Google Smartphone Decimeter Challenge

The problem

an introduction

GPS technologies in smartphones currently offer accuracy within **3-5 meters**. This level of precision is **insufficient** for various critical applications, such as **autonomous driving**, **precise navigation** in urban environments, **augmented reality**, and location-based services that require **decimeter-level accuracy** or better.

By using **advanced optimisation algorithms**, **machine learning**, and data from **multiple sensors**, we can ensure you don't get confused about whether or not to take that fly-over ;)

Our problem statement (and the goal of the competition)

How can we **utilize raw GNSS data** from standard smartphones to **achieve decimeter-level positioning accuracy**, enabling precise location services without the need for specialized equipment or additional sensors?

The problem

so much potential

Applications

The development of a solution capable of providing decimeter-level location accuracy could revolutionize several fields:

- **Autonomous** Vehicles & Drones
- Augmented Reality (AR)
- Outdoor & Indoor **Navigation**
- Asset **Tracking**
- **Geospatial Data Collection**
- **Aerospace** industry
- Agriculture
- Automotive industry
- Construction, Mining and Industrial projects
- **Defence** systems
- Mobile Mapping
- Timing Applications
- **Unmanned Systems**

Impact

- **Improved navigation** & **mapping services** *(i.e. you miss lesser turns)*
- **Better service** in for those **businesses that rely on location data** *(i.e. you don't waste time looking for your Uber)*
- **Advancements** in **overlaying digital information** on the physical world *(i.e. you quite literally get to live in your own reality)*
- Businesses increase operational **efficiency**, **reduce costs** & get new opportunities for services and applications

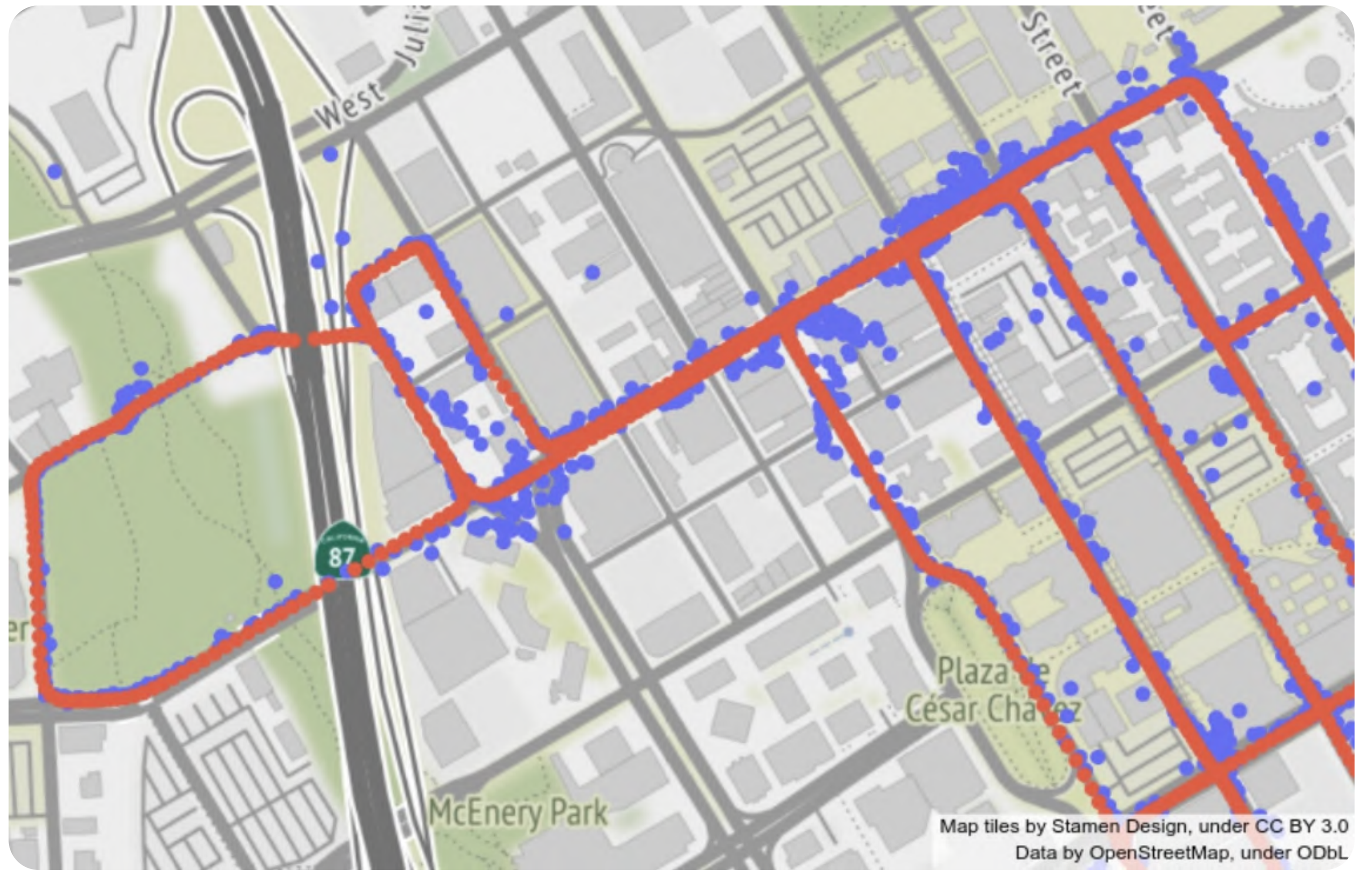
Our data-set

thankyou google!

To **predict** smartphone **positions** during specific trips using **satellite data** with the maximum amount of precision.

We have **raw GNSS measurement** and **raw inertial sensors' readings**, using several dual-frequency and ADR (Accumulated Delta Range) enabled **smartphones in driving scenarios**, collected in the US San Francisco Bay Area.

Each dataset comes with a **ground truth NMEA file** collected by NovAtel SPAN system in a **time synchronized manner**.


West Julian
Street
87
Plaza de César Chavez
McEnery Park
Map tiles by Stamen Design, under CC BY 3.0
Data by OpenStreetMap, under ODbL

Overlay of some values on a physical map

Data included

- Acceleration
- Rotation rates
- Altitude, and sometimes velocity.
- Latitude
- Longitude
- Signal Strength indicators

65

training routes

45 k

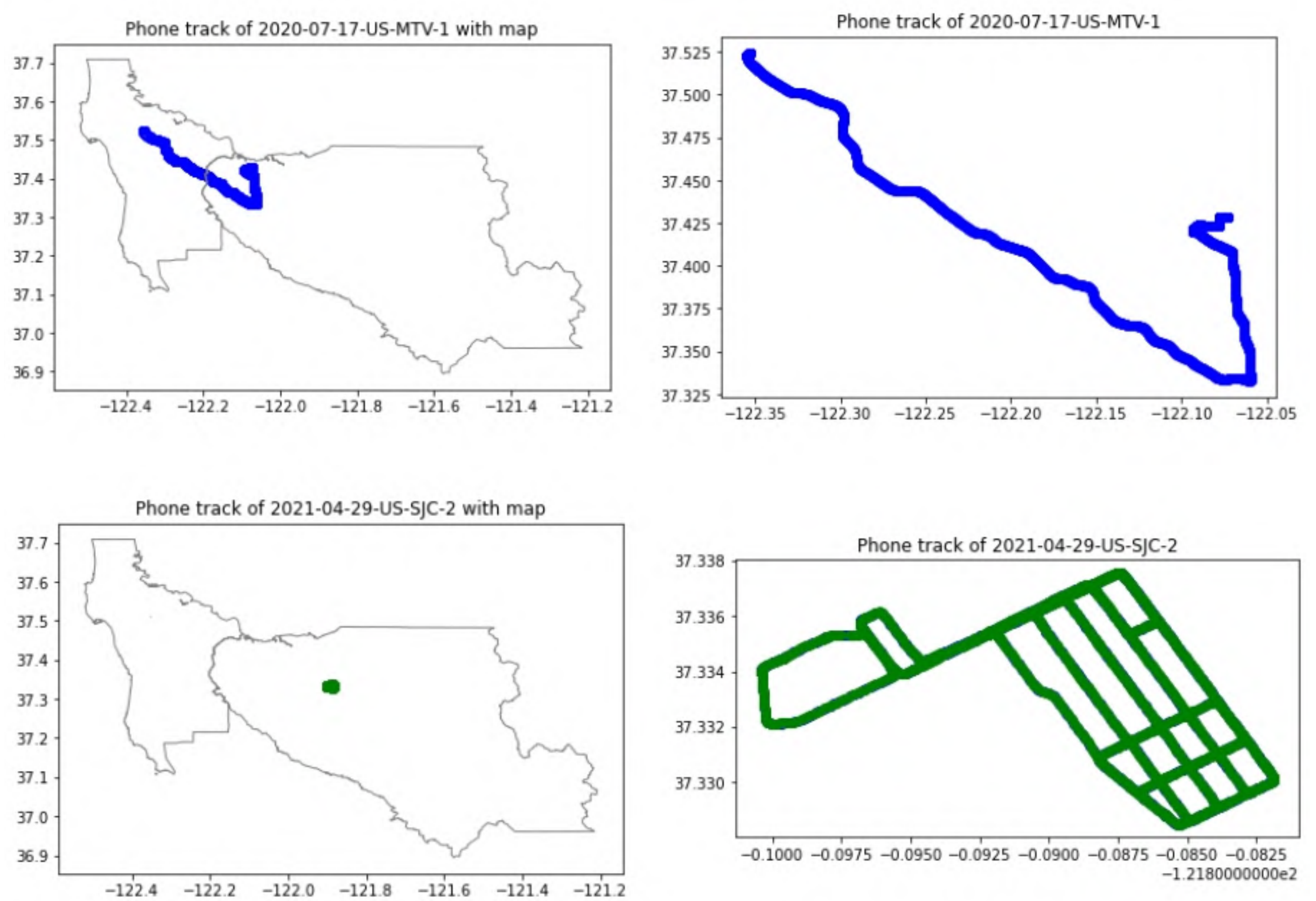
points per route

Our data-set

thankyou google!

The dataset contains over **55+ measurement fields**.
These include:

- **trip_ID**: unique identifier for each date, route and phone
- **utcTimeMillis**: at what timestamp was the message generated (acc. to user)
- **TimeNanos**: time acc. to GNSS receiver's clock
- **Cn0DbHz**: a measure of the strength of a received carrier signal relative to the strength of the received noise
- **SvAzimuthDegrees**: angle between satellite and earth's horizontal plane
- **SvPosition(X/Y/Z)EcefMeters**: X, Y and Z coordinates of the satellite in the Earth-centered Earth-fixed (ECEF) system
- **RawPseudorangeMeters**: the product of the speed of light and the time difference between the signal transmission time and the signal arrival time



Visualisations of 2 runs, with and without a map

Fu, Guoyu (Michael), Khider, Mohammed, van Diggelen, Frank, "Android Raw GNSS Measurement Datasets for Precise Positioning," Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020), September 2020.

Features pre-processing

finding the needle

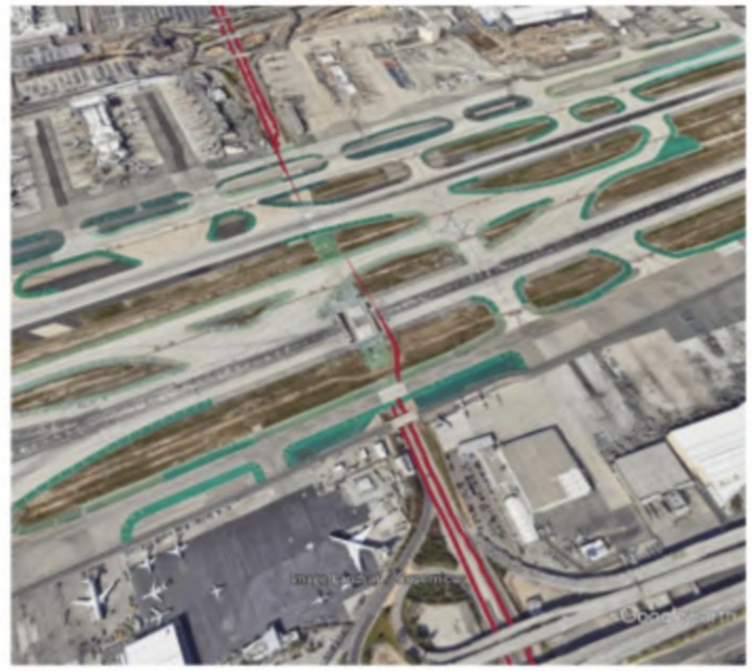
Measurements from GNSS chipsets of mobile phones are often **erroneous** or noisy. Due to this, some values have to be filtered out. Some of the indicatory features we used:

- **Carrier to Noise Ratio** is **above 30**: Values between **25-40 dB** are considered in the bracket of **good communication quality.**
- **Satellite Elevation in Degrees** is **above 5**: It is considered as a standard, an acceptable elevation angle. The higher the angle, the better.
- **Multipath Indicator** is **0**: This implies that the satellite signal has not bounced off of any surfaces (buildings, trees etc.) ensuring that the signal is error free in that respect.

A few more processes involved:

- **Temporal Alignment** based on **Timestamp** and **Trip ID**: Critical for matching the GNSS measurements closely with the actual ground truth locations.
- **Data Standardisation and Sizing**: To feed data into the model

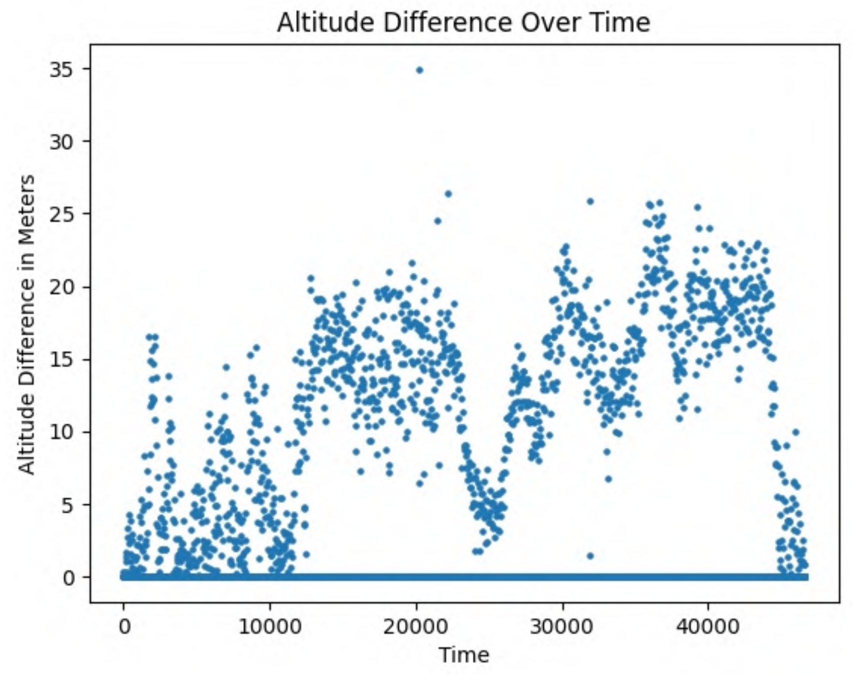
(a) Test/2022-04-01-US-LAX-3/]XiaomiMi8

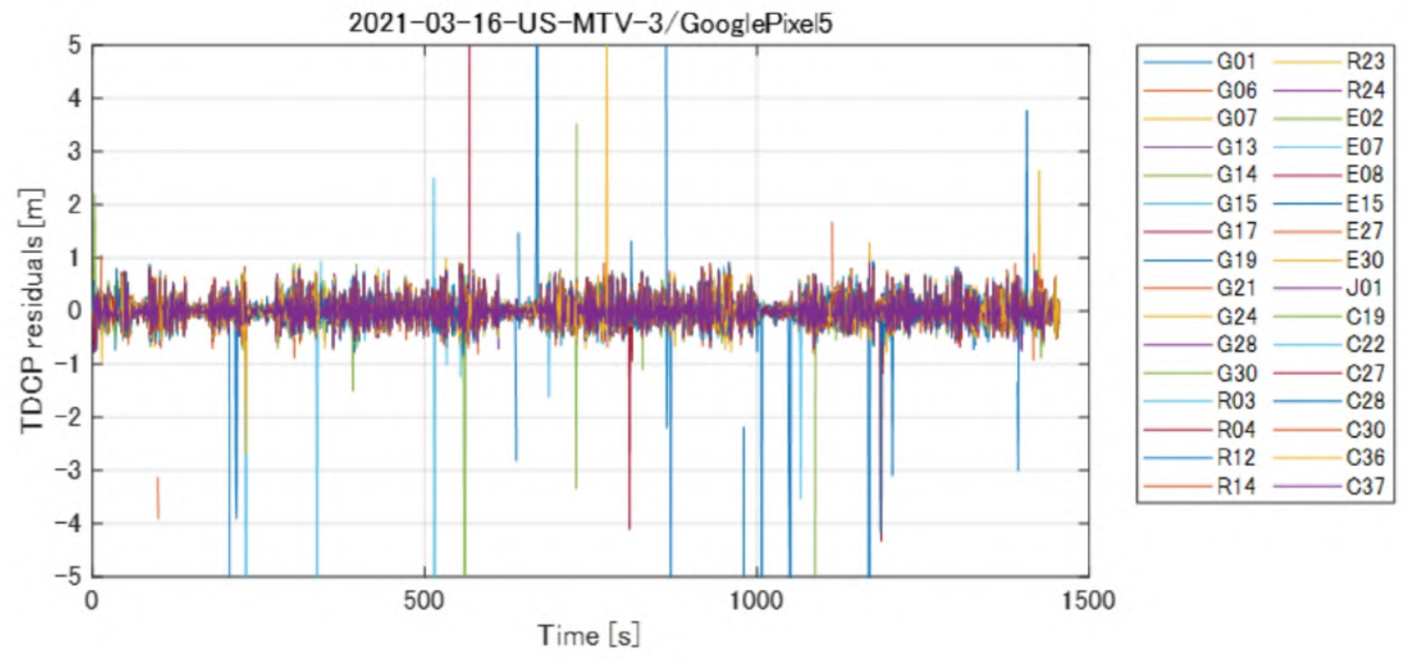

(b) Test/2022-02-23-US-LAX-3/XiaomiMi8

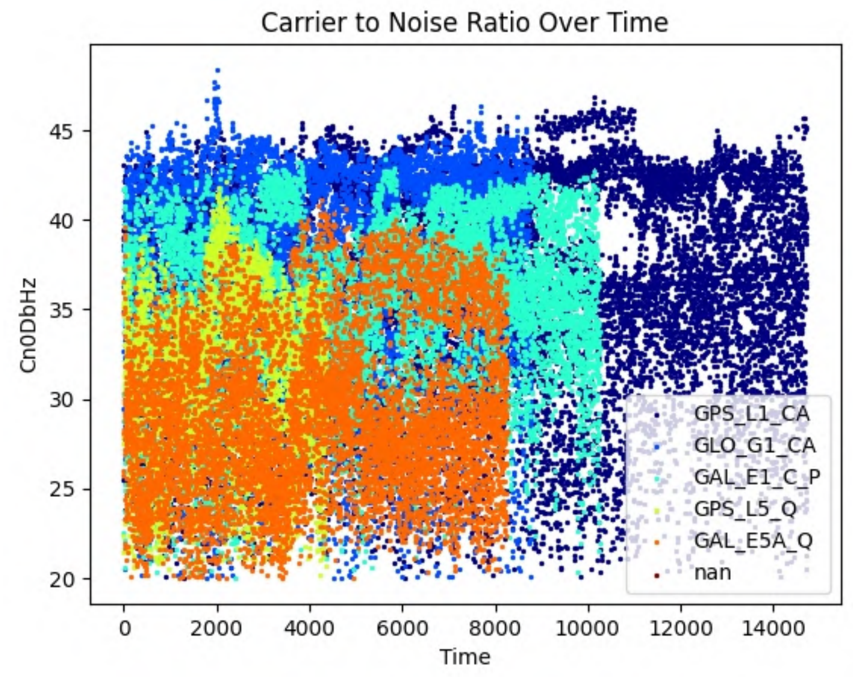
Suzuki T. Precise Position Estimation Using Smartphone Raw GNSS Data Based on Two-Step Optimization. Sensors (Basel). 2023 Jan 20;23(3):1205. doi: 10.3390/s23031205. PMID: 36772245; PMCID: PMC9919037.

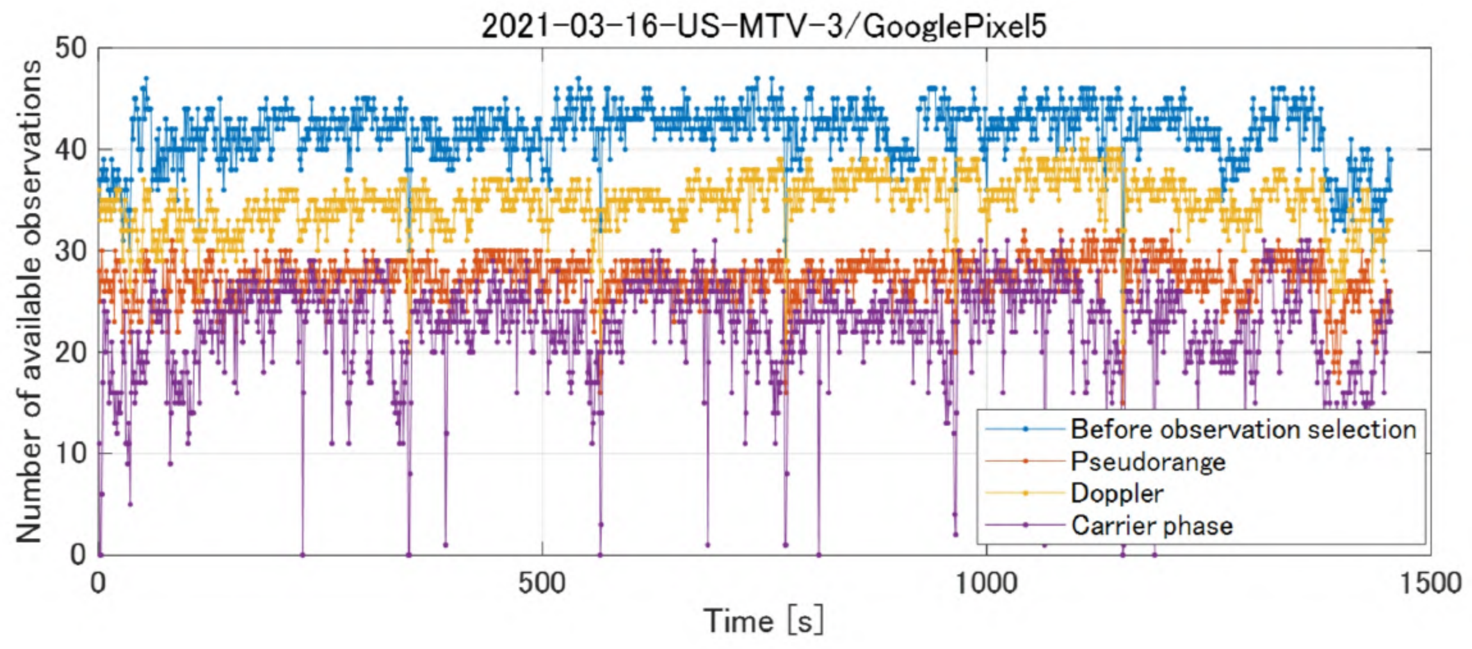
Features pre-processing

finding the needle



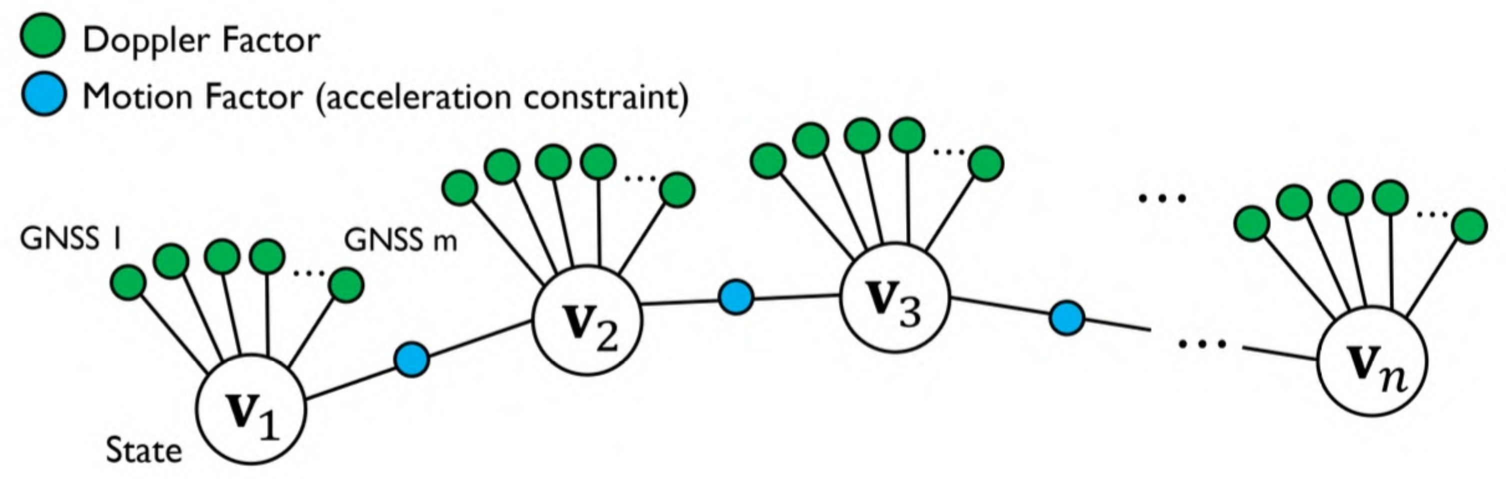






Literature Survey

Two-Step Factor Graph Optimisation



Velocity Estimation

Employs **Doppler shift measurements** to estimate the **receiver's velocity and clock drift**. By modelling the relationship between the observed Doppler shifts and the receiver's velocity, the process effectively captures the dynamics of the receiver's movement. [2]

Adjusts the values of the **velocity states** in a way that best fits the Doppler and motion **constraints** by minimising a cost function.

Suzuki T. Precise Position Estimation Using Smartphone Raw GNSS Data Based on Two-Step Optimization. Sensors (Basel). 2023 Jan 20;23(3):1205. doi: 10.3390/s23031205. PMID: [illegible]245; PMCID: PMC9919037.

Literature Survey

Two-Step Factor Graph Optimisation

Doppler Factor Error Function

$$e_{\mathrm{d},i}^{k} = \mathbf{H}_{\mathrm{v},i}^{k}\mathbf{V}_i - \left(\dot{\rho}_i^k - \mathbf{u}_i^k \mathbf{v}_{\mathrm{s},i}^k\right)$$

Represents the discrepancy between the measured values (from Doppler data) and the values predicted by the model (velocity states and assumed motion).

Pseudo-range rate from the Doppler of the satellite

$$\dot{\rho}_i^k = \mathbf{u}_i^k\left(\mathbf{v}_{\mathrm{s},i}^k - \mathbf{v}_i\right) + \dot{\mathbf{t}}_i$$

Motion Factor Error Function

$$\mathbf{e}_{\mathrm{m},i} = \mathbf{V}_{i+1} - \mathbf{V}_i$$

Operates under the assumption that changes in velocity (acceleration) between consecutive time steps are minimal under normal conditions.

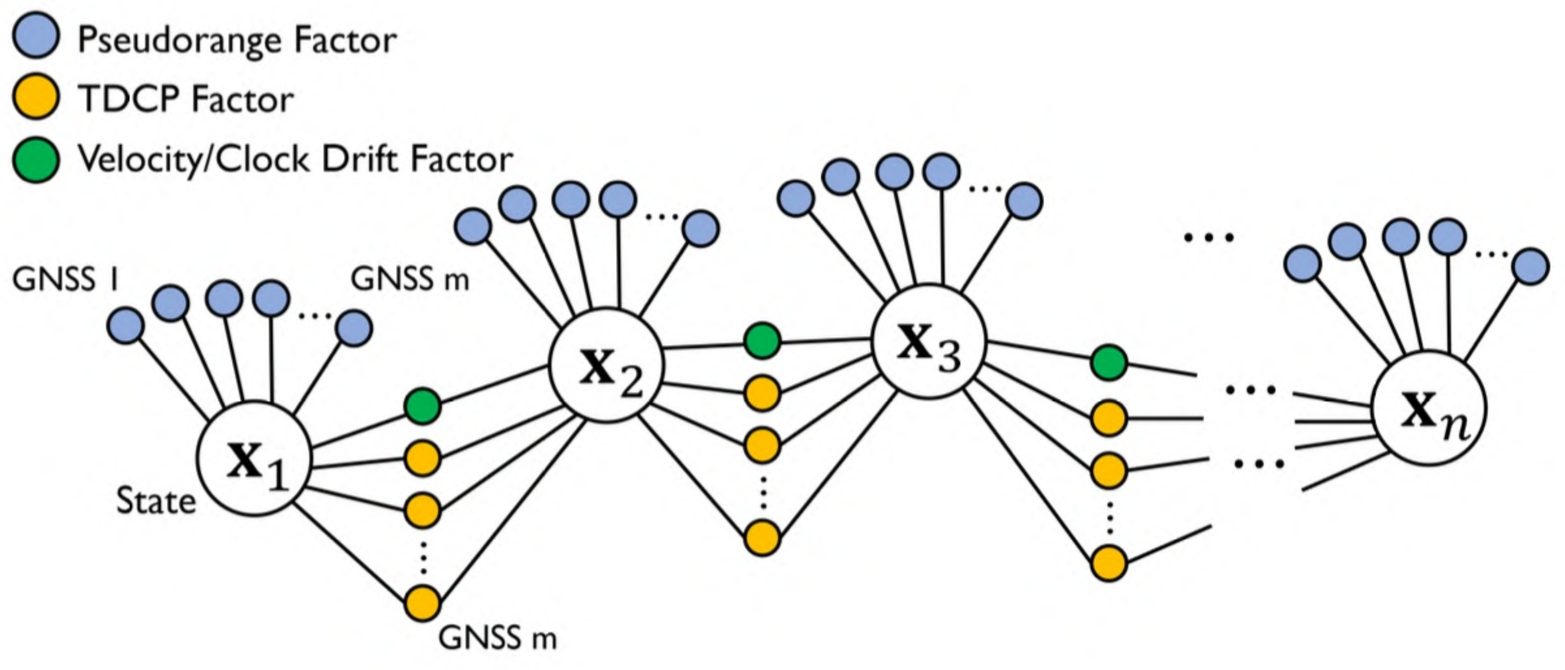
Objective Function to be Optimised

$$\hat{\mathbf{V}} = \underset{\mathbf{V}}{\operatorname{argmin}} \sum_i \|\mathbf{e}_{\mathrm{m},i}\|^2_{\Omega_{\mathrm{m},i}} + \sum_i \sum_k \left\| e_{\mathrm{d},i}^k \right\|^2_{\Omega_{\mathrm{d},i}^k}$$

Suzuki T. Precise Position Estimation Using Smartphone Raw GNSS Data Based on Two-Step Optimization. Sensors (Basel). 2023 Jan 20;23(3):1205. doi: 10.3390/s23031205. PMID: 36772245; PMCID: PMC9919037.

Literature Survey

Two-Step Factor Graph Optimisation



Position Estimation

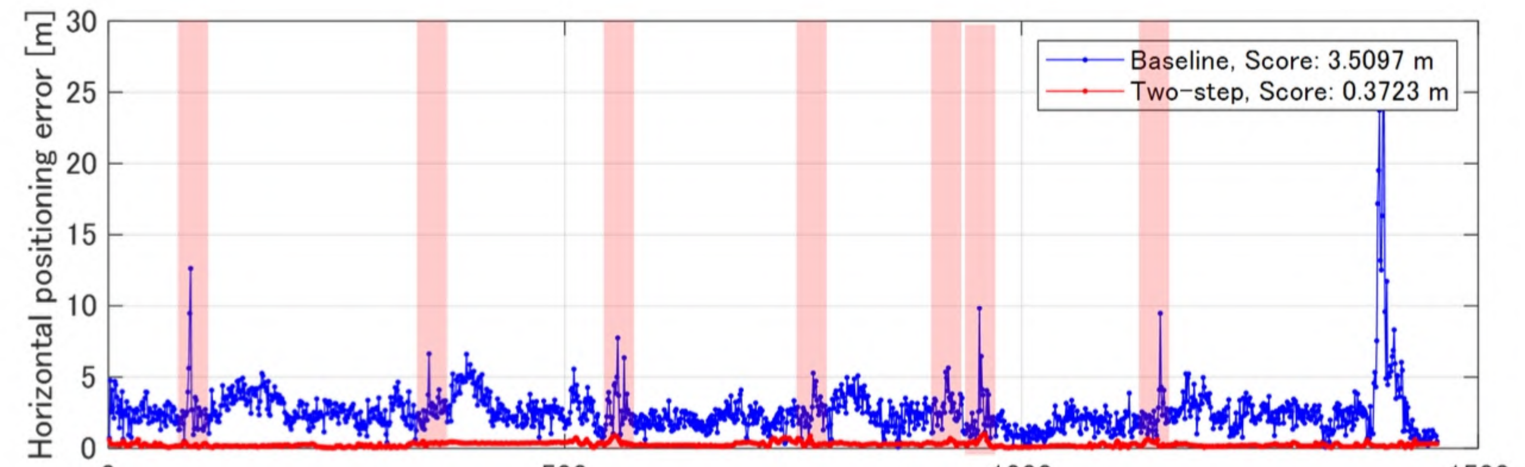
Utilises refined **velocity estimates** alongside corrected **pseudo-range** measurements to enhance the accuracy of the receiver's position estimation. This step **integrates the continuous velocity model with absolute position** measurements to achieve precise GNSS positioning.

Applies constraints from velocity/clock drift factors and pseudo-range corrections within an optimisation framework, effectively bridging the **temporal continuity** of the receiver's motion with spatial accuracy from GNSS observations.

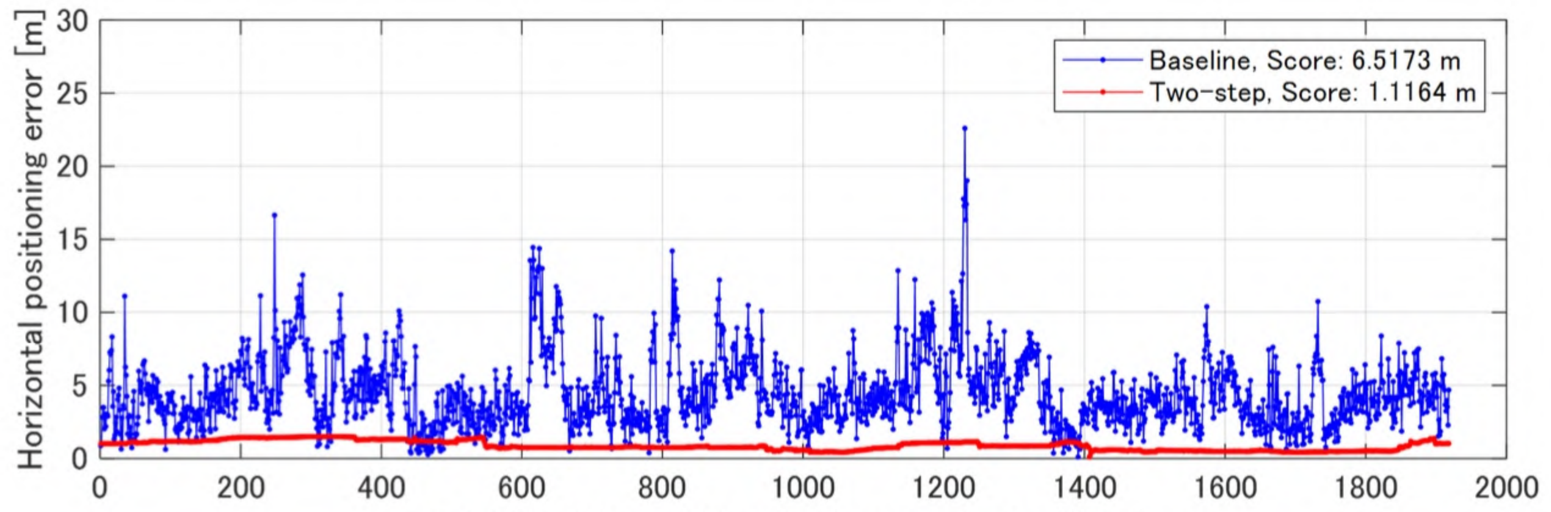
Suzuki T. Precise Position Estimation Using Smartphone Raw GNSS Data Based on Two-Step Optimization. Sensors (Basel). 2023 Jan 20;23(3):1205. doi: 10.3390/s23031205. PMID: 36772245; PMCID: PMC9919037.

Literature Survey

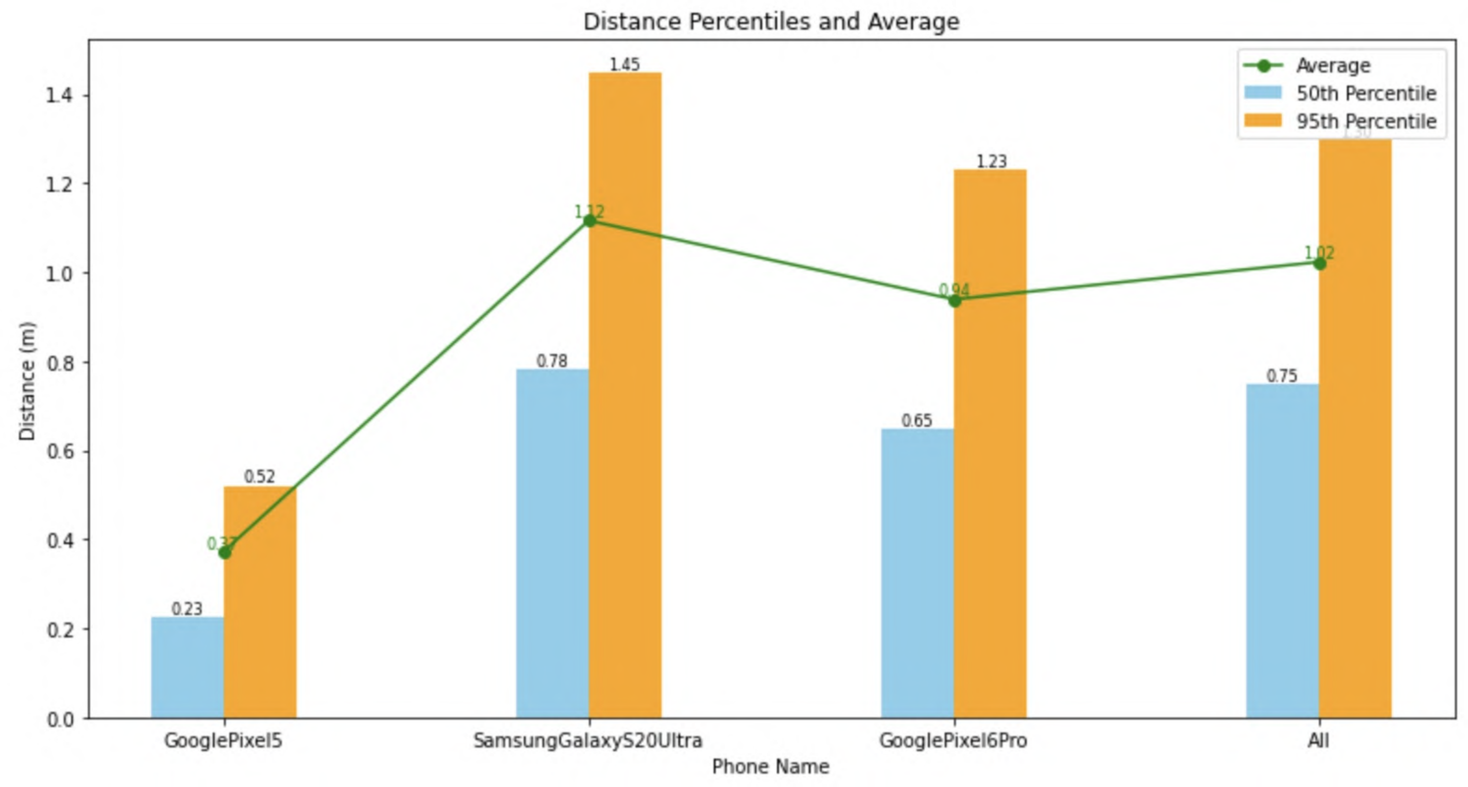
Two-Step Factor Graph Optimisation

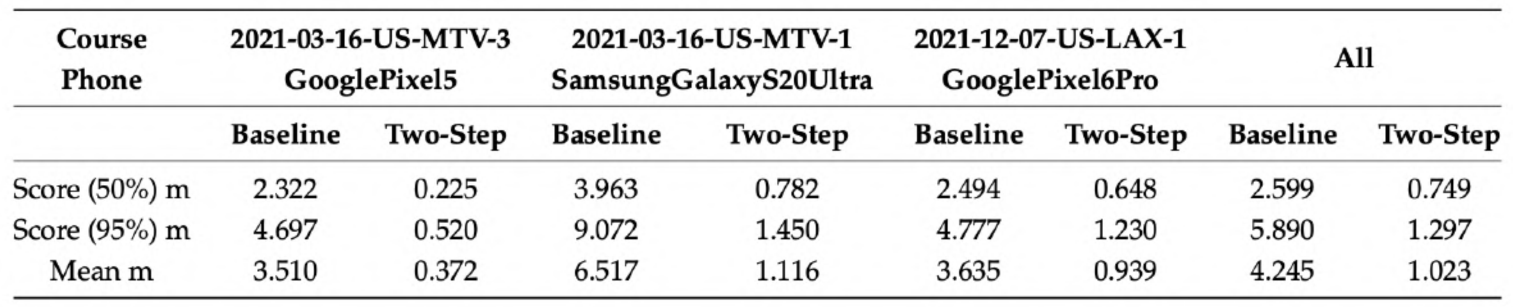


(a) 2021-03-16-US-MTV-3/GooglePixel5



(b) 2021-03-16-US-MTV-1/ SamsungGalaxyS20Ultra





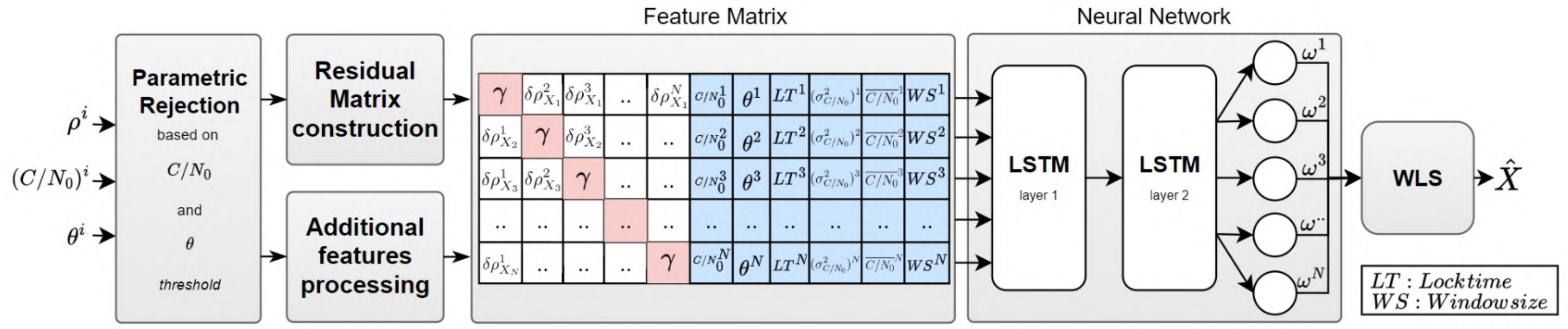
Course Phone	2021-03-16-US-MTV-3 GooglePixel5		2021-03-16-US-MTV-1 SamsungGalaxyS20Ultra		2021-12-07-US-LAX-1 GooglePixel6Pro		All	
	Baseline	Two-Step	Baseline	Two-Step	Baseline	Two-Step	Baseline	Two-Step
Score (50%) m	2.322	0.225	3.963	0.782	2.494	0.648	2.599	0.749
Score (95%) m	4.697	0.520	9.072	1.450	4.777	1.230	5.890	1.297
Mean m	3.510	0.372	6.517	1.116	3.635	0.939	4.245	1.023

Suzuki T. Precise Position Estimation Using Smartphone Raw GNSS Data Based on Two-Step Optimization. Sensors (Basel). 2023 Jan 20;23(3):1205. doi: 10.3390/s23031205. PMID: 36772245; PMCID: PMC9919037.

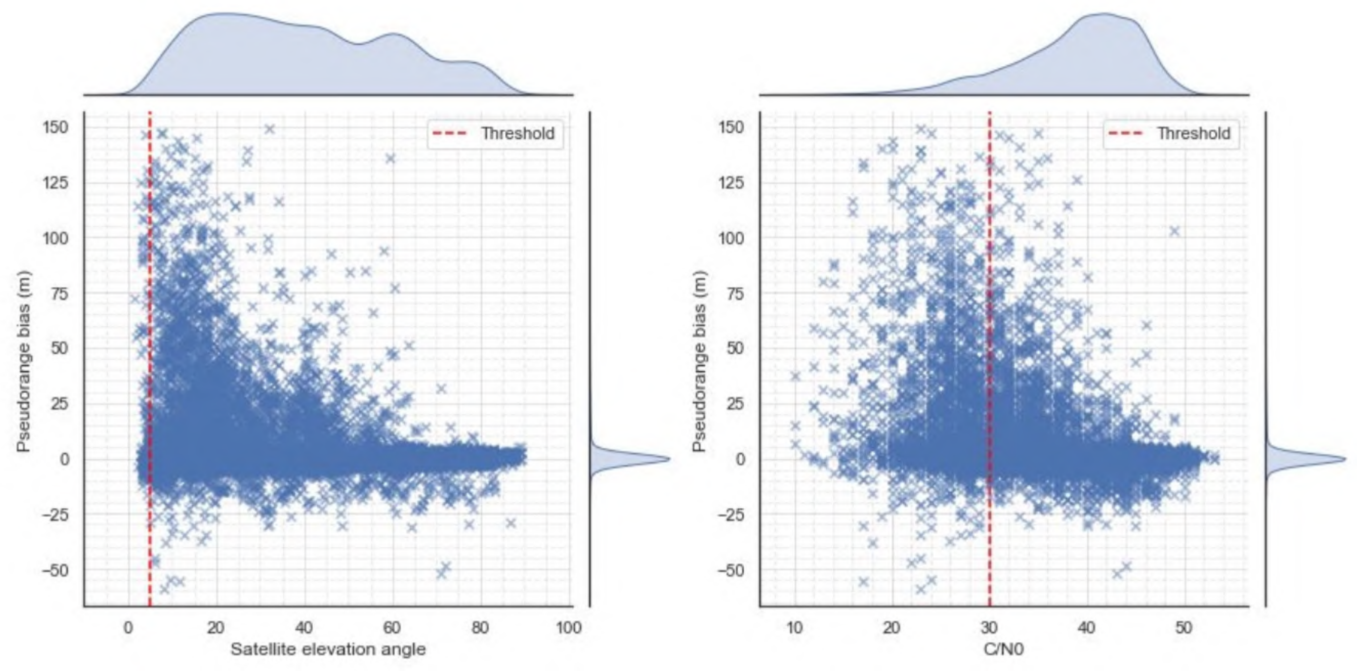
Literature Survey

LSTM-Based GNSS Localization Using Satellite Measurement Features Jointly with Pseudorange Residuals

- **Parametric Rejection** to remove unreliable satellite measurements that could negatively impact localization accuracy.
- **Residual Matrix Construction** excluding one satellite at a time to assess its impact.
- **Additional Features Processing** normalizing additional features to make them suitable for input.
- **Feature Matrix** combines both the other matrices into a comprehensive input matrix
- 2 layer **Neural Network**
- **Weighted Least Squares** algorithm to compute an accurate position estimate from weights.



Proposed Methodology



Parametric Rejection

Sbeity, I., Villien, C., Denis, B., & Belmega, E. V. (2024). LSTM-Based GNSS Localization Using Satellite Measurement Features Jointly with Pseudorange Residuals. *Sensors*, *24*(3), 833. https://doi.org/10.3390/s24030833

Literature Survey

Modelling and prediction of GNSS time series using LSTM and SVM machine learning approaches

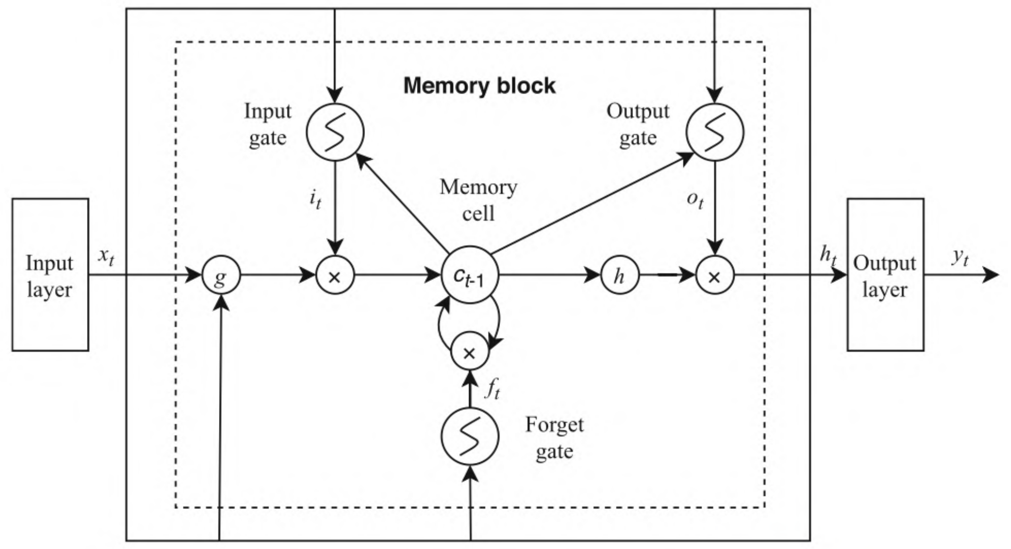


Fig. 1 LSTM neural network architecture (Ma et al. 2015)

$$
\begin{aligned}
i_n &= \sigma\left(W_{xi}x_n + W_{hi}h_{n-1} + W_{ci}c_{n-1} + b_i\right)\\
f_n &= \sigma\left(W_{xf}x_n + W_{hf}h_{n-1} + W_{cf}c_{n-1} + b_f\right)\\
c_n &= f_n c_{n-1} + i_n \tanh\left(W_{xc}x_n + W_{hc}h_{n-1} + b_c\right)\\
o_n &= \sigma\left(W_{xo}x_n + W_{ho}h_{n-1} + W_{co}c_n + b_o\right)\\
h_n &= o_n \tanh\left(c_n\right)
\end{aligned}
$$

$$
J(\omega) = \frac{\|\omega\|^2}{2} + C\sum_{i=1}^{n}\left(\zeta_n + \zeta_n^*\right)
$$

$$
s.t.\ \forall n : \begin{cases} y_n - \varphi(\omega, x_n) - s \le \varepsilon + \zeta_n \\ \varphi(\omega, x_n) + s - y_n \le \varepsilon + \zeta_n^* \\ \xi_n \ge 0 \\ \xi_n^* \ge 0 \end{cases}
$$

Linear Epsilon SVM

Gao, W., Li, Z., Chen, Q., Jiang, W., & Feng, Y. (2022). Modelling and prediction of GNSS time series using GBDT, LSTM and SVM machine learning approaches. *Journal of Geodesy*, 96(10). https://doi.org/10.1007/s00190-022-01662-5

Try Pitch

Methodology

our approach

Our approach employs a **dual-model strategy**, leveraging the strengths of **RTKLIB** for precise GNSS data processing and **Long Short-Term Memory (LSTM)** networks for predictive modeling.

1. **Initial Position Estimation Using LSTM**: The LSTM model is designed to capture temporal dependencies and patterns in the GNSS data, offering initial estimations of latitude and longitude.

2. **Supplementation with RTKLIB**: This serves as a reliable backup/baseline for instances where the LSTM model's output is suboptimal.

3. **Criteria-Based Validation**: After obtaining predictions from the LSTM, we evaluate these positions against specific criteria to assess their accuracy and validity.

4. **Selective Replacement Strategy**: If an LSTM-predicted position fails to meet the predefined criteria - indicating potential inaccuracies or errors - we replace it with the corresponding position calculated from RTKLIB.

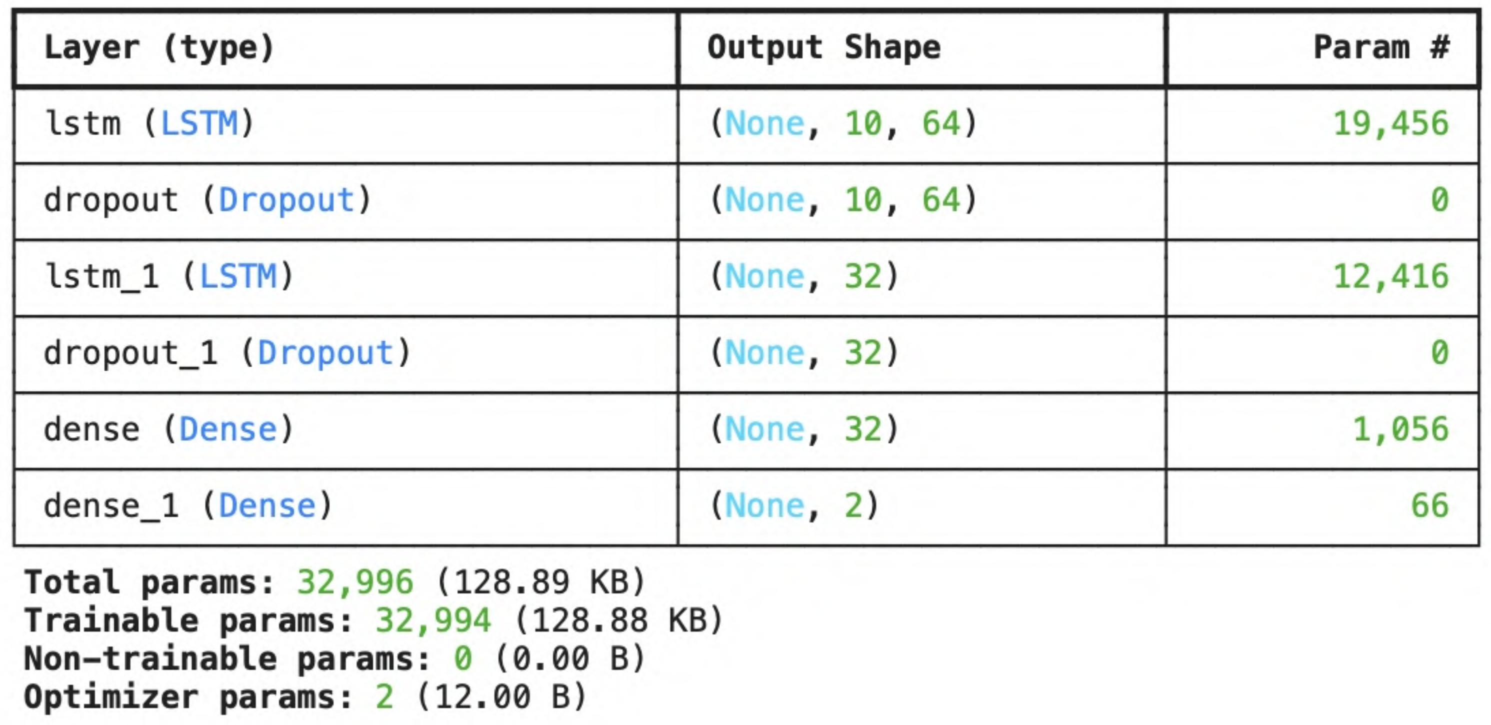
ML Methodology

LSTM

Model Type & Architecture

Layers Configuration:

- **LSTM Layers**: Two LSTM layers are used to process time-series data, capturing long-term dependencies in the sequence data effectively.
  - **First LSTM Layer**: Consists of 64 neurons.
  - **Second LSTM Layer**: Comprises 32 neurons, further refining the features extracted by the first LSTM layer.
- **Dropout Layers**: Positioned after each LSTM layer to prevent overfitting.
- **Dense Layers**: Two dense layers are included towards the end of the model:
  - **First Dense Layer**: Contains 32 neurons and serves to interpret the features from the LSTM layers.
  - **Output Dense Layer**: With just 2 neurons, corresponding to the predicted latitude and longitude.



Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 10, 64)	19,456
dropout (Dropout)	(None, 10, 64)	0
lstm_1 (LSTM)	(None, 32)	12,416
dropout_1 (Dropout)	(None, 32)	0
dense (Dense)	(None, 32)	1,056
dense_1 (Dense)	(None, 2)	66

```
Total params: 32,996 (128.89 KB)
Trainable params: 32,994 (128.88 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 2 (12.00 B)
```

- **Dense Layers Activation Functions**:
  - **ReLU Activation**: The first dense layer utilizes the ReLU for its efficiency and effectiveness in introducing non-linearity, improving the ability to learn complex patterns.
  - **Linear Activation**: The output layer employs a linear activation function.

ML Methodology

LSTM

Custom Loss Functions for Evaluation Metric: Recognising the limitations of traditional loss metrics, this function is specifically designed to align with Google's challenge metric. It emphasises the mean of the 50th and 95th percentile errors.

Inputs:

Dataset Composition and Split:

- The training dataset comprises 65 routes, which are divided into:
  - **Training Subset**: 80% of the data, with 20% of this subset further reserved for validation.
  - **Testing Subset**: The remaining 20% of the routes, used exclusively for evaluating the model's performance.

Shape of Input Data:

- Each batch fed into the model is structured as follows: (Epochs/run, Window size = 10, Number of features = 11). The model processes data in sequences of 10 time steps, each with 11 distinct features, facilitating the capture of temporal and spatial dependencies.

Outputs:

- The model predicts two continuous variables for each input sequence: latitude & longitude.

Methodology

RTKLIB

RTKLIB is an **open-source program tailored** for **precise positioning** with GNSS data. The software excels in **Real-Time Kinematic processing**, a GPS technology that enhances the precision of position data to within a few centimetres by using measurements from the **nearest base station**.

Data Handling and Processing:

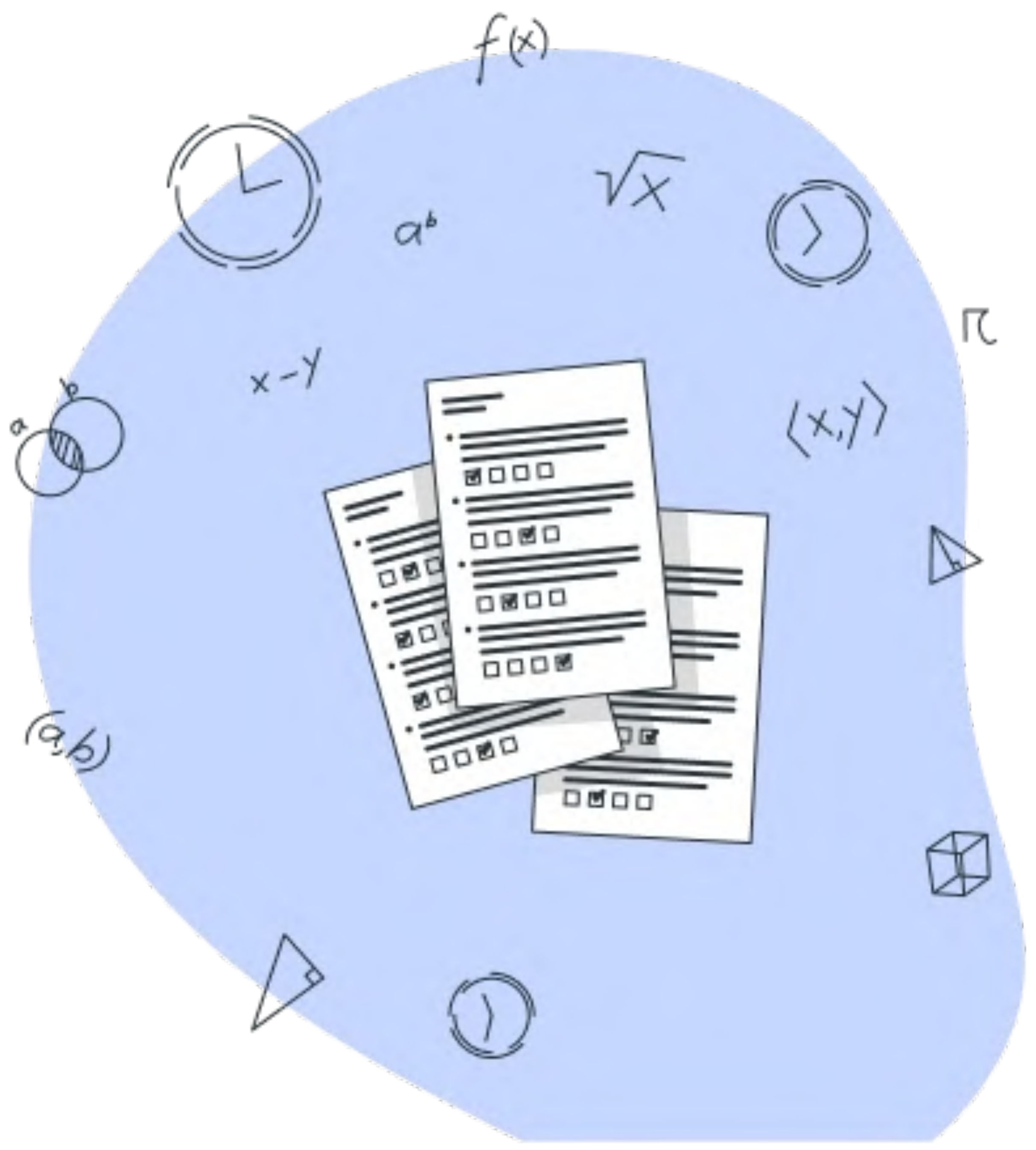
- **Navigation Data Acquisition**: Source navigation data and base station observations from EarthData (ensures robust & accurate positioning information).
- **Conversion to RINEX**: Transform raw GNSS data files from Android devices into RINEX (Receiver Independent Exchange Format) files.
- **Post-Processed Kinematics (PPK)**: Utilize RTKLIB for PPK operations to refine location estimates - processing GNSS data to correct positions post-recording, ensuring higher accuracy.

Output and Integration:

- **CSV Creation:** Convert the output from RTKLIB into CSV files, detailing all epochs in the test dataset to align with the timestamps.
- **Timestamp-Based Data Extraction:** From the CSV files, extract specific data corresponding to particular timestamps to synchronise with the ML model inputs.

Evaluation Metric

is it even working?



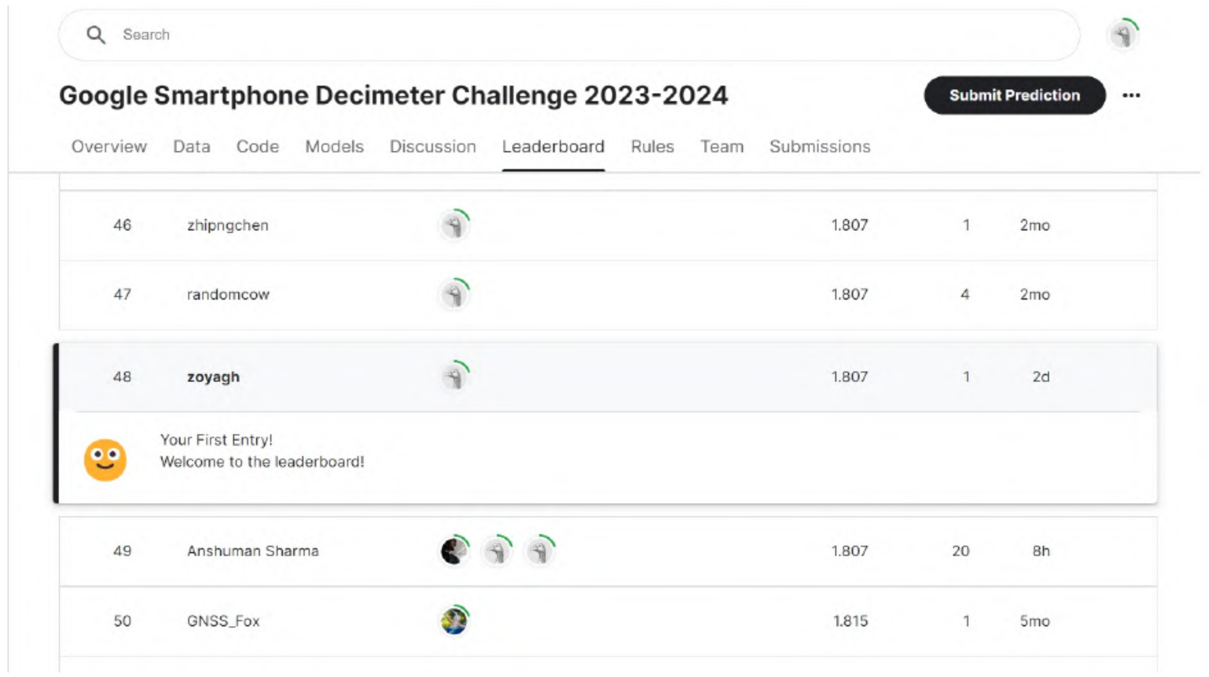
Submissions are scored on the **mean of the 50th** and **95th percentile distance errors**.

- For each smartphone, at every second, the horizontal distance is computed between the predicted latitude/longitude and the actual (ground truth) latitude/longitude.
- The **50th percentile (median) error** represents the typical error in the predictions.
- The **95th percentile error** reflects the worst-case errors in the predictions, showing how far off the predictions can be in less favorable circumstances.

Finally, the **mean of these average percentile values is calculated across all phones in the test set**.

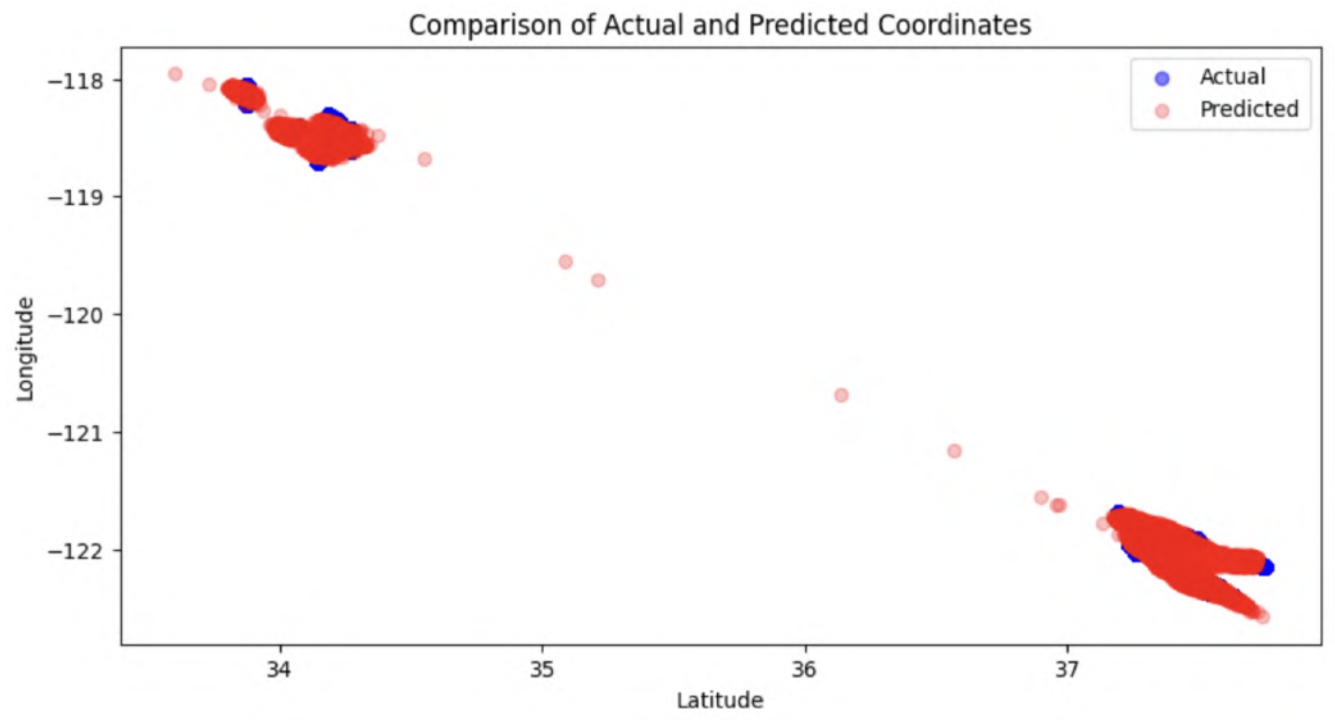
Our performance

yes, it is


Google Smartphone Decimeter Challenge 2023-2024

46	zhipngchen	1.807	1	2mo
47	randomcow	1.807	4	2mo
48	**zoyagh**	1.807	1	2d
49	Anshuman Sharma	1.807	20	8h
50	GNSS_Fox	1.815	1	5mo

Your First Entry!
Welcome to the leaderboard!



```
Calculated 50th Percentile: 1.6943902570771678
Calculated 95th Percentile: 2.06232662302294
Calculated Mean: 1.878358440050054
```

- Currently, we rank **48th on the global leaderboard** (out of ~300 teams).
- 10 other people **share a score** of **1.807m** however this is based on submission time
- These **rankings start from 39th** on the global leaderboard
- Most **other ML solutions** report a score of **7+** on the Kaggle leaderboard

0.99

accuracy in our 1st epoch!?
unfortunately not :(

Deployment

can we use this?

It can be used to improve navigation around campus for new visitors or students, and track university assets like vehicles, equipment etc. It can also be used for the implementation of more realistic virtual tours.

- **Hardware Setup:** Install GNSS receivers at strategic locations around the campus to collect raw satellite data.
- **Data Pipeline:** Establish a robust data pipeline that can handle streaming data from GNSS receivers to the servers where RTKLIB processes the data and the LSTM model runs predictions.
- **Integration:** Ensure the system integrates seamlessly with existing university databases and software systems, possibly using APIs.
- **User Interface**: Develop user-friendly interfaces that allow students, staff, and researchers to access and utilise the data easily. Possibly an app for real-time campus navigation or a dashboard for monitoring research data.
- **Feedback Loop**

Possible challenges in deployment

will it really work out?

- **Cybersecurity threats**
  Larger systems with more extensive data integration **increase the potential attack surface** for cybersecurity threats.
- **Data quality and availability**
  Ensuring **consistent quality** and **availability** of GPS data **across the campus** can be challenging due to physical obstructions, interference, and multi-path effects that degrade GPS signals.
- **Costs**
  Initial setup, ongoing maintenance, and upgrades can be costly. **Budget constraints** might **limit the extent** of **deployment** and the **quality** of the equipment and software used. In addition to this, people will have to be trained to use the new system.
- **Regulatory, security and privacy issues**
  Different regions have different **regulations for accessing satellite data** which need to be adhered to. Also, handling and storing location data involve addressing **privacy concerns** and complying with **data protection regulations**.

Challenges for us

i give up :'(

- **Domain Knowledge**
  Unfortunately, we have none.
  Our solution: research-research-research!

- **Features Processing**
  Noisy and **erroneous** data, with outliers making it difficult to train models on it.
  Our Solution: Using relevant thresholding methods and filters to clean the data meticulously

- **Choice of Features**
  Incorrect feature selection leads to poor model performance for LSTM model.
  Our Solution: More research for industry standards and analysis on what features work best

- **Hyperparameter Tuning**
  The model needs to be **trained multiple times** using different parameters which is **computationally heavy** and **time consuming**.
  Our Solution: Make more google accounts

End