Preliminary statistical time series features are extracted from vibration data captured by two sensors placed perpendicular to each other inside the machine. They are described as follows

Mean = [mean latex formula], Root Mean Square = [rms latex formula], Variance, Standard Deviation, Minimum Absolute Value, Maximum Absolute Value, Peak-to-Peak Value, Impulse Factor

We now propose a novel algorithm to construct a health indicator using the extracted features. This algorithm can be mathematically described in the following manner.

Given a preliminary feature signal **s(t)** for a specific bearing, the signal is first smoothed using the Savitzky-Golay filter. This filter is applied iteratively to reduce noise:

**SnFiltered(t)= SGFilter( s(t), window\_size, 2)**

Where:

**SnFiltered(t)** is the smoothed signal.

**SGFilter** denotes the Savitzky-Golay filter

**n** is the number of times the filter is applied to the raw signal

**2** is the polyOrder of the smoothing filter

here, the value of **window\_size** is dynamically calculated using the length of the signal

**window\_size = floor(N / preSmoothingFactor)**, where **preSmoothingFactor** is a tunable parameter

**N** is the size of the signal s(t)

The smoothed signal is then segmented into non-overlapping windows. Each window wi​ of size T is defined as:

**Wi(t)= SnFiltered(ti:ti+T)**

Where:

**ti** is the starting index of the ith window.

**T = max(10, N / windowFactor)**, is the window length and **windowFactor** is a tunable parameter

For each window **wi**, the local trend is computed using linear regression optimized using OLS

**Wireg​(t)=βi​t+αi​**

Where **βi​** is the slope and **αi​** is the intercept of the regression line

the signal within this window bounds is now recalculated as follows

if the slope **βi**is negative and is not the first calculated window, then the adjusted signal is described as

**Wirecalculated(t) = Wi(t); βi >= 0,**

**Wi(t) / ( Wireg(t) / elevationFactor ); else**

Where **elevationFactor** is a tunable parameter

Further, the Savitzky-Golay filter is iteratively applied again to this window to minimize any noise that is magnified by the window readjustment computation

**Wifiltered(t) = SGFilter( Wirecalculated(t), window\_size, 2) ; if postSmoothingFactor > 0 and βi < βthresh**

**Wirecalculated(t) ; else**

Where:

**Window\_size = floor( Wirecalculated(t) / postSmoothingFactor )**

**postSmoothingFactor** and **βthresh** are tunable parameters

**2** is the polyOrder of the smoothing filter

Finally, if the recalculated window was not the first window, then the signal within bounds is vertically readjusted to keep the signal continuous and is described as follows

**sireconstructed(t) = Wifiltered(t) + ( sireconstructed(ti-1 + T - 1) - Wifiltered(t0) ) ; if i > 0**

**Wifiltered(t) ; if i = 0**

Here,

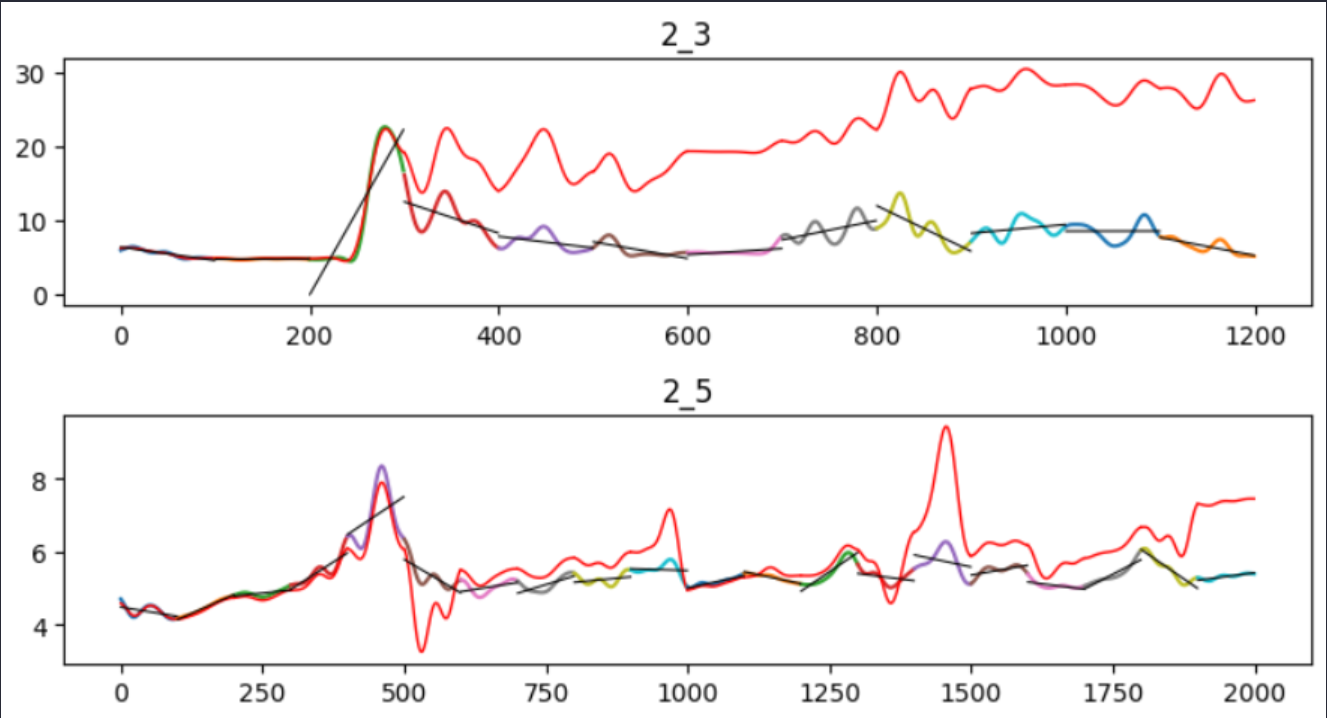
**sireconstructed(ti-1 + T - 1)** is the value of the last datapoint of the previous window signal

**Wifiltered(t0)** is the first datapoint of the current signal window signal

some considerations kept in mind while setting tunable parameters

Smaller **preSmoothingFactor** values result in large window sizes, and do not expressing trends in a desirable fashion. Over a large window, a signal can have both increasing and decreasing trends, which can result in a flat regression slope giving us no information at all.

Larger **preSmoothingFactor** values result in smaller window sizes. However, very small windows over-emphasize trends and end up amplifying noise instead. This also results in suboptimal computation times, justifying the tried and tested constant 10 when calculating T (above)



After entensive testing, the final tunable parameters used for this algorithm are as follows

*preSmoothingFactor* = 45

*windowFactor* = 100

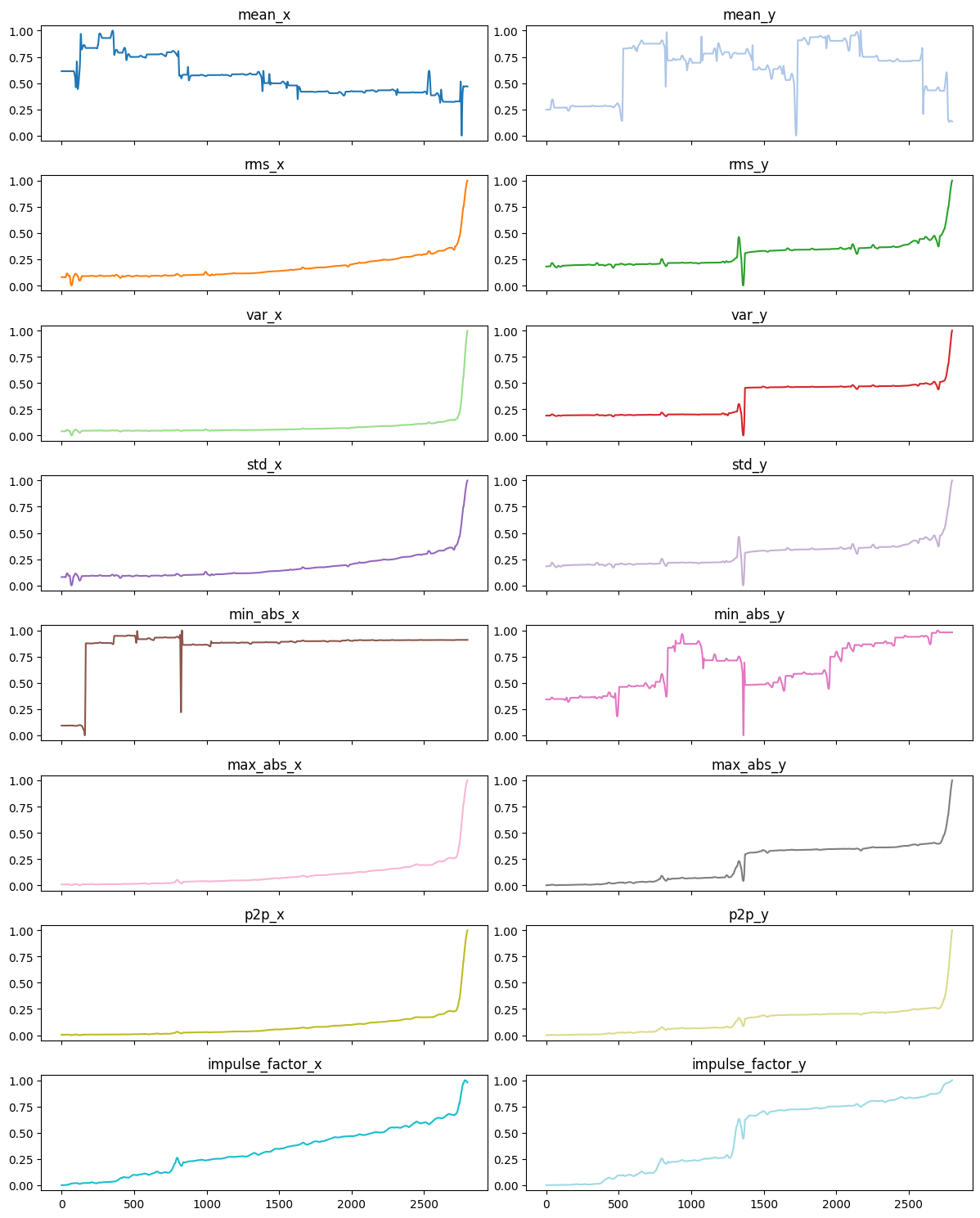
*slopeThreshold,* βthresh = 0.4

*elevationFactor* = 20.1

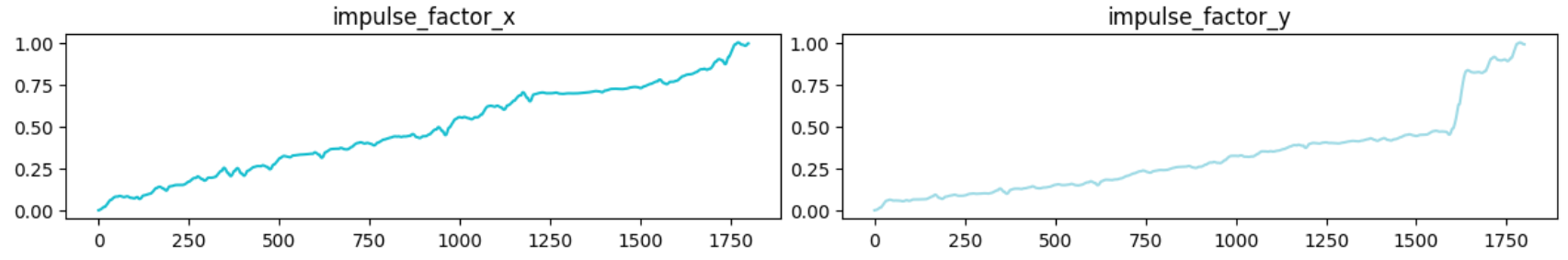
*postSmoothingFactor* = 3

This algorithm is tested on all extracted statistical time series features. The results on the ball bearing 1\_1 are as follows

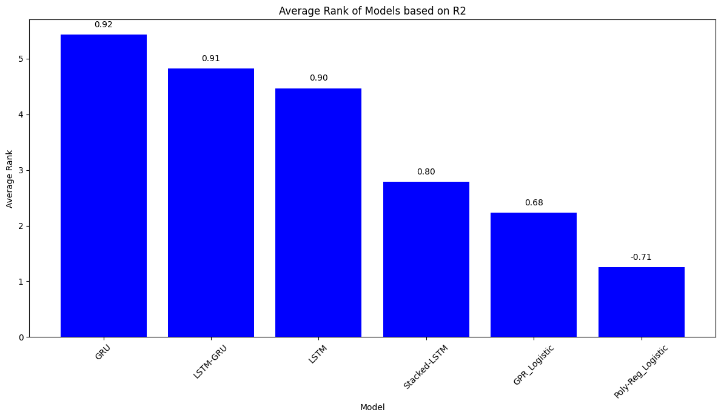
Here, we can notice that impulse factor shows us a relatively higher promise in trend for constructing the degradation index of ball bearing 1\_1. A similar pattern is observed across all ball bearings. There are instances where data from the latter sensor is not as reliable as the former. Taking this into account, we will only consider the first sensor data for further experiments.

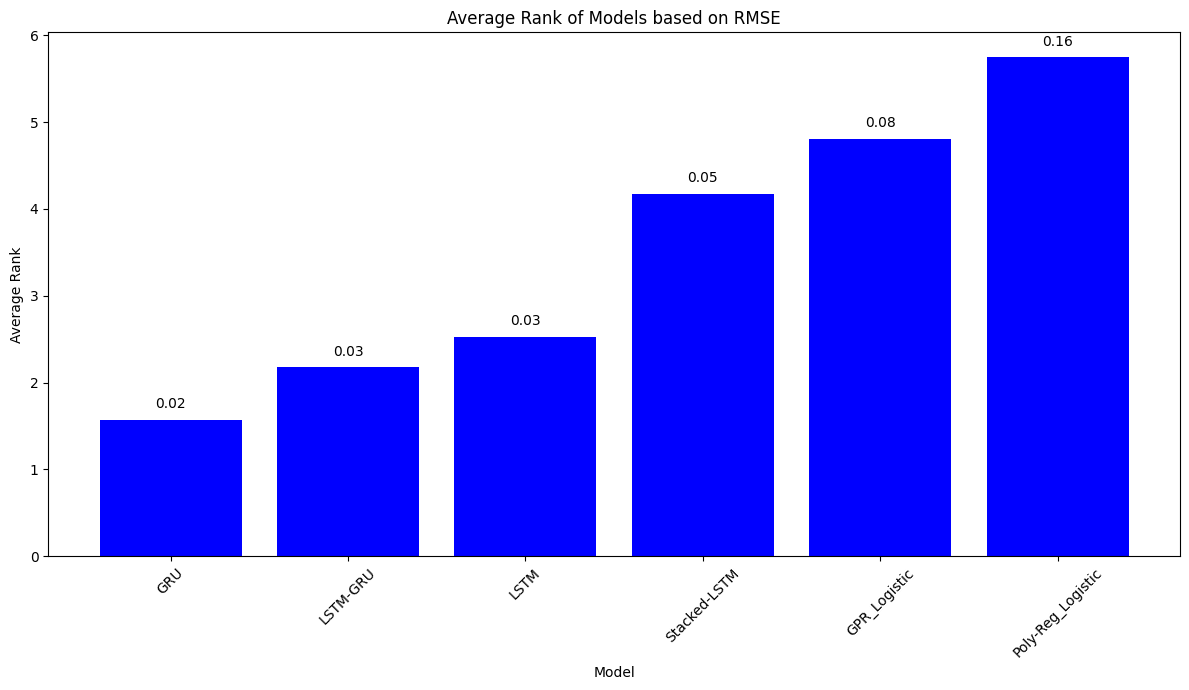


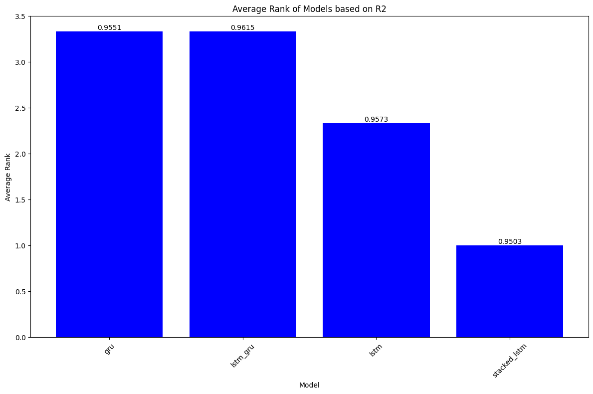
Best Degradation index produced by Impulse factor using SASD

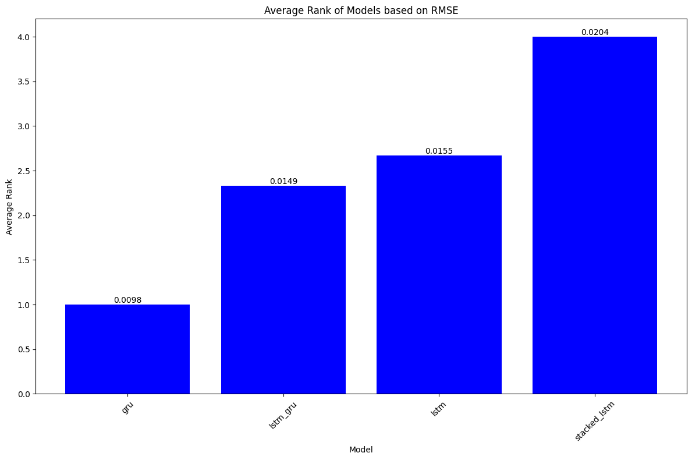


Y is Not reliable

Results (Pronostia RUL Prediction)



Results (Pronostia Generalization)



Note: Numbers above bars are the average metrics (RMSE / R2)