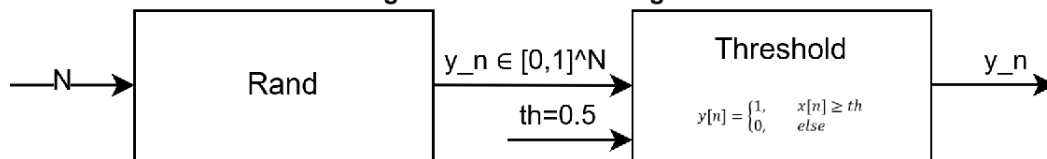

Table of Contents

.....	1
Section (A) - Adding new utils	1
GreaterThan block	5
Section (B) - Implementing BinaryClassifier	6
Section (C)	7

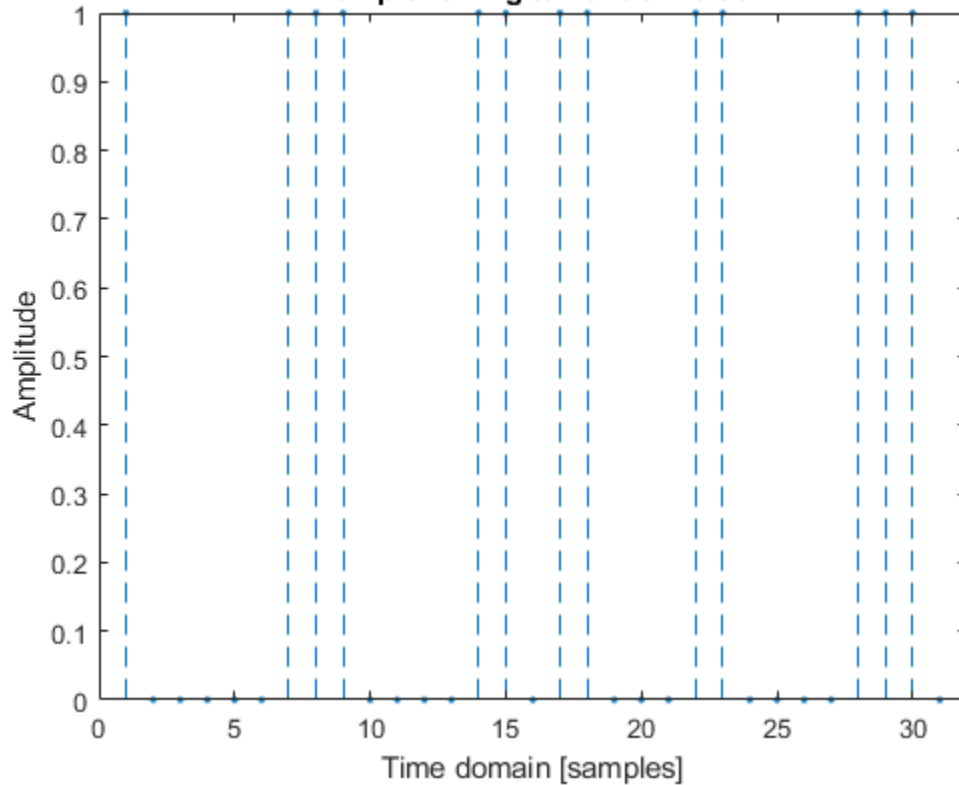
Section (A) - Adding new utils

We have implemented new utils using blocks from previous homeworks. Here are the diagram blocks and examples:

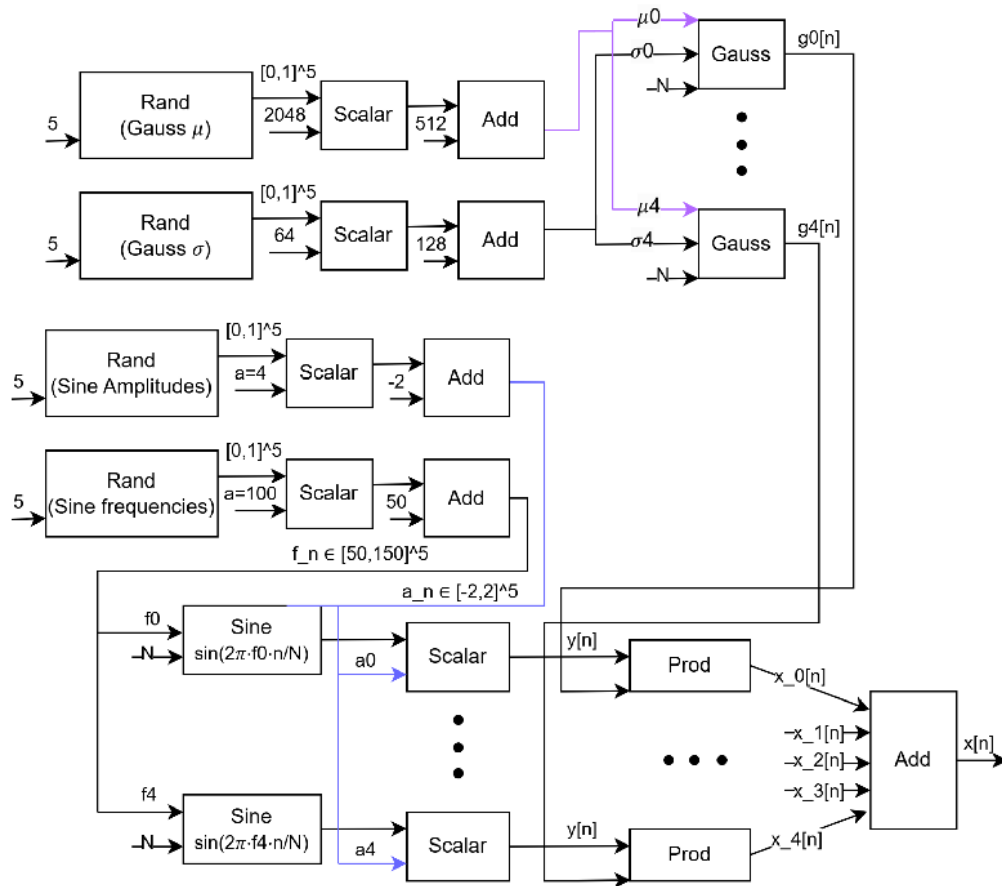
DigitalRandom block diagram



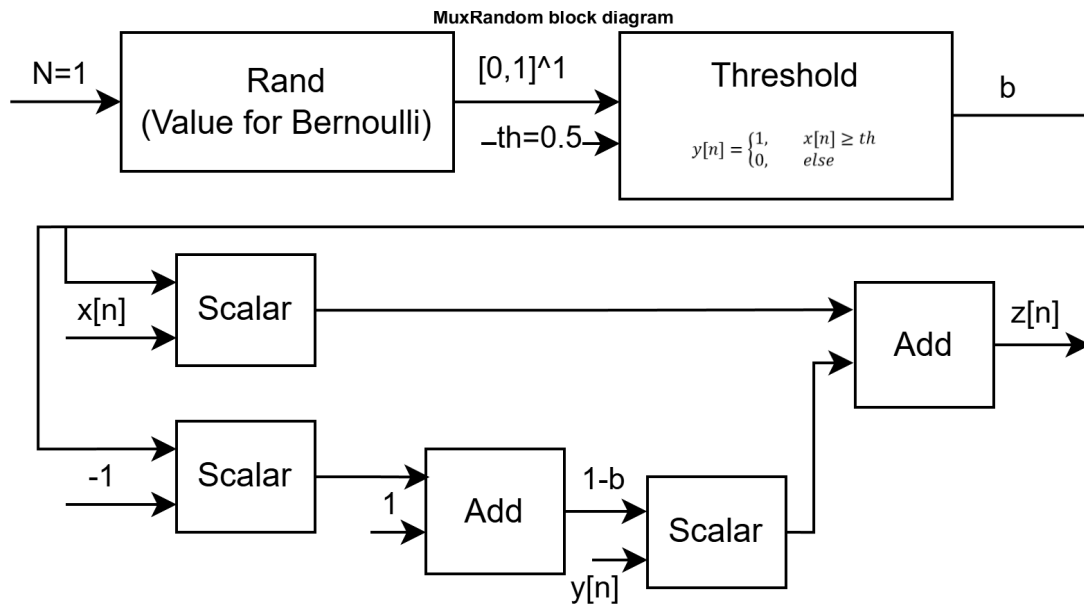
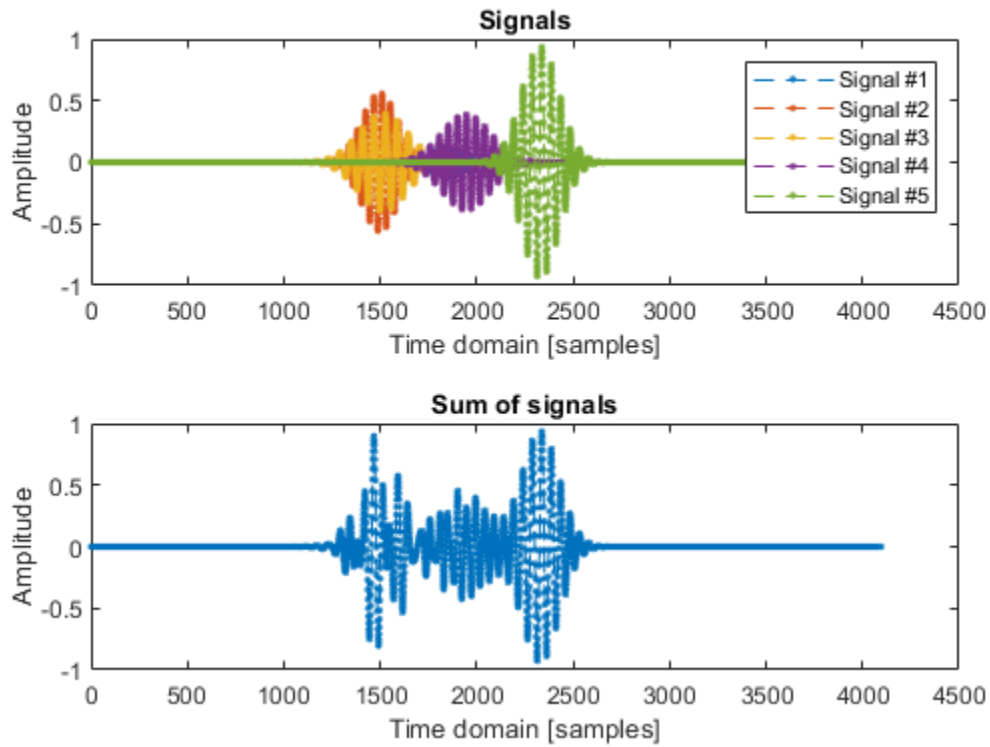
Example for DigitalRandom block



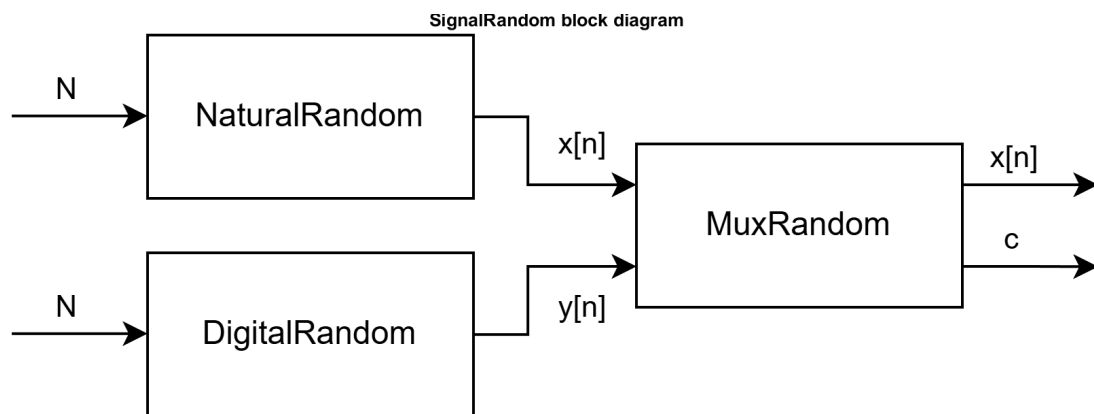
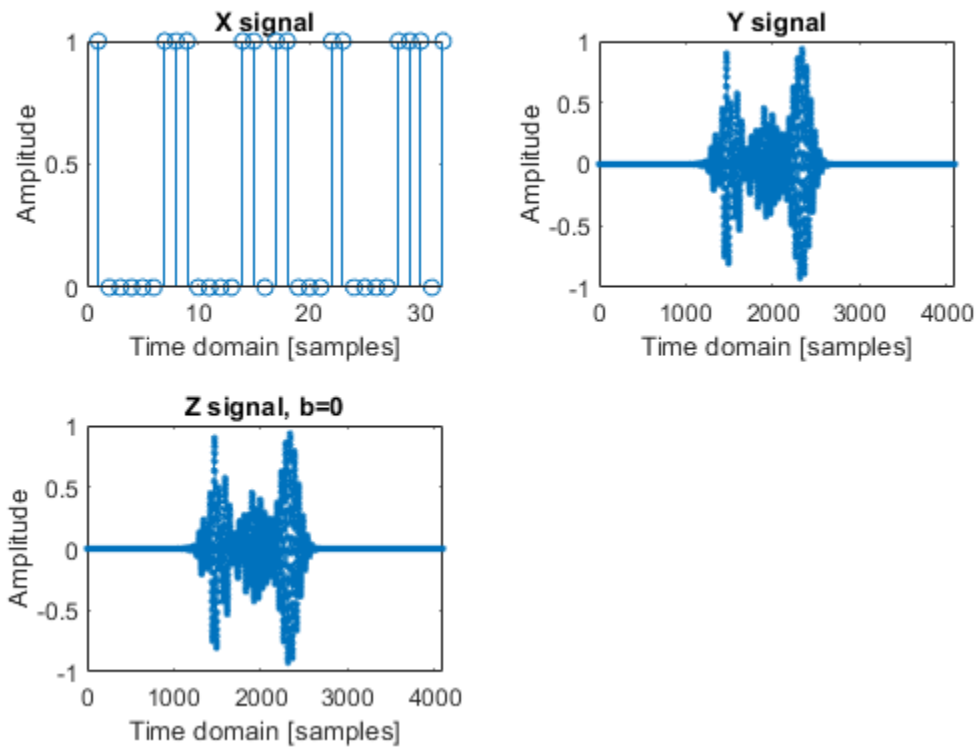
NaturalRandom block diagram

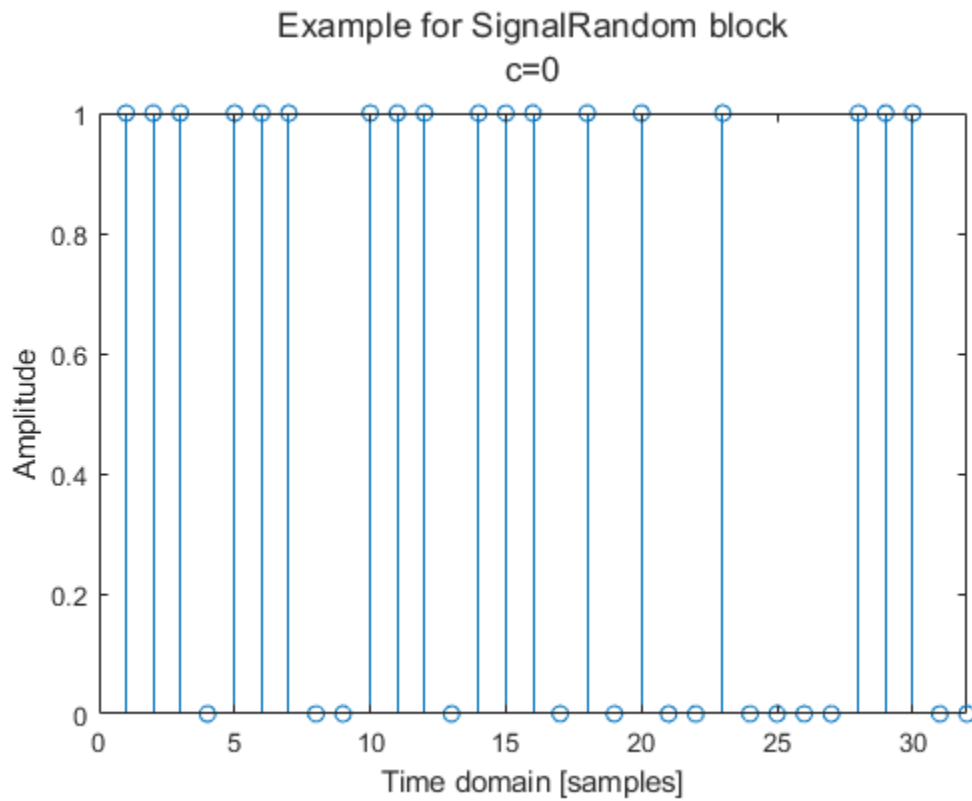


Example for NaturalRandom block



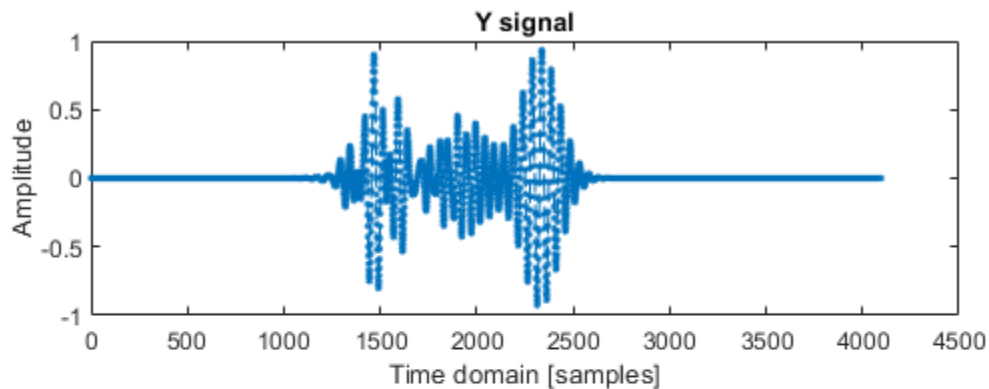
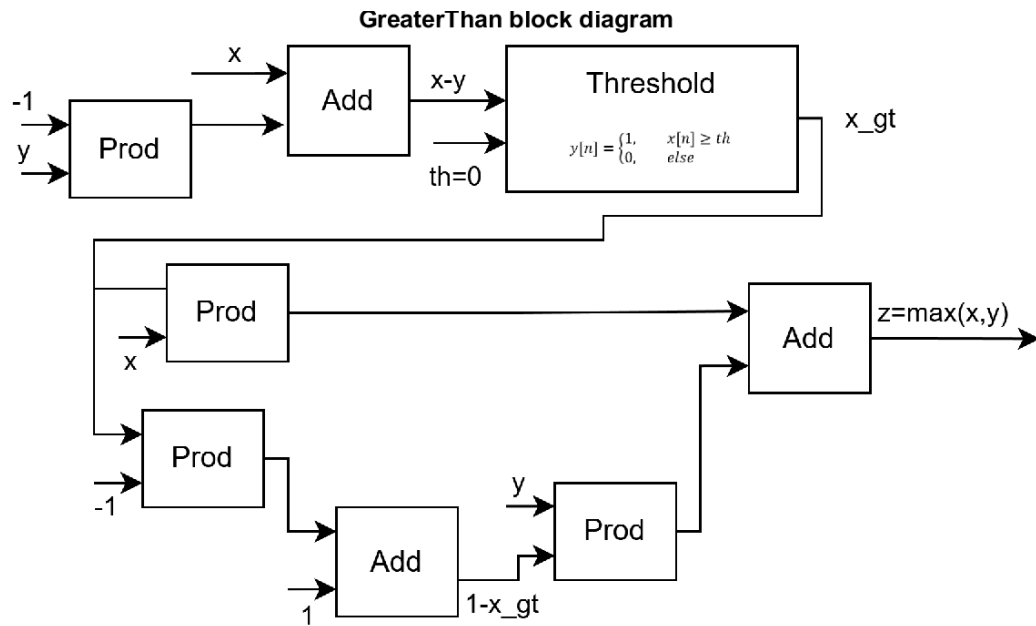
Example for MuxRandom block





GreaterThan block

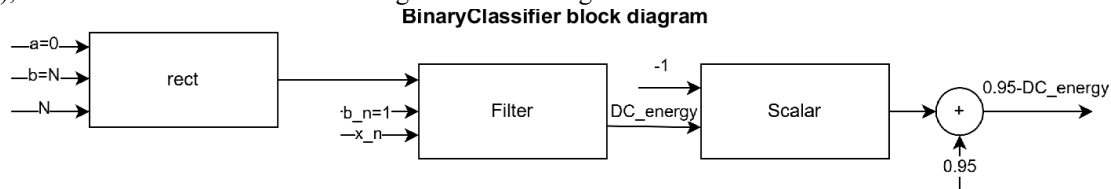
As Example for GreaterThan block, we select 2 signals - Natural signal from Natural signal block and signal of zeros (Using Sine block with DC frequency). We expect that applying GreaterThan block will return only positive values, the negative one will be replaced with zeros.



Section (B) - Implementing BinaryClassifier

As we can see, the goal is to classify two types of signals: digital and natural. Each type of signal has distinct characteristics, some of which can be helpful in the classification process. One key difference between these signal

types is that digital signals contain a DC frequency component, whereas natural signals do not. The DC energy tends to the mean of the digital symbols which equals to 0.5. For the natural signals there is no DC because we choose the sine frequency and gaussian s.t. the mean over time equals to zero. Considering the constraints above, we offer the following classifier - if the signals DC energy is above 0.5 - it's of digital type otherwise it's natural. The mean of the signal can be approximated using the DC base function of Haar wavelet, which can be defined using Rect block from HW1 and applied using Filter function (it's equivalent to FIR filter - $\text{ones}(1, N)/N$ where N is the input signal length). The logic is opposite to the desired threshold defined in the question ($c \geq 0.5 \rightarrow$ natural otherwise digital), so if we look at $1 - DC_{energy}$ we get the desired logic.



Section (C)

We generate 30 random signals belonging to two categories: natural and digital. Each signal is subjected to one of three noise levels:

- No noise
- $\sigma^2 = 0.1^2$
- $\sigma^2 = 0.5^2$

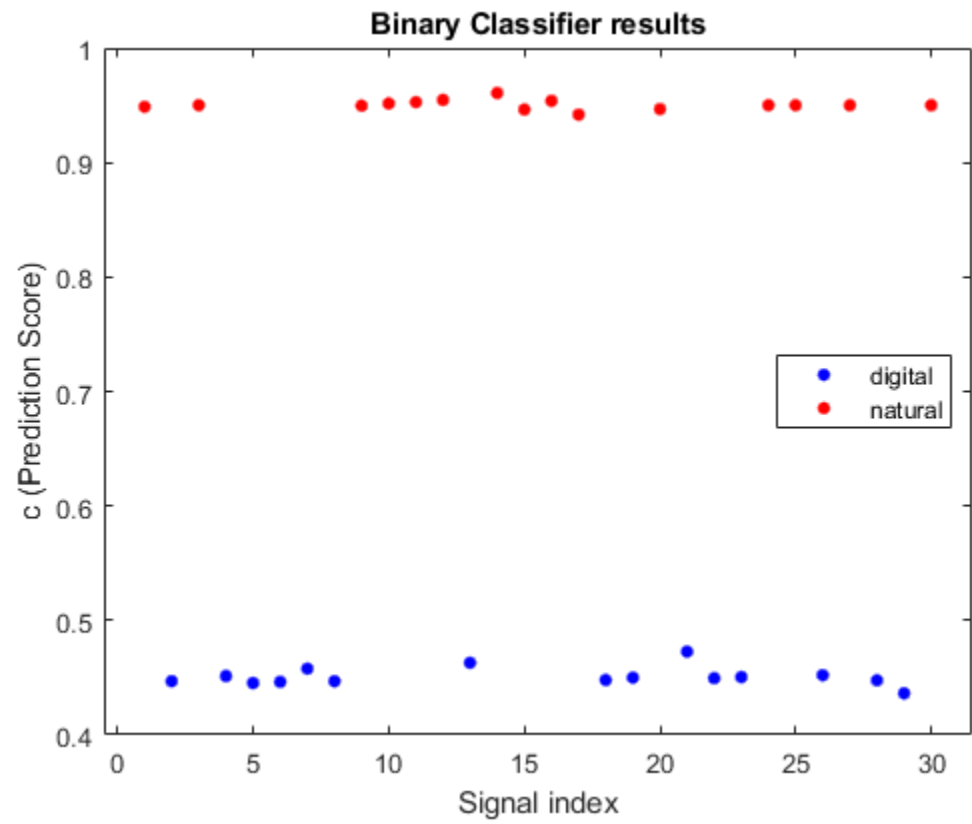
Each signal is then processed by a BinaryClassifier, which predicts the signal type and outputs a classification score c :

- $c \leq 0.5$ - Natural signal
- $c > 0.5$ - Digital signal

Finally, we visualize the classification scores against the signal index, using color to distinguish between signal types:

- Red - Natural signals
- Blues - Digital signals

The results demonstrate that the DC component, extracted using the Haar wavelet, is an effective feature for signal type prediction. The BinaryClassifier achieves perfect classification accuracy.



Published with MATLAB® R2024a