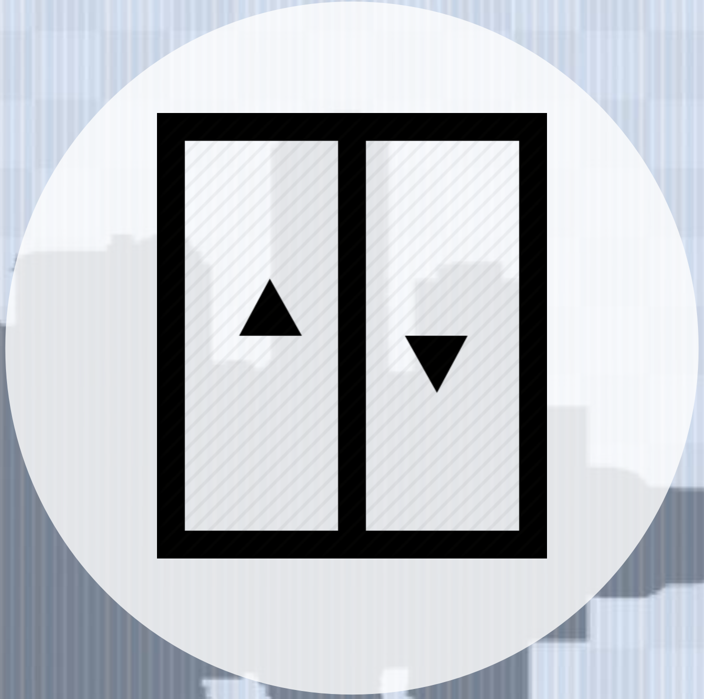
Classifying Vertical Mobility Patterns



Dr. ELIAS WILLEMS
Prof ROLAND BOUFFANAIS

K VIDYAA
AJAYKUMAR MANIVANNAN

AIM

To classify vertical mobility patterns in a largely unlabeled dataset

Why?

To understand the network underlying cities

"Because networks determine the rates at which energy and resources are delivered to cells, they set the pace of all physiological processes"

- Scale, *Geoffrey West.*

To predict Human Behavior

"Many of our daily actions are quite repetitive in nature, making them foreseeable."

"Once priorities come into play, randomness is out, and bursts take place. "

- Bursts, *Albert-Laszlo Barabasi*

Steps

Data description

Data Visualization

Optimization

Feature extraction

Numerical differentiation, Regression and Interpolation

Classification

Optimization

Data Description

Data statistics

Labelled data (6 hours)
No. of Training data: 641 (60 %)
No. of Validation data: 428 (40 %)

Unlabelled data (18 hours)
No. of Test data: 5246

Ground Truth (GT)	Training data (60%)	Validation data (40%)	Transportation (Vehicle)	Ambulation (Non-vehicle)
Train ride + Train stop	74 + 21	51 + 13	✓	
Bus ride + Bus stop	116 + 72	73 + 49	✓	
Car ride + Car stop	48 + 7	27 + 6	✓	
Walking	199	126		✓
Vertical Mobility (VM)	45	43		✓
Idle	59	40		✓
Total data points	641	428	557	512

Recording Frequency – Every 13~18 seconds

Data Description

Description of features

Environmental variables	Motion Sensor variables	Derived variables
Humidity, Temperature, IR temperature, Barometric Pressure, Light, Noise	MeanMag, MeanGyr, STDGyr, STDAcc, MaxAcc, STDMag, MaxGyr.	Step Count, Latitude, Longitude

Gyr – Gyroscope; Acc – Accelerometer; Mag – Magnetometer.
Mean – Mean of the magnitude in 100HZ; Max – Max of the magnitude in 100HZ.

Snapshot of data

	index	TIMESTAMP	HUMIDITY	TEMPERATURE	IRTEMPERATURE	PRESSURE	HEIGHT	\
0	1	1542950608	57.5	32.88	26.37	101112	16.81	
1	2	1542950624	60.0	32.89	24.04	101098	17.91	
2	3	1542950639	57.5	32.86	23.31	101114	16.65	
3	4	1542950656	57.0	32.86	23.31	101112	16.81	
4	5	1542950672	56.0	32.85	22.59	101104	17.44	

	LIGHT	NOISE	NumWifi	...	ACCURACY	GT	GT_binary_1	GT_binary_2	\
0	0	58	16	...	110.0	Bus ride	Vehicle	HDA	
1	2190	58	15	...	152.0	Bus ride	Vehicle	HDA	
2	2420	55	1	...	-2.0	Bus ride	Vehicle	HDA	
3	2310	55	4	...	144.0	Bus ride	Vehicle	HDA	
4	2280	55	6	...	163.0	Bus stop	Vehicle	HDA	

	lat_interp_regres	lon_interp_regres	velocity	dsteps-1	dp-1	dt-1
0	1.310810	103.854753	16.07	0.0	1.0	18.0
1	1.311330	103.855120	15.92	0.0	-14.0	16.0
2	1.311807	103.855460	15.66	0.0	16.0	15.0
3	1.312333	103.855840	15.25	0.0	-2.0	17.0
4	1.312806	103.856188	14.70	0.0	-8.0	16.0

Data Visualization

t-SNE – t - distributed stochastic neighbor embedding

It is a non-linear technique for dimensionality reduction.

It helps visualize high dimensional data into lower dimensional space.

It uses probability distributions (Gaussian) that defines the relationship between the points in high dimensional space and recreates them (student t-distribution) in lower dimension.

t-SNE optimizes this using gradient descent with a non-convex loss function.

Data Visualization

t-SNE – t - distributed stochastic neighbor embedding

Step 1: Probability (Gaussian) of picking point xj as the neighbor of point xi.

$$p_{ij} = \frac{\exp(-||x_i - x_j||^2 / 2\sigma_i^2)}{\sum_{k \neq l} \exp(-||x_k - x_l||^2 / 2\sigma_i^2)}$$

Step 2: Re-creating probability distribution (Student t-distribution) in low dimensional space as yi.

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l} (1 + ||y_k - y_l||^2)^{-1}}$$

Step 3: Optimization is done by gradient descent on the KL-divergence between the distributions p and q.

$$\frac{\delta J}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1}$$

A positive gradient represents an attraction, whereas a negative gradient represents a repulsion between the points. This “push-and-pull” eventually makes the points settle down in the low-dimensional space.

Disadvantage:

Non-convex loss functions means the solutions might get stuck in local optima.

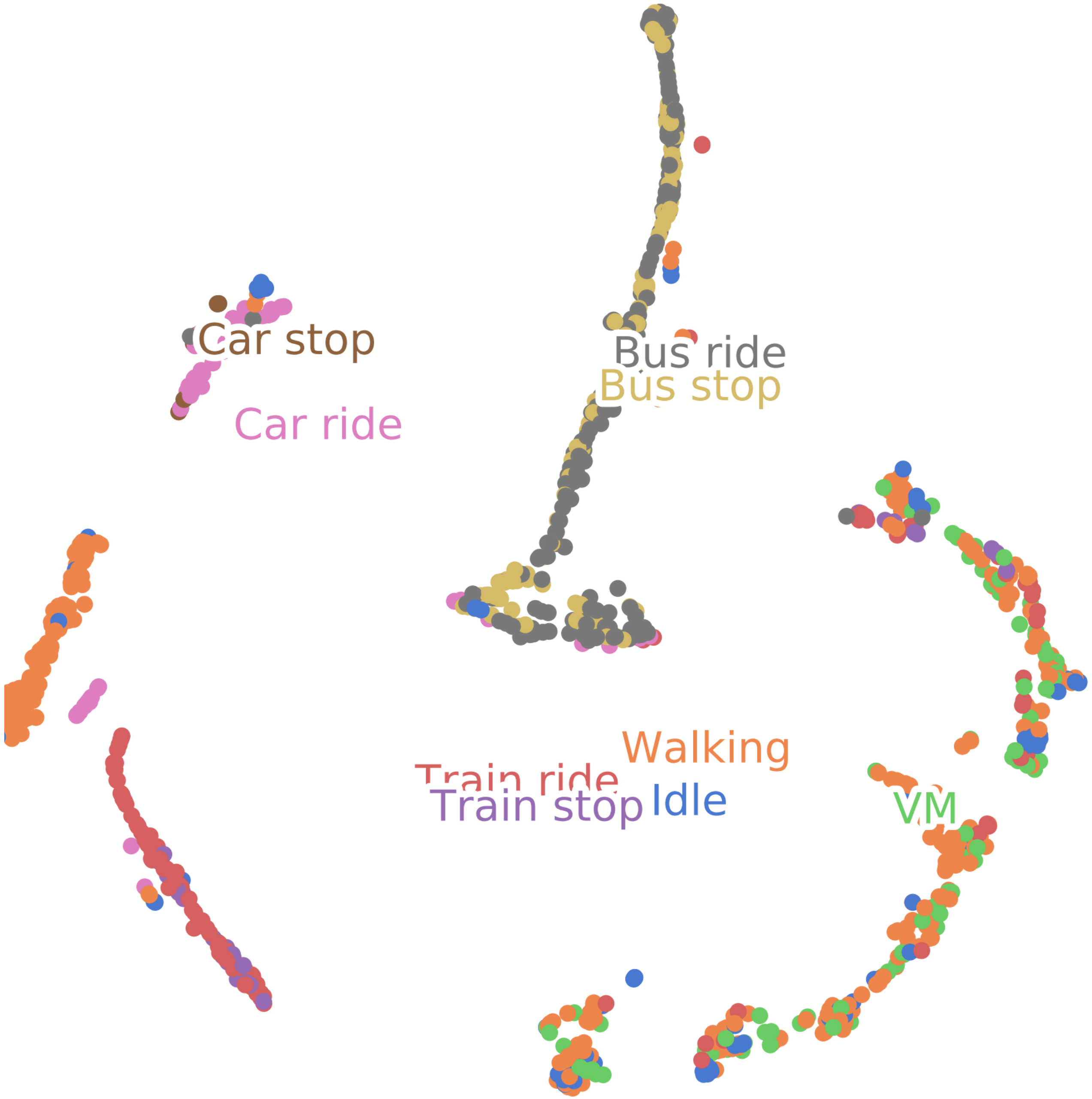
Data Visualization: t-SNE

Features: Environmental sensor variables

- Humidity
- Temprature
- IR temperature
- Barometric Pressure
- Light
- Noise

Results:

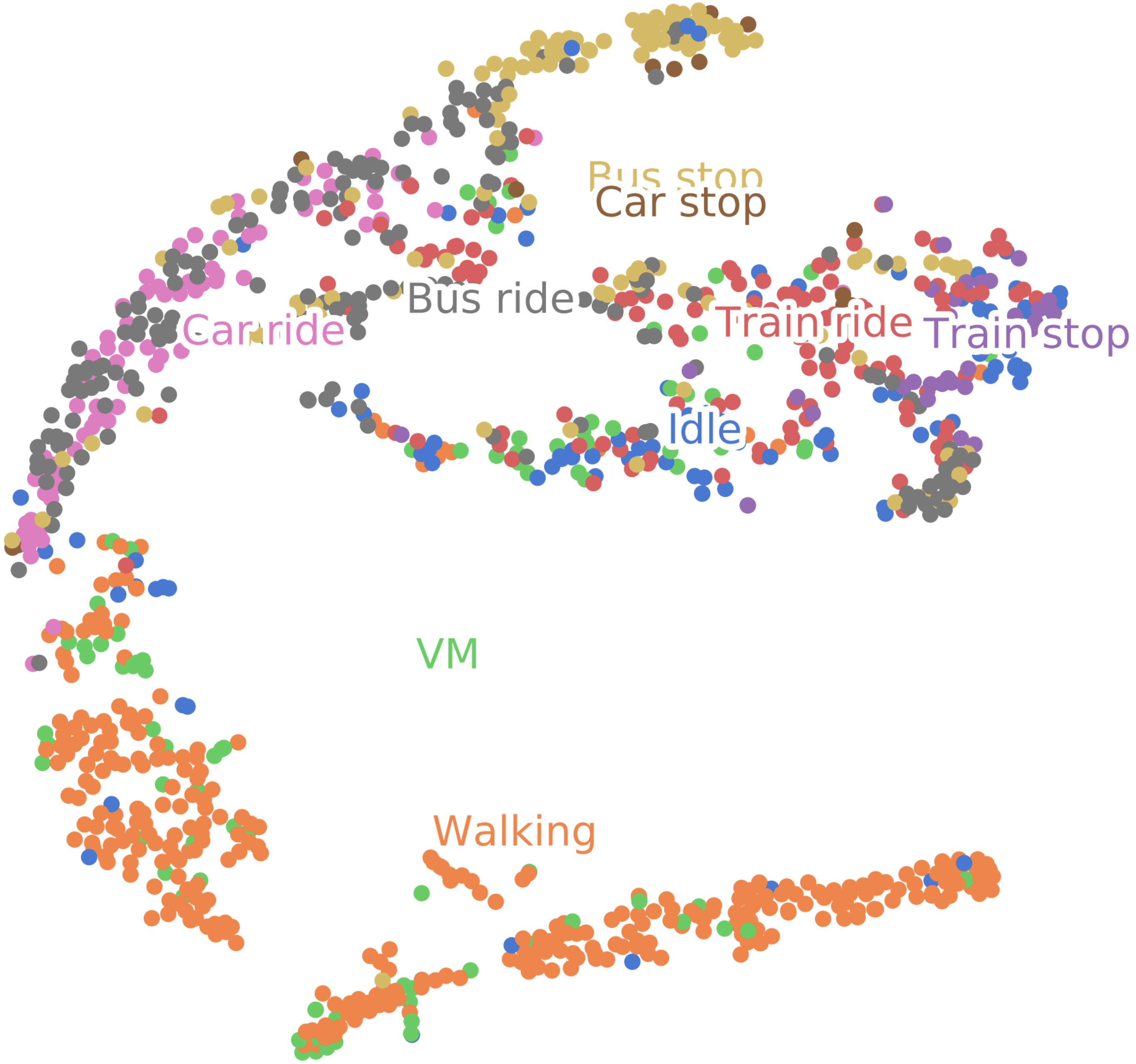
Using Environmental variables will group data by experiment ID.



Data Visualization: t-SNE

Features: Motion sensor variables

- MeanMag
- MeanGyr
- STDGyr
- STDAcc
- MaxAcc
- STDMag
- MaxGyr



Results:

Using Motion sensor variables makes the model less dependent on time-varying conditions.

Feature extraction

Regression and Interpolation of WIFI localization data

Feature

WIFI localization data → Interpolation → Regression → Interpolated and Regressed location data

Extracted Feature

Interpolation is performed to offset failed localization (missing data)

Method: Piecewise Cubic spline Interpolation

Regression is performed to offset localization inaccuracy.

Method: Piecewise Natural Cubic regression splines.

Feature extraction:

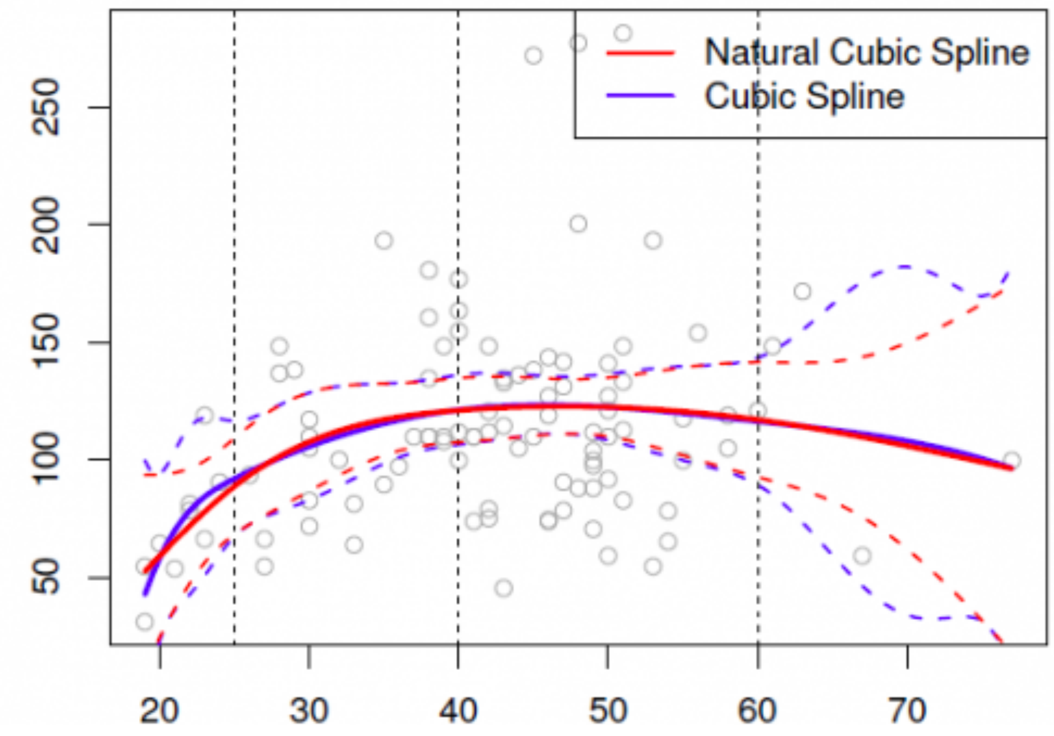
Interpolation: Piecewise Cubic spline Interpolation
Regression : Piecewise Natural Cubic regression splines.

Natural Cubic Spline — is a piece-wise cubic polynomial that is twice continuously differentiable. It is considerably 'stiffer' than a polynomial in the sense that it has less tendency to oscillate between data points *.

Segments: each segment of the spline curve is a cubic polynomial.

At the Knots: the first derivative and the second derivative is continuous

At the end points: The second derivative of the spline at end points are zero. Since these end condition occur naturally in the beam model, the resulting curve is known as the natural cubic spline.



Knots at 25,40 and 60 **

** https://gerardnico.com/data/type/number/function/natural_spline

* https://towardsdatascience.com/numerical-interpolation-natural-cubic-spline-52c1157b98ac

Feature extraction:

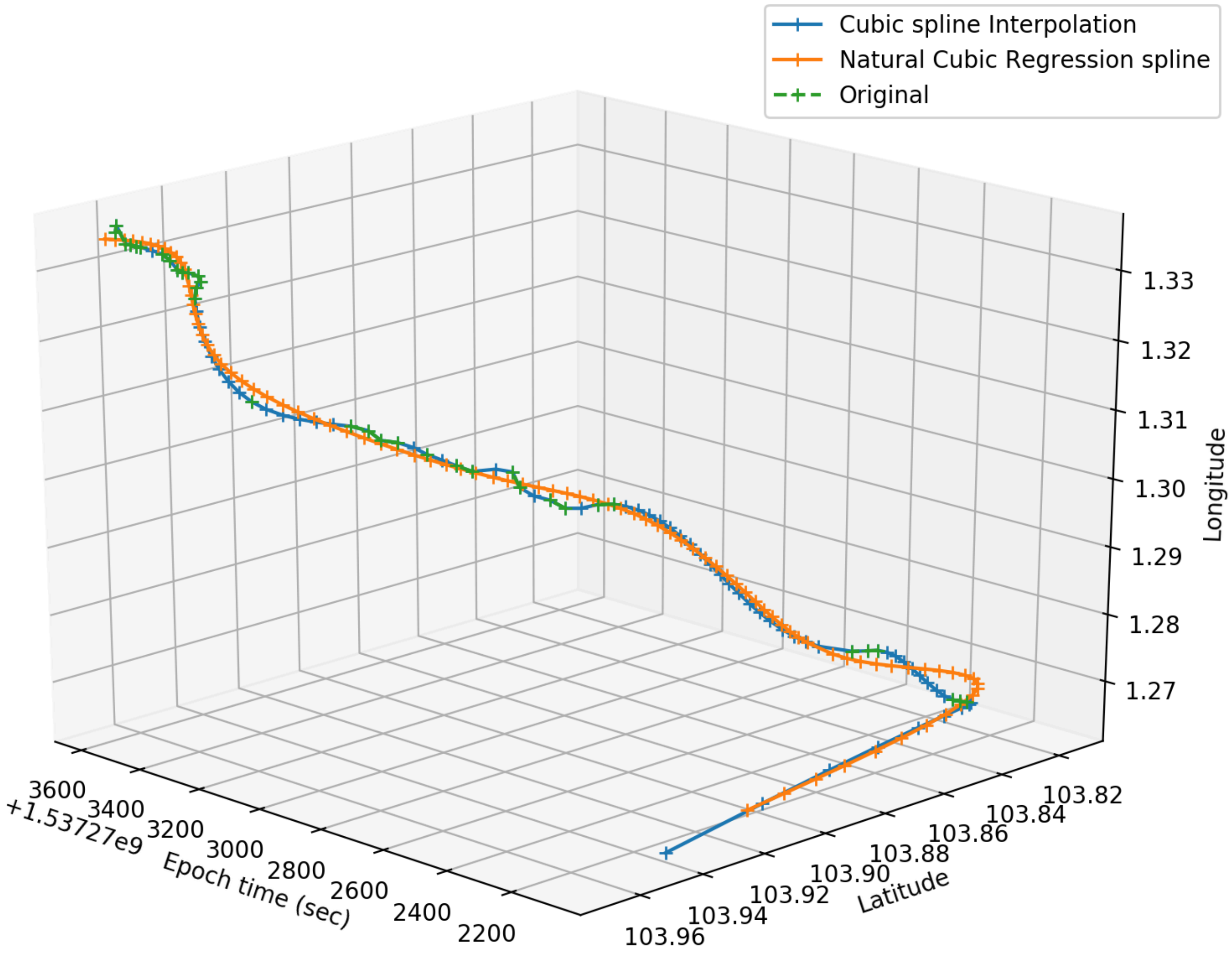
Results:

During travel, WIFI localization sometimes fails and results in missing data.

Here most of the data is interpolated.

Regression and Interpolation of WIFI localization data

Location: Harbor front to SUTD



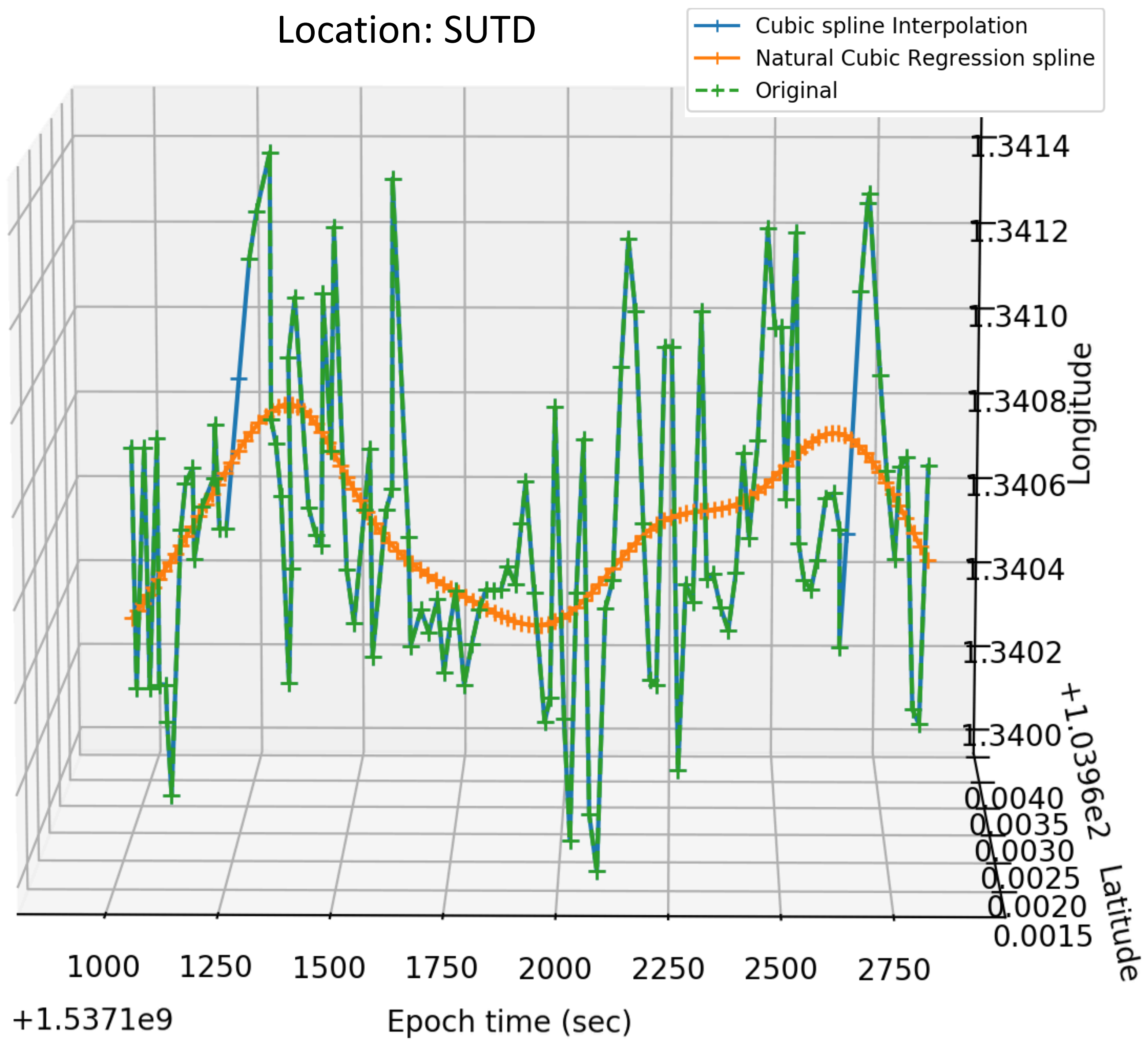
Regression and Interpolation of WIFI localization data

Feature extraction:

Results:

Within a small environment, WIFI localization data is very erratic due to its inaccuracy (± 150m).

Here regression splines smoothens the location data.

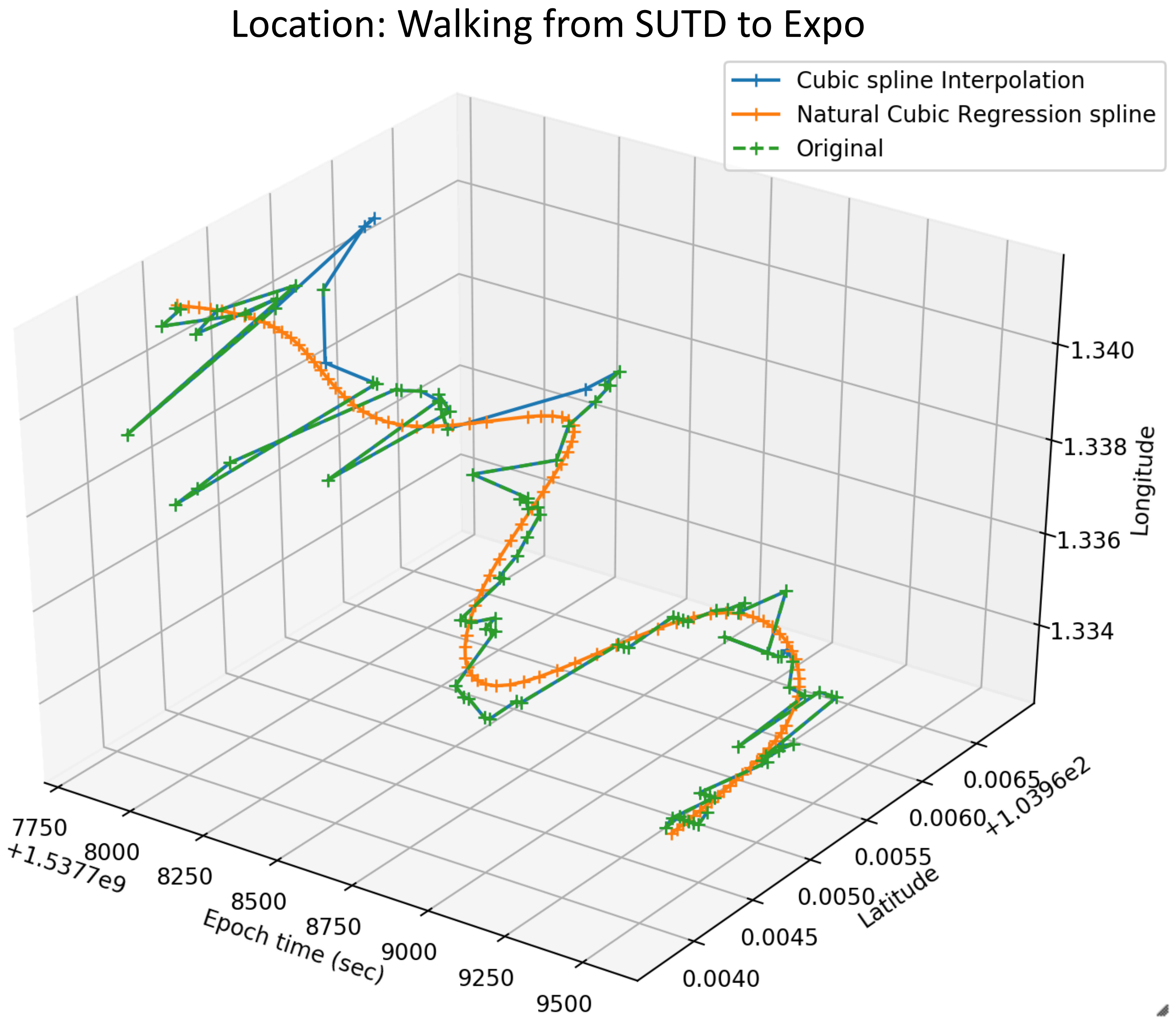


Feature extraction:

Results:

Within a small environment, WIFI localization data is very erratic due to its inaccuracy (± 150m).

If original location data is used to calculate velocity of travel, it can be misleading.



Feature extraction

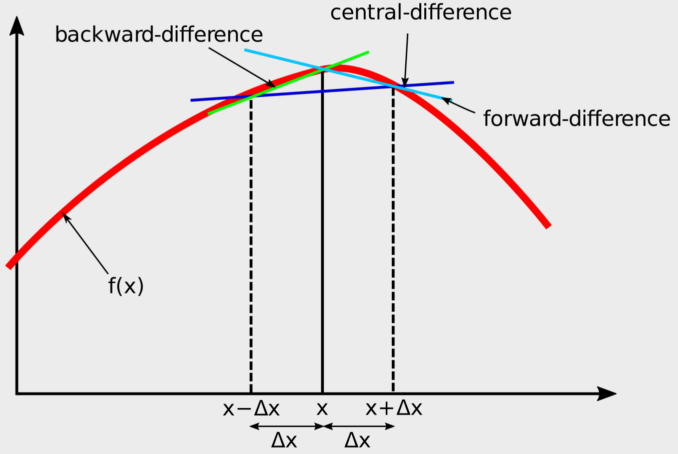
Numerical differentiation

Backward difference:

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h}$$

Centered difference:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2\text{h}}$$



For irregularly spaced grid, polynomial interpolation is required. This is performed by the gradient function in NumPy library *.

$$\hat{f}_i^{(1)} = \frac{h_s^2 f(x_i + h_d) + (h_d^2 - h_s^2) f(x_i) - h_d^2 f(x_i - h_s)}{h_s h_d (h_d + h_s)} + \mathcal{O}\left(\frac{h_d h_s^2 + h_s h_d^2}{h_d + h_s}\right)$$

* https://docs.scipy.org/doc/numpy/reference/generated/numpy.gradient.html

Feature extraction

Numerical differentiation

Results:

Feature	Pre-processing	Numerical differentiation	Extracted Feature
Step count	NIL	Backward and Centered difference	$\frac{d(step\ count)}{dt}$
Barometric Pressure	NIL	Backward and Centered difference	$\frac{dp}{dt}$
Location data	Interpolation and Regression	Backward and Centered difference	$\frac{dx}{dt}$

* https://docs.scipy.org/doc/numpy/reference/generated/numpy.gradient.html

Data Visualization: t-SNE

Features:

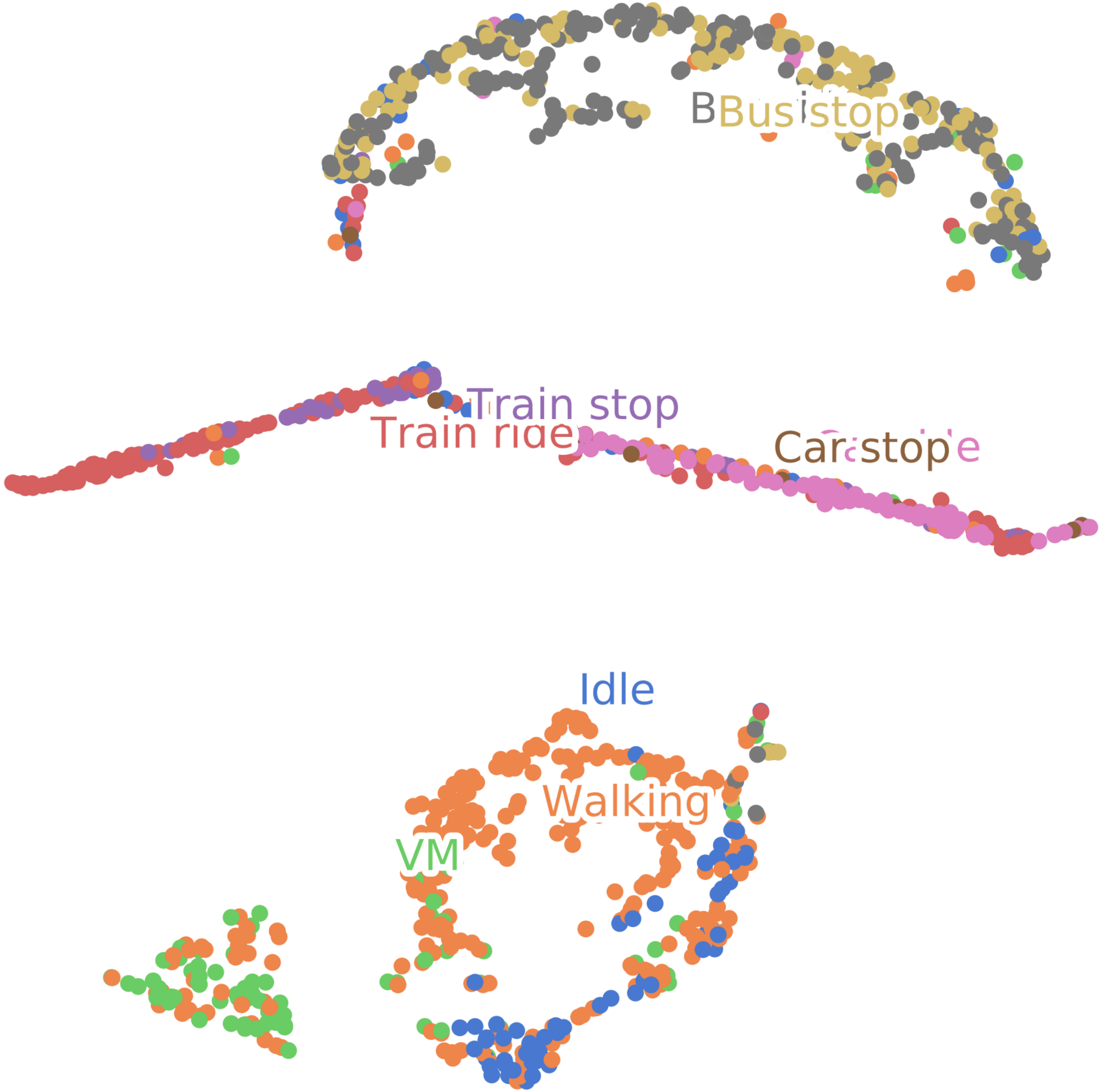
Backward difference of Barometric Pressure, Step count and location data.

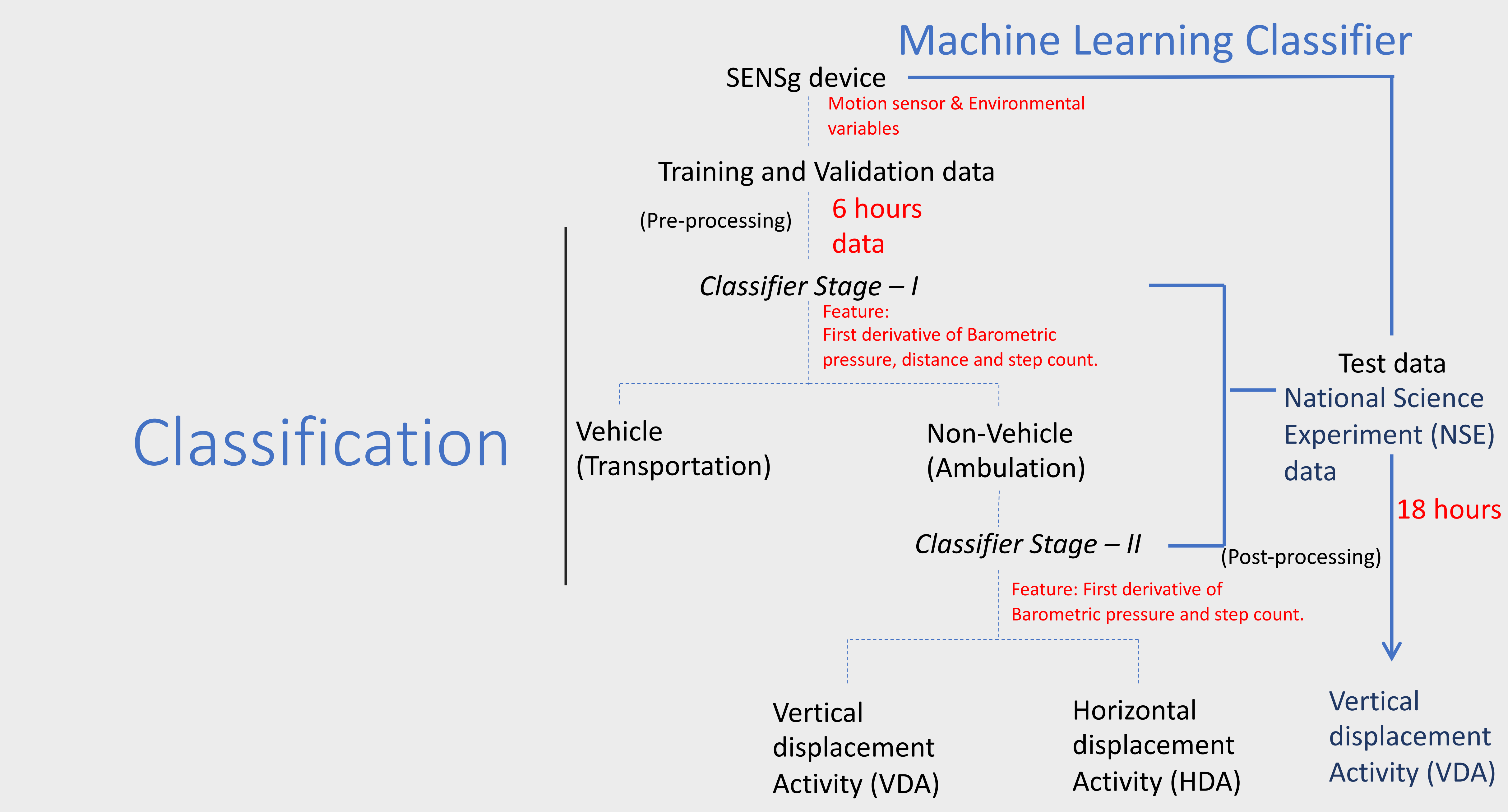
- $\frac{d(step\ count)}{dt}$
- $\frac{dp}{dt}$
- $\frac{dx}{dt}$

Results:

It shows clear separation of Transportation and ambulation data.

The above extracted features are user-independent and device-independent.




Machine Learning Classifier
SENSg device
Motion sensor & Environmental variables
Training and Validation data
(Pre-processing)
6 hours data
Classifier Stage – I
Feature:
First derivative of Barometric pressure, distance and step count.
Classification
Vehicle (Transportation)
Non-Vehicle (Ambulation)
Classifier Stage – II
Feature: First derivative of Barometric pressure and step count.
Vertical displacement Activity (VDA)
Horizontal displacement Activity (HDA)
Test data
National Science Experiment (NSE) data
18 hours
(Post-processing)
Vertical displacement Activity (VDA)

Classification

Classifier

Boosting is an ensemble technique in which the predictors are not made independently, but sequentially.

AdaBoost (Adaptive Boosting)

- Looking for the global optimum by solving a sequence of subproblems in a greedy manner.
- AdaBoost builds an additive logistic regression model.
- It has an exponential loss function.

XG Boost (e**X**treme **G**radient **B**oosting)*

- It uses a gradient descent algorithm to minimize the loss when adding new models.

Objective function obj(θ)=$L(\theta)$+$\Omega(\theta)$

L – Loss function

Ω - Regularization

θ - Parameter

* https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d

Classification Results on Validation data

Classifier Stage I – Ambulation vs Transportation

Classifier	Features	Overall accuracy (%)	Predicted labels (%)	Precision (%)	Recall (%)	F1 score (%)	Number of data
AdaBoost	Original features	91.35	Ambulation	97	85	91	209
			Transportation	87	98	92	219
	t-SNE applied features	91.82	Ambulation	97	86	91	209
			Transportation	88	97	92	219
XG Boost	Original features	94.15	Ambulation	94	94	94	209
			Transportation	94	95	94	219
	t-SNE applied features	90.65	Ambulation	92	89	90	209
			Transportation	89	93	91	219

Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$

Precision = $\frac{TP}{TP+FP}$

Recall = $\frac{TP}{TP+FN}$

Results:
XG Boost classifier gives the highest overall accuracy of 94.15% for the original feature set.

Classification Results on Validation data

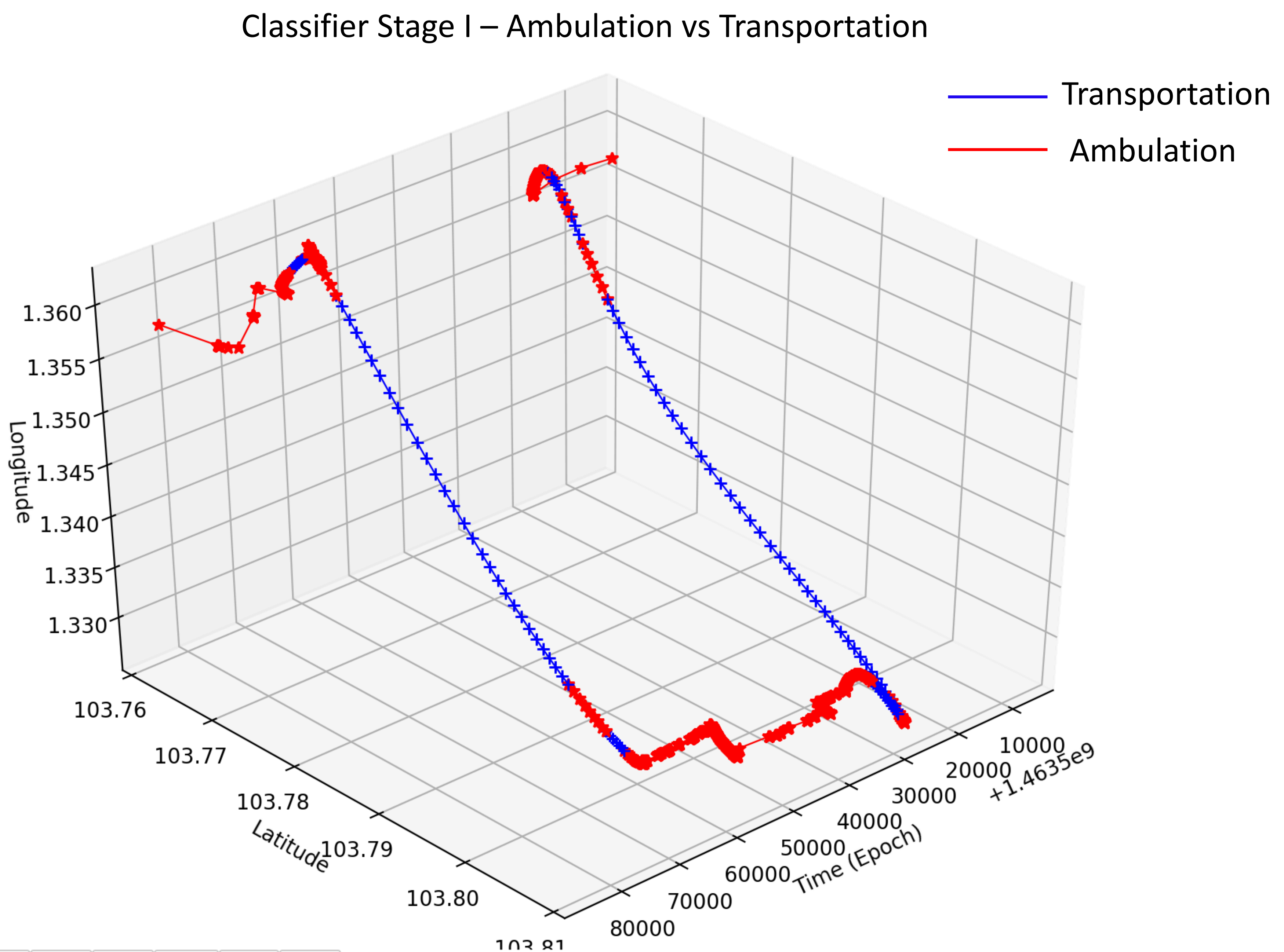
Classifier Stage II – Vertical displacement activities (VDA) vs Horizontal displacement activities (HDA)

Classifier	Features	Overall accuracy (%)	Predicted labels (%)	Precision (%)	Recall (%)	F1 score (%)	Number of data
AdaBoost	Original features	92.19	HDA	92	97	95	143
			VDA	93	81	86	62
	t-SNE applied features	91.21	HDA	90	98	94	143
			VDA	94	76	84	62
XG Boost	Original features	92.19	HDA	92	97	95	143
			VDA	93	81	86	62
	t-SNE applied features	92.19	HDA	93	97	95	143
			VDA	91	82	86	62

Results:
AdaBoost classifier gives the highest precision (Avoiding False positives) and recall (Finding True positives) for classifying VDA.

Although recall (% of positive samples found) of VDA is low, they correspond to 95% of total pressure changes i.e. **it accounts for 95% of the vertical mobility**.

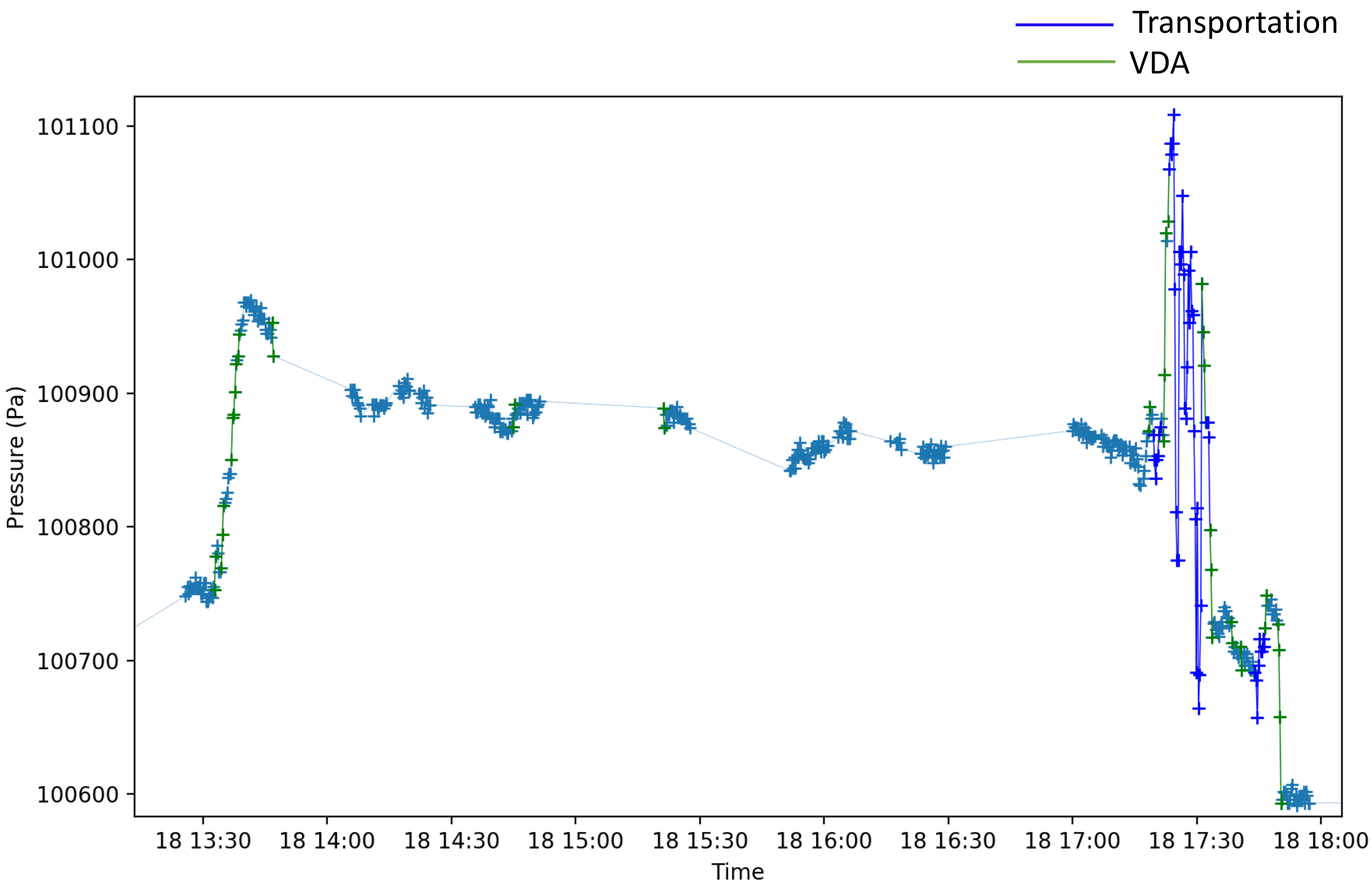
Classification Results on Unlabeled Test data (NSE database)



Sample shown:
1 person for over
18 hours

Classification Results on Unlabeled Test data (NSE database)

Classifier Stage II – Vertical displacement activities (VDA) vs Horizontal displacement activities (HDA)



Sample shown:
1 person for
over 18 hours

Conclusion

Data Visualization – Optimization

- t-SNE method is used to do dimensionality reduction and visualize the data. The t-SNE applied variables are then used as features for classifier.

Feature extraction – Numerical differentiation, Regression and Interpolation

- Cubic spline Interpolation and Natural cubic Regression splines is used to interpolate the missing data and smoothen the inaccurate localization.
- Numerical differentiation is used to derive first derivatives of select variables and used as feature for classifier.

Classification - Optimization

- AdaBoost and XG Boost classifiers are used to perform classification of the data in two stages – Ambulation vs Transportation and VDA vs HDA. The trained classifiers are then used to predict the test data (NSE database).

Thank you

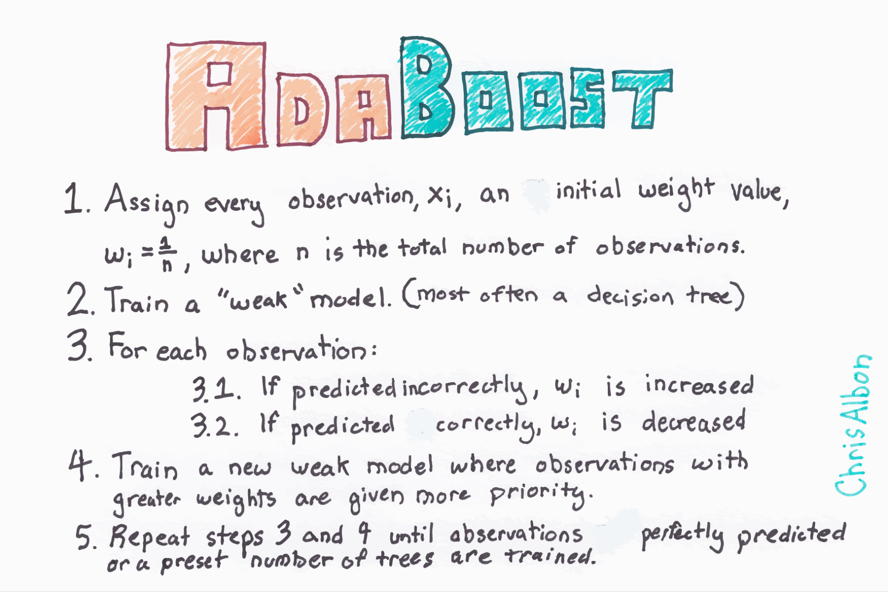
References

- West, Geoffrey B. *Scale: The Universal Laws of Growth, Innovation, Sustainability, and the Pace of Life in Organisms, Cities, Economies, and Companies*. , 2018. Print.
- Albert-Laszlo Barabasi. 2010. Bursts: The Hidden Pattern Behind Everything We do. Dutton Adult.
- http://rob.schapire.net/papers/explaining-adaboost.pdf
- https://people.eecs.ku.edu/~jhuan/EECS940_S12/slides/boosting.pdf
- https://chrisalbon.com/machine_learning/trees_and_forests/adaboost_classifier/
- https://doi.org/10.1038/nature06958
- https://www.analyticsvidhya.com/blog/2018/03/introduction-regression-splines-python-codes/
- https://www.datacamp.com/community/tutorials/introduction-t-sne
- http://mlexplained.com/2018/09/14/paper-dissected-visualizing-data-using-t-sne-explained/

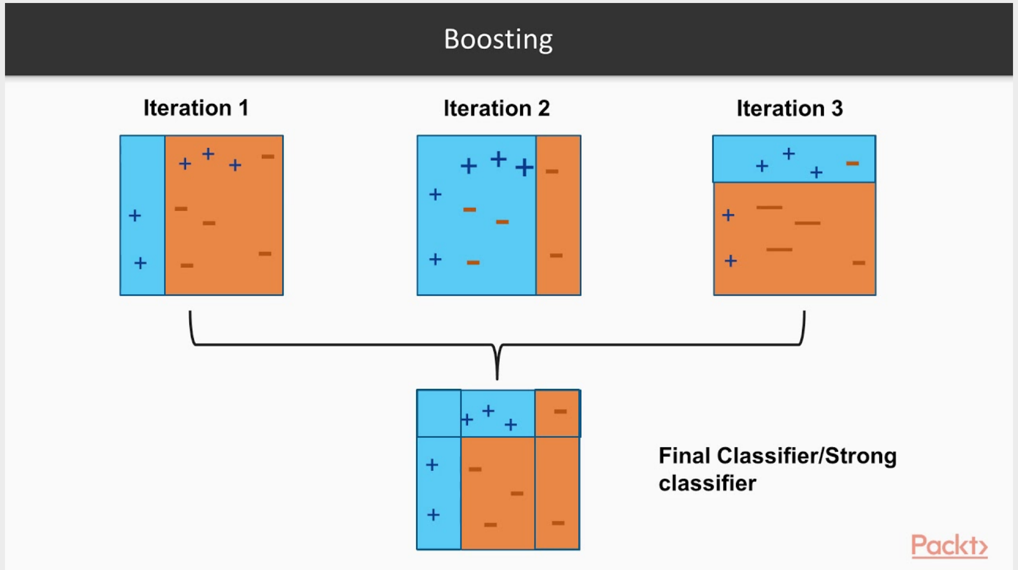
Supplementary slides

AdaBoost Classifier with decision trees

Combines several weak and inaccurate classifiers to create highly accurate prediction



Adaptive Boosting classifier is the first successful boosting algorithm for binary classification problems.



Advantages:
-> Resistant to Over-fitting
-> Forces classifier to work on hard-to-classify data.

Performance evaluation

The classifier performance is generally evaluated by metrics such as accuracy, precision, recall and F-measure. A confusion matrix is used to check the True positives (TP), True negatives (TN), False positives (FP) and False negatives (FN) for each class.

Accuracy is given by $\frac{(TP+TN)}{(TP+TN+FP+FN)}$. It is simply the ratio of correct predictions.

The precision is given by $\frac{TP}{TP+FP}$. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative.

The recall is the ratio $\frac{TP}{TP+FN}$. The recall is intuitively the ability of the classifier to find all the positive samples.

The F-beta score is given by $F_\beta = \frac{(1+\beta^2) precision * recall}{\beta * precision + recall}$. It can be interpreted as a weighted harmonic mean of the precision and recall, where an F-beta score reaches its best value at 1 and worst score at 0. The F-beta score weights recall more than precision by a factor of beta. $\beta == 1.0$ means recall and precision are equally important.