|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Number of trees | Root Node | Node 1 | | Node 2 | | | |
| 1 | 67.75 | 27.9 | 140.5 | 18.5 | 93.5 | 34.5 | 382.0 |
| 2 | 94.98 | 78.5 | 281.5 | 48.5 | 134.5 | 166.5 | 363.9 |
| 3 | 90.5 | 28.5 | 336.5 | 18.5 | 58.5 | 137.5 | 740.5 |
| 4 | 140.5 | 112.7 | 359.5 | 26.5 | 512.7 | 247.0 | 294.5 |
| 5 | 118.5 | 92.5 | 217.5 | 51.28 | 42.5 | 216.5 | 281.5 |

**Random Forest trained coefficients:**

**Random Forest Accuracy vs. Latency vs. Max depth:**

|  |  |  |  |
| --- | --- | --- | --- |
| Number of trees | Accuracy  (%) | FPGA Latency (ns) | ARM Cortex  Latency (ns) |
| 1 | 82.71 | 3.00 | **0.0029** |
| 2 | 83.00 | 3.18 | **0.0051** |
| 3 | 83.28 | 3.24 | **0.0087** |
| 4 | 83.28 | 3.29 | **0.0016** |
| 5 | 84.13 | 3.32 | **0.0145** |

|  |
| --- |
| **Algorithm1** Modified Random Forest Algorithm |
| **Input:** Cp-close prices; Hp-High prices; Lp -High prices; N-Total number of Test Data; TrainCi-Trained Coefficients; J – tree number; FX-Input Feature; respectively X =1,2  **Output:** D signal  **for** i = 0 to N **do**  F1(i) = Hp (i) - Cp (i);  F1(i)  = F1(i) << 4;    F2(i) = Cp (i) - Lp (i);  F2(i)  = F2(i)<< 4;  F = [F1i, F2i];  **for** c = 0 to J **do**  **if**F (1) == TrainC1 **then**  if F (2) <= TrainC2  if F (2) <= TrainC3  V(i) -= 1;  else  V(i) += 1;  end if  else  if F (1) <= TrainC4  V(i) += 1;  else  V(i) -= 1;  end if  end if  else  if F (2) <= TrainC5  if F (1) <= TrainC6  V(i) -= 1;  else  V(i) -= 1;  end if  else if  if F (1) <= TrainC7  V(i) += 1;  else  V(i) -= 1;  end if  end if  end if  end for  **if** V(i) > 0  D(i) = 1;  **else**  D(i) = 0;  **end**  end for |
|  |