A Financial Asset Price Simulation DEVS implementation

**Author: Guillermo Trabes**

**Student ID: 101141466**

**Part one: Conceptual model**

Financial assets are liquid assets that gets its value from a contractual right or ownership claim. Cash, stocks, bonds, mutual funds, and bank deposits are all are examples of financial assets. The prices of financial assets evolve over time and past data of their values can be used to predict their future value.

The aim of the proposed model is to implement a financial asset price simulation with the DEVS formalism. The model will take as an input a sequence of time series data (date and price) for a given financial asset and it will return as an output the forecast of the future price for the following day of the input series. The work is based on [1] where a simulation with 2 different techniques are presented to perform the simulation over stock prices: Bootstrap and Monte Carlo. In this work we will focus on the Bootstrap method.

The conceptual model is as follows:

APS

Bootstrap generator

Random Selector

Daily return generator

Prediction Generator

The Daily Return Generator model will receive the time series data and it will generate the daily returns for each point in the series with the following formula , where is daily return, is the financial asset price at a given day and is the price the day before. Then it will send to the Bootstrap generator coupled model a set with the daily return values and the last date and price to perform the prediction.

Inside the boostrap generator coupled model, the random selector module will pick one of the return values at random with a uniform distribution probability and it will send the result and the last date and price from the original series to the prediction generator.

The prediction generator module will generate a value prediction with the following formula: , where where is the previous return value picked at random and is ~𝑁(0,1) , the predicted value generated will be send to the top model. The result will be the predicted value for the next day from the data analysed.

REFERENCES

[1] Pažický, Martin. (2017). Stock Price Simulation Using Bootstrap and Monte Carlo. Scientific Annals of Economics and Business. 64. 155-170.

**Part two: Formal Specification and Experimentation Strategy**

## Formal model

As explained by the conceptual model, the top model for this system will be composed by 2 DEVS models: daily\_return\_generator and boostrap\_generator. The model daily\_return\_generator is a DEVS atomic model and the model boostrap\_generator is a coupled DEVS model, composed by the random\_selector and prediction\_generator DEVS atomic models.

In the model’s definition a datatype is used, called time\_series\_element, which is composed by a tuple: <date, price>, where: date is a string representing a correct date on the Gregorian calendar with the format “YYYY-MM-DD”, and price is a real number.

The daily\_return\_generator takes as input a set of elements of time\_series\_elements type with the date and price taken from a financial asset and returns a set of time\_series\_elements type with the daily return for every value on the set received, and also adds on the set the date and price of the last element received, which will be used for predictions. The formal definition for the atomic models is as follows:

daily\_return\_generator: <X, Y, S, δint, δext, λ, ta>, where:

* X= { drg\_in: time\_series\_element }
* Y= { drg\_out: time\_series\_element }
* S = {
  + received\_elements integer
  + daily\_returns: finite set of time\_series\_element
  + output\_needed: boolean
  + last\_element: time\_series\_element

}

* δint : any state -> output\_needed=false
* δext: for(every i in X) {

if(received\_elements\_number==0){last\_element = i}

else { daily\_return.add ( ln(last\_element/i))}

received\_elements++

}

last\_element = element in X with larger date value

output\_needed= true

* λ: drg.out ={daily\_returns, last\_element}
* ta: if (output\_needed == true) {ta=0} else {ta= infinity}

The random\_selector model takes as input a set of elements of time\_series\_elements type with a date and daily\_return taken from a financial asset, and the last date and price for the asset. This model returns as output 2 elements of type time\_series\_element, the first one is a value picked at random with a uniform distribution from the set of daily returns received, and the second one is last date and price value received as input. The formal description for the model is as follows:

random\_selector <X, Y, S, δint, δext, λ>, where:

* X= { rs\_in: time\_series\_element}
* Y= { rs\_out: time\_series\_element}
* S = {
* random\_choice: time\_series\_element
* last\_element: time\_series\_element
* output\_needed;

}

* δint : { any state -> output\_needed=false}
* δext: {

/\* picks random value \*/

random\_value = rs\_in[uniform\_distribution(0..rs\_in.size()-1)]

last\_element= rs\_in[last\_value];

output\_needed=true;

}

* λ: return(random\_value, last\_element)
* ta: if (output\_needed == true) { ta=0} else {ta= infinity}

The prediction\_generator model takes as input 2 elements of type time\_series\_element. The first one is a date and a return and the second one is date and a price. This model generates as output 1 element of type time\_series\_element, which is a prediction for the next day of the second input element received. The formal description for the model is as follows:

prediction\_generator <X, Y, S, δint, δext, λ>:

* X= {pg:in: time\_series\_element, last\_date: series\_element}
* Y= {pg.out: time\_series\_element}
* S = {
  + prediction: time\_series\_element
  + output\_needed: boolean

}

* δint : { any state -> output\_needed=false}
* δext: {

prediction.price = last\_date.price + enormal\_value(0,1)\*random\_return.price

prediction.date = nextday(last\_date.date)

output\_needed=true

}

* λ: return(random\_value, last\_element)
* ta: if (output\_needed == true) { ta=0} else {ta= infinity}

The formal definition for the coupled models is the following:

Boostrap\_predictor:

* X = {in\_returns: set of time\_series\_element}
* Y = {out\_prediction: set of time\_series\_element}
* D = {random\_selector, prediction\_generator}
* Md = {I\_random\_selector, I\_prediction\_generator}
* EIC = {(self.in\_returns, random\_selector.rs\_in)}
* EOC = {(predictor\_generator.pg\_out, self.out\_prediction)}
* IC = {(random\_selector.rs\_out, prediction\_generator.pg\_in)}

APS:

* X = {in: set of time\_series\_element}
* Y = {out: set of time\_series\_element}
* D = {daily\_return\_generator, boostrap\_predictor}
* Md = {I\_daily\_return\_selector, I\_bootstrap\_predictor}
* EIC = {(self.in, daily\_return\_generator.drg\_in)}
* EOC = {(boostrap\_predictor.out\_prediciton, self.out)}
* IC = {(daily\_return\_generator.drg\_out, boostrap\_predictor.in\_returns)

## Experimentation Strategy

To experiment and test a bottom up strategy will be used. First the atomics models (daily\_return\_generator, random\_selector and prediction\_generator) will be tested. Then, the coupled models will be tested incrementally, first the boostrap\_predictor and finally the complete APS model.

For each of the models the inputs will change in size and data to show that the models behave correctly.

For the daily\_return\_generator model, the experiments with the following inputs will be performed:

1. 2018-10-25 29334.0

The expected result is an error, as the model does not have enough information to perform a daily return value

1. 2018-10-25 29334.0

2018-10-26 29369.0

2018-10-29 28898.0

2018-10-30 29425.0

2018-10-31 29491.0

The expected result is the daily returns from 2018-10-26 to 2018-10-31 and the last date and price:

2018-10-26 0.0011

2018-10-29 -0.016

2018-10-30 0.018

2018-10-31 0.0022

2018-10-31 29491.0

For the random\_selector model, the experiments with the following inputs will be performed:

1. 2018-10-26 0.0011

2018-10-31 29491.0

The expected result is:

2018-10-26 0.0011

2018-10-31 29491.0

There is only one value to choose at random

1. 2018-10-29 -0.016

2018-10-30 0.018

2018-10-31 0.0022

2018-10-31 29491.0

The expected result is a random selection from one of the 3 first elements followed by the fourth one. Several executions for this experiment will be performed to check the results.

For the prediction\_generator model, the experiments with the following inputs will be performed:

1. 2018-10-26 0.15

2018-10-31 29491.109375

The expected result is:

2018-11-01 value

Where value is a real number calculated with a formula involving a random distribution, a statistical analysis should be performed to validate the result, which is out of the scope of this work. Nevertheless, several experiments will be performed with this input to test the randomness of the component and the results in variation in price are expected to be near 0.

1. 2018-10-26 5.3

2018-10-31 29491.109375

The expected result is:

2018-11-01 value

As in the previous case the expected result for value is random but the result should have a greater variance in the predicted prices comparing them with the previous experiment. Several experiments will be executed with this input as well.

For the bootstrap\_generator model, the experiments with the following inputs will be performed:

1. 2018-10-26 0.15

2018-10-31 29491

The expected result is:

2018-11-01 value

Where value is a real number calculated with a formula involving a random distribution. The results in variation in price are expected to be near 0 as they are all predicted from a unique value from the list.

1. 2018-10-29 -0.16

2018-10-30 0.18

2018-10-31 0.22

2018-10-31 29491

The expected result is:

2018-11-01 value

As in the previous case the expected result for value is random but the result should have a greater variance in the predicted prices, comparing them with the previous experiment. Several experiments will be executed with this input as well.

Finally, for the complete APS model the experiments with the following input will be performed:

1. 2018-10-25 29334.990234

2018-10-26 29369.869141

2018-10-29 28898.240234

2018-10-30 29425.789063

2018-10-31 29491.109375

2018-11-01 31308.630859

2018-11-02 31419.259766

2018-11-05 31429.300781

2018-11-07 31404.689453

2018-11-08 30736.410156

2018-11-09 29912.490234

2018-11-12 29423.300781

2018-11-13 28543.429688

2018-11-14 29372.539063

2018-11-15 30473.820313

2018-11-16 31109.250202

As discussed on previous examples the result from the experiment will be random, this test will be performed to check if the algorithm returns correctly a time\_series\_type value.

**Part three: Implementation for the Asset Price Simulator in Cadmium**

The code for the complete model can be found in in the .zip file attached. The structure for the models is the same as the one for the Alternate Bit Protocol presented in the Cadmium tutorial. The source code from the atomic models is located on the atomics folder and the source code to test the different components is in the tests folder. The source code for the complete coupled model is on the folder main\_top.

The scripts used to perform the experiments are located on the bin directory. Every script requires an input data file, for all the experiments performed and showed here, the input files can be found in the folder input\_data. In this section the results of executing the experiments proposed on last section will be show.

The readme file on the projects details the steps to execute the simulations.

## Experimental Results

All the traces show in this section correspond to the messages log from cadmium.

daily\_return model:

To execute this model the main file main\_daily\_return\_model\_test.cpp was created, which compiles the DAILY\_RETURN\_TEST executable in the bin folder, to execute this file 2 scripts were created at the folder scripts: daily\_return\_test1.sh and daily\_return\_test2.sh

1. The result of executing this experiment is the following:

Command:

./daily\_return\_test1.sh

Result:

DAILY\_RETURN\_TEST: test/../atomics/daily\_return\_generator.hpp:72: void daily\_return\_generator<TIME>::external\_transition(TIME, cadmium::make\_message\_bags<std::tuple<daily\_return\_generator\_defs::in> >::type) [with TIME = NDTime; cadmium::make\_message\_bags<std::tuple<daily\_return\_generator\_defs::in> >::type = std::tuple<cadmium::message\_bag<daily\_return\_generator\_defs::in> >]: Assertion `false && "Error: at least two prices are needed to compute a daily return"' failed. Aborted (core dumped)

As expected the result is an error indicating the cause.

1. The trace of execution in the messages log from execute the experiment is the following:

Command:

/daily\_return\_test2.sh

Log:

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out:{}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-25 29334, 2018-10-26 29369, 2018-10-29 28898, 2018-10-30 29425, 2018-10-31 29491}] generated by model input\_reader

00:00:00:000

[daily\_return\_generator\_defs::out: {2018-10-26 0.00119244, 2018-10-29 -0.0161673, 2018-10-30 0.0180723, 2018-10-31 0.00224048, 2018-10-31 29491}] generated by model daily\_return\_generator1

The logs show the expected result.

random\_selector model:

To execute this model the main file main\_random\_selector\_model\_test.cpp was created, which compiles the RANDOM\_SELECTOR\_TEST executable in the bin folder, to execute this file 2 scripts were created in the folder scripts for the experiments: random\_selector\_test1.sh and random\_selector\_test2.sh

1. The result of executing this experiment is the following:

Command:

./random\_selector\_test1.sh

Log:

00:00:00:000

00:00:00:000 [cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out:{}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-26 0.0011, 2018-10-31 29491}] generated by model input\_reader

00:00:00:000

[random\_selector\_defs::out: {2018-10-26 0.0011, 2018-10-31 29491}] generated by model random\_selector1

As expected the result for the random choice is the only element available on the set to pick and the value for the last date and price is transmitted correctly.

1. To show the randomness the experiment was executed several times. The traces of execution in the messages log from execute the experiment is the following.

Command:

./random\_selector\_test2.sh

Execution 1:

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-29 -0.016, 2018-10-30 0.018, 2018-10-31 0.0022, 2018-10-31 29491}] generated by model input\_reader

00:00:00:000

[random\_selector\_defs::out: {2018-10-30 0.018, 2018-10-31 29491}] generated by model random\_selector1

Execution 2:

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-29 -0.016, 2018-10-30 0.018, 2018-10-31 0.0022, 2018-10-31 29491}] generated by model input\_reader

00:00:00:000

[random\_selector\_defs::out: {2018-10-29 -0.016, 2018-10-31 29491}] generated by model random\_selector1

Execution 3:

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-29 -0.016, 2018-10-30 0.018, 2018-10-31 0.0022, 2018-10-31 29491}] generated by model input\_reader

00:00:00:000

[random\_selector\_defs::out: {2018-10-29 -0.016, 2018-10-31 29491}] generated by model random\_selector1

Execution 4:

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-29 -0.016, 2018-10-30 0.018, 2018-10-31 0.0022, 2018-10-31 29491}] generated by model input\_reader

00:00:00:000

[random\_selector\_defs::out: {2018-10-30 0.018, 2018-10-31 29491}] generated by model random\_selector1

Execution 5:

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-29 -0.016, 2018-10-30 0.018, 2018-10-31 0.0022, 2018-10-31 29491}] generated by model input\_reader

00:00:00:000

[random\_selector\_defs::out: {2018-10-29 -0.016, 2018-10-31 29491}] generated by model random\_selector1

The logs show the expected result, the model chooses between the different elements.

prediction\_generator model:

To execute this model the main file main\_prediction\_generator\_model\_test.cpp was created, which compiles the PREDICTION\_GENERATOR\_TEST executable in the bin folder, to execute this file 2 scripts were created in the folder scripts for the experiments: prediction\_generator\_test1.sh and prediction\_generator\_test2.sh

1. The result of executing this experiment is the following:

Command:

./prediction\_generator\_test1.sh

Execution1

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-26 0.15, 2018-10-31 29491.1}] generated by model input\_reader

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-01 29492.2}] generated by model prediction\_generator1

Execution 2

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-26 0.15, 2018-10-31 29491.1}] generated by model input\_reader

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-01 29492.1}] generated by model prediction\_generator1

Execution 3

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-26 0.15, 2018-10-31 29491.1}] generated by model input\_reader

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-01 29492}] generated by model prediction\_generator1

Execution 4

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-26 0.15, 2018-10-31 29491.1}] generated by model input\_reader

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-01 29492.2}] generated by model prediction\_generator1

Execution 5

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-26 0.15, 2018-10-31 29491.1}] generated by model input\_reader

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-01 29492.3}] generated by model prediction\_generator1

As expected, the model advances the date and generates a prediction.

Command:

./prediction\_generator\_test2.sh

Execution 1:

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-26 5.3, 2018-10-31 29491.1}] generated by model input\_reader

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-01 31616.5}] generated by model prediction\_generator1

Execution 2:

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-26 5.3, 2018-10-31 29491.1}] generated by model input\_reader

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-01 29491.1}] generated by model prediction\_generator1

Execution 3:

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-26 5.3, 2018-10-31 29491.1}] generated by model input\_reader

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-01 29604.1}] generated by model prediction\_generator1

Execution 4:

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-26 5.3, 2018-10-31 29491.1}] generated by model input\_reader

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-01 29502.4}] generated by model prediction\_generator1

Execution 5:

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-26 5.3, 2018-10-31 29491.1}] generated by model input\_reader

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-01 29495.1}] generated by model prediction\_generator1

The logs show the expected result, the model generates predicitons with the correct date and with a higher variability than in experiment 1.

bootstrap\_generator model:

To execute this model the main file main\_bootstrap\_generator\_model\_test.cpp was created, which compiles the BOOTSTRAP\_GENERATOR\_TEST executable in the bin folder, to execute this file 2 scripts were created in the folder scripts for the experiments: bootstrap\_generator\_test1.sh and bootstrap\_generator\_test2.sh

1. The result of executing this experiment is the following:

Command:

./bootstrap\_generator\_test1.sh

Execution 1

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-26 0.15, 2018-10-31 29491}] generated by model input\_reader

00:00:00:000

[random\_selector\_defs::out: {2018-10-26 0.15, 2018-10-31 29491}] generated by model random\_selector1

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-01 29492.1}] generated by model prediction\_generator1

Execution 2

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-26 0.15, 2018-10-31 29491}] generated by model input\_reader

00:00:00:000

[random\_selector\_defs::out: {2018-10-26 0.15, 2018-10-31 29491}] generated by model random\_selector1

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-01 29491.9}] generated by model prediction\_generator1

Execution 3

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-26 0.15, 2018-10-31 29491}] generated by model input\_reader

00:00:00:000

[random\_selector\_defs::out: {2018-10-26 0.15, 2018-10-31 29491}] generated by model random\_selector1

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-01 29491.7}] generated by model prediction\_generator1

The results are the expected.

1. To show the randomness the experiment was executed several times. The traces of execution in the messages log from execute the experiment is the following.

Command:

./bootstrap\_generator\_test2.sh

Execution 1:

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-29 -0.16, 2018-10-30 0.18, 2018-10-31 0.22, 2018-10-31 29491}] generated by model input\_reader

00:00:00:000

[random\_selector\_defs::out: {2018-10-31 0.22, 2018-10-31 29491}] generated by model random\_selector1

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-01 29491.8}] generated by model prediction\_generator1

Execution 2:

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-29 -0.16, 2018-10-30 0.18, 2018-10-31 0.22, 2018-10-31 29491}] generated by model input\_reader

00:00:00:000

[random\_selector\_defs::out: {2018-10-30 0.18, 2018-10-31 29491}] generated by model random\_selector1

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-01 29491.9}] generated by model prediction\_generator1

Execution 3:

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-29 -0.16, 2018-10-30 0.18, 2018-10-31 0.22, 2018-10-31 29491}] generated by model input\_reader

00:00:00:000

[random\_selector\_defs::out: {2018-10-29 -0.16, 2018-10-31 29491}] generated by model random\_selector1

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-01 29492}] generated by model prediction\_generator1

Execution 4:

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-29 -0.16, 2018-10-30 0.18, 2018-10-31 0.22, 2018-10-31 29491}] generated by model input\_reader

00:00:00:000

[random\_selector\_defs::out: {2018-10-31 0.22, 2018-10-31 29491}] generated by model random\_selector1

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-01 29492.3}] generated by model prediction\_generator1

Execution 5:

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-29 -0.16, 2018-10-30 0.18, 2018-10-31 0.22, 2018-10-31 29491}] generated by model input\_reader

00:00:00:000

[random\_selector\_defs::out: {2018-10-30 0.18, 2018-10-31 29491}] generated by model random\_selector1

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-01 29492.2}] generated by model prediction\_generator1

The logs show the expected result, the model generates different predictions.

APS model:

To execute this model the main file main.cpp was created in the folder top\_model, which compiles the APS executable in the bin folder , to execute this file 1 scripts was created in the folder scripts for the experiment: APS.sh

1. The result of executing this experiment is the following:

Command:

./APS.sh

Execution 1

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-25 29335, 2018-10-26 29369.9, 2018-10-29 28898.2, 2018-10-30 29425.8, 2018-10-31 29491.1, 2018-11-01 31308.6, 2018-11-02 31419.3, 2018-11-05 31429.3, 2018-11-07 31404.7, 2018-11-08 30736.4, 2018-11-09 29912.5, 2018-11-12 29423.3, 2018-11-13 28543.4, 2018-11-14 29372.5, 2018-11-15 30473.8, 2018-11-16 31109.2}] generated by model input\_reader

00:00:00:000

[daily\_return\_generator\_defs::out: {2018-10-26 0.00118828, 2018-10-29 -0.0161886, 2018-10-30 0.0180908, 2018-10-31 0.00221737, 2018-11-01 0.059805, 2018-11-02 0.00352727, 2018-11-05 0.00031953, 2018-11-07 -0.000783376, 2018-11-08 -0.0215093, 2018-11-09 -0.0271718, 2018-11-12 -0.0164892, 2018-11-13 -0.0303601, 2018-11-14 0.0286334, 2018-11-15 0.0368078, 2018-11-16 0.0206372, 2018-11-16 31109.2}] generated by model daily\_return\_generator1

00:00:00:000

[random\_selector\_defs::out: {2018-11-14 0.0286334, 2018-11-16 31109.2}] generated by model random\_selector1

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-17 31110.3}] generated by model prediction\_generator1

Execution 2

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-25 29335, 2018-10-26 29369.9, 2018-10-29 28898.2, 2018-10-30 29425.8, 2018-10-31 29491.1, 2018-11-01 31308.6, 2018-11-02 31419.3, 2018-11-05 31429.3, 2018-11-07 31404.7, 2018-11-08 30736.4, 2018-11-09 29912.5, 2018-11-12 29423.3, 2018-11-13 28543.4, 2018-11-14 29372.5, 2018-11-15 30473.8, 2018-11-16 31109.2}] generated by model input\_reader

00:00:00:000

[daily\_return\_generator\_defs::out: {2018-10-26 0.00118828, 2018-10-29 -0.0161886, 2018-10-30 0.0180908, 2018-10-31 0.00221737, 2018-11-01 0.059805, 2018-11-02 0.00352727, 2018-11-05 0.00031953, 2018-11-07 -0.000783376, 2018-11-08 -0.0215093, 2018-11-09 -0.0271718, 2018-11-12 -0.0164892, 2018-11-13 -0.0303601, 2018-11-14 0.0286334, 2018-11-15 0.0368078, 2018-11-16 0.0206372, 2018-11-16 31109.2}] generated by model daily\_return\_generator1

00:00:00:000

[random\_selector\_defs::out: {2018-11-12 -0.0164892, 2018-11-16 31109.2}] generated by model random\_selector1

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-17 31110.2}] generated by model prediction\_generator1

Execution 3

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-25 29335, 2018-10-26 29369.9, 2018-10-29 28898.2, 2018-10-30 29425.8, 2018-10-31 29491.1, 2018-11-01 31308.6, 2018-11-02 31419.3, 2018-11-05 31429.3, 2018-11-07 31404.7, 2018-11-08 30736.4, 2018-11-09 29912.5, 2018-11-12 29423.3, 2018-11-13 28543.4, 2018-11-14 29372.5, 2018-11-15 30473.8, 2018-11-16 31109.2}] generated by model input\_reader

00:00:00:000

[daily\_return\_generator\_defs::out: {2018-10-26 0.00118828, 2018-10-29 -0.0161886, 2018-10-30 0.0180908, 2018-10-31 0.00221737, 2018-11-01 0.059805, 2018-11-02 0.00352727, 2018-11-05 0.00031953, 2018-11-07 -0.000783376, 2018-11-08 -0.0215093, 2018-11-09 -0.0271718, 2018-11-12 -0.0164892, 2018-11-13 -0.0303601, 2018-11-14 0.0286334, 2018-11-15 0.0368078, 2018-11-16 0.0206372, 2018-11-16 31109.2}] generated by model daily\_return\_generator1

00:00:00:000

[random\_selector\_defs::out: {2018-10-29 -0.0161886, 2018-11-16 31109.2}] generated by model random\_selector1

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-17 31110.3}] generated by model prediction\_generator1

Execution 4

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-25 29335, 2018-10-26 29369.9, 2018-10-29 28898.2, 2018-10-30 29425.8, 2018-10-31 29491.1, 2018-11-01 31308.6, 2018-11-02 31419.3, 2018-11-05 31429.3, 2018-11-07 31404.7, 2018-11-08 30736.4, 2018-11-09 29912.5, 2018-11-12 29423.3, 2018-11-13 28543.4, 2018-11-14 29372.5, 2018-11-15 30473.8, 2018-11-16 31109.2}] generated by model input\_reader

00:00:00:000

[daily\_return\_generator\_defs::out: {2018-10-26 0.00118828, 2018-10-29 -0.0161886, 2018-10-30 0.0180908, 2018-10-31 0.00221737, 2018-11-01 0.059805, 2018-11-02 0.00352727, 2018-11-05 0.00031953, 2018-11-07 -0.000783376, 2018-11-08 -0.0215093, 2018-11-09 -0.0271718, 2018-11-12 -0.0164892, 2018-11-13 -0.0303601, 2018-11-14 0.0286334, 2018-11-15 0.0368078, 2018-11-16 0.0206372, 2018-11-16 31109.2}] generated by model daily\_return\_generator1

00:00:00:000

[random\_selector\_defs::out: {2018-11-01 0.059805, 2018-11-16 31109.2}] generated by model random\_selector1

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-17 31110.3}] generated by model prediction\_generator1

Execution 5

00:00:00:000

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {}] generated by model input\_reader

00:00:00:000

[cadmium::basic\_models::pdevs::iestream\_input\_defs<time\_series\_element>::out: {2018-10-25 29335, 2018-10-26 29369.9, 2018-10-29 28898.2, 2018-10-30 29425.8, 2018-10-31 29491.1, 2018-11-01 31308.6, 2018-11-02 31419.3, 2018-11-05 31429.3, 2018-11-07 31404.7, 2018-11-08 30736.4, 2018-11-09 29912.5, 2018-11-12 29423.3, 2018-11-13 28543.4, 2018-11-14 29372.5, 2018-11-15 30473.8, 2018-11-16 31109.2}] generated by model input\_reader

00:00:00:000

[daily\_return\_generator\_defs::out: {2018-10-26 0.00118828, 2018-10-29 -0.0161886, 2018-10-30 0.0180908, 2018-10-31 0.00221737, 2018-11-01 0.059805, 2018-11-02 0.00352727, 2018-11-05 0.00031953, 2018-11-07 -0.000783376, 2018-11-08 -0.0215093, 2018-11-09 -0.0271718, 2018-11-12 -0.0164892, 2018-11-13 -0.0303601, 2018-11-14 0.0286334, 2018-11-15 0.0368078, 2018-11-16 0.0206372, 2018-11-16 31109.2}] generated by model daily\_return\_generator1

00:00:00:000

[random\_selector\_defs::out: {2018-11-13 -0.0303601, 2018-11-16 31109.2}] generated by model random\_selector1

00:00:00:000

[prediction\_generator\_defs::out: {2018-11-17 31110.2}] generated by model prediction\_generator1

As expected the models generates predictions for the price on the next day.