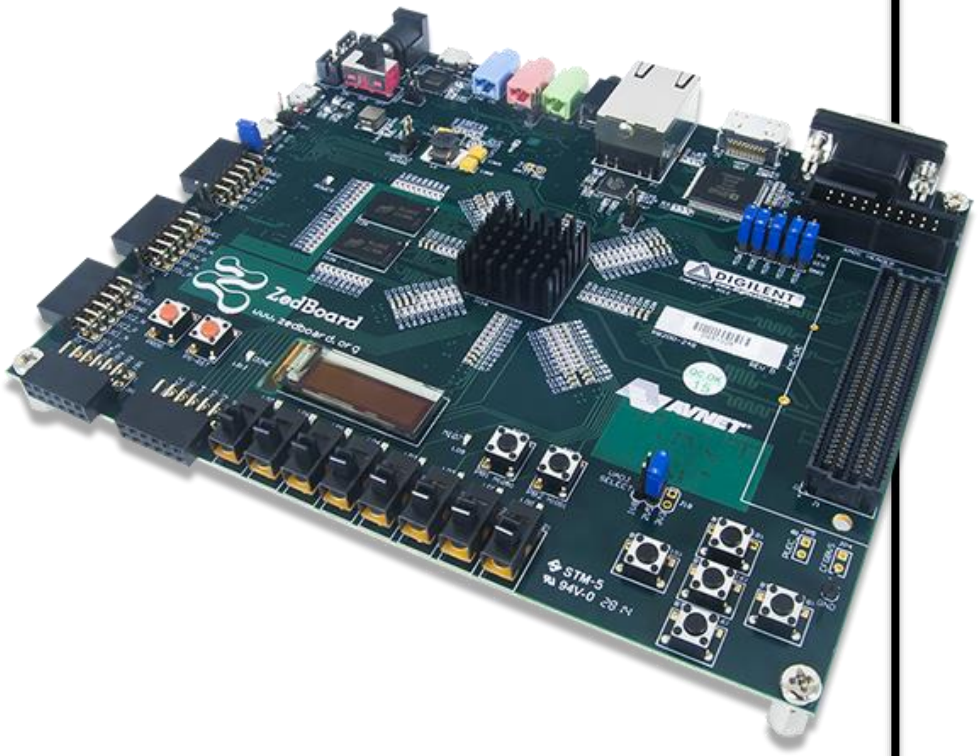
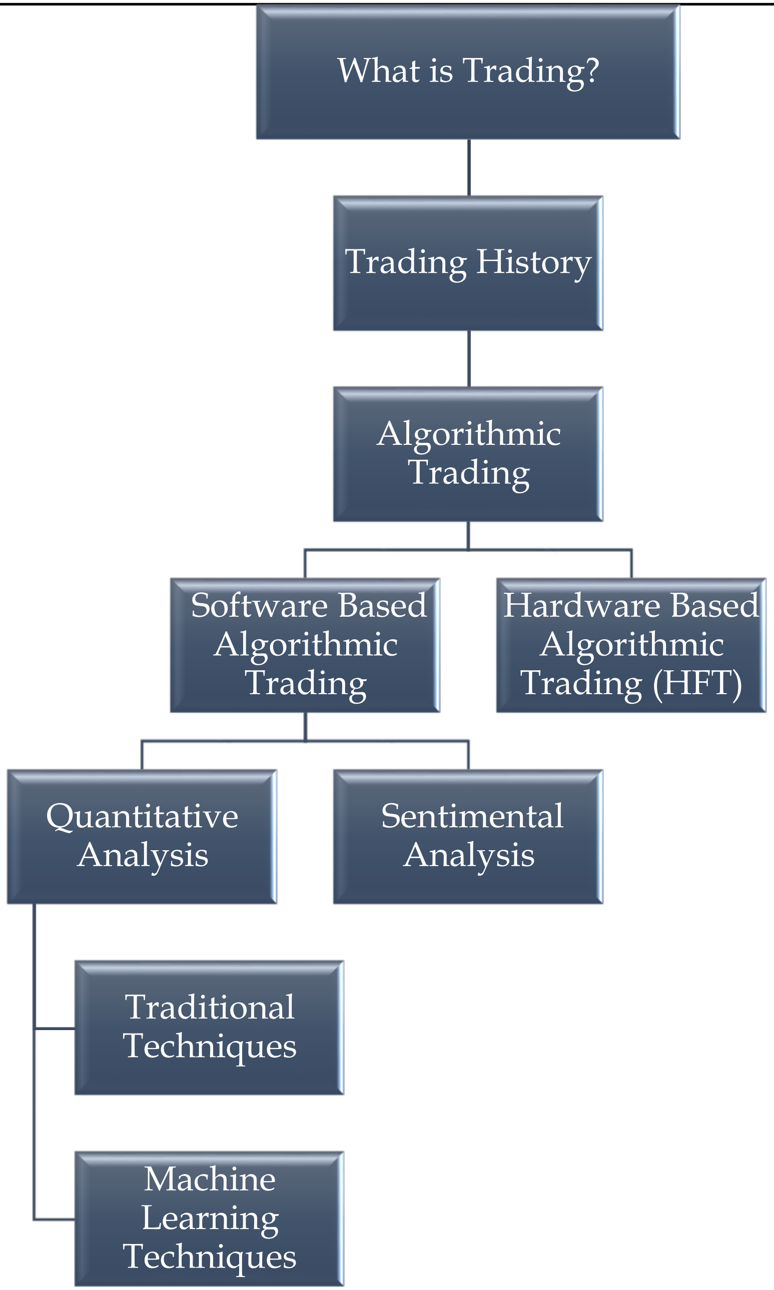
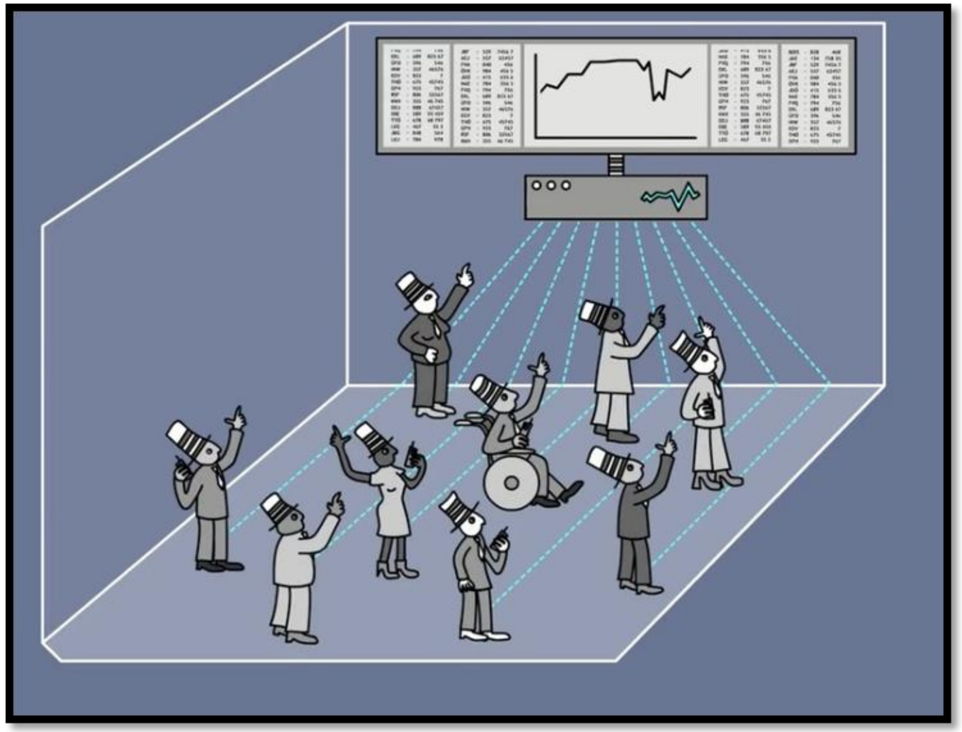
Developing and Verifying Speed Trading Algorithms on FPGAs

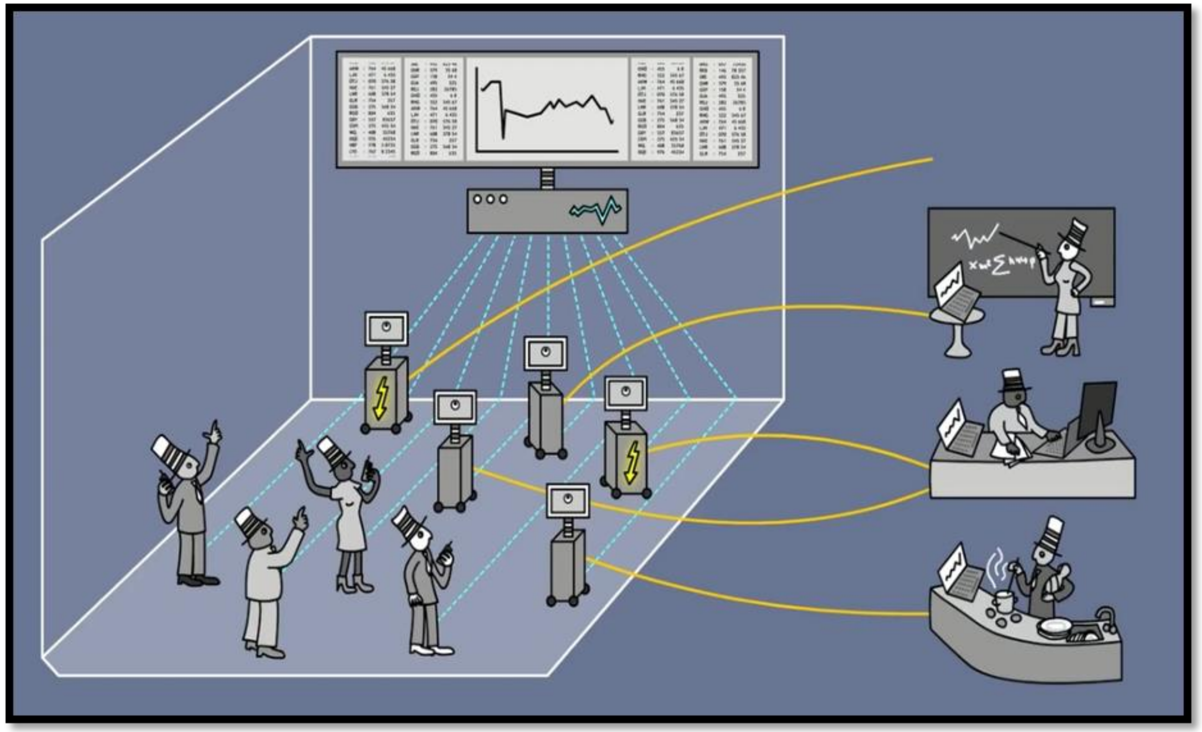


Overview


What is Trading?
Trading History
Algorithmic Trading
Software Based Algorithmic Trading
Hardware Based Algorithmic Trading (HFT)
Quantitative Analysis
Sentimental Analysis
Traditional Techniques
Machine Learning Techniques

What is Trading?





20th Century

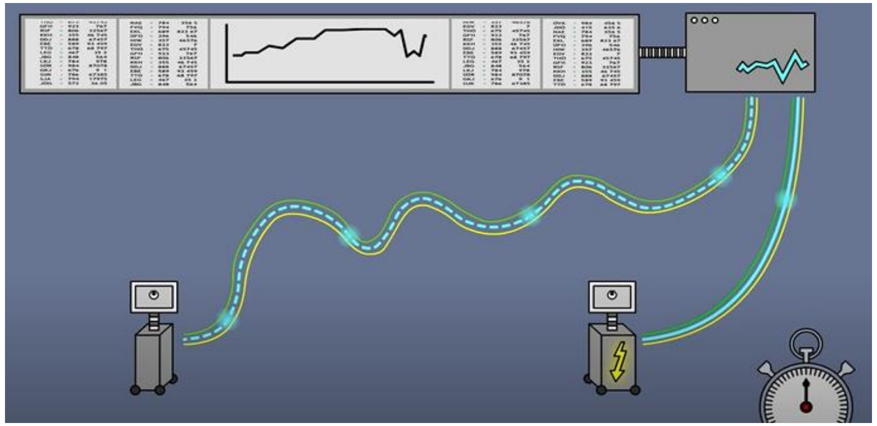
21st Century

Traditional Trading → **Algorithmic Trading**

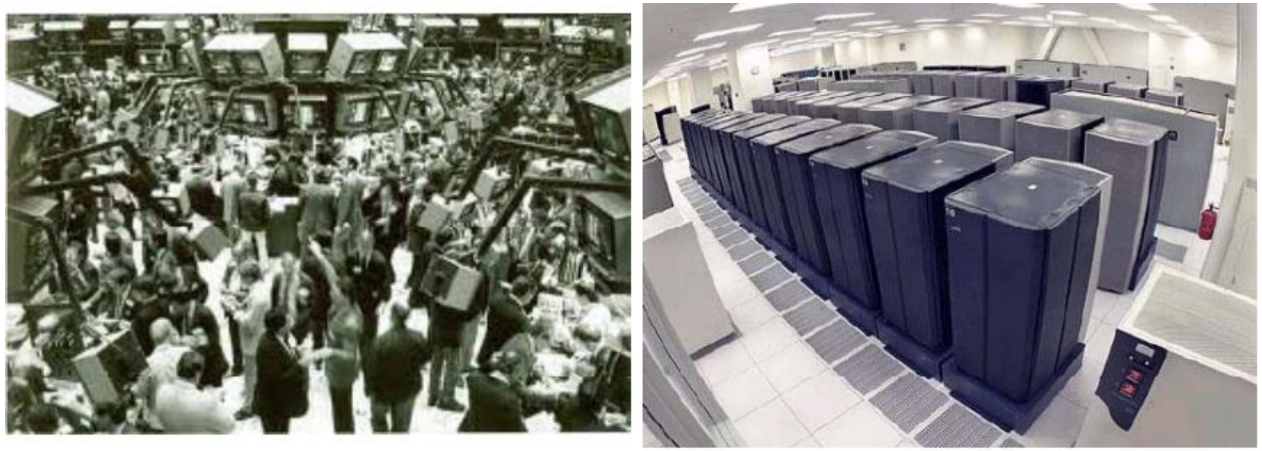
What is HFT (High Frequency Trading)?

High Frequency Trading

- The core difference between them is that Algorithmic Trading (AT) is designed for the long-term, while High-Frequency Trading (HFT) allows one to buy and sell at a very fast rate. The reason we shifted form AT to HFT is Speed.
- Usually, the latency should be between 300 – 800 nanoseconds.
- Typically short holding period about 22 seconds



High Frequency Trading System



20th century   21st century

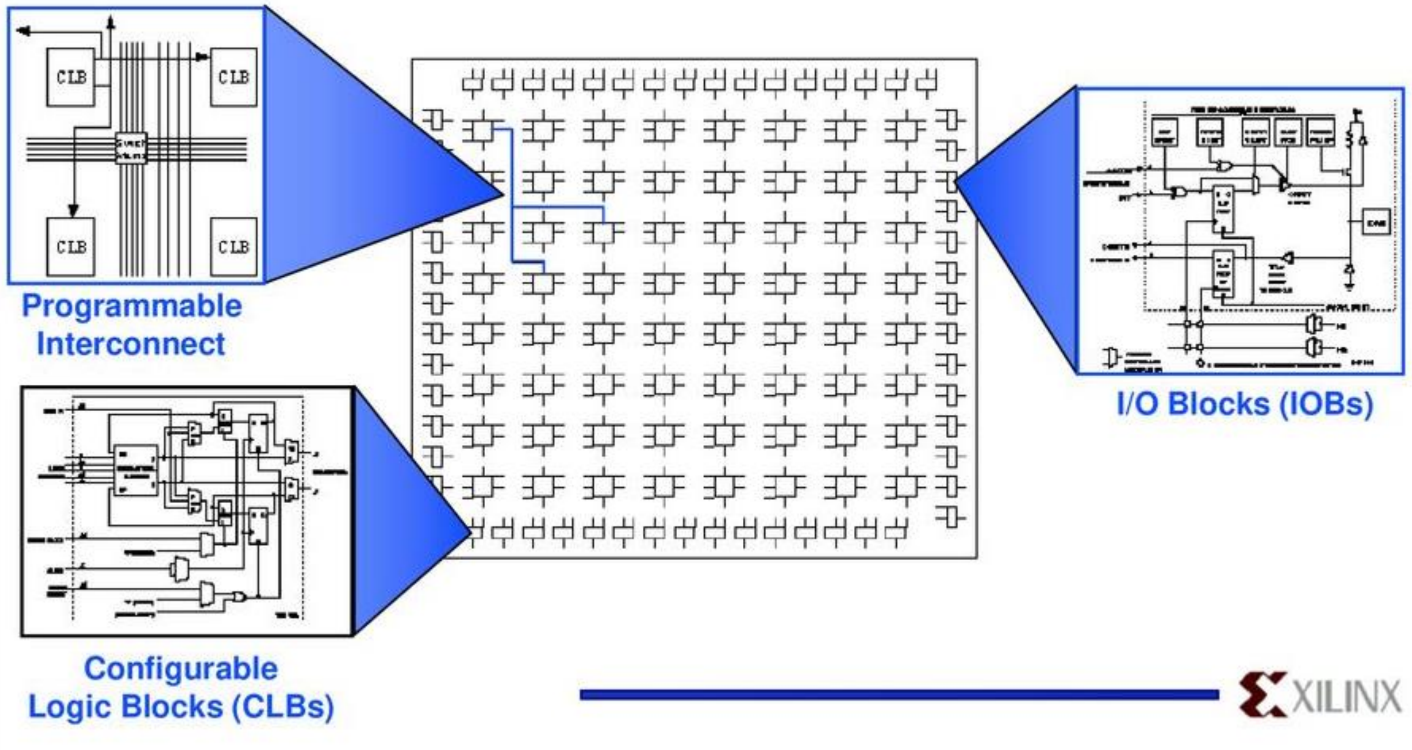
Evolution of High Frequency Trading

What is FPGA
(Field Programmable Gate Arrays)?

Introduction to FPGA

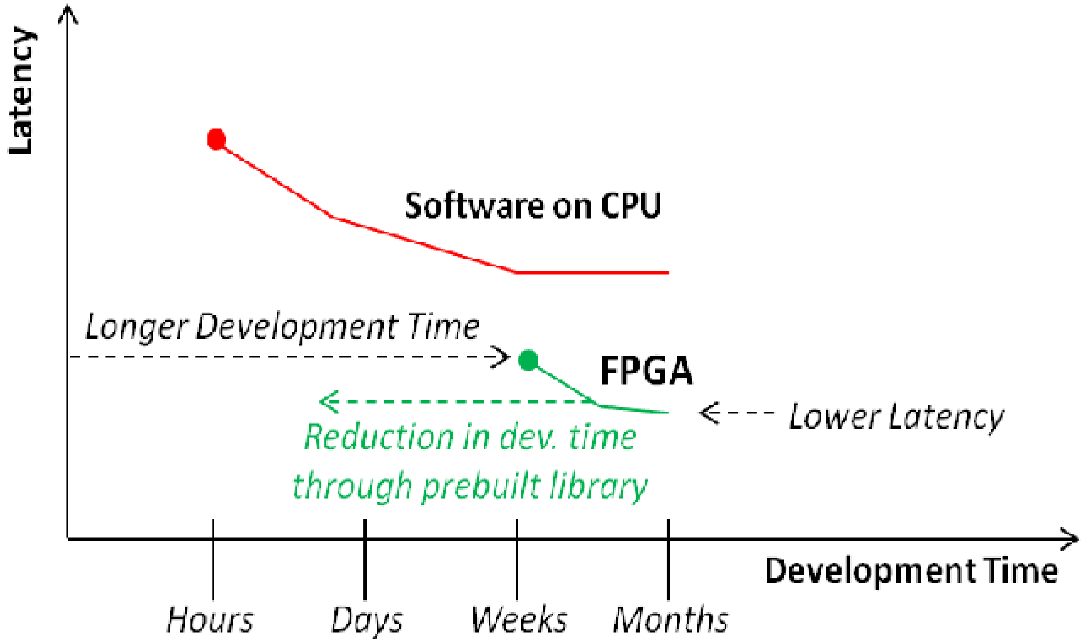
- ❑ Reprogrammable chip contains thousands of Logic Gates
- ❑ Software defined Hardware
- ❑ HDL language is used (VHDL or Verilog)

- ❑ Main Components
- o CLB (Configurable Logic Gates)
- o Connection Blocks (Wires)
- o Switch boxes
- o Input/ Output Blocks



Why we use FPGA for HFT?

- ❑ HFT system requires extremely low latency in response to market updates which is achieved by FPGA.
- ❑ FPGA chips enable them to execute certain types of trading algorithms up to 1000 times faster than traditional software solutions.

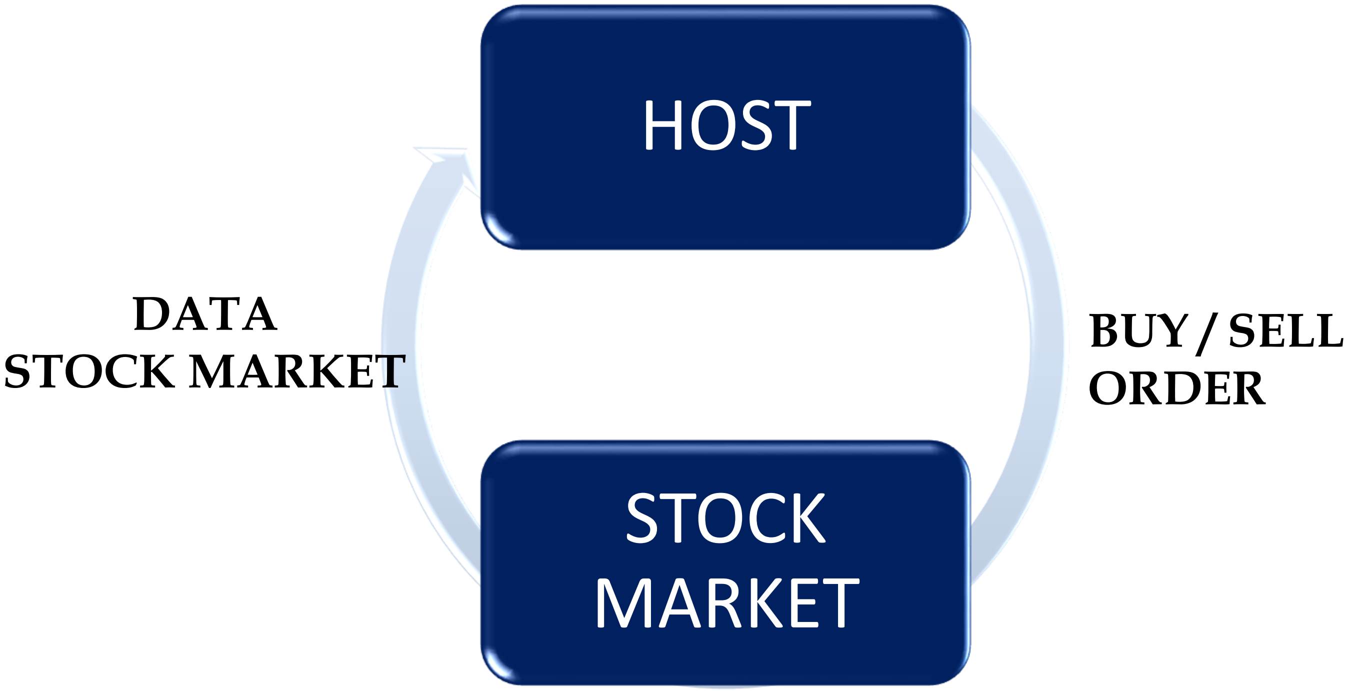


Methods of Stock Market Prediction

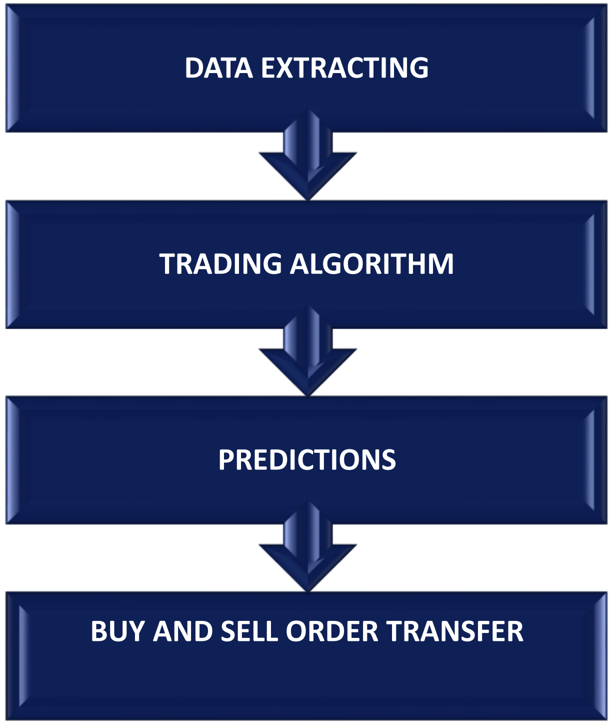
Methods of Stock Market Prediction

An automated trading system (ATS) uses a computer program to create buy and sell orders and automatically submits the orders to a market center or exchange.

1. Traditional Techniques
2. Machine Learning Techniques



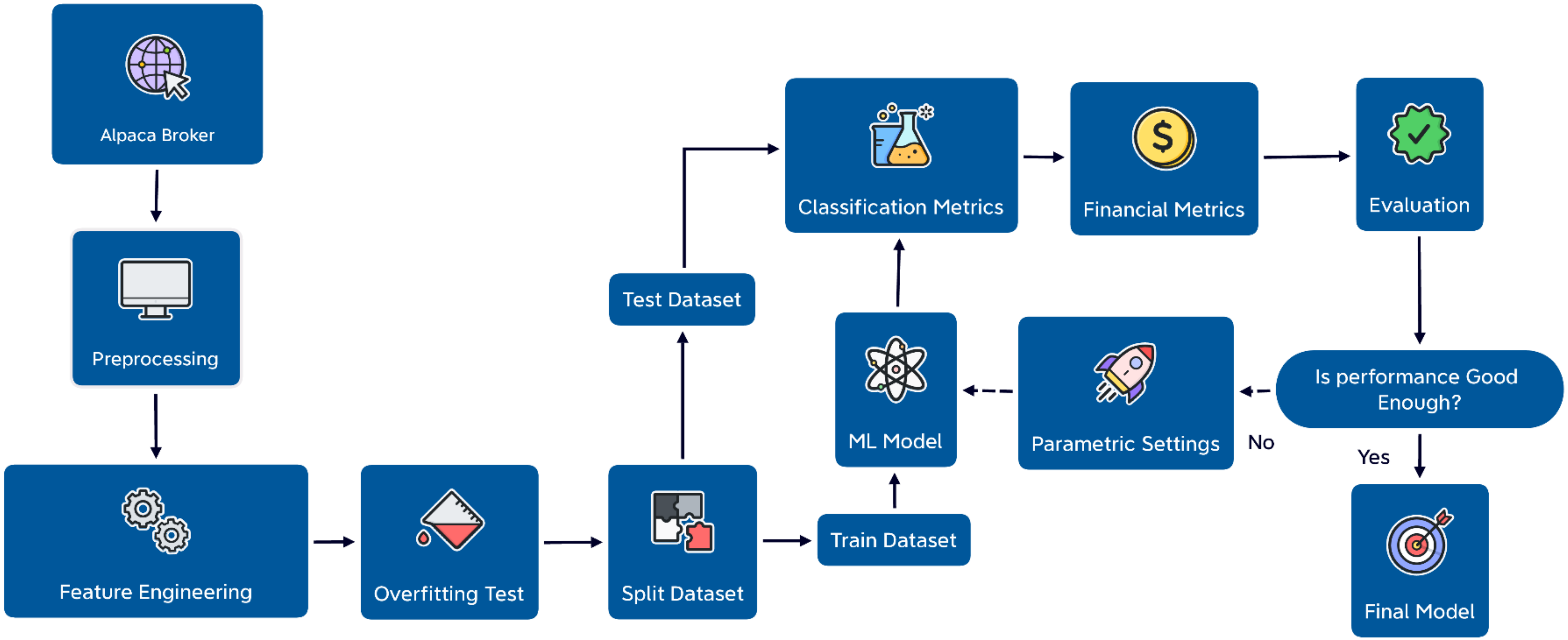
Overall trading overview



Overall host process

Software Implementation

Methodology



Machine Learning Algorithms

- K Nearest Neighbors (KNN)
- Support Vector Machine (SVM)
- Random Forest
- Logistic regression (LR)
- Artificial Neural Network (ANN)
- ADA Boost
- XG Boost
- Naïve Bayes
- Decision Tree

Research Contribution

- A performance comparison of nine ML models trained using the traditional methodology for stock market prediction using both performance metrics and financial system simulations.

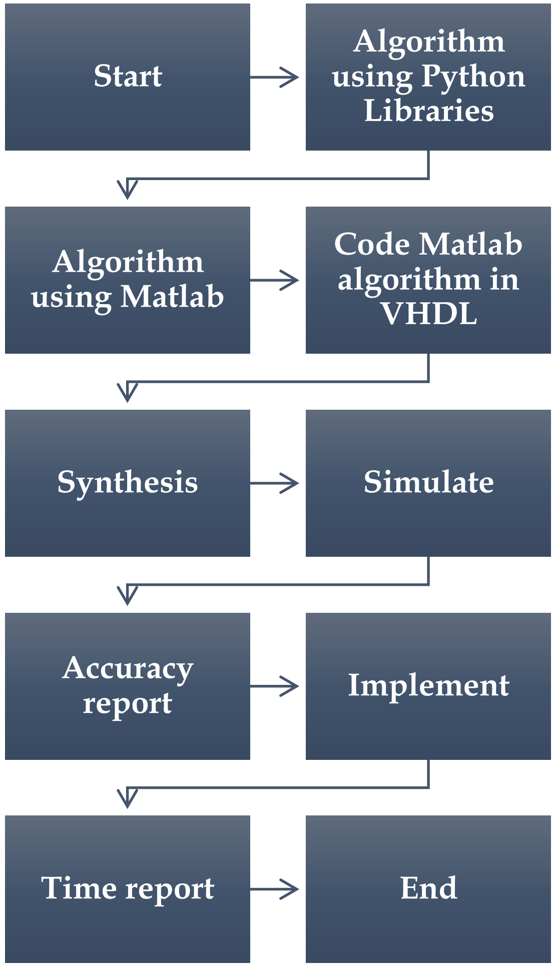
- Proposing a novel strategy to train the ML models for financial markets that perform much better than the traditional methodologies.

- Proposing a novel financial system simulation that provides financial performance metrics like returns, maximum drawdown and risk-to-reward ratio for each ML model.

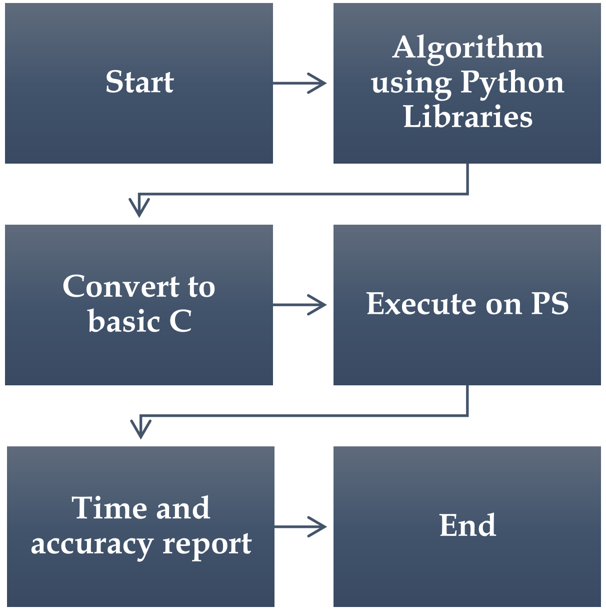
Hardware Implementation

Hardware Implementation

Methodology

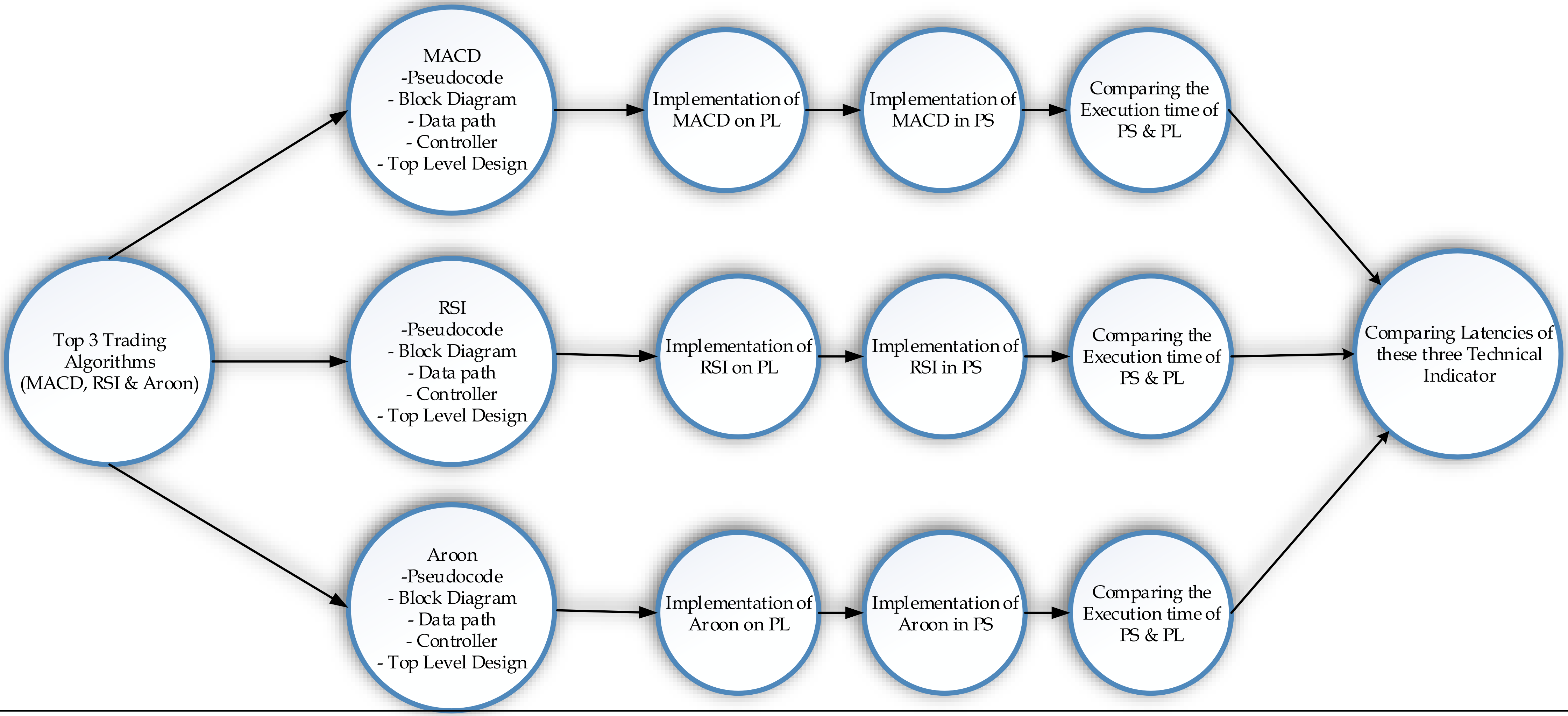


Programmable Logic (PL) overview



Processing System (PS) overview

Hardware Implementation
Methodology



Implemented Trading Algorithms on FPGA

- ❑Moving Average Convergence Divergence (MACD)

- ❑Relative Strength Index (RSI)

- ❑Aroon Indicator

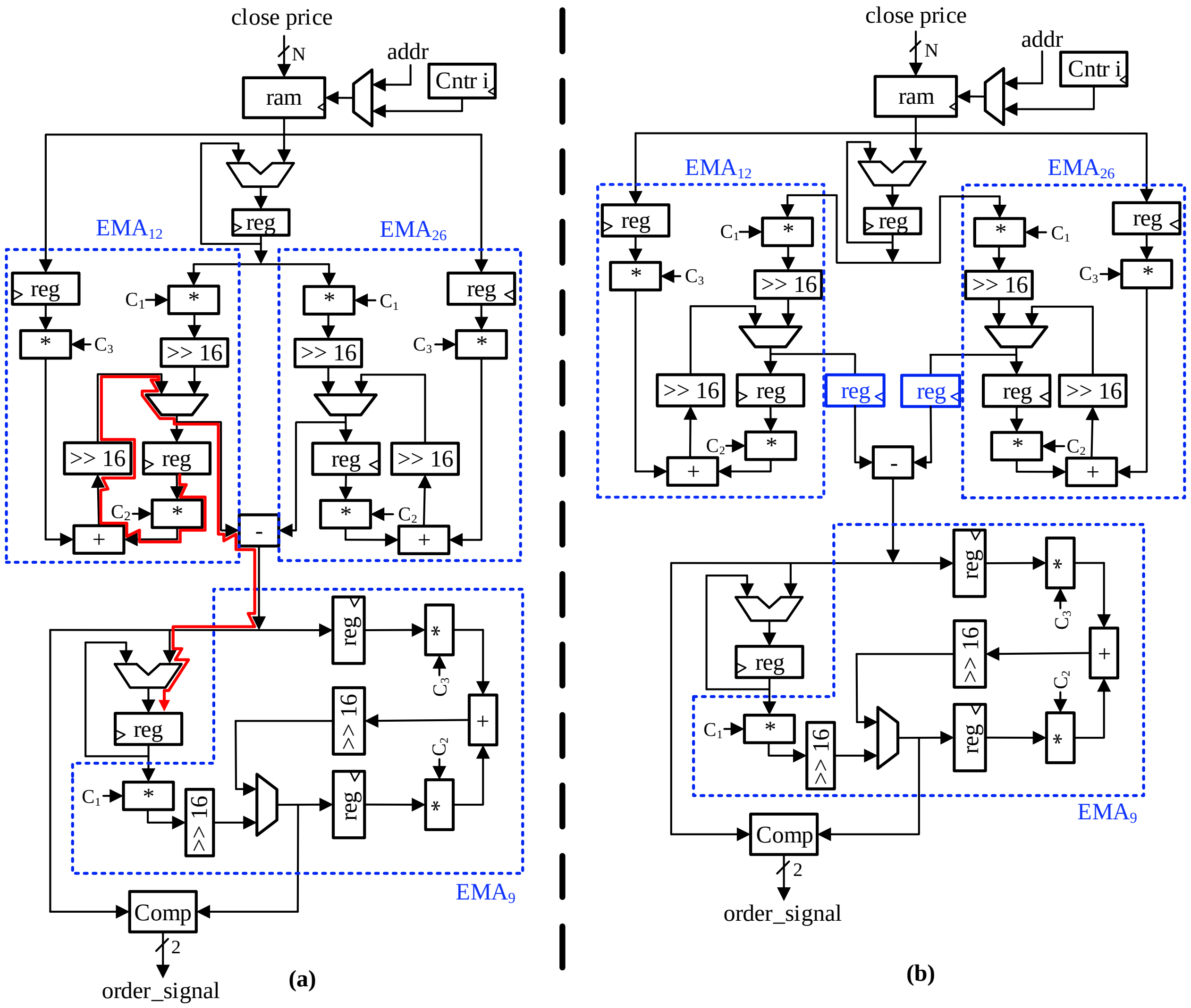
MACD Analysis

Algorithm 1 Modified MACD Algorithm

Input: C_p - close prices of all cyrprocurrencies; N - number of close prices; $C_{12,j}$, $C_{26,j}$, $C_{9,j}$ - modified smoothing factors for EMA_{12}, EMA_{26} and EMA_9 respectively; $1 \leq j \leq 3$

Output: MACD, $EMA_9 \in$ MACD signal

for $i = 0$ *to* N **do**
  $Sum = \sum_{n=1}^{26} C_p(i)$
  if $i == 12$ **then**
    $EMA_{12}(i) = Sum * C_{12,1}$
    $EMA_{12}(i) = EMA_{12}(i) >> 16$
  else if $i > 12$ **then**
    $EMA_{12}(i) = C_p * C_{12,2} + EMA_{12}(i-1) * C_{12,3}$
    $EMA_{12}(i) = EMA_{12}(i) >> 16$
  else
    $EMA_{12}(i) = 0.$
  end if
  if $i == 26$ **then**
    $EMA_{26}(i) = Sum * C_{26,1}$
    $EMA_{26}(i) = EMA_{26}(i) >> 16$
  else if $i > 26$ **then**
    $EMA_{26}(i) = C_p * C_{26,2} + EMA_{26}(i-1) * C_{26,3}$
    $EMA_{26}(i) = EMA_{26}(i) >> 16$
  else
    $EMA_{26}(i) = 0.$
  end if
end for
$Sum = \emptyset$
for $i = 26$ *to* N **do**
  MACD $(i) = EMA_{12}(i) - EMA_{26}(i)$.
  $Sum = \sum_{n=1}^{9}$ MACD(i).
  if $i == 34$ **then**
    $EMA_9(i) = Sum * C_{9,1}$
    $EMA_9(i) = EMA_9(i) >> 16$
  else if $i > 34$ **then**
    $EMA_9(i) = MACD(i) * C_{9,2} + EMA_9(i-1) * C_{9,3}$
    $EMA_9(i) = EMA_9(i) >> 16$
  else
    $EMA_9(i) = 0.$
  end if
end for



RSI Analysis

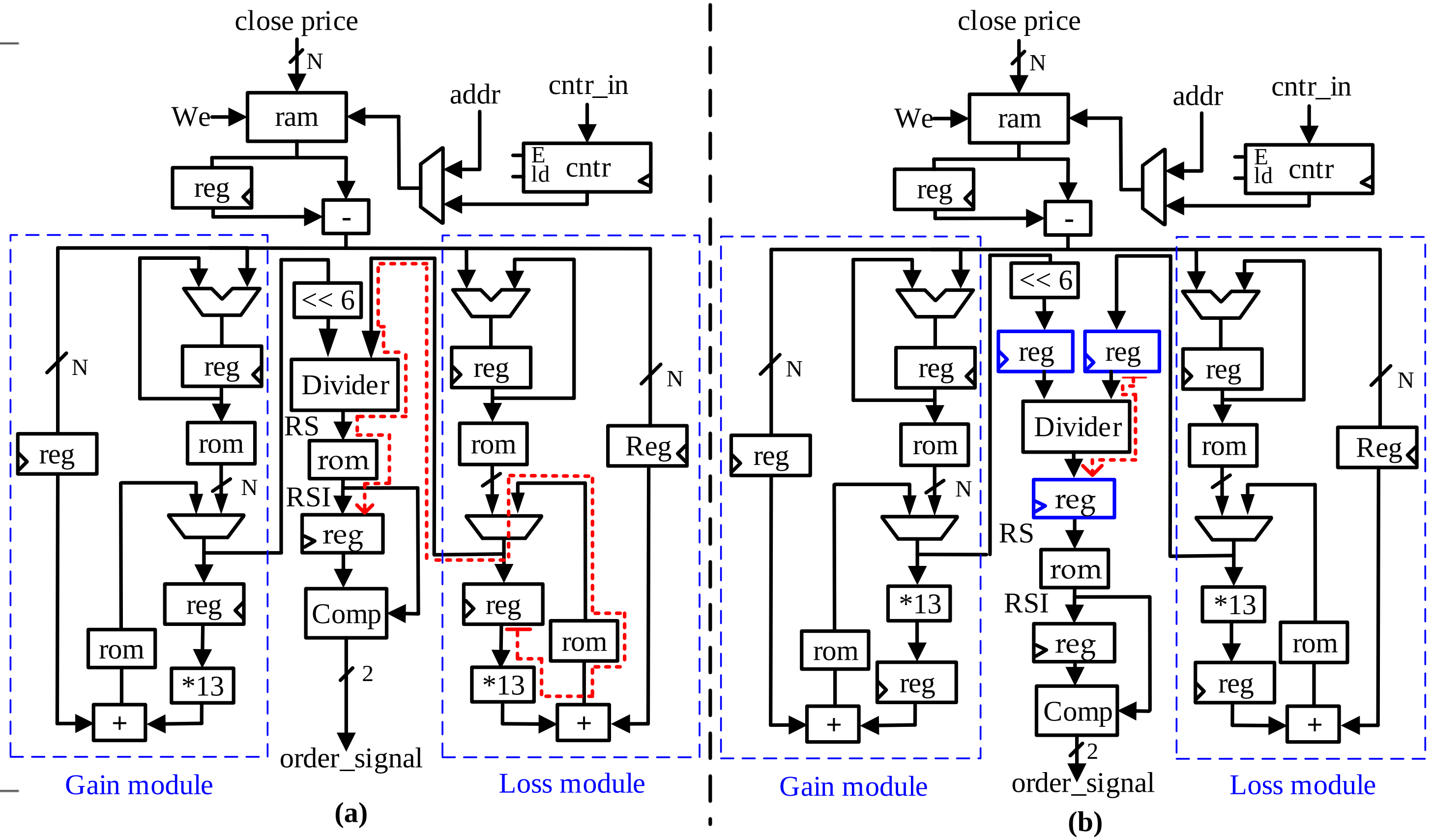
Algorithm 2 RSI Algorithm

Input: $Cp_i \in Close\ Prices$

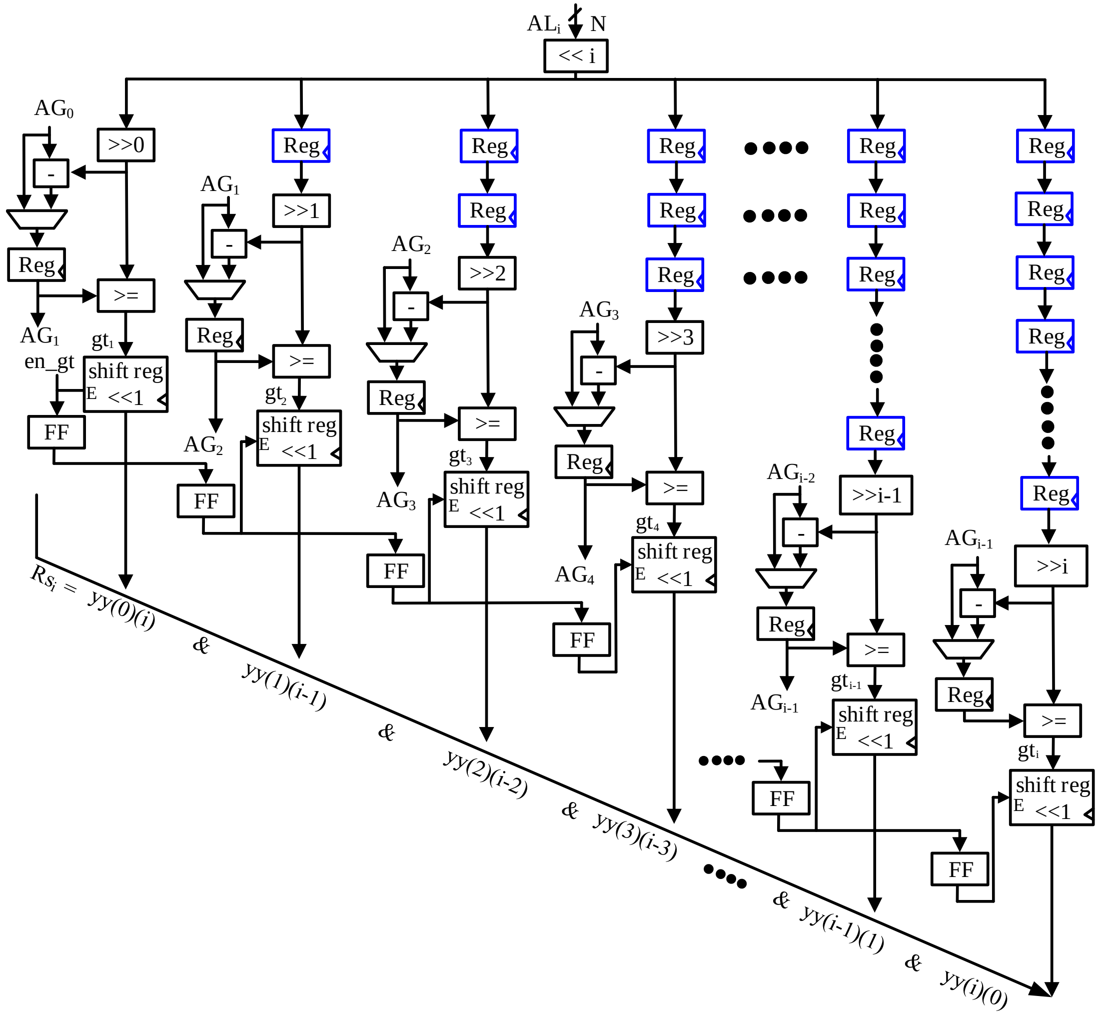
$Cp_i = (Cp_0, Cp_1 ... Cp_a), 0 \leq i \leq a$

Output: RSI_j

for $i = 1$ *to* a **do**
  if $Cp_i - Cp_{i-1} > 0$ **then** $Gain_i = Cp_i - Cp_{i-1}$
  else $Gain_i = 0$
  end if
  if $Cp_i - Cp_{i-1} < 0$ **then** $Loss_i = |Cp_i - Cp_{i-1}|$
  else $Loss_i = 0$
  end if
end for
for $j = 14$ *to* a **do**
  if $j = 14$ **then** $RS_{14} = \frac{AG_{14}}{AL_{14}}$
  else $RS_j = \frac{(AG_{j-1}*13+Gain_j)/14}{(AL_{j-1}*13+Loss_j)/14}$
  end if
  $RSI_j = 100 - \frac{100}{1+RS_j}$
end for



RSI Analysis (Fully Pipelined Divider)



Design	Delay (ns)	Freq. (MHz)	Clock Cycles	Latency (ns)
d1	33.7	29.6	365	12,318
d2	4.4	227.2	4,380	19,272
d3	5.5	181.8	377	2,074
RSI variants based on the dividers used				
$RSI_{ppl\text{-}d1}$	35.2	28.4	368	12,957
$RSI_{ppl\text{-}d2}$	8.9	112	4,383	39,228
$RSI_{ppl\text{-}d3}$	8.5	117	380	3,245

Performance of three divider variants followed by the performance of entire RSI architectures based on these dividers.

Notation

d1 - combinational divider

d2 - 2-stage pipelined divider

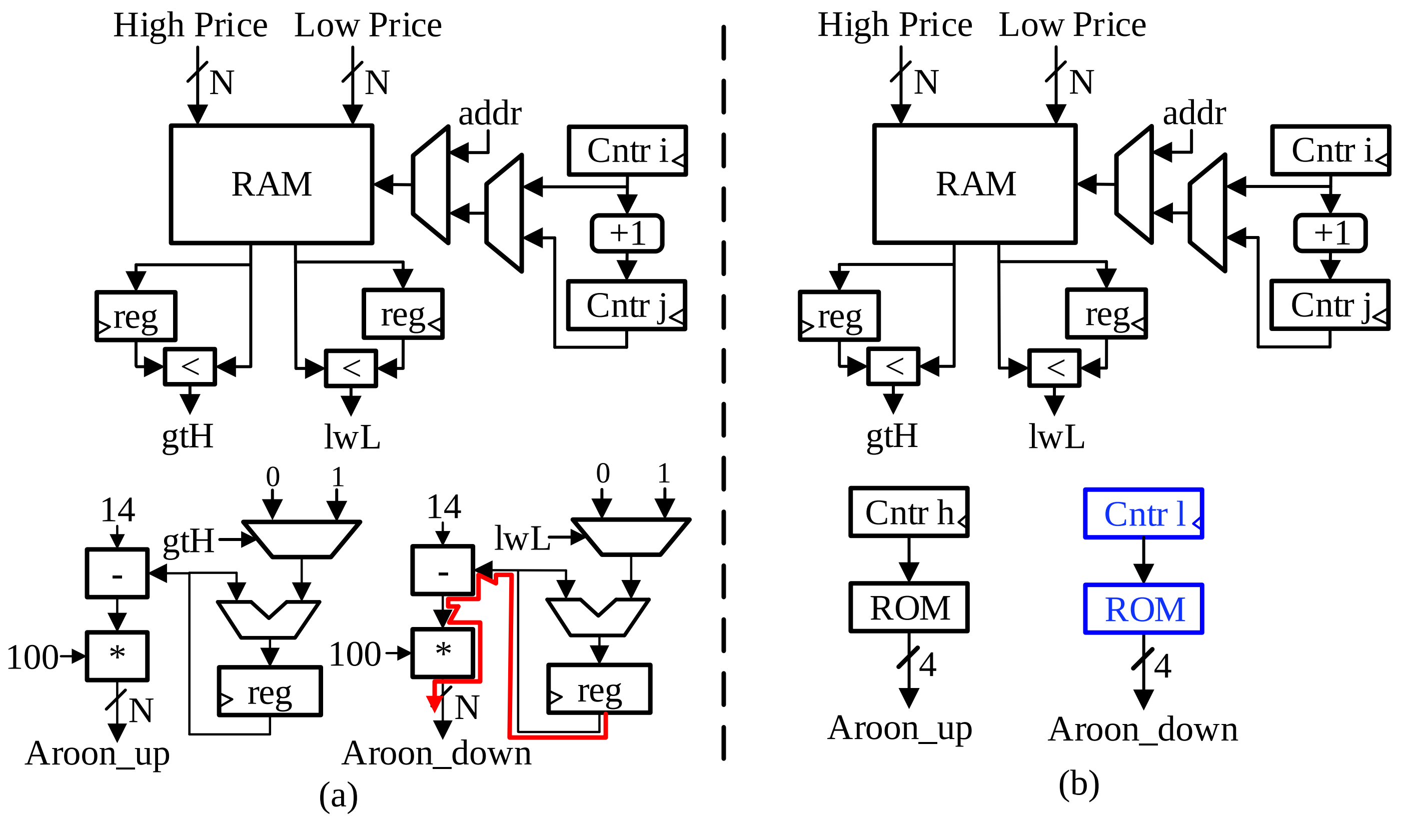
d3 - Fully pipelined divider

Aroon Analysis

Algorithm 3 Aroon Algorithm

Input: C_H: high prices of all cryptocurrencies
C_L: low prices of all cryptocurrencies
F_s: frame size of time period
N: number of required high and low prices
Output: Aroon_{up}, Aroon_{down}

for $i = 0$ to $N - F_s$ **do**
  $\text{Previous}_{high} = C_H(i)$
  $\text{Previous}_{low} = C_L(i)$
  for $j = i + 1$ to $i + F_s$ **do**
    if $C_H(j) > \text{Previous}_{high}$ **then**
      $\text{Previous}_{high} = C_H(j)$, $\text{Count}_{up} = \emptyset$
    else
      $\text{Count}_{up} = \text{Count}_{up} + 1$
    end if
    if $C_L(j) < \text{Previous}_{low}$ **then**
      $\text{Previous}_{low} = C_L(j)$, $\text{Count}_{down} = \emptyset$
    else
      $\text{Count}_{down} = \text{Count}_{down} + 1$
    end if
    if $j == i + F_s$ **then**
      $\text{Period}_{up} = \text{Count}_{up}$
      $\text{Period}_{down} = \text{Count}_{down}$
      $\text{Aroon}_{up}(j) = \frac{14 - Period_{up}(j)}{14} * 100$
      $\text{Aroon}_{down}(j) = \frac{14 - Period_{down}(j)}{14} * 100$
      $\text{Count}_{up} = \emptyset$, $\text{Count}_{down} = \emptyset$
    end if
  end for
end for



Results

Software Results

Financial Performance

Prediction Models	Cum Return (%)	Annual Return (%)	Max Draw down (%)	Sharpe Ratio	Ending Capital (USD)
Decision Tree	48.35	6.82	-48.11	0.36	14835
Logistic Regression	83.69	10.71	-59.35	0.47	18369
KNN	14.00	2.22	-59.25	0.23	11400
Naive Bayes	-19.16	-3.50	-53.85	0.10	8084
Random Forest	189.66	19.48	-37.21	0.68	28966
ADA Boost	135.91	15.44	-45.76	0.58	23591
SVM	104.10	12.69	-44.23	0.51	20417
XG Boost	130.37	14.99	-35.79	0.57	23037
ANN	154.46	16.92	-55.77	0.62	25446

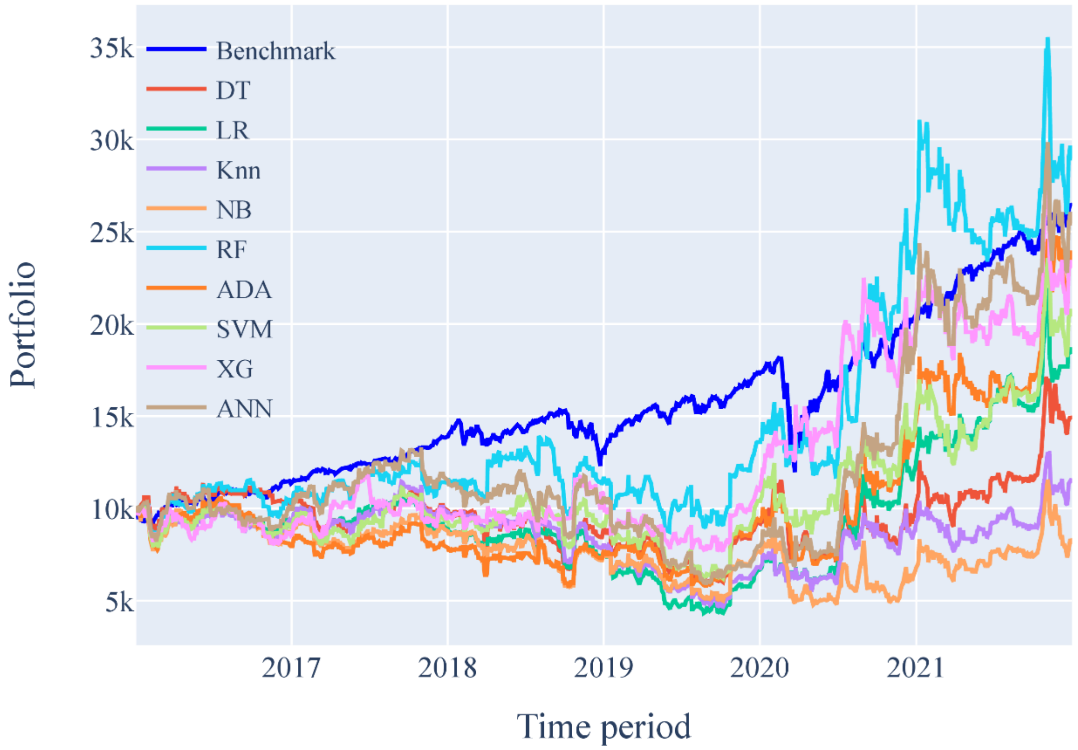
Financial performance of ML models for Tesla Inc. stocks on 1-day time frame

Prediction Models	Cum Return (%)	Annual Return (%)	Max Draw down (%)	Sharpe Ratio	Ending Capital (USD)
Decision Tree	67.51	9.02	-12.74	0.73	16751
Logistic Regression	86.59	11.00	-21.17	0.68	18659
KNN	47.67	6.74	-19.33	0.51	14767
Naive Bayes	-0.36	-0.06	-20.85	0.05	9964
Random Forest	153	16.80	-35.09	0.79	25300
ADA Boost	92.73	11.60	-21.23	0.71	19273
SVM	79.52	10.29	-17.04	0.77	17952
XG Boost	101.77	12.46	-18.01	0.74	20177
ANN	122.94	14.36	-19.33	0.91	22294

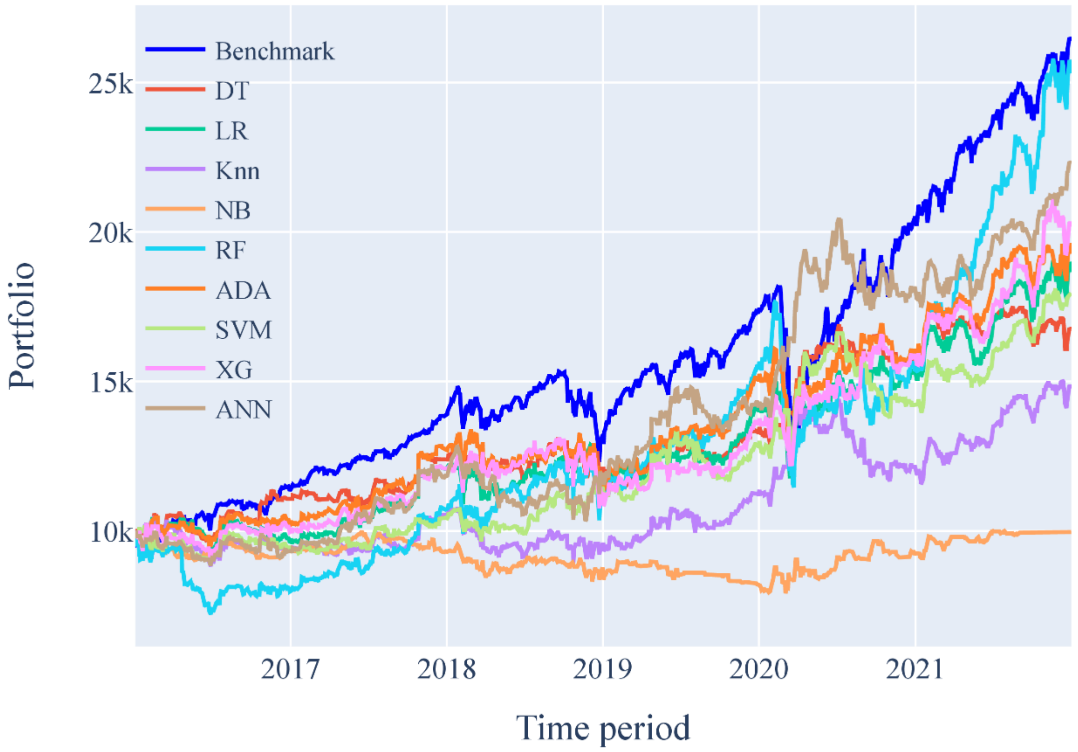
Financial performance of ML models for Tesla Inc. stocks on 15-min time frame

Software Results

Financial Performance



Portfolio Analysis of ML models on Tesla Inc. stocks for 1-day time frame



Portfolio Analysis of ML models for Tesla Inc. stocks on the proposed 15-min time interval strategy

Software Results

Classification Performance

ML Models	Accuracy (%)	F1_score (%)	ROC_AUC (%)	Precision (%)	Recall (%)
Decision Tree	83.01	83.00	83.58	83.50	83.50
Logistic Regression	85.51	85.50	85.77	85.50	86.00
KNN	79.15	79.12	79.49	79.50	79.50
Naive Bayes	73.49	70.10	70.50	79.50	70.50
Random Forest	84.45	85.11	85.13	85.00	85.50
ADA Boost	83.74	84.53	84.97	85.00	84.00
SVM	82.68	82.51	82.82	82.50	83.00
XG Boost	84.80	85.52	85.45	85.50	85.00
ANN	84.45	84.50	90.95	84.50	84.50

Classification metrics for Tesla Inc. stocks for 1-day time frame data

ML Models	Accuracy (%)	F1_score (%)	ROC_AUC (%)	Precision (%)	Recall (%)
Decision Tree	88.10	88.50	88.96	88.00	88.50
Logistic Regression	90.60	90.55	90.52	90.50	90.50
KNN	80.53	80.50	80.37	81.00	80.00
Naive Bayes	81.54	81.50	81.77	82.50	81.50
Random Forest	91.27	91.00	91.28	92.00	91.50
ADA Boost	90.93	91.02	91.03	91.50	91.00
SVM	88.59	88.50	88.49	89.00	88.50
XG Boost	90.93	91.00	91.53	91.00	90.50
ANN	89.93	90.00	90.63	90.00	90.00

Classification metrics for Tesla Inc. stocks for the proposed 15-min time interval strategy

Software Results

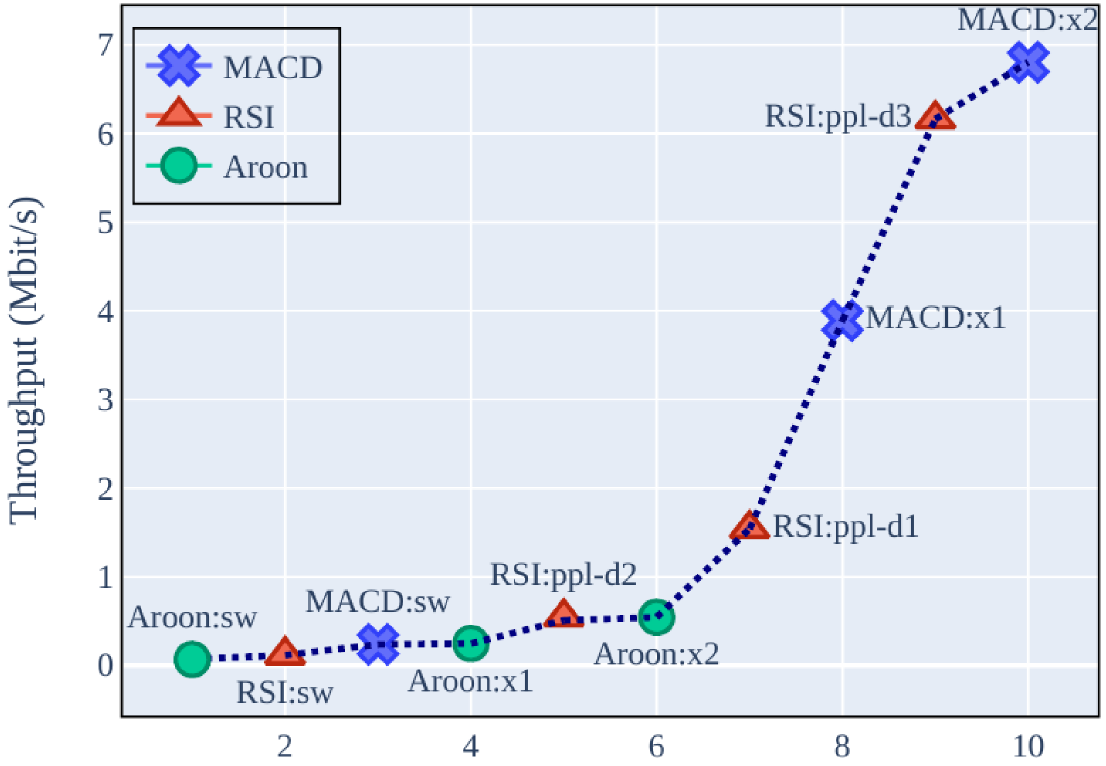
Classification Performance

Hardware Results

Arch.	Clock Cycles	Critical Path (ns)	Freq. (MHz)	Resource Utilization (LUTs, BRAMs, DSPs)	Latency (ns)	Throughput (Mbits/s)
				MACD		
x1	367	14.0	71.3	(419, 0, 9)	5,143	3.9
x2	368	7.7	129.6	(252, 1, 9)	2,839	7.0
				RSI		
ppl-d1	368	35.2	28.4	(1488, 0, 2)	12,957	1.5
ppl-d2	4383	8.9	112.1	(1004, 1, 0)	39,228	0.5
ppl-d3	377	8.5	117.0	(1227, 0, 0)	3,219	6.2
				AROON		
x1	5616	14.3	69.8	(619, 0, 2)	80,510	0.3
x2	5616	6.5	153.1	(124, 1, 0)	36,683	0.6

Results for the hardware architectures of all variants of the technical indicators, i.e., MACD, RSI and Aroon.

Hardware Results

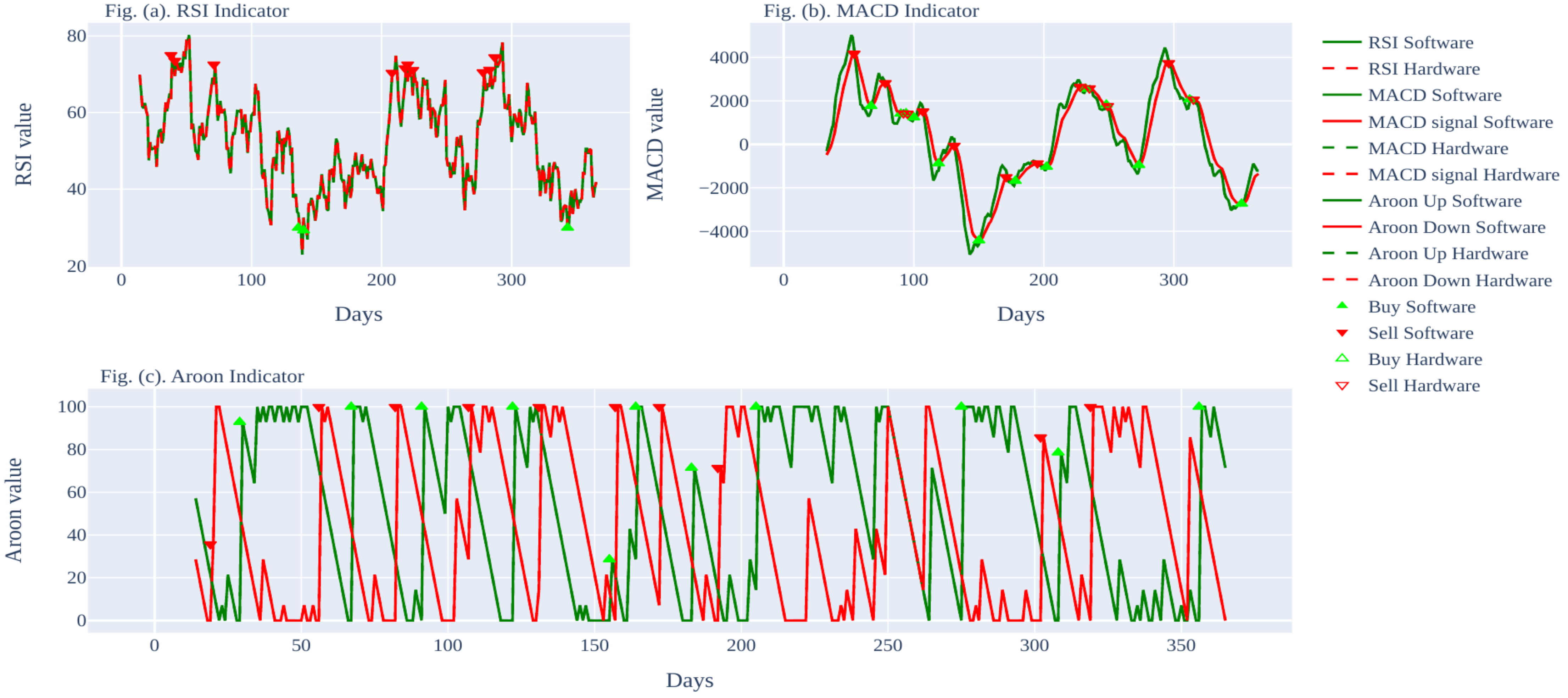


Highest performing software and hardware architectures for all three indicators

Algorithm	Arch.	Latency (ns)	Throughput (Mbits/s)	Speedup
MACD	SW x2	84,000 2,839	0.24 7.0	30* SW
RSI	SW ppld3	170,000 3,219	0.1 6.2	52* SW
Aroon	SW x2	289,000 36,683	0.1 0.6	32* SW

Comparison of the best hardware architecture from each of the technical indicators with the software implementation of the respective algorithm

Hardware Results



A Comparison of the accuracy achieved by software vs. hardware for a single day time frame on Bitcoin

Contribution to the Field

Contribution to the Field

1. A Performance Comparison of Machine Learning Models for Stock Market Prediction with Novel Investment Strategy
(Partially accepted in PLOS ONE Journal).

2. Speed vs. Efficiency: A Framework for High-Frequency Trading Algorithms on FPGA using Zynq SoC Platform
Under review in Journal of Computational Science (Elsevier).

3. High-Performance FPGA Architectures for Low-Latency Machine Learning Models in Stock Market Prediction: An In-Depth Complexity Analysis **(Current Research).**

Team

INTELLI FPGA
AI/FPGA Inovations

Dr. Umar Sharif
Doctorate in Post Quantum Cryptography

Dr. Zaka Ullah
Doctorate in Power Electronics

Abbas Ali
FPGA Engineer

Abdullah Shah
FPGA Engineer

Azaz Hassan Khan
FPGA Engineer