be further classified as being near the vehicle (e.g. in unloading or gas refilling) or away from it, and driving can be further classified as transit, looking for parking, stopped at signal, etc.

## REFERENCES


[1] Sina Dabiri and Kevin Heaslip. 2018. Inferring transportation modes from GPS trajectories using a convolutional neural network. *Transportation research part C: emerging technologies* 86 (2018), 360–371.

[2] Sina Dabiri, Chang-Tien Lu, Kevin Heaslip, and Chandan K Reddy. 2019. Semi-supervised deep learning approach for transportation mode identification using GPS trajectory data. *IEEE Transactions on Knowledge and Data Engineering* 32, 5 (2019), 1010–1023.

[3] Emre Eftelioglu, Gil Wolff, et al. 2022. GPS Activity Classification Anonymized GPS Trajectories. (2022). https://github.com/amazon-research/goal-gps-ordered-activity-labels

[4] Yuki Endo, Hiroyuki Toda, Kyosuke Nishida, and Akihisa Kawanobe. 2016. Deep feature extraction from trajectories for transportation mode estimation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 54–66.

[5] Xiang Jiang, Erico N de Souza, Ahmad Pesaranghader, Baifan Hu, Daniel L Silver, and Stan Matwin. 2017. Trajectorynet: An embedded gps trajectory representation for point-based classification using recurrent neural networks. *arXiv preprint arXiv:1705.02636* (2017).

[6] Carson K Leung, Peter Braun, and Alfredo Cuzzocrea. 2019. AI-based sensor information fusion for supporting deep supervised learning. *Sensors* 19, 6 (2019), 1345.

[7] Hongbin Liu and Ickjai Lee. 2017. End-to-end trajectory transportation mode classification using Bi-LSTM recurrent neural network. In *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*. IEEE, 1–5.

[8] Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* (2016).

[9] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[10] Asif Nawaz, Huang Zhiqiu, Wang Senzhang, Yasir Hussain, Izhar Khan, and Zaheer Khan. 2020. Convolutional LSTM based transportation mode learning from raw GPS trajectories. *IET Intelligent Transport Systems* 14, 6 (2020), 570–577.

[11] Helmut Schmid. 1994. Part-of-speech tagging with neural networks. *arXiv preprint cmp-lg/9410018* (1994).

[12] Li Shen and Peter R Stopher. 2014. Review of GPS travel survey and GPS data-processing methods. *Transport reviews* 34, 3 (2014), 316–334.

[13] Ranjeeth Siddakatte, Ali Broumandan, and Gérard Lachapelle. 2017. Performance evaluation of smartphone GNSS measurements with different antenna configurations. In *Proceedings of the International Navigation Conference*.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[15] Atro Voutilainen. 2003. Part-of-speech tagging. *The Oxford handbook of computational linguistics* (2003), 219–232.

[16] Jin Wang, Qingmei Zhao, and Chen Wang. 2017. DeepTravel: Online transport mode identification based on low-rate sampling sensors through deep neural network. In *2017 4th International Conference on Systems and Informatics (ICSAI)*. IEEE, 1486–1491.

[17] Linlin Wu, Biao Yang, and Peng Jing. 2016. Travel mode detection based on GPS raw data collected by smartphones: a systematic review of the existing methodologies. *Information* 7, 4 (2016), 67.

[18] Zhengxuan Wu, Xiyu Zhang, Tan Zhi-Xuan, Jamil Zaki, and Desmond C. Ong. 2019. Attending to Emotional Narratives. *IEEE Affective Computing and Intelligent Interaction (ACII)*.

[19] Guangnian Xiao, Qin Cheng, and Chunqin Zhang. 2019. Detecting travel modes using rule-based classification system and Gaussian process classifier. *IEEE Access* 7 (2019), 116741–116752.

[20] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. 2019. A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation* 31, 7 (2019), 1235–1270.

[21] Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, and Wei-Ying Ma. 2008. Understanding mobility based on GPS data. In *Proceedings of the 10th international conference on Ubiquitous computing*. 312–321.