

Issues with the Signal Boosting framework of multiplying multiple signals which have been normalised to a scale off 0 to 1:

1. When multiplying the 1-0 normalised signals, you lose the statistical meaning behind the Z-scores (Gaussian distribution).
2. It suffers from an insane amount of alpha decay.

Fix to issue 1.

When normalising the signals via the Z-score (as previously done) we already have everything needed to perform a signal boosting operation. Simply set all values below 0 to 0 and multiply the positive values remaining.

For the bottom signal, replace values above 0 with 0 and perform the multiplicative aggregation with an absolute value operation around the signal (to get positive numbers only).

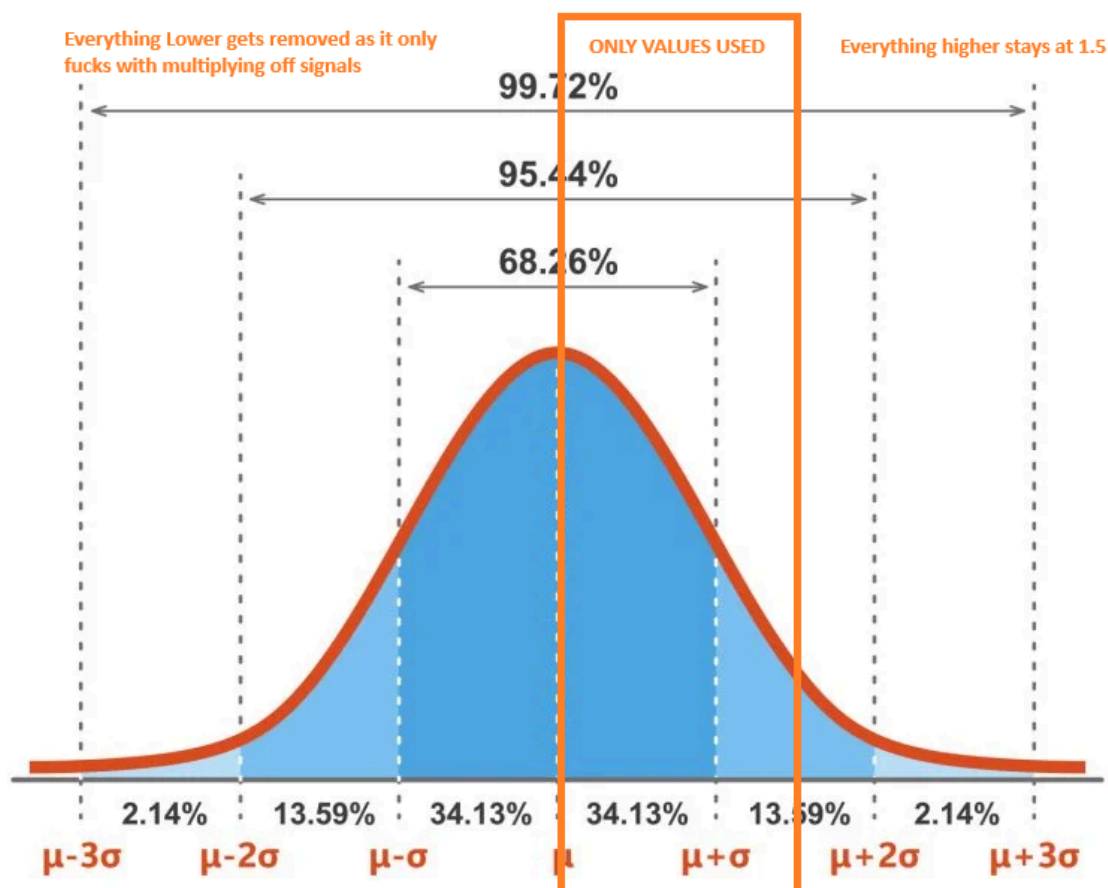
Potential fix to issue 2.

Our goal now is to make the signal as even as possible over time and different peaks off the market, bottom signals stay consistent and you can use the same process I will describe for the top signals as well.

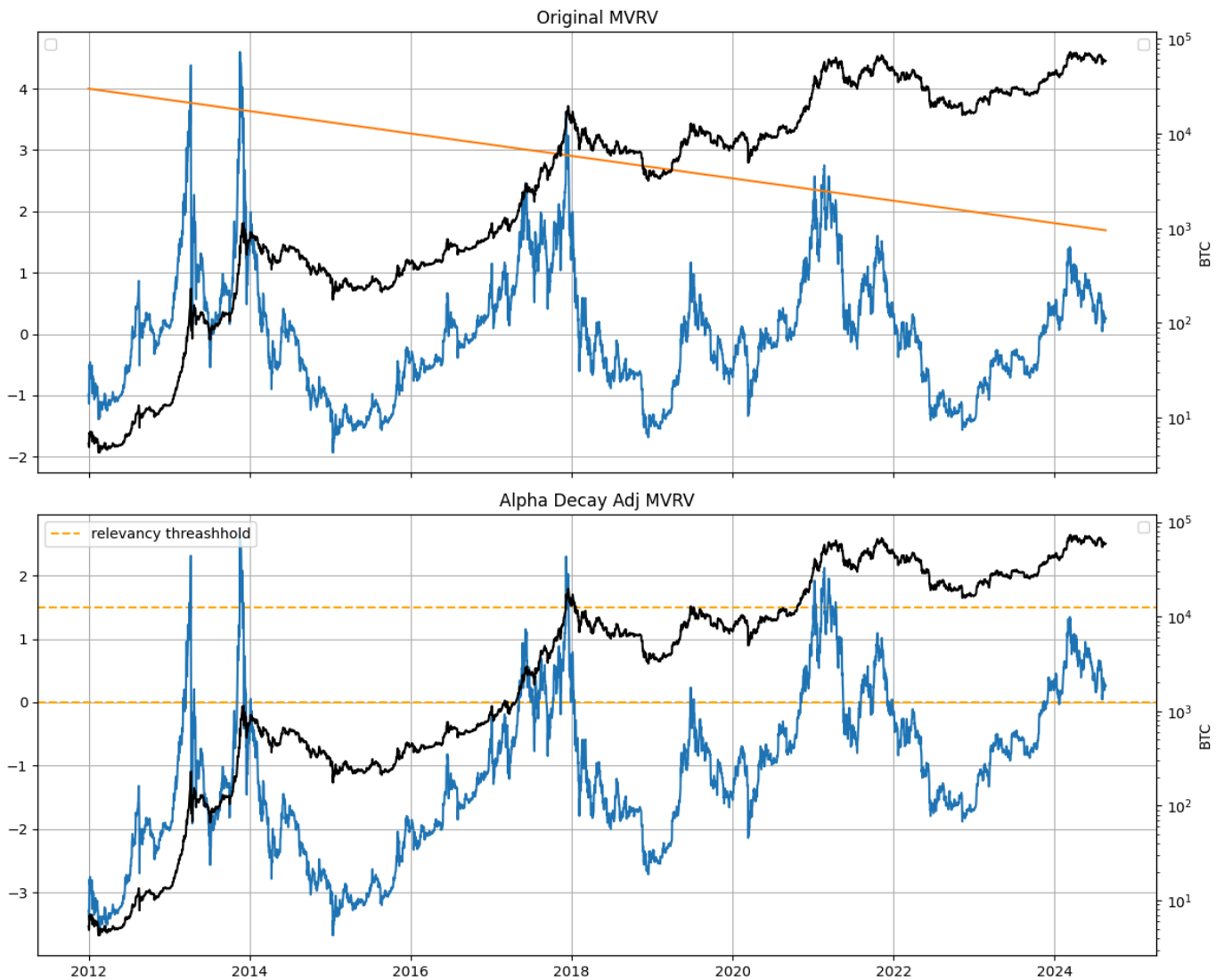
The first order of business is to not use standard deviations above a certain threshold in order to mute earlier volatile signals but still have a good signal, 1.5 std. is what I have chosen. (IMG1)

The second way is to manually adjust all signals to alpha decay by finding a linear line (multivariable and power law also works) that describes the alpha decay and then subtracting the signal by the values off the line. (IMG2)

\*Once this is done, the multiplication of your signals *should* give an amazing signal for tops and bottoms.



In the **upper** part of the image, the adjustment line is found



In the **lower** part of the Image the MVRV has been subtracted by the value of the line and the relevant numbers are within the orange line, removing numbers below and above should not be an Issue.

The **result** is a good signal, the only issue is that at the first top the top indicators didn't always line up and contaminated the signal.

The Top signal has 10 indicators, the Bot has 8.

