11th INTERNATIONAL HEAVY HAUL CONFERENCE

Advancing Heavy Haul Technologies and Operations in a Changing World

2 – 6 September 2017 • Cape Town • South Africa

IHHA 2017
CAPE TOWN
02 – 06 September 2017

Institute of Railway Technology

Advancing the Railway Industry through Technology for over 45 Years

Machine Learning Approaches for Tamping Effectiveness Prediction

IHHA 2017
CAPE TOWN
02 – 06 September 2017

Chang Wei **TAN**[1], Paul **REICHL**[2], Geoff **WEBB**[1], Francois **PETITJEAN**[1],

[1]*Faculty of I.T. Monash University*

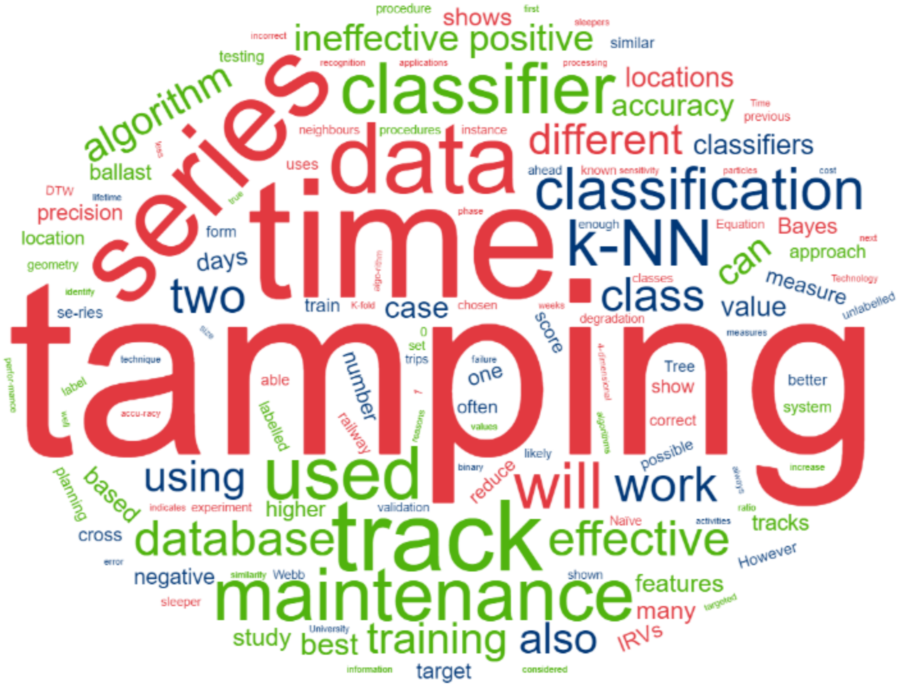
[2]*Institute of Railway Technology, Monash University*

chang.tan@monash.edu
www.monash.edu/irt



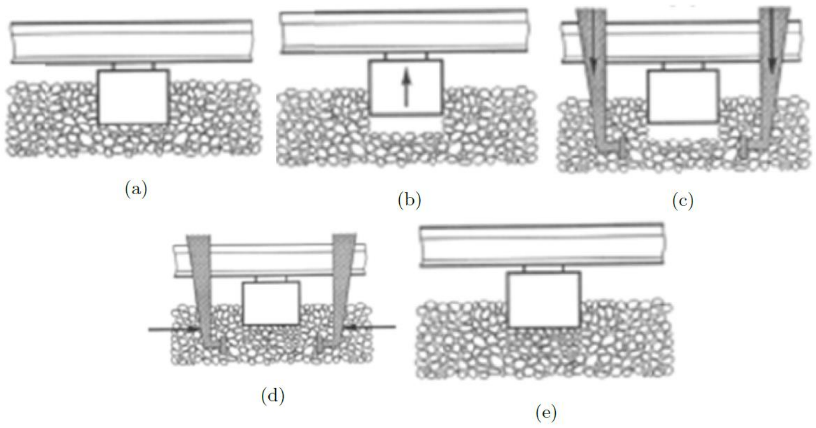

Outline

1. Tamping Maintenance
2. Motivation
3. Methodology
   - Data
   - Machine Learning Techniques
4. Results
5. Conclusion & Future Work



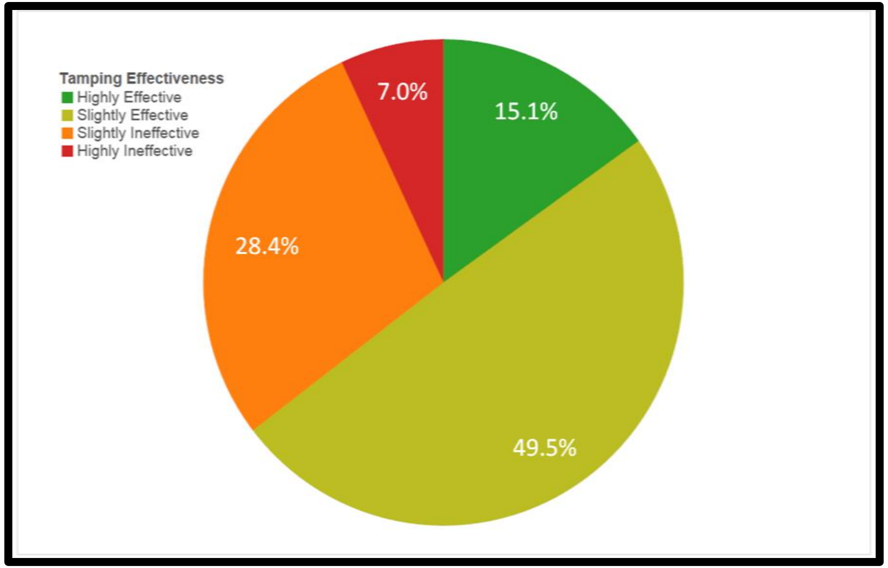
Tamping Maintenance

- Reset track geometry by rearranging the ballast particles



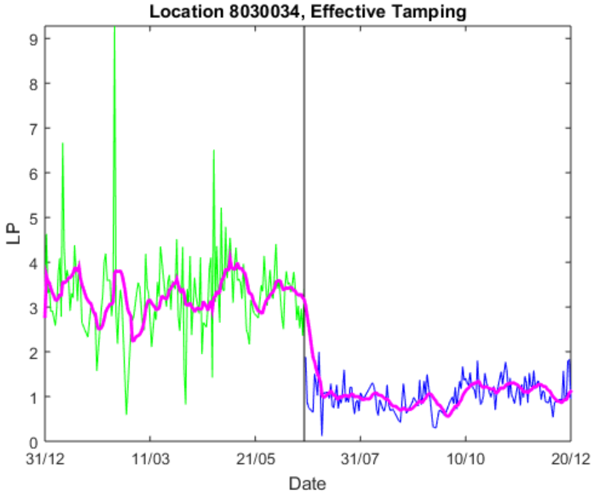
Motivation

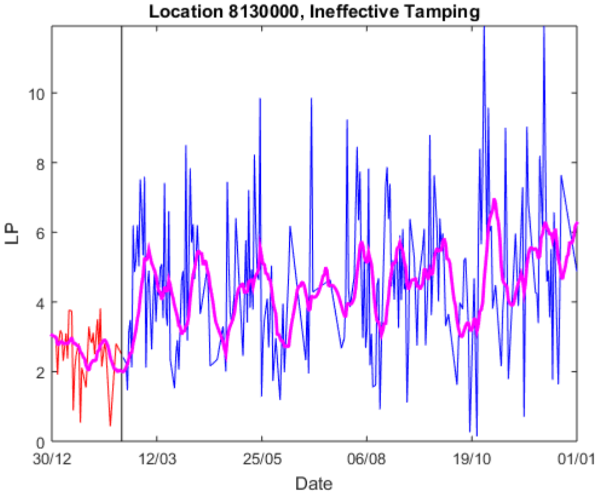
- 30-40% of the tamping maintenance are shown to be ineffective
- Ineffective tamping reduces tracks' life-cycle



What can we do?

- Predict Tamping Effectiveness
- A recent work calculates tamping effectiveness using ratio of average responses before and after tamping.





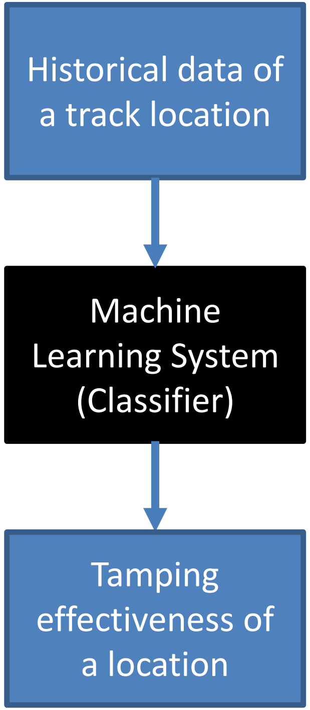
Significance

- Challenging problem due to many complex phenomena
  - Weather
  - Soil properties
- Important for 3 reasons
  1. Minimise maintenance cost and time
  2. Reduce unplanned downtime
  3. Avoid cost of failure recovery

Predict if tamping will be effective for a track location

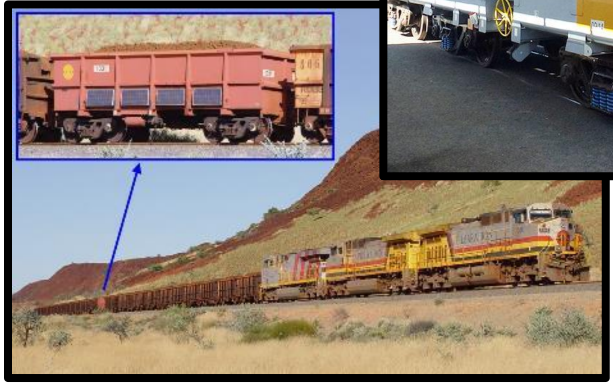
How to do it?

1. Data pre-processing & filtering
   – Convert acquired data from Instrumented Revenue Vehicles (IRVs) into time series for each track location
2. Train a classifier
   – *k*-Nearest Neighbours
   – Classification tree
   – Naïve Bayes
3. Evaluate and cross-validate the performance of the classifier
4. Predict tamping effectiveness

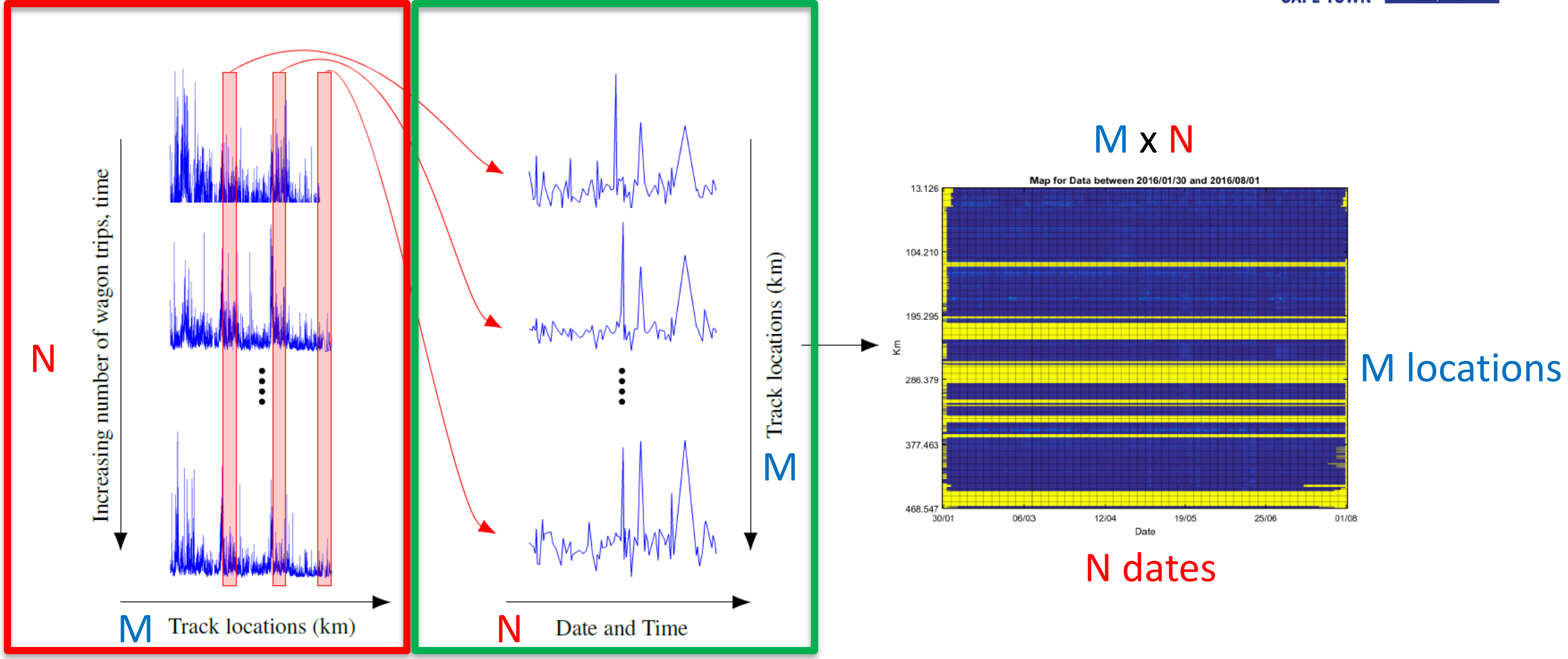


1. Data – Instrumented Revenue Vehicles

- Continuous monitoring system that uses measured wagon’s dynamic activity to infer information about track condition
- Installed on normal revenue wagons
  - Cheap
  - Do not affect normal operations
- Measures:
  - Speed
  - Axle Load
  - In Train Forces
  - Spring Nest Deflection (SND)



1. Data – IRVs Time Series

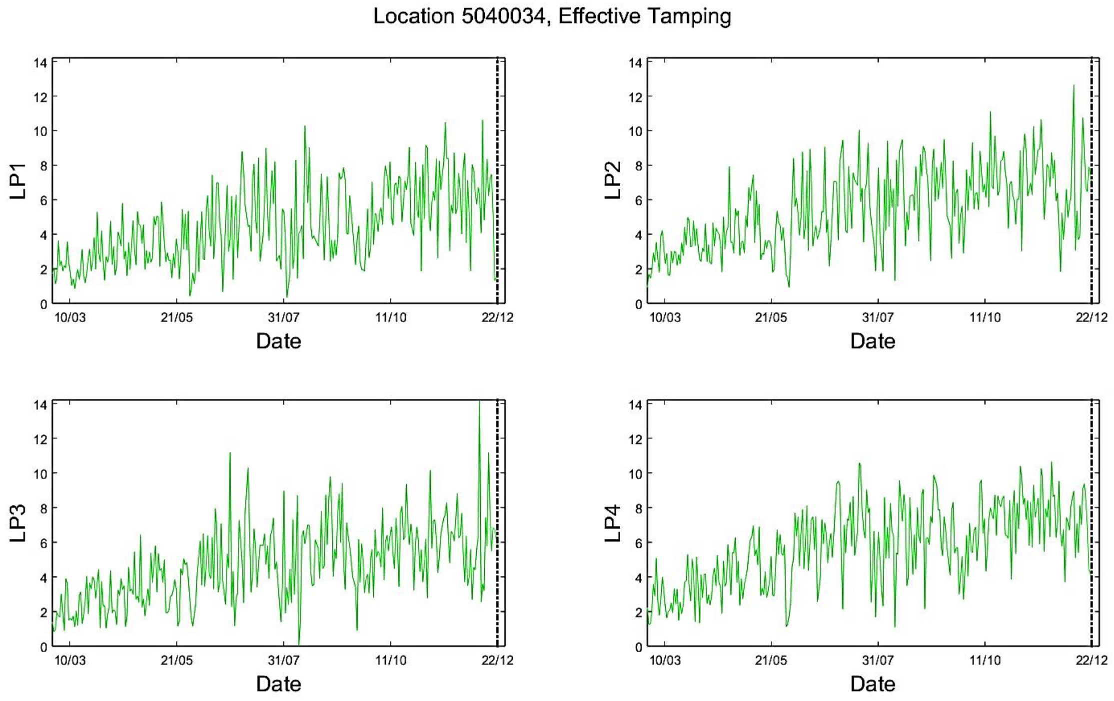


(a) Raw data from each trip
(b) Time series per location
(c) Dataset in the form of matrix

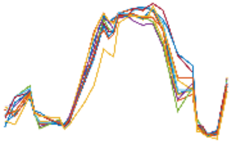
1. Data – IRVs Time Series

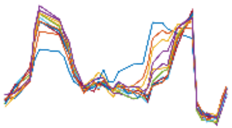
- 6 months multivariate sensor data from IRVs
- Multiple trips form a 4 dimensional time series for each track location
- Filters
  - Locations without tamping
  - Locations with less than 30 trips
- Pre-process
  - Linearly interpolate missing data
  - Label each location with tamping effectiveness
    - Effective and Ineffective
- Resample data to days with the average of the day

1. Data – IRVs Time Series

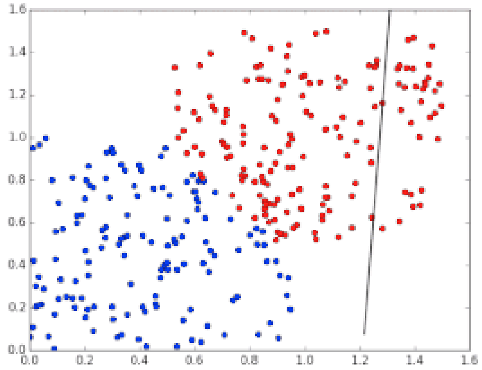


2. Train a Classifier





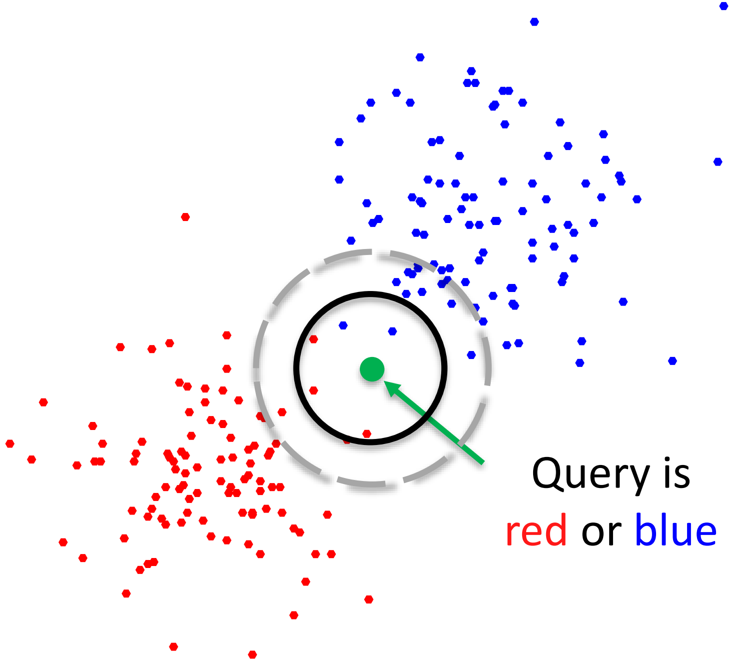
- Supervised machine learning
  - Train a classifier using labelled data
  - Training time
    - Learn the characteristics of the labelled data
  - Testing time
    - Label a query object using the learnt characteristics from the training data
- Time series classification
  - *k*-Nearest Neighbour (*k*-NN) Classifier



Hello

2. Train a Classifier – *k*-NN Classifier

- State-of-the-art for time series classification
  - Performed the best over a wide range of time series dataset
- Simple and effective algorithm that does not require training
- Outperforms others with large database size
- Search for the k^{th} most similar objects (time series)
- Labels the query using the majority label of the k^{th} nearest neighbours
- Uses a similarity measure

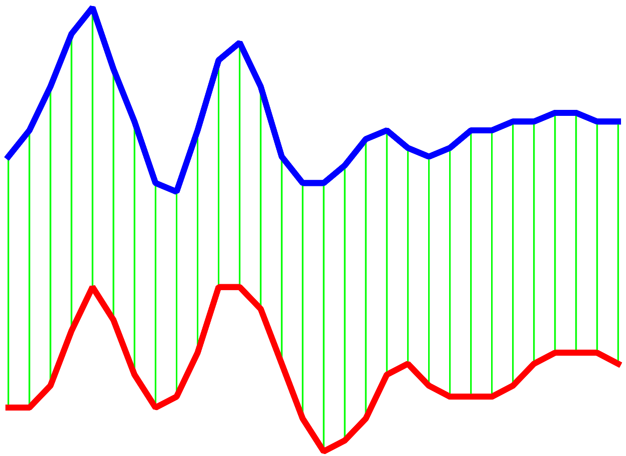


2. Train a Classifier – Comparing Time Series

- Calculate a distance between the two time series
- Simplest distance measure is Euclidean Distance (ED)

$$\mathrm{ED}(x, y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$

- Euclidean distance cannot handle distortions in time axis, different length
- Time series are often shifted in time (distortions) with different length

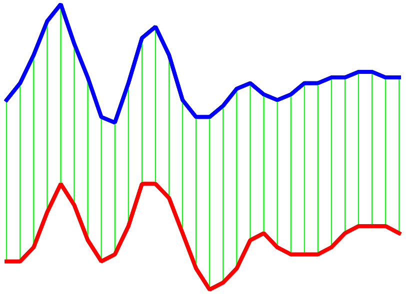


2. Train a Classifier – Dynamic Time Warping

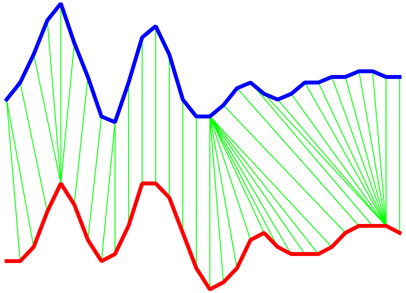
- Robust to time axis distortions and can handle different length
- Find alignment between two time series
- Given two time series *X* and *Y* of length *n* and *m*

$$\mathrm{DTW}(X,Y) = \mathrm{ED}(x_1,y_1) + \min\begin{pmatrix}\mathrm{DTW}\big(\mathrm{Tail}(X),\mathrm{Tail}(Y)\big)\\ \mathrm{DTW}(\mathrm{Tail}(X),Y)\\ \mathrm{DTW}\big(X,\mathrm{Tail}(Y)\big)\end{pmatrix}$$

- $\mathrm{Tail}(X) = \{x_2, x_3, \ldots, x_n\}$



(a)



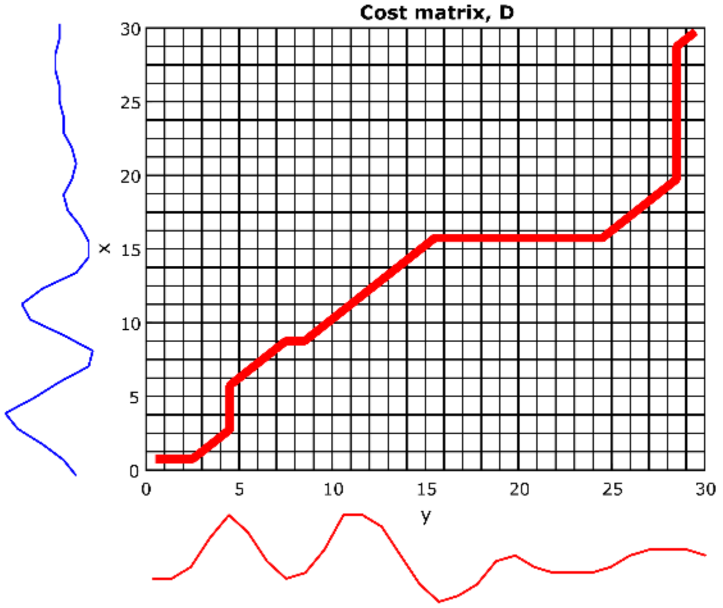
(b)

2. Train a Classifier – Dynamic Time Warping

- Dynamic Programming
- Constructs *n×m* cost matrix *D* and find a warping path (red) that minimises the alignment cost of the two time series

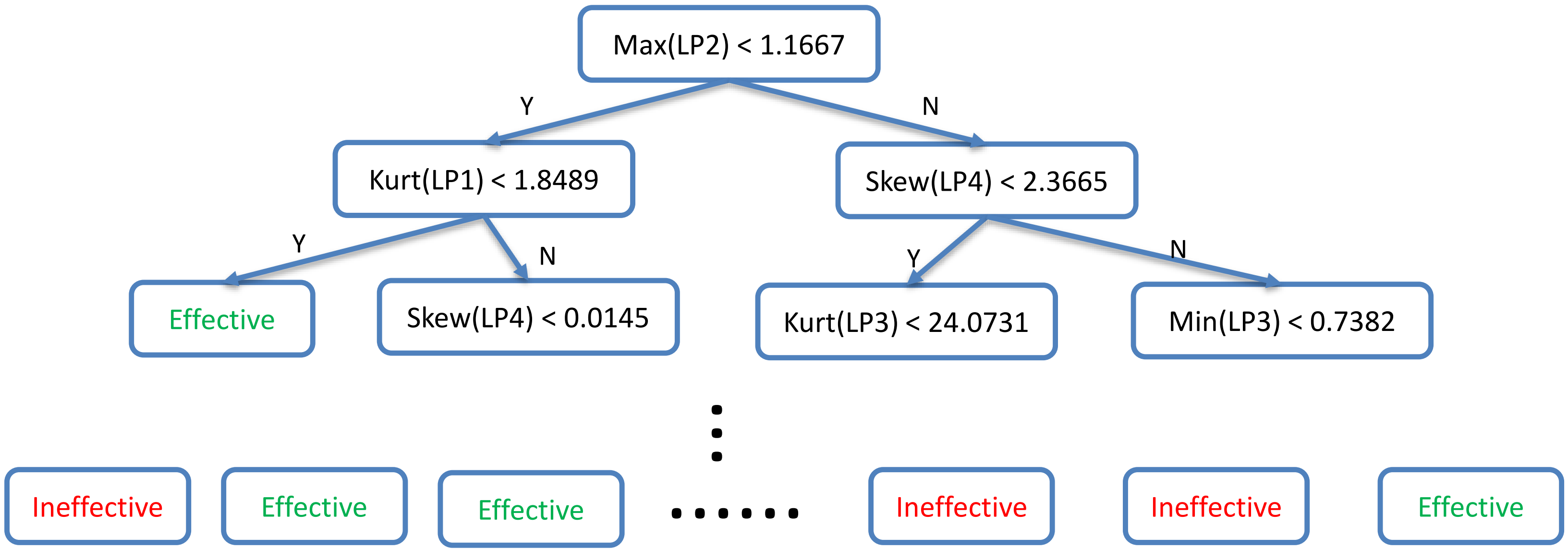
$$D(i,j) = \mathrm{ED}(x_i, y_j) + \min\begin{pmatrix} D(i-1, j-1) \\ D(i-1, j) \\ D(i, j-1) \end{pmatrix}$$

$$\mathrm{DTW}(X,Y) = D(n,m)$$



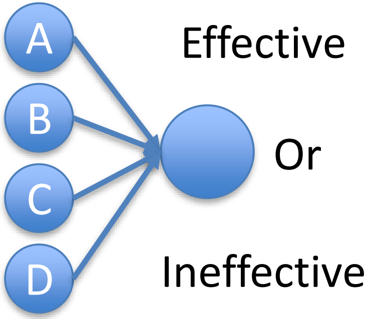
2. Train a Classifier - Classifiers

- Classification Tree (CART)
  - Build a decision tree to classify the query based on some rules (features)



2. Train a Classifier - Classifiers

- Naïve Bayes
  - Classifies the query based on the posterior of each class observed from the features of training database
  - Given a set of features, what is the probability of tamping being effective/ineffective?



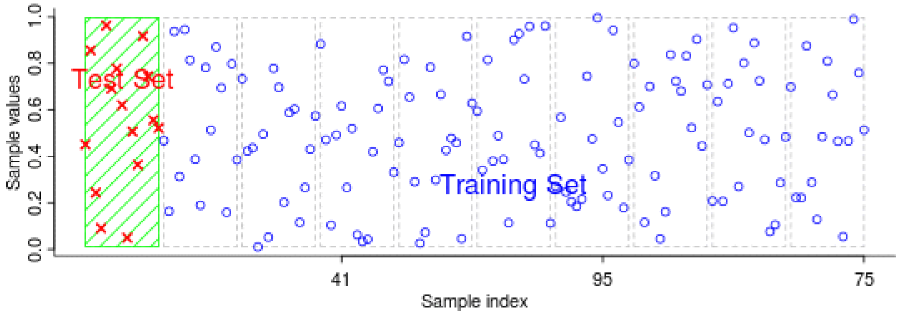
$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

$$\text{posterior(eff)} = \frac{\text{P(eff)p}(\mu \mid \text{eff})\text{p}(\sigma^2 \mid \text{eff}) \cdots \text{p(min} \mid \text{eff)}}{\text{P(eff)p}(\mu \mid \text{eff}) \cdots + \text{P(ineff)p}(\mu \mid \text{ineff}) \cdots}$$

3. Evaluation and Validation

- Binary classification
- Compute performance metric for both effective and ineffective tamping
  - Accuracy, $A = \frac{TP+TN}{TP+FP+TN+FN}$
  - Precision, $P = \frac{TP}{TP+FP}$
  - Sensitivity, $S = \frac{TP}{TP+FN}$
  - F_1 Score, $F_1 = 2 \cdot \frac{P \cdot S}{P+S}$
- 10-fold cross validation
  - Split and validate the training database

Actual\Predict	Effective tamping	Ineffective tamping
Effective tamping	True positive (TP)	False negative (FN)
Inffective tamping	False positive (FP)	True negative (TN)



4. Tamping Effectiveness Prediction with *k*-NN

Algorithm 11: ε = predict_effectiveness(Q, D)

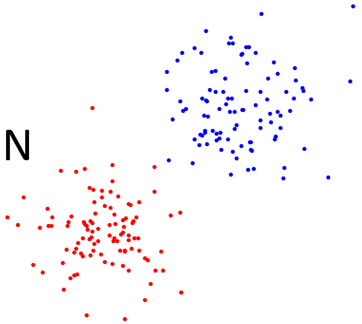
Input: Q: Query time series
Input: D: Training dataset
Output: ε: Tamping effectiveness

```
knn = Ø ;
knn.distance = ∞
for all C ∈ D do
    d = DTW(Q, C)
    if d < max(knn.distance) then
        remove the furthest neighbour from knn
        knn.add(C, d)
    end
end
return ε = mode(knn.class)
```

List for *k*-NN

Update the list if nearest than the furthest distance

Majority class



A. Selecting the best classifier

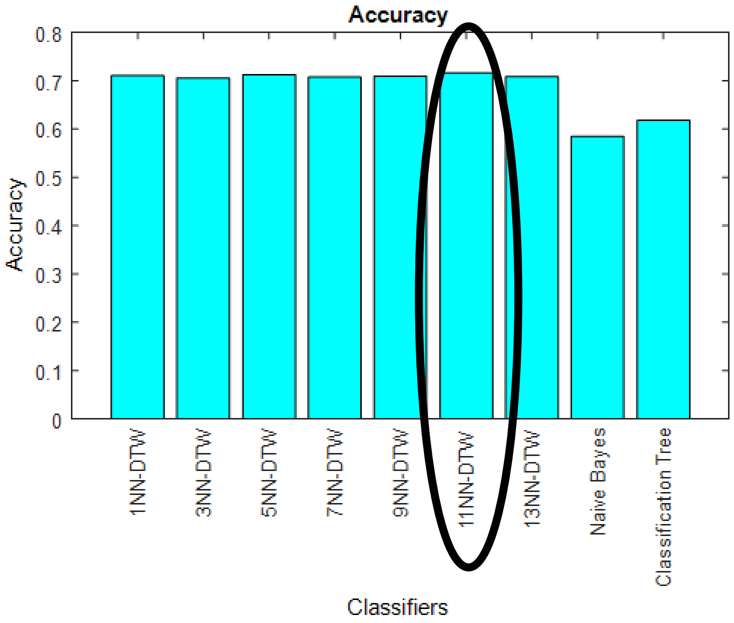
B. Early prediction of tamping effectiveness

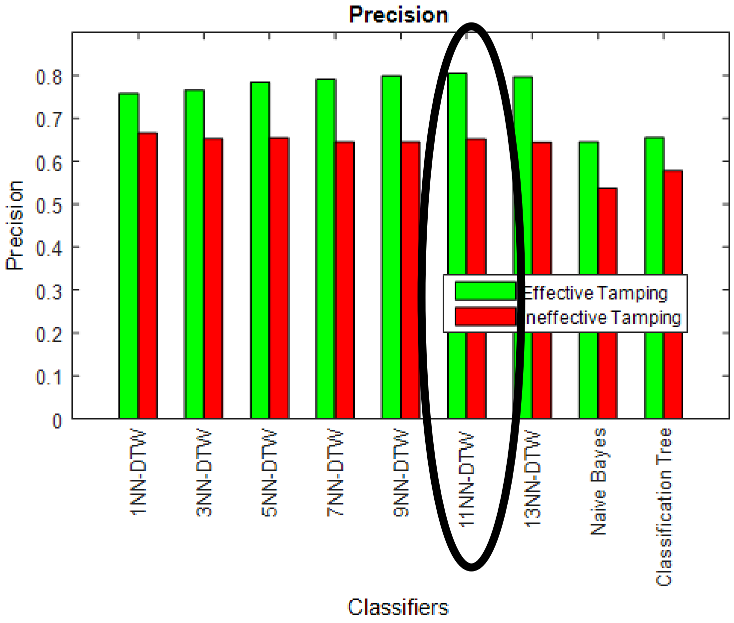
Results – Selecting the Best Classifier

- State-of-the-art *k*-NN works well for univariate time series
- Show that *k*-NN works well even for multivariate time series in predicting tamping effectiveness
- Compare with Naïve Bayes and Classification Tree
  - *k* = 1,3,5,7,9,11,13 for *k*-NN
  - 6 statistical features
    - Mean, Standard Deviation, Skewness, Kurtosis, Maximum and Minimum for Naïve Bayes and Classification Tree.

Results – Selecting the Best Classifier

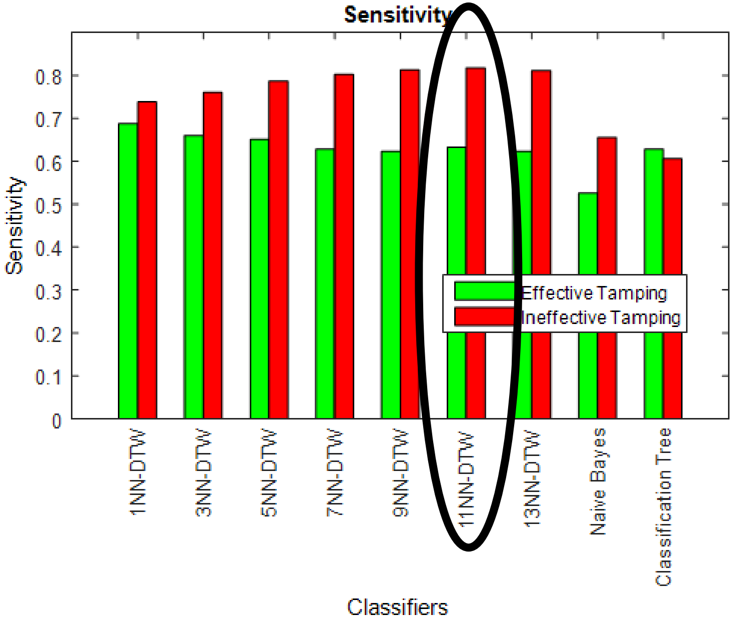
- 11-NN has highest accuracy than other two
- 11-NN has highest precision in predicting both effective and ineffective tamp than other two

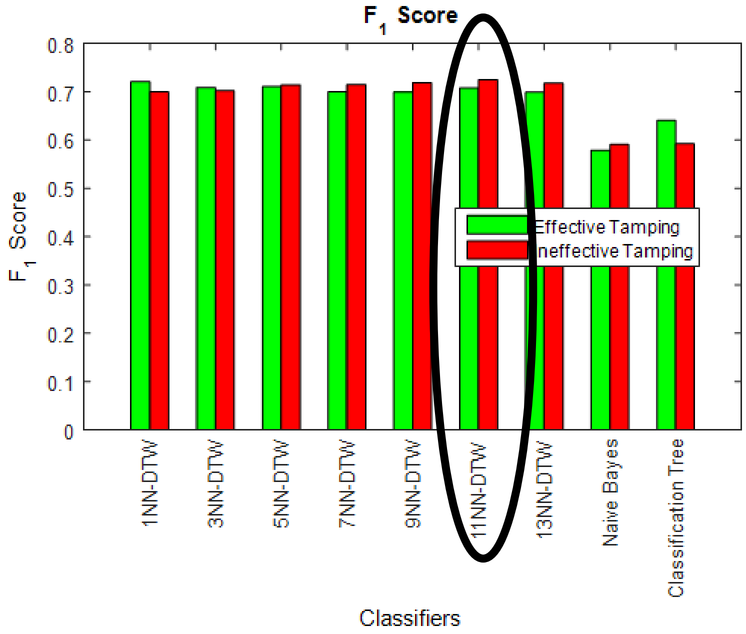




Results – Selecting the Best Classifier

- 11-NN has higher Sensitivity in predicting both effective and ineffective tamp than other two
- 11-NN has higher F_1 Score in predicting both effective and ineffective tamp than other two





Results – Selecting the Best Classifier (Why *k*-NN)

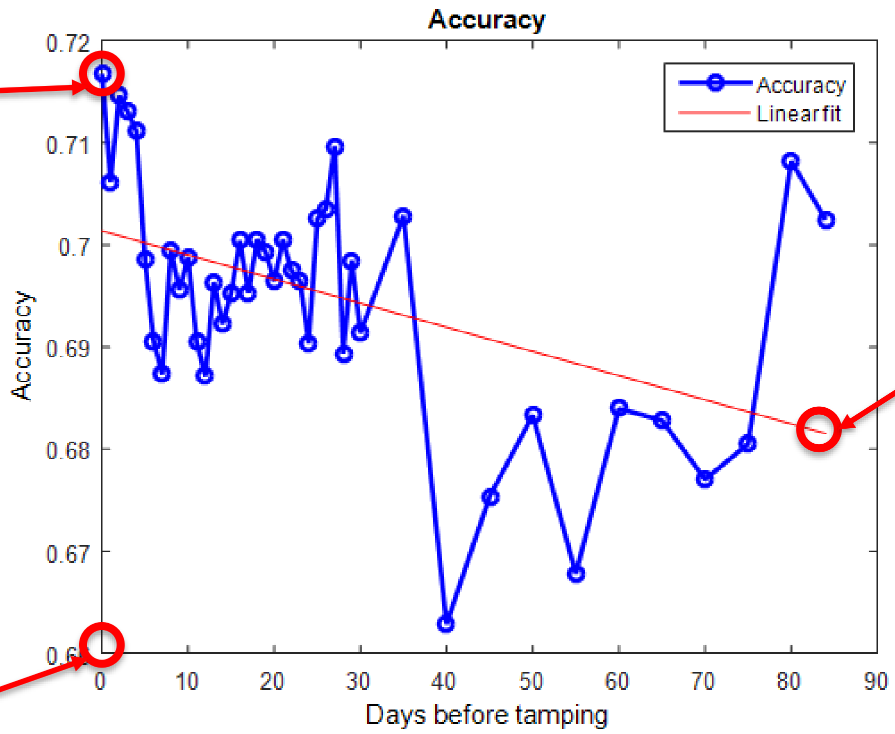
- Many algorithms uses static features
  - Naïve Bayes
  - Classification Trees
- In most applications, features change over time
- “Incorrect” to classify by observing only the static features
- *k*-NN will not be affected
- Select *k*-NN for the prediction

Results – Early Prediction of Tamping Effectiveness

- Predict tamping effectiveness as early as possible, typically 12 weeks
- The earlier we know about the effectiveness, the better the maintenance can be planned.
- Procedure
  1. Truncate the existing time series by the number of days before tamping
  2. Predict the tamping effectiveness of the query location with 11-NN-DTW
  3. Repeat with days ranging from 0 to 84 days (12 weeks)

Results – Early Prediction of Tamping Effectiveness

72%



66%

68%, Predicting tamping effectiveness 12 weeks (84 days) ahead

Conclusion and Future Work

- Present a tamping effectiveness prediction system using *k*-NN-DTW.
- Showed that 11-NN-DTW gives good prediction.
- The system is able to give good prediction 12 weeks before tamping.
- Significance
  - Improve efficiency of railway track maintenance
  - Can be used with other maintenance procedures
- Future work
  - Optimise with more data
  - Predicting an actual response
  - Other features and classifiers

Thank you!

IHHA 2017
CAPE TOWN
02 – 06 September 2017
11th International Heavy Haul Association Conference

Contact: Chang Wei Tan
Email : chang.tan@monash.edu
Web : www.monash.edu/irt