ETH*zürich*

DEPARTMENT OF CIVIL, ENVIRONMENTAL AND GEOMATIC ENGINEERING

Data driven identification and classification of rail surface defects

Aiyu Liu

Supervised by: Cyprien Hoelzl, Prof. Eleni Chatzi

Tuesday 4th February, 2020

Acknowledgements

This thesis would not be possible without the help of my supervisors, Cyprien Hoelzl and professor Eleni Chatzi.

Contents

Chapter 1

Introduction

1.1 Motivation

Railway companies need to continuously and sufficiently maintain the train and the train tracks and opti- mally predict the future defects in order to have a more punctual and more effective train system. However, the current systems are expensive, time consuming and ineffective.

1.2 Objective

1.3 Defects

Defect can be of any type, which defects do we want to focus on

insert pictures

1.4 Data

Data is provided by SBB

Chapter 2

Design and Implementation

First we need to analyse the data,

2.1 Peak windows

To find

2.2 Neural network architecture

Trained a neural network, although we were only able to achieve max Based on the analysis we

2.3 Visualisation

Chapter 3

Evaluation

3.1 Results

3.2 Discussion

Chapter 4

Conclusion and future work

4.1   Conclusion

4.2   Future work

-
-

4.3   TODO

- very fast speed, overlap between switch and ins, old vs new rail, ax1 arrow 2 arrow 3 arrow 4

- 3D plots?

- `pd.options.display.max_rows = 15`

whats this

```python
def conv(df):
    """
    has to be series
    """
    return np.vstack([v for v in df])


dup_ins = s_features.ins_joints.copy()[['accelerations']]
dup_swi = s_features.switches.copy()[['accelerations']]
dup_def = s_features.defects.copy()[['accelerations']]

dup_ins['accelerations'] = np.sum(conv(dup_ins.accelerations),1)
dup_swi['accelerations'] = np.sum(conv(dup_swi.accelerations),1)
dup_def['accelerations'] = np.sum(conv(dup_def.accelerations),1)

# s_features.ins_joints[['vehicle_speed(m/s)', 'Axle', 'campagin_ID']].duplicated() or

idx_ins = dup_ins.accelerations.duplicated()
idx_swi = dup_swi.accelerations.duplicated()
idx_def = dup_def.accelerations.duplicated()
new_ins = s_features.ins_joints[~idx_ins]
new_swi = s_features.switches[~idx_swi]
new_def = s_features.switches[~idx_def]

print("Duplcated samples: ", len(dup_ins) - len(new_ins))
print("Duplcated samples: ", len(dup_swi) - len(new_swi))
print("Duplcated samples: ", len(dup_def) - len(new_def))
```

test sample

```python
ii = pd.DataFrame([
    [np.array([1,2]),2],
    [np.array([1,2]),2],
    [np.array([1,2]),2]])


x = a
[u,I,J] = unique(x, 'rows', 'first')
hasDuplicates = size(u,1) < size(x,1)
ixDupRows = setdiff(1:size(x,1), I)
dupRowValues = x(ixDupRows,:)

s_features.ins_joints.timestamps[:2].duplicated()
```

Appendix A

Appendix

Include the src files?