Performance of different machine learning techniques for forecasting of particle accelerator interlocks

Status Update

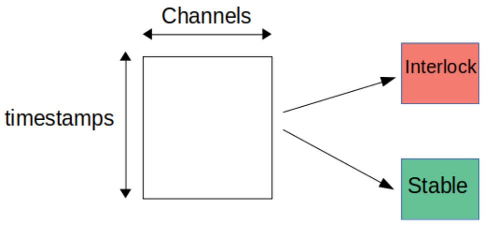
Mélissa Zacharias

February 24, 2020

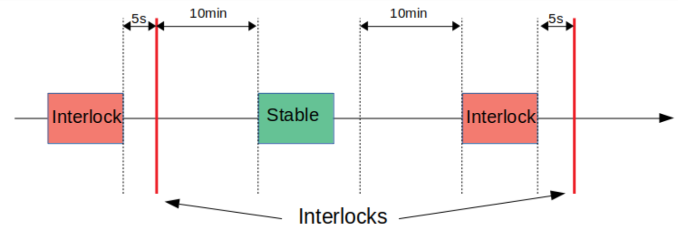
Formulating the problem

Classification approach: what gets classified?

"windows" of a multivariate timeseries

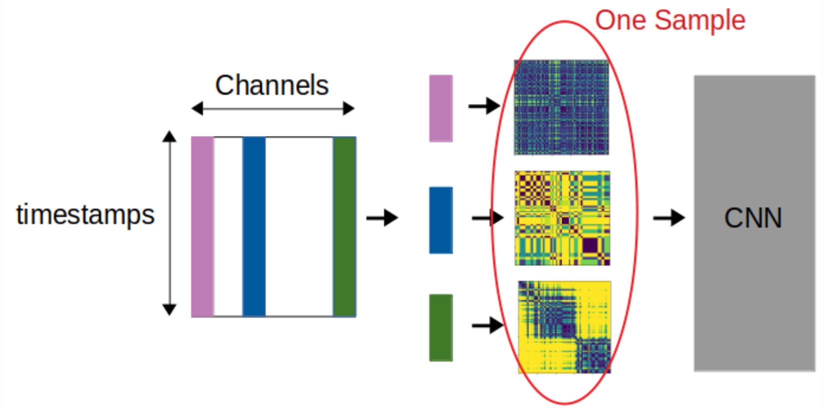


what are stable and interlock windows?



Input

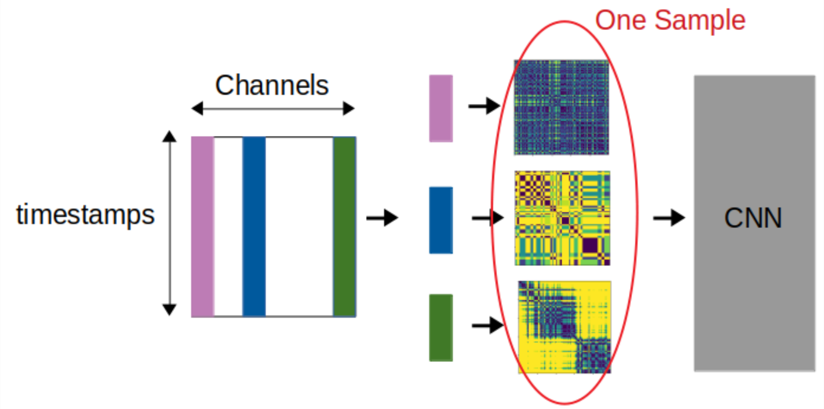
Recurrence Plots of the data windows



CNN

Input

Recurrence Plots of the data windows



Time Series Classification → Image Classification

Recurrence Plots

What is a recurrence plot(RP)?

Visualization of the recurrence of a state $\overrightarrow{x}_i$ in phase space.

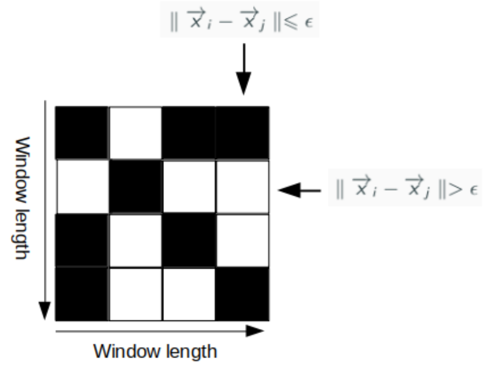
$$R_{i,j} = \Theta(\epsilon - \parallel \overrightarrow{x}_i - \overrightarrow{x}_j \parallel) \tag{1}$$

Recurrence Plots

What is a recurrence plot(RP)?

Visualization of the recurrence of a state $\overrightarrow{x}_i$ in phase space.

$$R_{i,j} = \Theta(\epsilon - \| \overrightarrow{x}_i - \overrightarrow{x}_j \|) \tag{1}$$

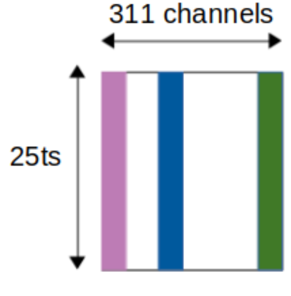


Architecture

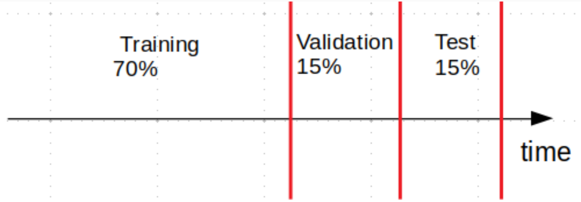
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
recurrence_plot (RecurrenceP (None, 25, 25, 311)       1
_________________________________________________________________
conv2d (Conv2D)              (None, 25, 25, 32)        89600
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 13, 13, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 7, 7, 64)          0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 7, 7, 32)          18464
_________________________________________________________________
dense (Dense)                (None, 7, 7, 128)         4224
_________________________________________________________________
global_max_pooling2d (Global (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 129
=================================================================
Total params: 130,914
Trainable params: 130,914
Non-trainable params: 0
```

CNN

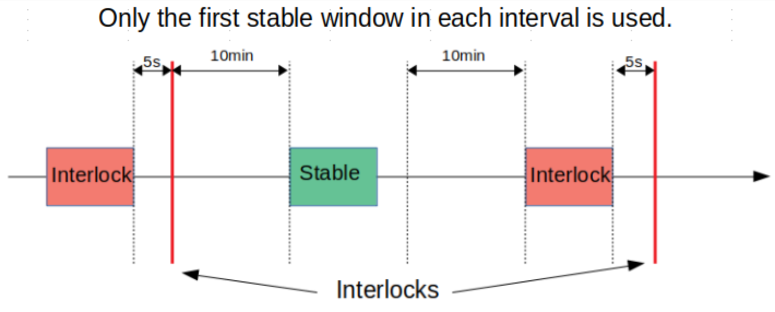
Input



Training validation and test splits



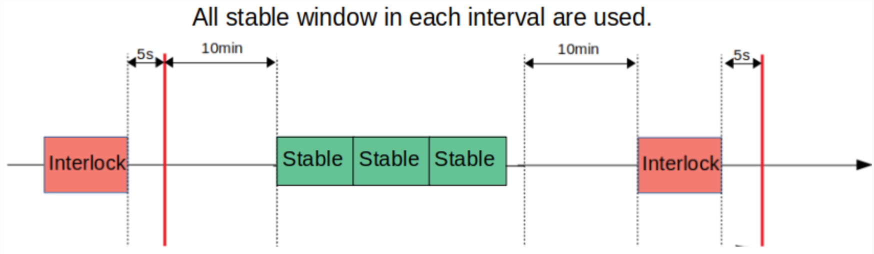
Balanced Data



Test results for model trained on balanced data and tested on balanced data:

	Predicted Stable	Predicted Interlock
True Stable	186	1
True Interlock	0	347

Real Distribution



Test results for model trained on balanced data and tested on data following the real distribution:

	Predicted Stable	Predicted Interlock
True Stable	64140	73459
True Interlock	0	345

GUI

GUI for live predictions!

Any Questions?

More Network Parameters

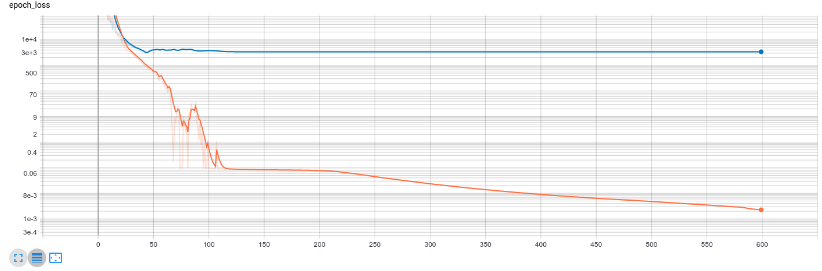
Optimizer Adam
Learning rate 0.01
Batchsize 2048
Loss binary crossentropy

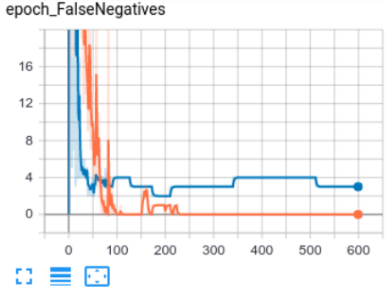
Sample sizes:

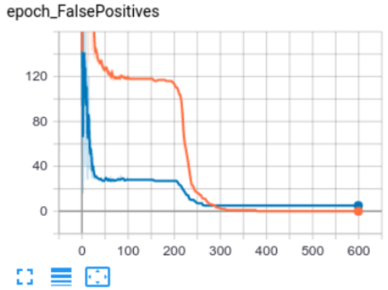
Balanced: Training: 802 Stables and 1669 Interlocks
Test: 187 Stables and 347 Interlocks

Real distribution:
Test: 137599 Stable and 345 Interlocks

loss function and metric of the presented model







Incentive

The Interlock system makes up $\sim 20\%$ of the total beam time loss

If interlocks can be predicted,we can prevent them

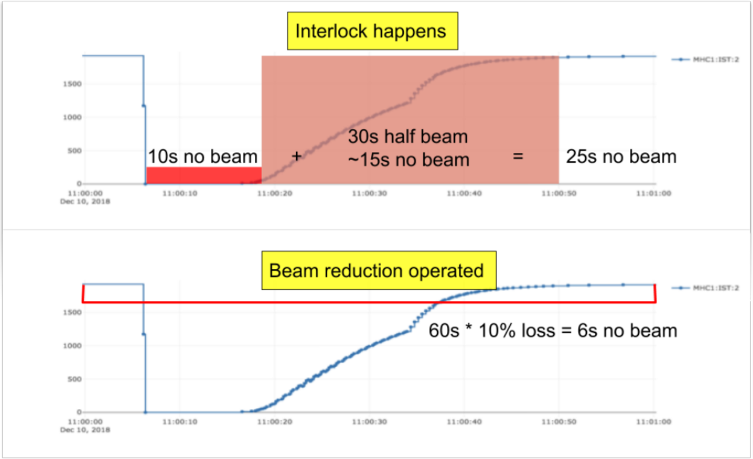


Figure by Li Sichen

Sanity Checks

Classification of Time-Series Images Using Deep Convolutional Neural Networks

Nima Hatami, Yann Gavet and Johan Debayle

Ecole Nationale Superieure des Mines de Saint-Etienne,
SPIN/LGF CNRS UMR 5307, 158 cours Fauriel, 42023 Saint-Etienne, France

ABSTRACT

Convolutional Neural Networks (CNN) has achieved a great success in image recognition task by automatically learning a hierarchical feature representation from raw data. While the majority of Time-Series Classification (TSC) literature is focused on 1D signals, this paper uses Recurrence Plots (RP) to transform time-series into 2D texture images and then take advantage of the deep CNN classifier. Image representation of time-series introduces different feature types that are not available for 1D signals, and therefore TSC can be treated as texture image recognition task. CNN model also allows learning different levels of representations together with a classifier, jointly and automatically. Therefore, using RP and CNN in a unified framework is expected to boost the recognition rate of TSC. Experimental results on the UCR time-series classification archive demonstrate competitive accuracy of the proposed approach, compared not only to the existing deep architectures, but also to the state-of-the art TSC algorithms.

Keywords: Convolutional Neural Networks (CNN), Time-Series Classification (TSC), Deep Learning, Recurrence Plots (RP)

Compare the CNN performance and recurrence plot algorithm to results of the paper

Sanity Checks

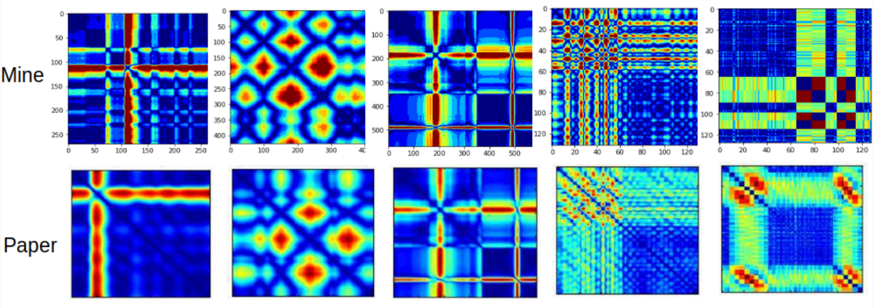
Datasets are available from the UCR archive.

Classification results (error rates)

	Paper	Mine	
Coffee	0	0.0	[[14 0] [0 13]]
Wafer	0	0.0055	[[636 29] [5 5493]]
ECG200	0	0.17	[[29 7] [10 53]]
GunPoint	0	0.154	[[62 12] [11 64]]
Lightning2	0	0.25 (needs more epochs)	[[13 15] [0 32]]
Yoga	0	0.40 (needs more epochs)	[[635 757] [465 1142]]

Sanity Checks

Recurrence plot comparison



binary_crossentropy:

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

y_i: label (0/1)

p(y_i): predicted probability that y_i is 1