

The Use of Cognitive Digital Twins on an IoT System for Edge Resilience and Anomaly Detection

Contributors:

- Meriem Smati
- Boubou Thiam Niang
- Jannik Laval

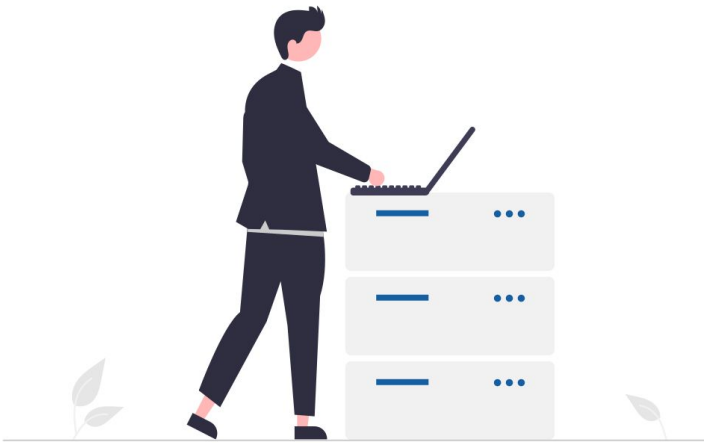
Presented By:

- Meriem Smati

Part I

Section I.1 : Problem Statement, Objectives and Backgrounds

Part I.1: Problem Statement



- Resilience.
- Maintenance.
- IoT Systems.

How to use Digital Twins for IoT Systems resilience?

Part I.1: Objectives

- Define the DT's concepts.
- Analyze different articles to create a new improved framework.

Part I.1: Different Involved Domains

- Internet of Things
- Artificial Intelligence
- Machine Learning
- Deep Learning
- Digital Twins

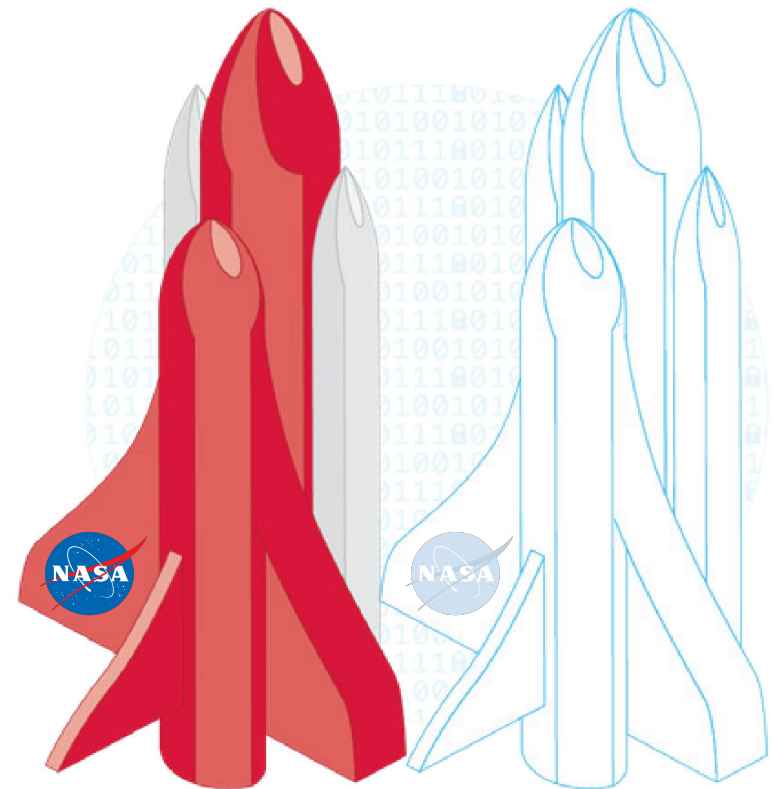
Part I.1: Digital Twins Background

1970

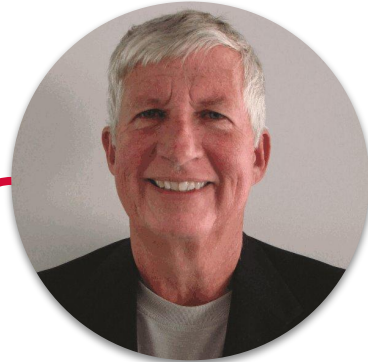
- The concept of a “Twin” has its roots in the NASA Apollo program of the 1970s.



- Create a replica of space vehicles on Earth.



Part I.1: Digital Twins Background



Michael Grieves



University of Michigan

2003

- Michael Grieves proposed the idea of a “Digital Twin” in his Product Life-Cycle Management (PLM) course.

Part I.1: Digital Twins Background

Michael Grieves, a professor at the University of Michigan, coins the term "digital twin" to describe a virtual model of a physical object.

GE introduces its digital twin technology for jet engines, allowing for real-time monitoring and predictive maintenance.

Siemens introduces its MindSphere cloud platform for digital twins, enabling companies to create, simulate, and analyze their products and processes.

1960s

2002

2010s

2015

2017

2018

The concept of "mirrored systems" emerges in NASA's space program to monitor and control spacecraft remotely.

Advances in sensors, big data, and cloud computing make it easier to create and manage digital twins of complex systems like buildings, factories, and entire cities.

The Industrial Internet Consortium publishes a reference architecture for digital twins, promoting standardization and interoperability.

DTs continue to evolve, becoming more sophisticated and integrated with other technologies like AI, ML & AR. They are increasingly used across industries for a wide range of applications.

Section 1.2 : State of the Art

Part I.2: State of The Art

State of the Art

- 
- Digital Twins Concepts
 - Digital Twins for resilience

Part I.2.1: Digital Twin's Principales

Digital Twin's Definitions

Digital Twins lack a universal definition and highly depend on the use case and the domain of application.

Part I.2.1: Digital Twin's Principales

Digital Twin's Definitions

In Aerospace:

A DT is an integrated multiphysics, multiscale, probabilistic simulation of an as-built vehicle that uses the physical models and other relevant information to accurately replicate the life and behavior of its corresponding flying counterpart.

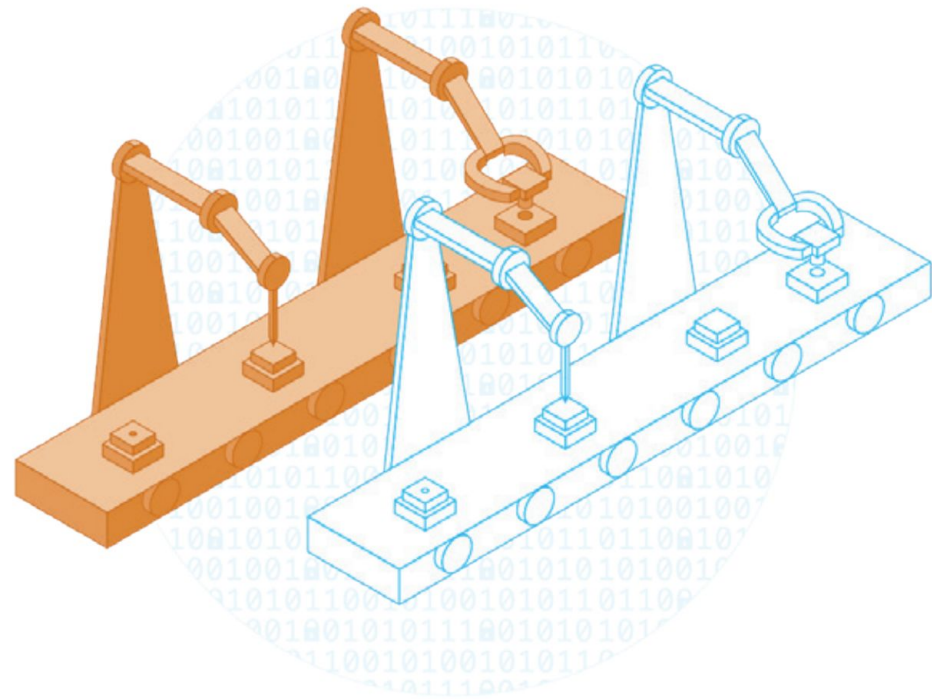


Part I.2.1: Digital Twin's Principales

Digital Twin's Definitions

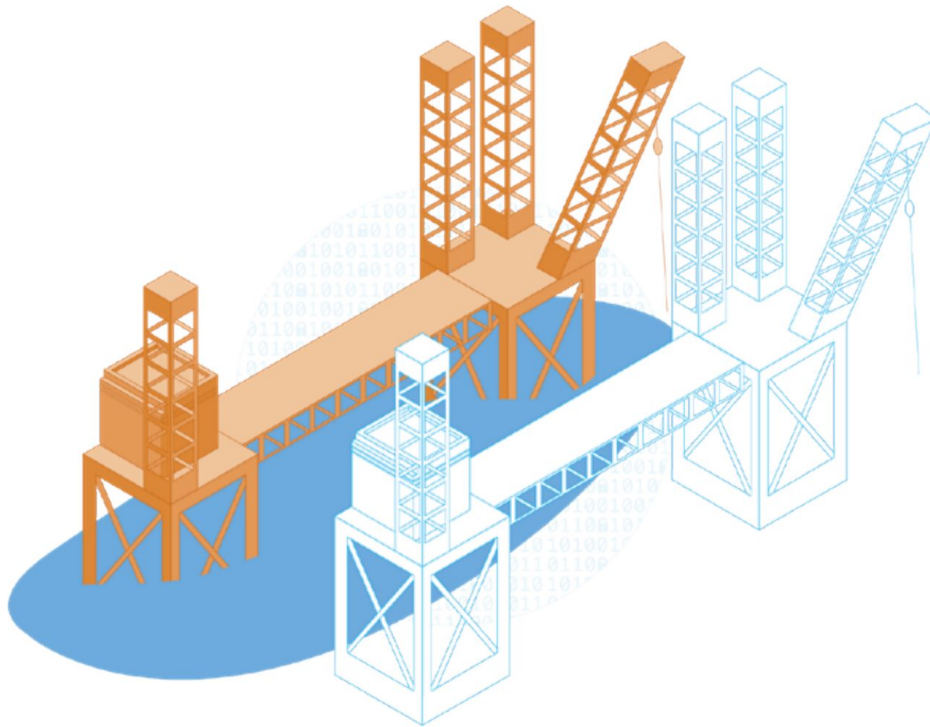
In the Industry:

A DT is an evolving digital profile of the historical and current behavior of a physical object that helps optimize business performance. It is based on massive, cumulative, real-time, real-world data measurements across an array of dimensions.



Part I.2.1: Digital Twin's Principales

Digital Twin's Definitions



In Engineering:

A DT is a digital replica of physical assets, processes, and systems that can be used for various purposes, such as simulation, optimization, and monitoring.

Part I.2.1: Digital Twin's Principales

Digital Twin's Definitions

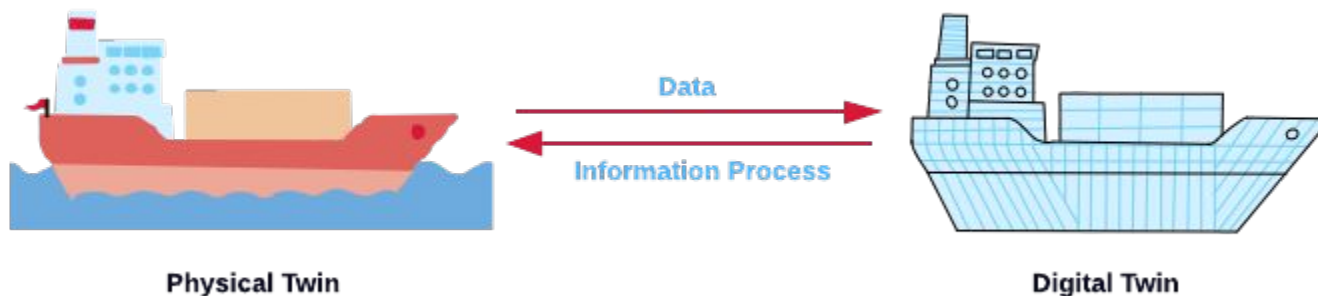
And much more other definitions have been conducted in Healthcare, Manufacturing, Agriculture etc.

Part I.2.1: Digital Twin's Principales

Digital Twin's Definitions

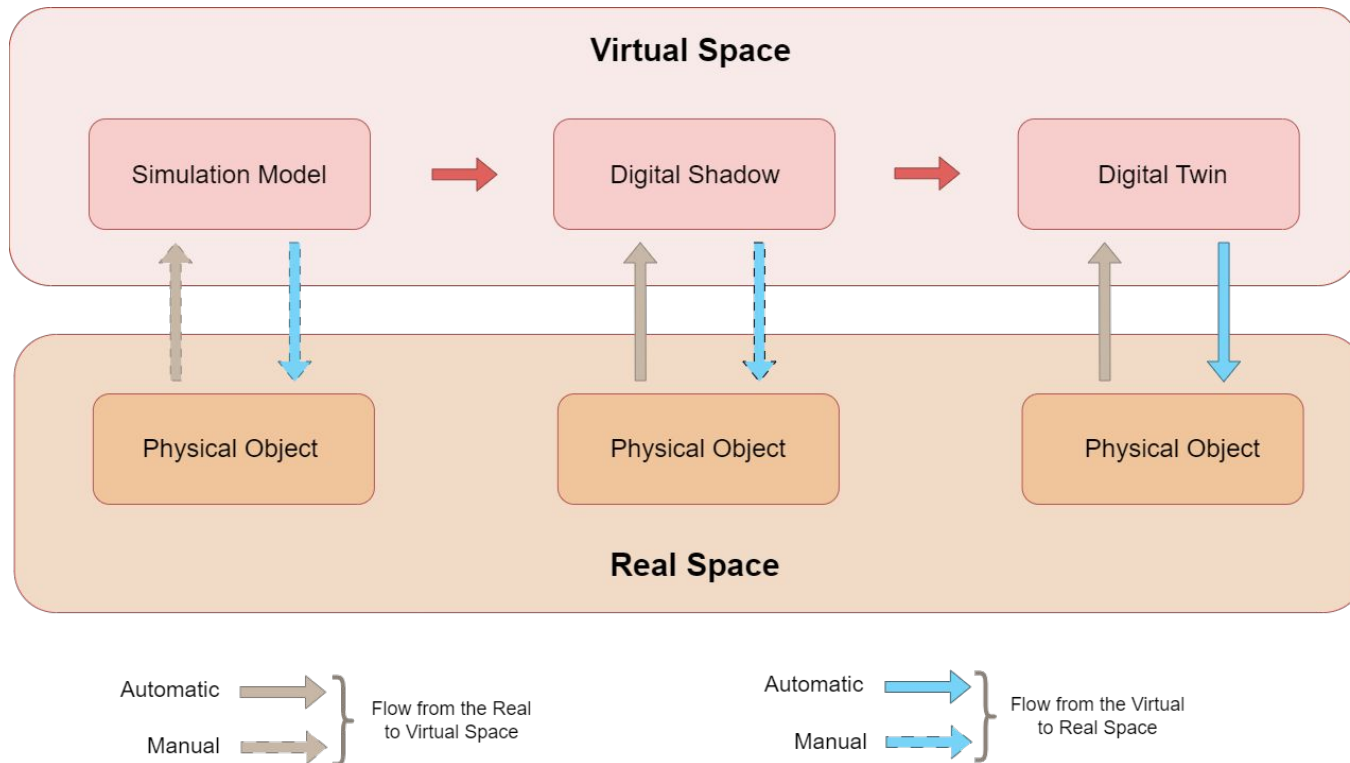
We can conclude the following definition for “Digital Twins in the IoT Domain”:

“The combination of virtual machines and computer-based models that enable the simulation, emulation, or mirroring of the behavior and characteristics of a physical entity”.



Part I.2.1: Digital Twin's Principales

Digital Twin's Predecessors



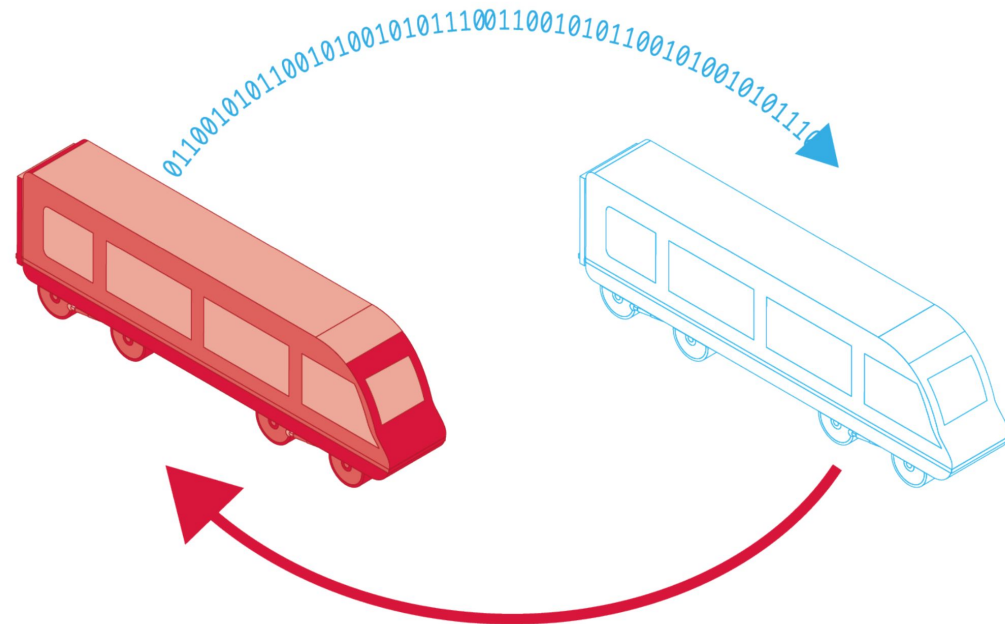
Part I.2.1: Digital Twin's Principales

Digital Twin's Components

Required Components

- Physical Asset
- Digital Asset
- Continuous Relation

Bijection

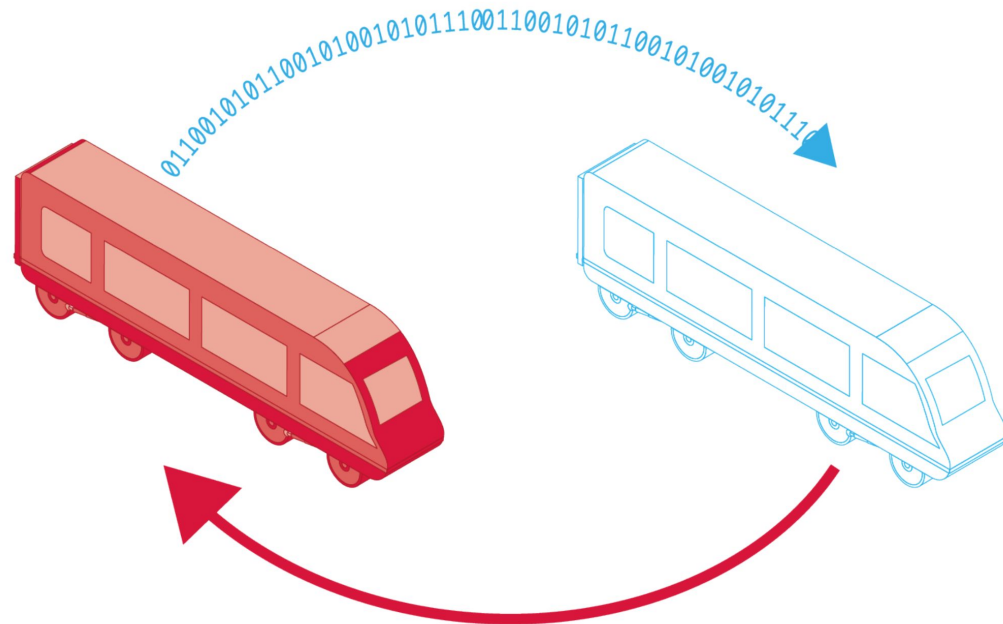


Part I.2.1: Digital Twin's Principales

Digital Twin's Components

Optional Components

- Time Continuous Data
- Time-series data
- Knowledge Database
- Security
- ML
- Evaluation metrics
- IoT
- etc.

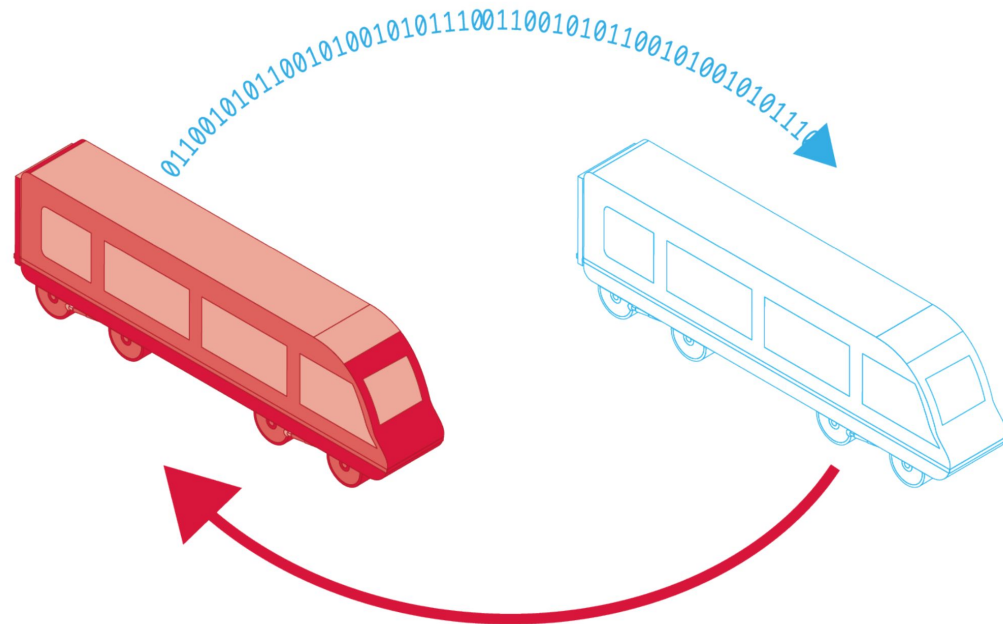


Part I.2.1: Digital Twin's Principales

Digital Twin's Components

Optional Components

- Time Continuous Data
- Time-series data
- Knowledge Database
- Security
- ML
- Evaluation metrics
- IoT
- etc.



Part I.2.1: Digital Twin's Principales

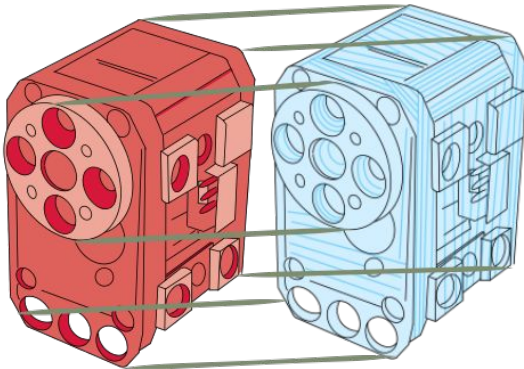
Digital Twin's Types

- 
- Level of granularity
 - Twin's dynamism

Part I.2.1: Digital Twin's Principales

Digital Twin's Types

Level of granularity



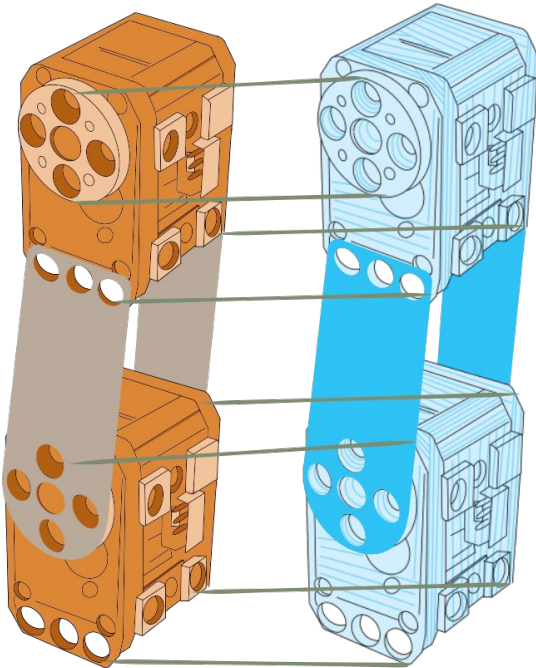
Components Twins:

The basic unit of a DT and the smallest example of a functioning component.

Part I.2.1: Digital Twin's Principales

Digital Twin's Types

Level of granularity



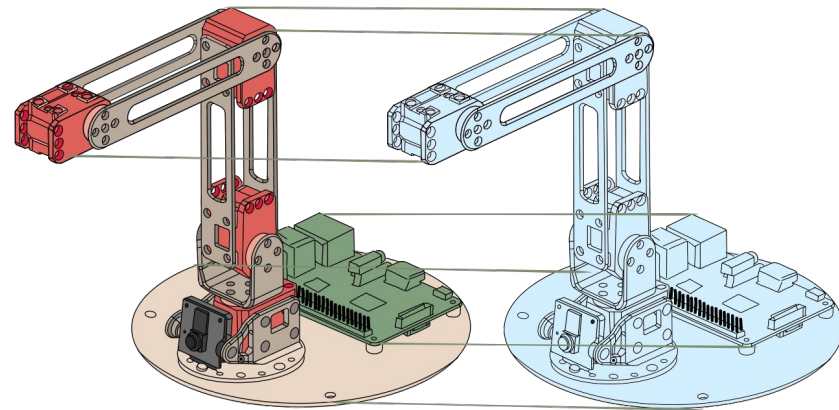
Assets Twins:

When two or more components work together.

Part I.2.1: Digital Twin's Principales

Digital Twin's Types

Level of granularity



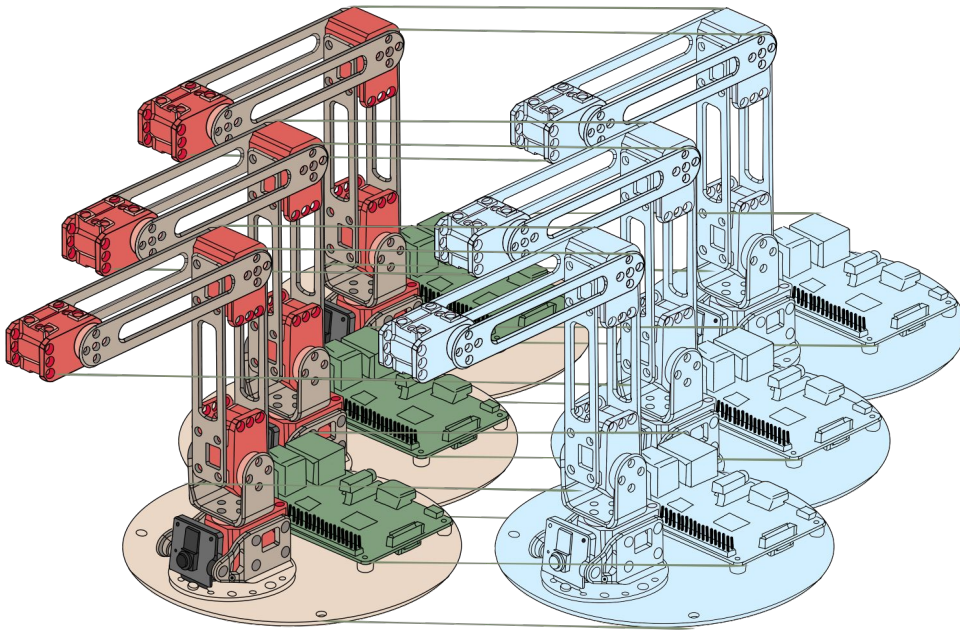
System Twins / Unit Twins:

It enables to detect different assets connected to form a whole functioning system.

Part I.2.1: Digital Twin's Principales

Digital Twin's Types

Level of granularity



Process Twins:

It is the macro level of magnification. It is the digitalization of entire business processes.

Part I.2.1: Digital Twin's Principales

Digital Twin's Types

Twin's Dynamism

a Dynamic DT:

Fed by live data flows from a physical asset. Insights and programmed instructions from the digital twin can then impact the physical twin using real-time control mechanisms.

a Static DT:

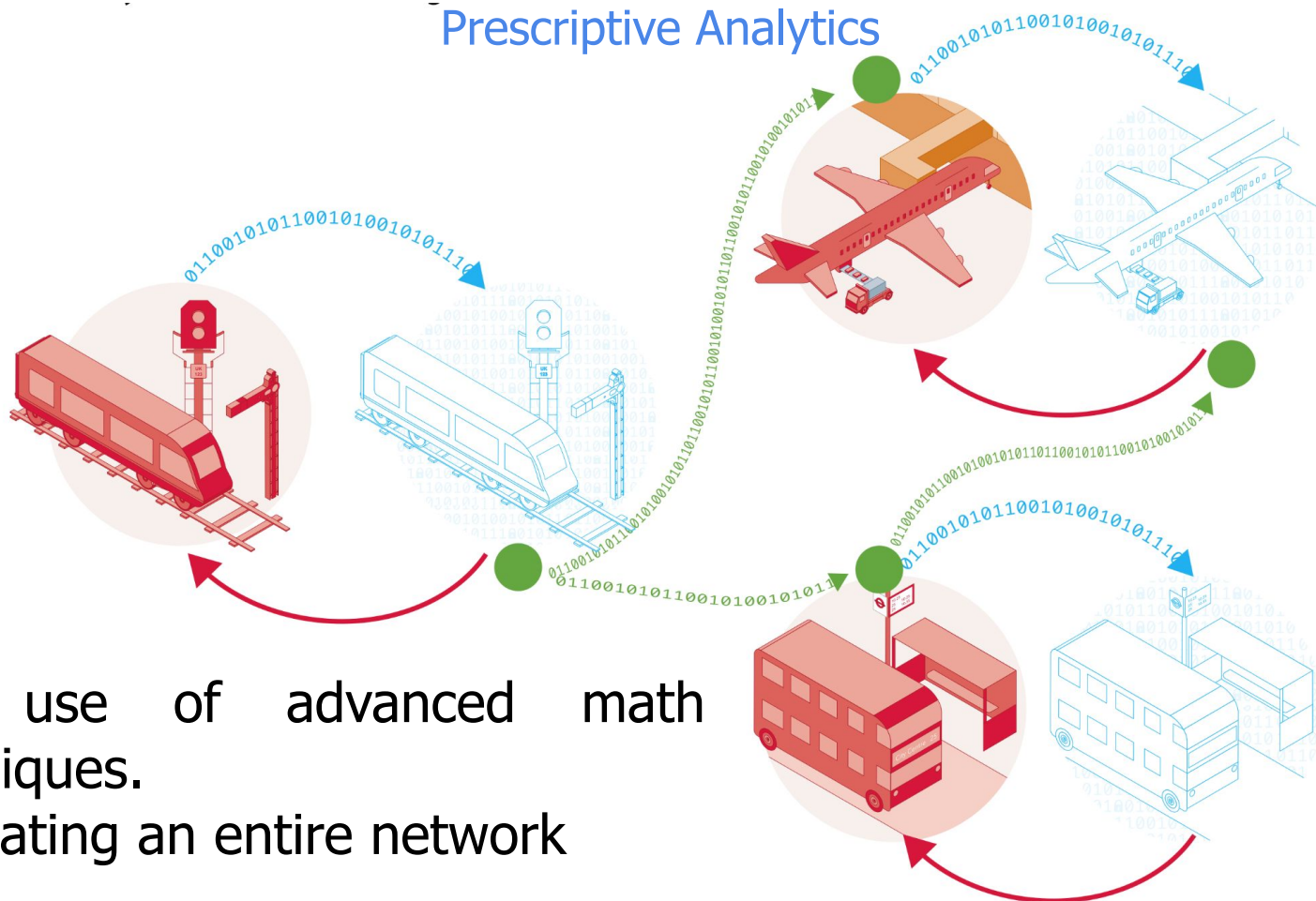
Changes periodically as long-term data about a physical asset are added in.

Part I.2.1: Digital Twin's Principales

Digital Twins with ML and DL

Two widely used ML areas in DT

Prescriptive Analytics



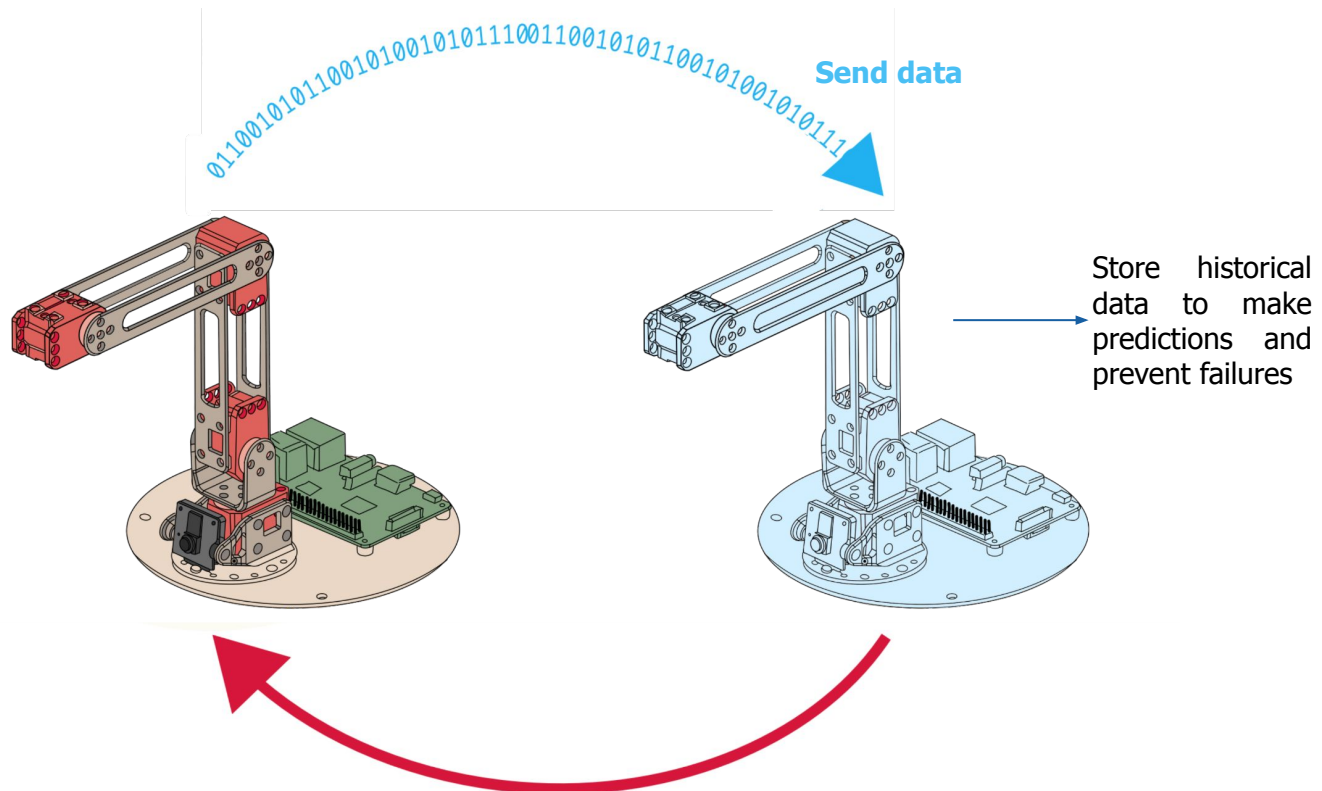
- The use of advanced math techniques.
- Simulating an entire network

Part I.2.1: Digital Twin's Principales

Digital Twins with ML and DL

Two widely used ML areas in DT

Diagnostic and Predictive Analytics



- Diagnose potential problems and predict future behaviors of the system.

Part I.2.2: Digital Twins for resilience

Numerous of Articles

In CPSs:

creation of conceptual
framework



Predict potential failures
and recommend actions
to prevent them,

In Production:

Use CDT for detecting
failures



Predict potential failures
and recommend actions
to prevent them,

In Healthcare:

Resilience of material



Predict potential failures

Part I.2.2: Digital Twins for resilience

Contribution

Creation of a CSDT

Cognitive Super-Digital Twin

Part II

Section II.1 : Realization

The scenario involves an IoT system aligned with the Physical Twin, featuring a total of four sensors that help detect whether the room is occupied or not:

- A temperature sensor.
- A humidity sensor.
- A light sensor.
- A CO2 sensor.

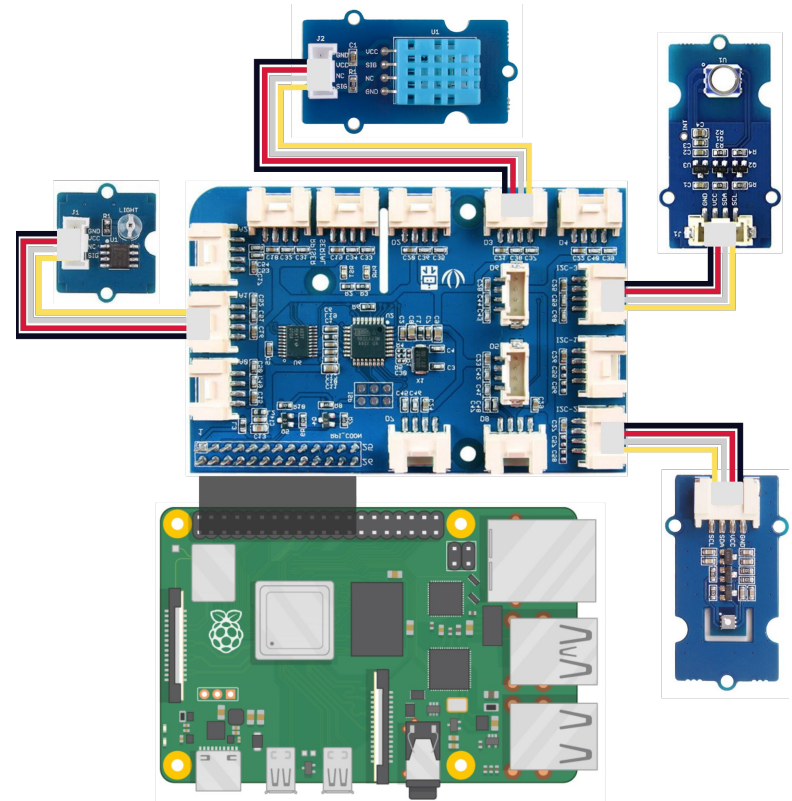
Each Sensor corresponds to a feature in the external dataset (Occupancy Detection Dataset).

The robot poppy ergo jr moves depending on the result of the model and the collected data.

Part II.1.2: Used Hardware and Technologies

First subsystem

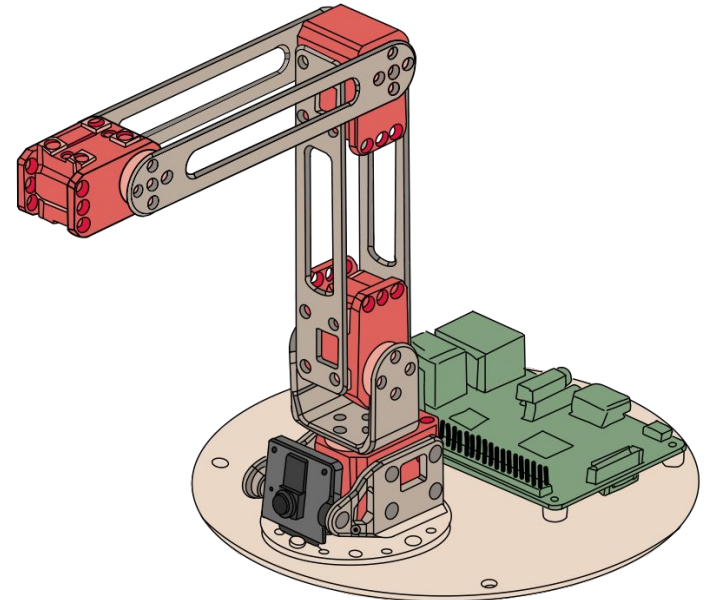
- Raspberry Pi 3 Model B.
- GrovePi+ add-on board.
- Grove Temperature & Humidity Sensor (DHT11).
- Grove Barometer (High-Accuracy)
- Grove Light Sensor.
- Grove VOC and eCO2 Gas Sensor (SGP30).



Part II.1.2: Used Hardware and Technologies

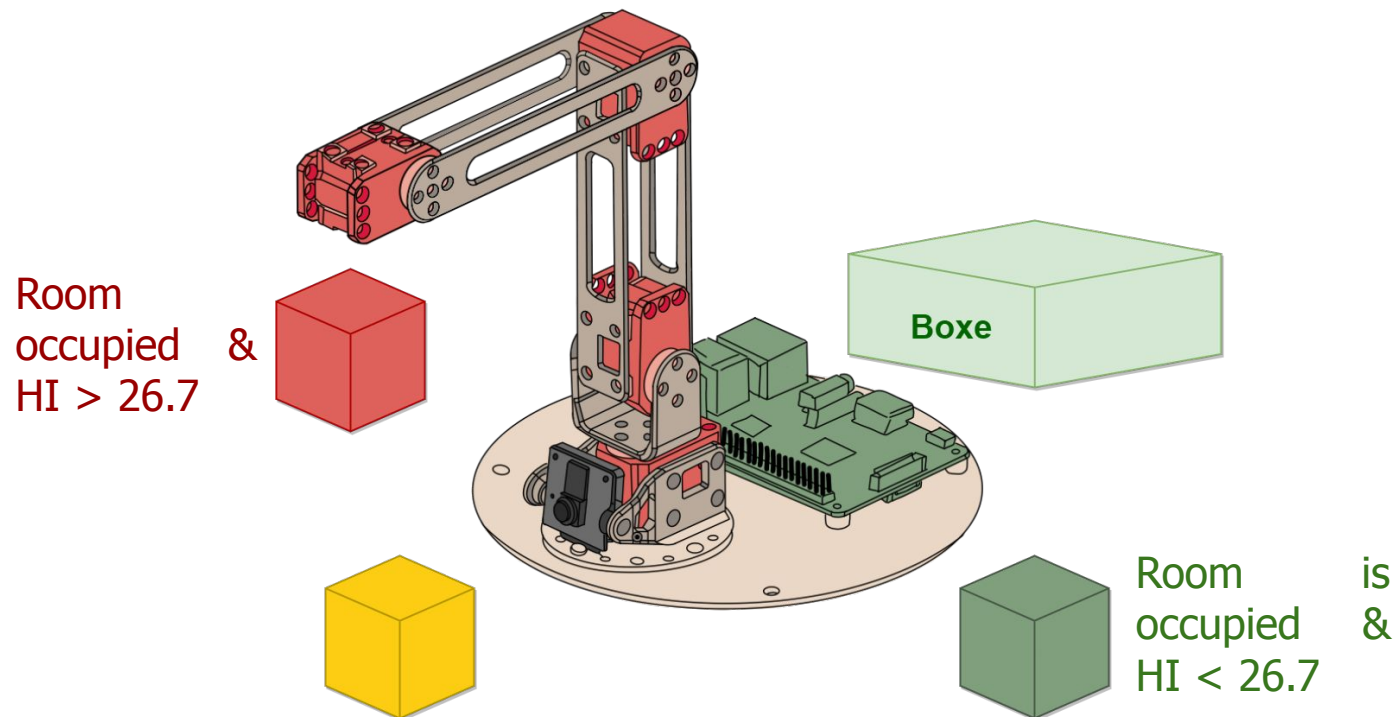
Second Subsystem - Poppy Ergo Jr

- Raspberry Pi 3 Model B.
- Six motors.
- 3D printed parts.



Part II.1.2: Used Hardware and Technologies

Second Subsystem - Poppy Ergo Jr



Part II.1.2: Used Hardware and Technologies

Second Subsystem - Poppy Ergo Jr

$$\begin{aligned} HI = & c_1 + c_2 \cdot T + c_3 \cdot RH + c_4 \cdot T \cdot RH \\ & + c_5 \cdot T^2 + c_6 \cdot RH^2 + c_7 \cdot T^2 \cdot RH \\ & + c_8 \cdot T \cdot RH^2 + c_9 \cdot T^2 \cdot RH^2 \end{aligned}$$

where :

HI is the Heat Index

T is the temperature in Celsius

RH is the relative humidity in percentage

c_1, c_2, \dots, c_9 are the coefficients specific to the equation

$$c_1 = -8.78469475556$$

$$c_2 = 1.61139411$$

$$c_3 = 2.33854883889$$

$$c_4 = -0.14611605$$

$$c_5 = -0.012308094$$

$$c_6 = -0.0164248277778$$

$$c_7 = 0.002211732$$

$$c_8 = 0.00072546$$

$$c_9 = -0.000003582$$

Part II.1.2: Used Hardware and Technologies

RabbitMQ - MQTT

- It is an extension to RabbitMQ that enables support for the MQTT protocol. MQTT is a lightweight messaging protocol designed for efficient communication between devices or client applications in constrained or unreliable networks.



Part II.1.2: Used Hardware and Technologies

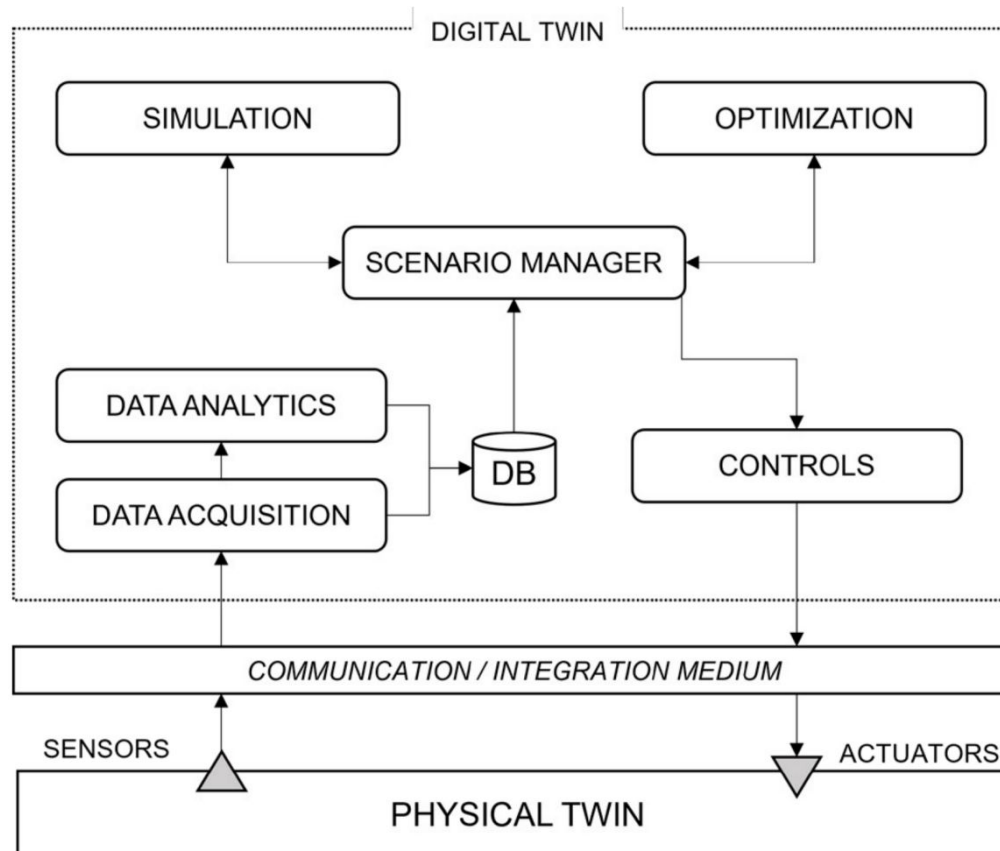
InfluxDB

- Open-source time series database written in Go programming language for storing and retrieving time series data.

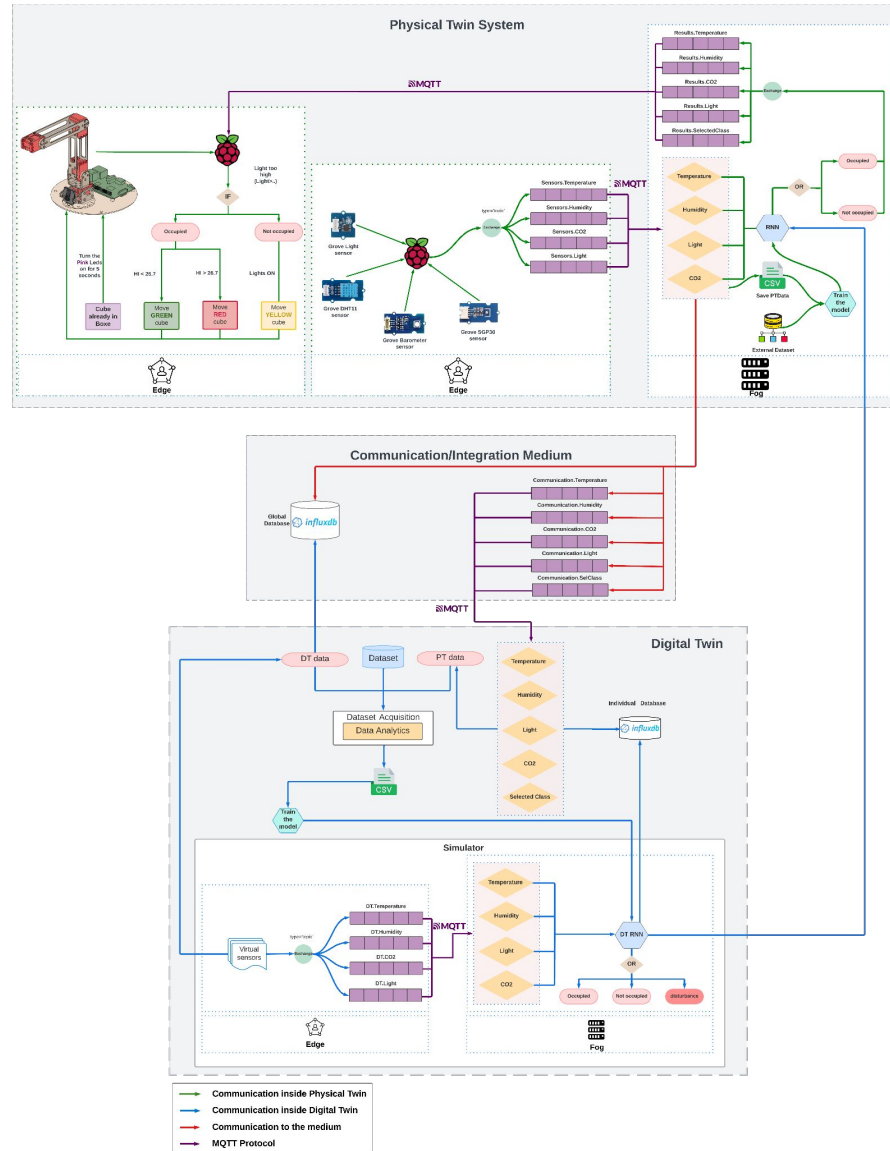


Part II.1.3: General Architecture

Inspired from:

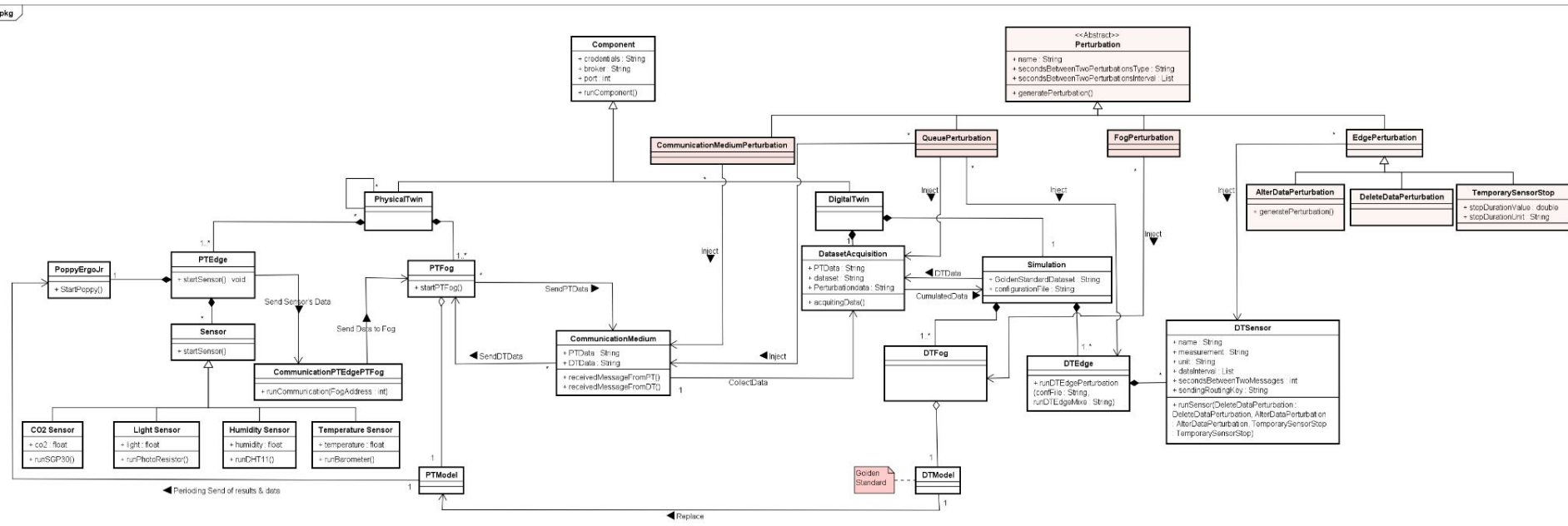


Part II.1.3: General Architecture



Part II.1.3: Class Diagram

pkg



Part II.1.4: Sources of Data

The Digital Twin has Three sources of data:

- External Dataset (Occupancy Detection Dataset).
- Physical Twin's Data.
- Disturbed Data.



Part II.1.4: Sources of Data

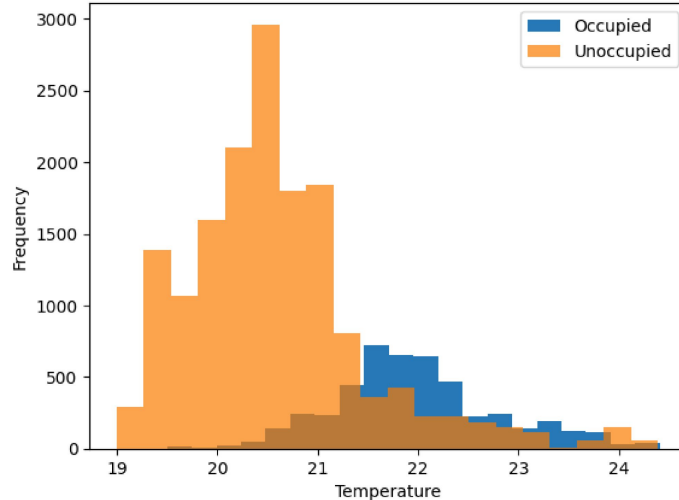
External Dataset

Date	Temperature	Humidity	Light	CO2	Occupancy
2015-02-02 14:19:00	23.7	26.272	585.2	749.2	1
2015-02-02 14:19:59	23.718	26.29	578.4	760.4	1
2015-02-02 14:21:00	23.73	26.23	572.66	769.66	1
2015-02-02 14:22:00	23.7225	26.125	493.75	774.75	1
2015-02-02 14:23:00	23.754	26.2	488.6	779	1

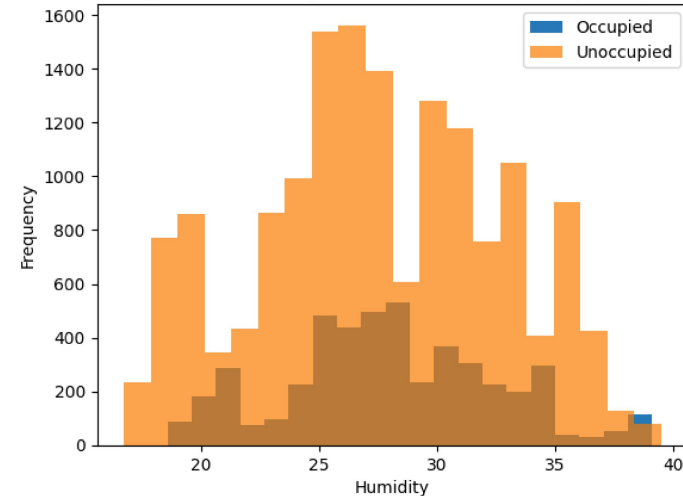
Part II.1.4: Sources of Data

External Dataset

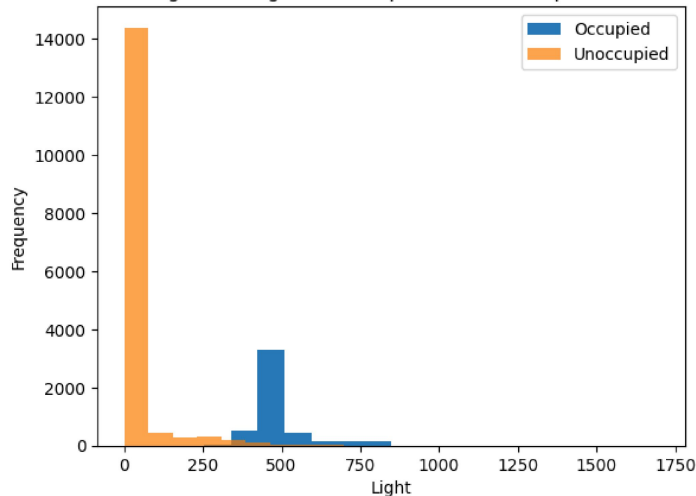
Histogram of Temperature for Occupied and Unoccupied States



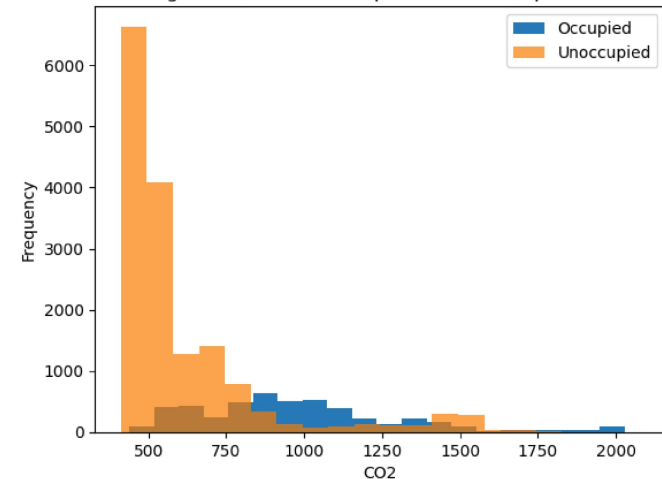
Histogram of Humidity for Occupied and Unoccupied States



Histogram of Light for Occupied and Unoccupied States



Histogram of CO2 for Occupied and Unoccupied States



Part II.1.4: Sources of Data

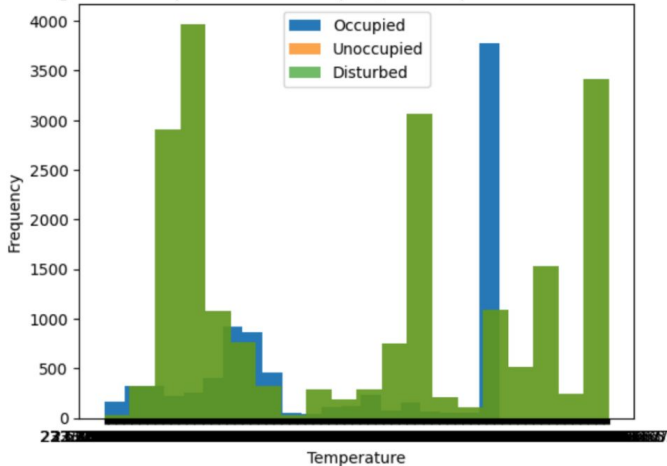
Physical Twin's Dataset

Date	Temperature	Humidity	Light	CO2	Occupancy
2023-06-09 15:33:00	26.12	17.0	766.0	809.0	1
2023-06-09 15:33:33	26.18	17.2	766.0	808.4	1
2023-06-09 15:34:03	25.73	18.23	572.66	769.66	1
2023-06-09 15:34:25	25.74	18.125	600.75	774.75	1
2023-06-09 15:35:06	25.754	17.2	568.6	779	1

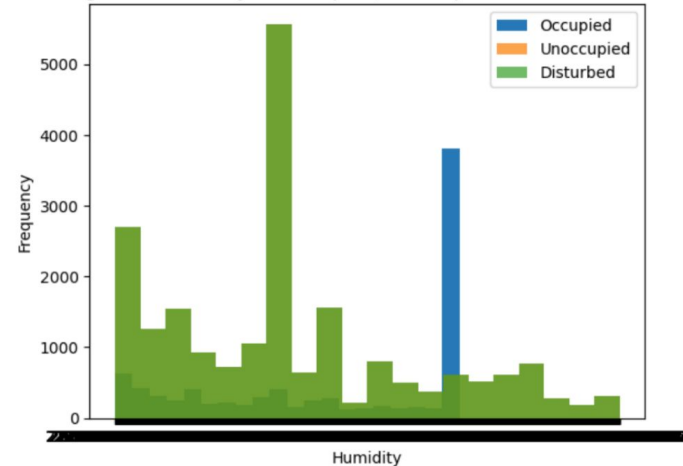
Part II.1.4: Sources of Data

Physical Twin's Dataset

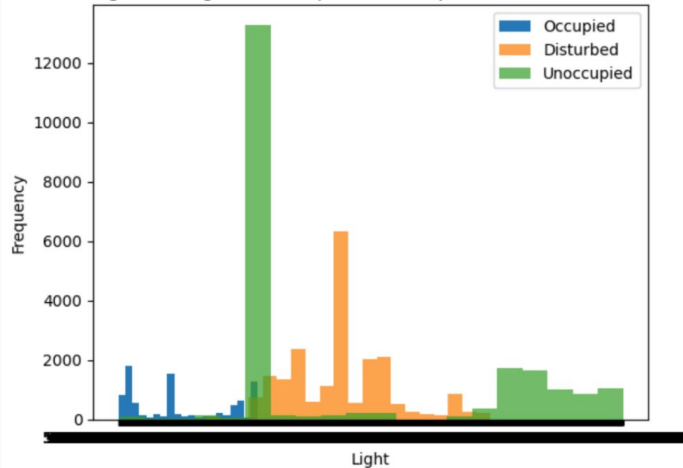
Histogram of Temperature for Occupied, Unoccupied and Disturbed States



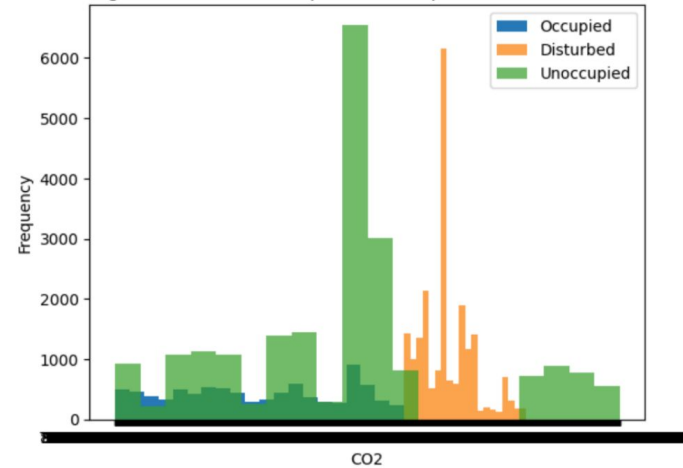
Histogram of Humidity for Occupied, Unoccupied and Disturbed States



Histogram of Light for Occupied, Unoccupied and Disturbed States



Histogram of CO2 for Occupied, Unoccupied and Disturbed States



Part II.1.4: Sources of Data

Disturbed Data

```
{
  "Sensor name": "Temperature 1",
  "Measurement": "Temperature",
  "topic": "DT/temperature",
  "Unit": "°C",
  "Data type": "double",
  "DTData": [
    {
      "Occupancy": 0,
      "Data Interval": [19.1, 24.39]
    },
    {
      "Occupancy": 1,
      "Data Interval": [20.29, 26]
    },
    {
      "Occupancy": 2,
      "Data Interval": [
        [-100, 10],
        [50, 1000]
      ]
    }
  ]
}
```

Part II.1.4: Selecting a Model for the Physical Twin

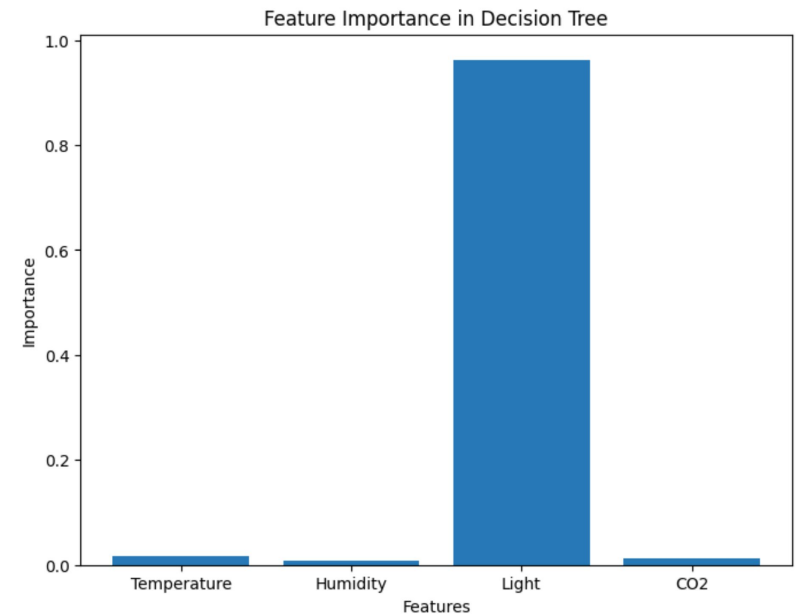
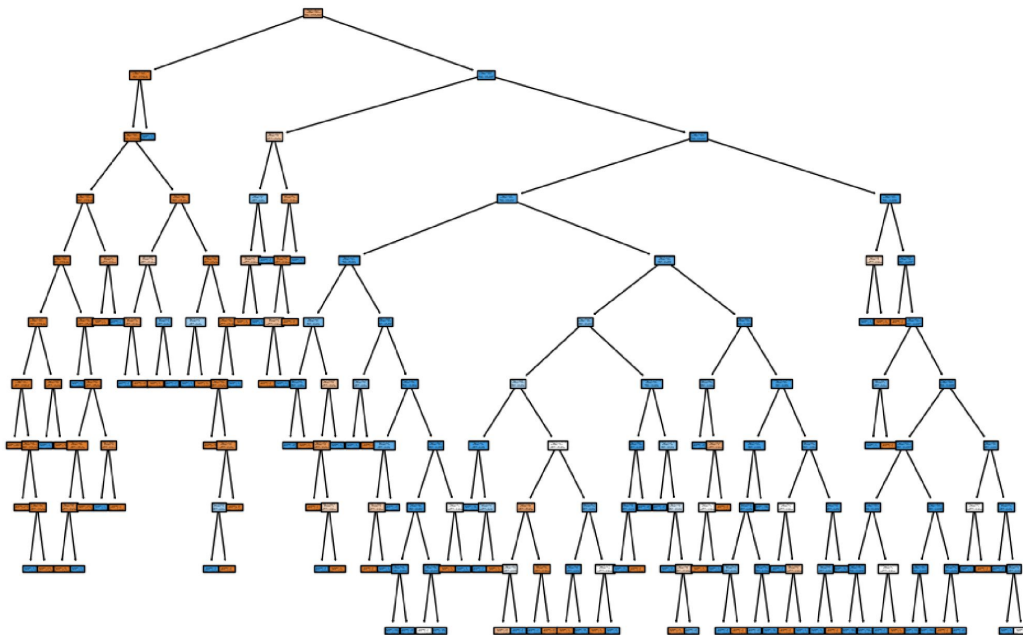
Train ML & DL models on the External dataset.

- Decision Trees.
- Random Forests.
- KNN.
- Naive Bayes.
- RNN.
- MLP.

The training is done either with GridSearch or RandomSearch

Part II.1.4: Selecting a Model for the Physical Twin

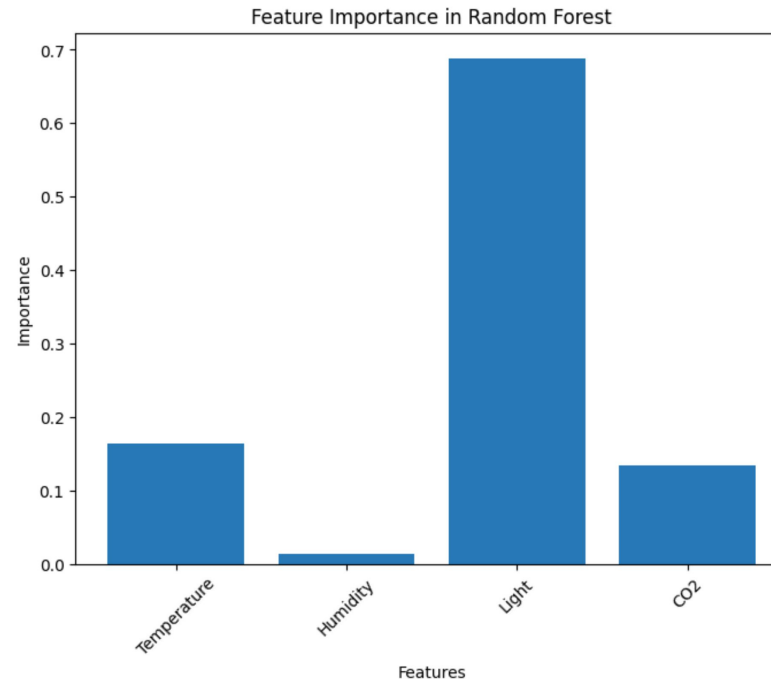
Physical Twin - Decision Tree



- Accuracy: 99.14%
- Precision: 97.73%
- Recall: 98.47%
- F1_Score: 98.10%

Part II.1.4: Selecting a Model for the Physical Twin

Physical Twin - Random Forest

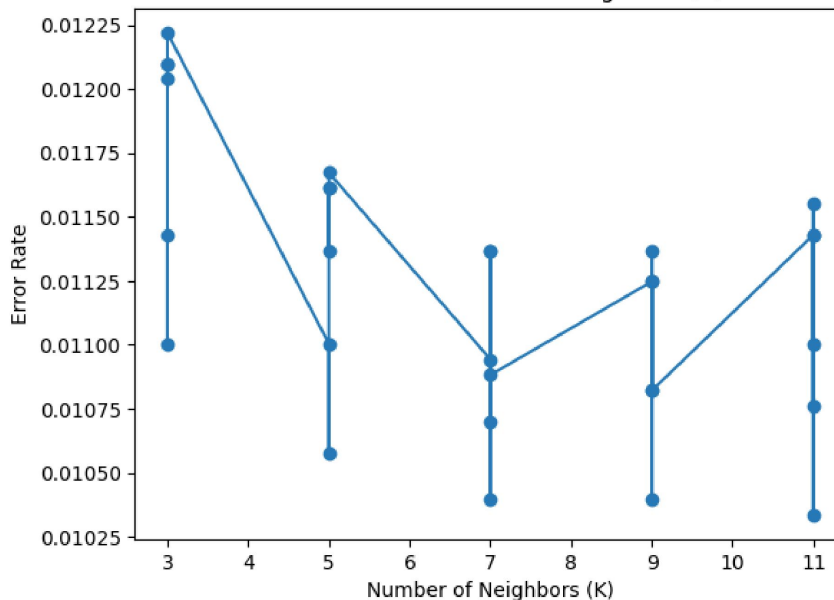


- Accuracy: 99.34%
- Precision: 97.85%
- Recall: 98.23%
- F1_Score: 98.54%

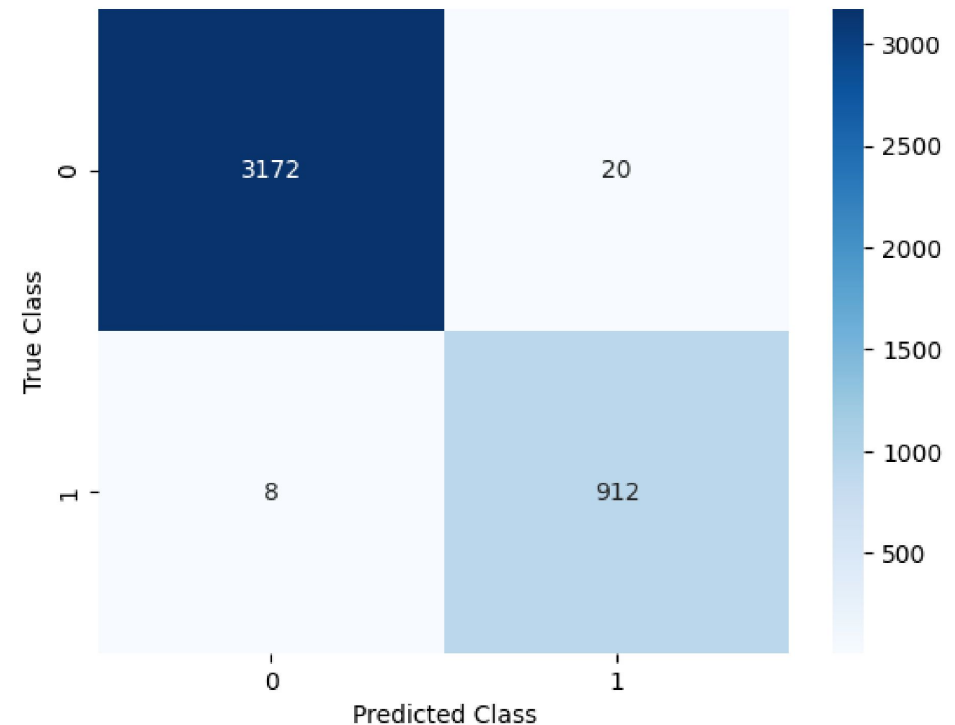
Part II.1.4: Selecting a Model for the Physical Twin

Physical Twin - KNN

Error Rate vs. Number of Neighbors (K)



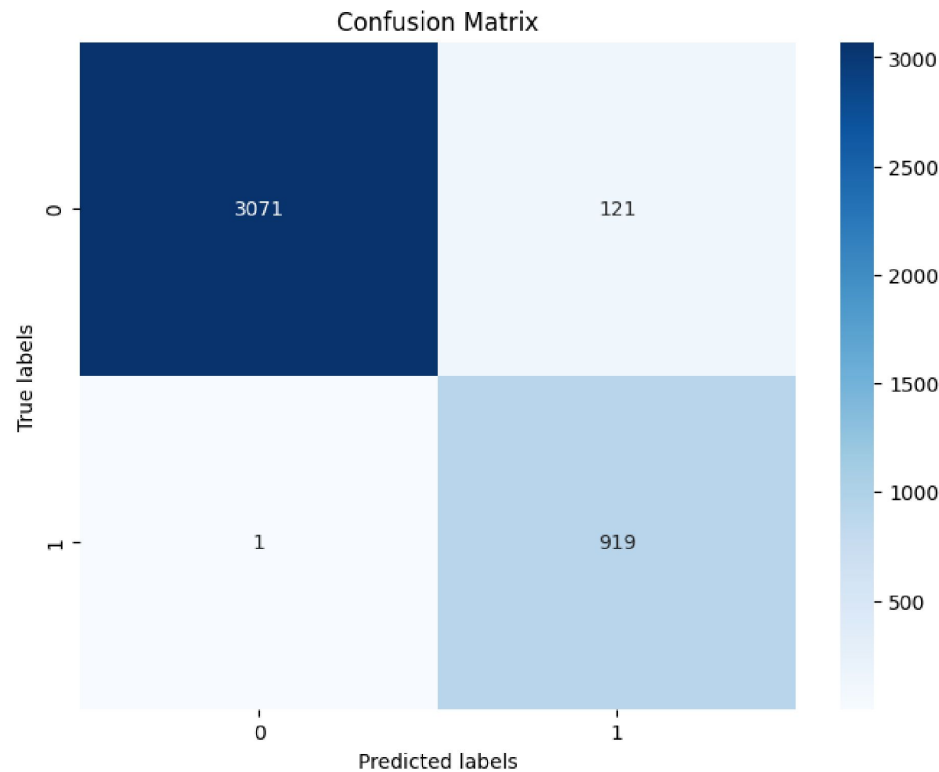
Confusion Matrix



• Accuracy: 99.31% • Precision: 97.85% • Recall: 99.13% • F1_Score: 98.48%

Part II.1.4: Selecting a Model for the Physical Twin

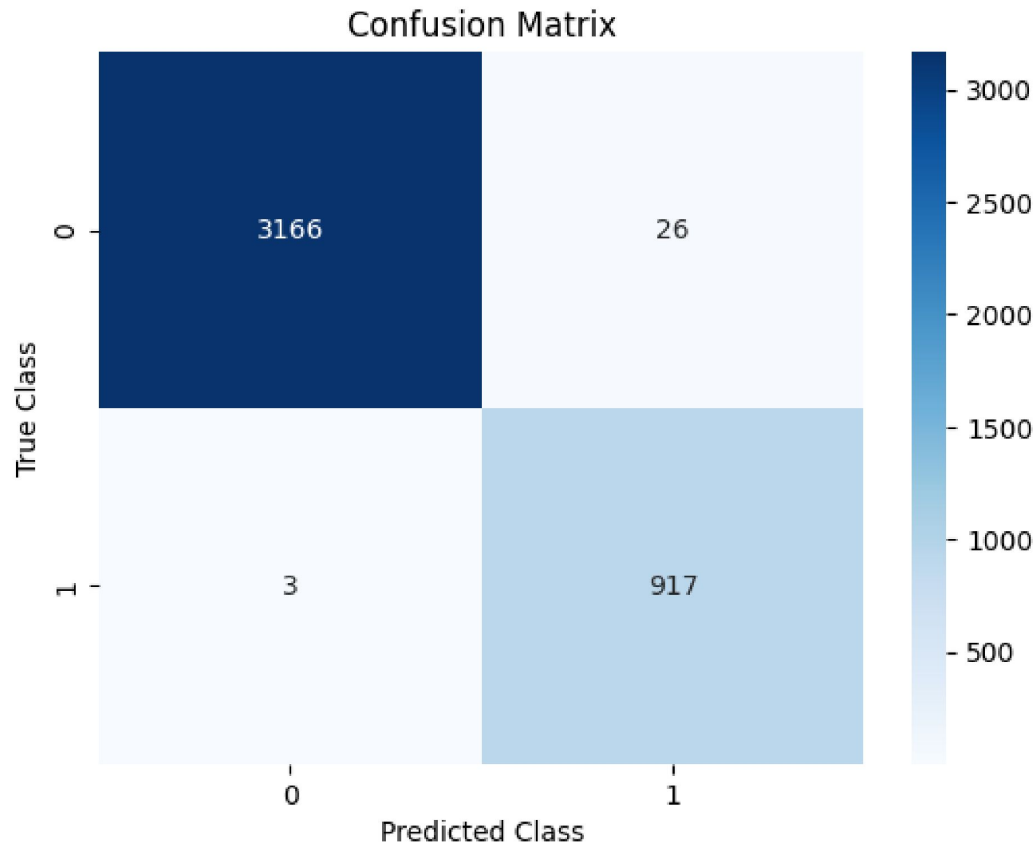
Physical Twin - Naive Bayes



- Accuracy: 97.03%
- Precision: 98.36%
- Recall: 99.89%
- F1_Score: 93.77%

Part II.1.4: Selecting a Model for the Physical Twin

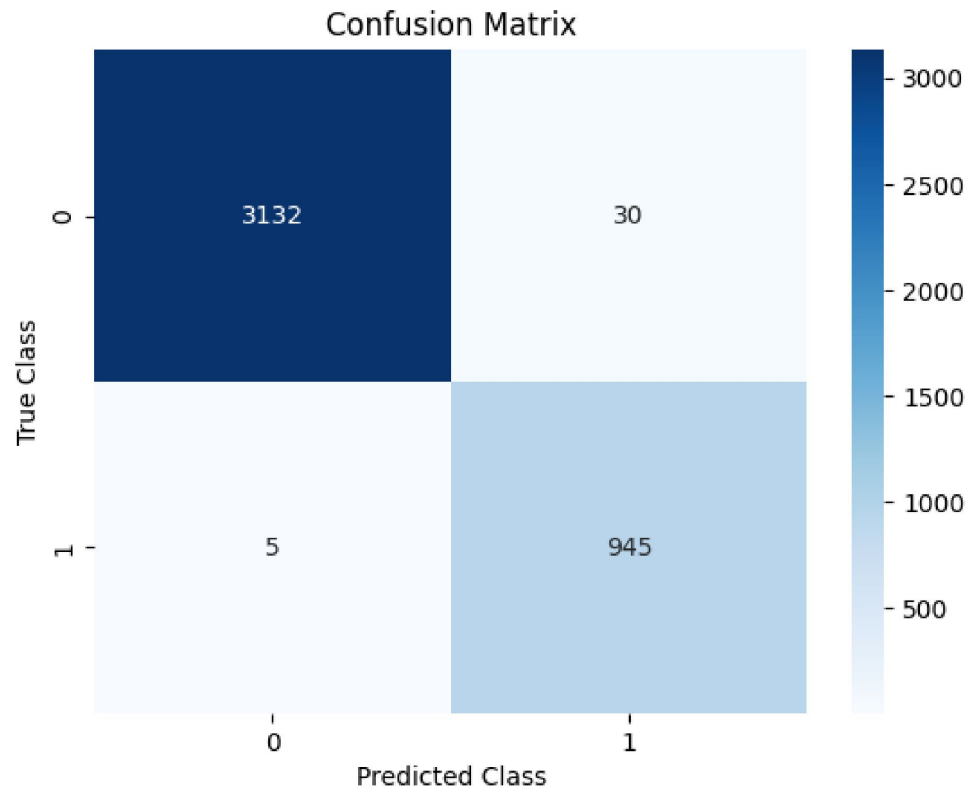
Physical Twin - MLP



- Accuracy: 97.27%
- Precision: 97.34%
- Recall: 99.46%
- F1_Score: 98.39%

Part II.1.4: Selecting a Model for the Physical Twin

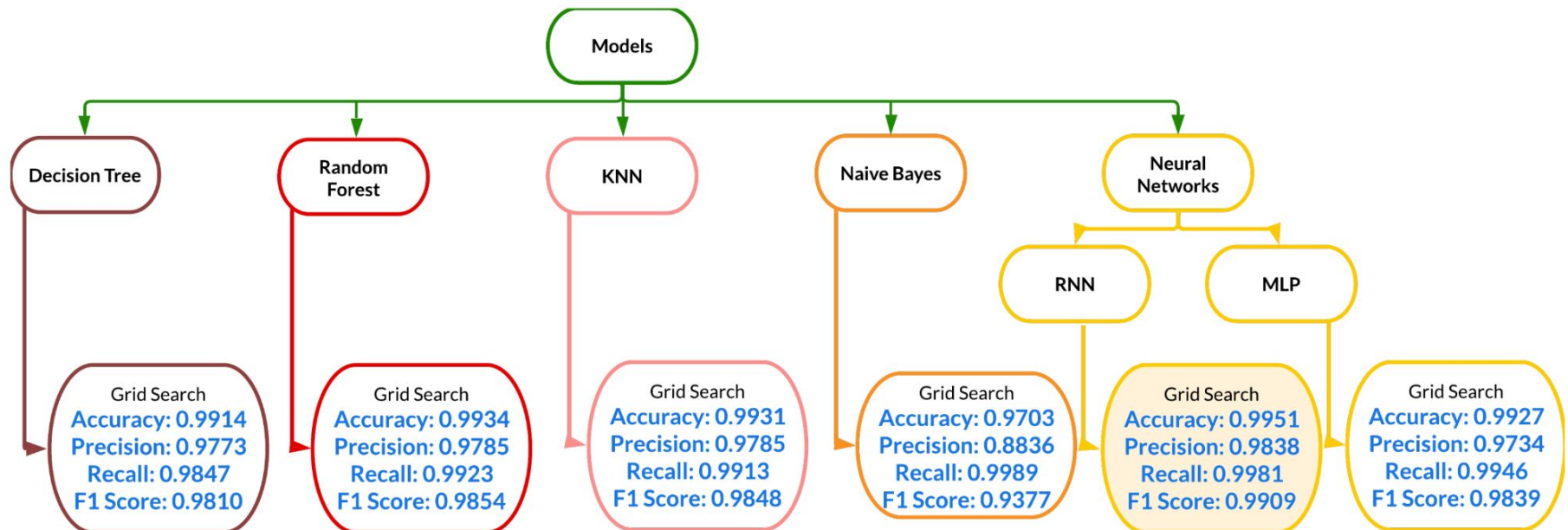
Physical Twin - RNN



- Accuracy: 99.51%
- Precision: 97.38%
- Recall: 99.81%
- F1_Score: 98.09%

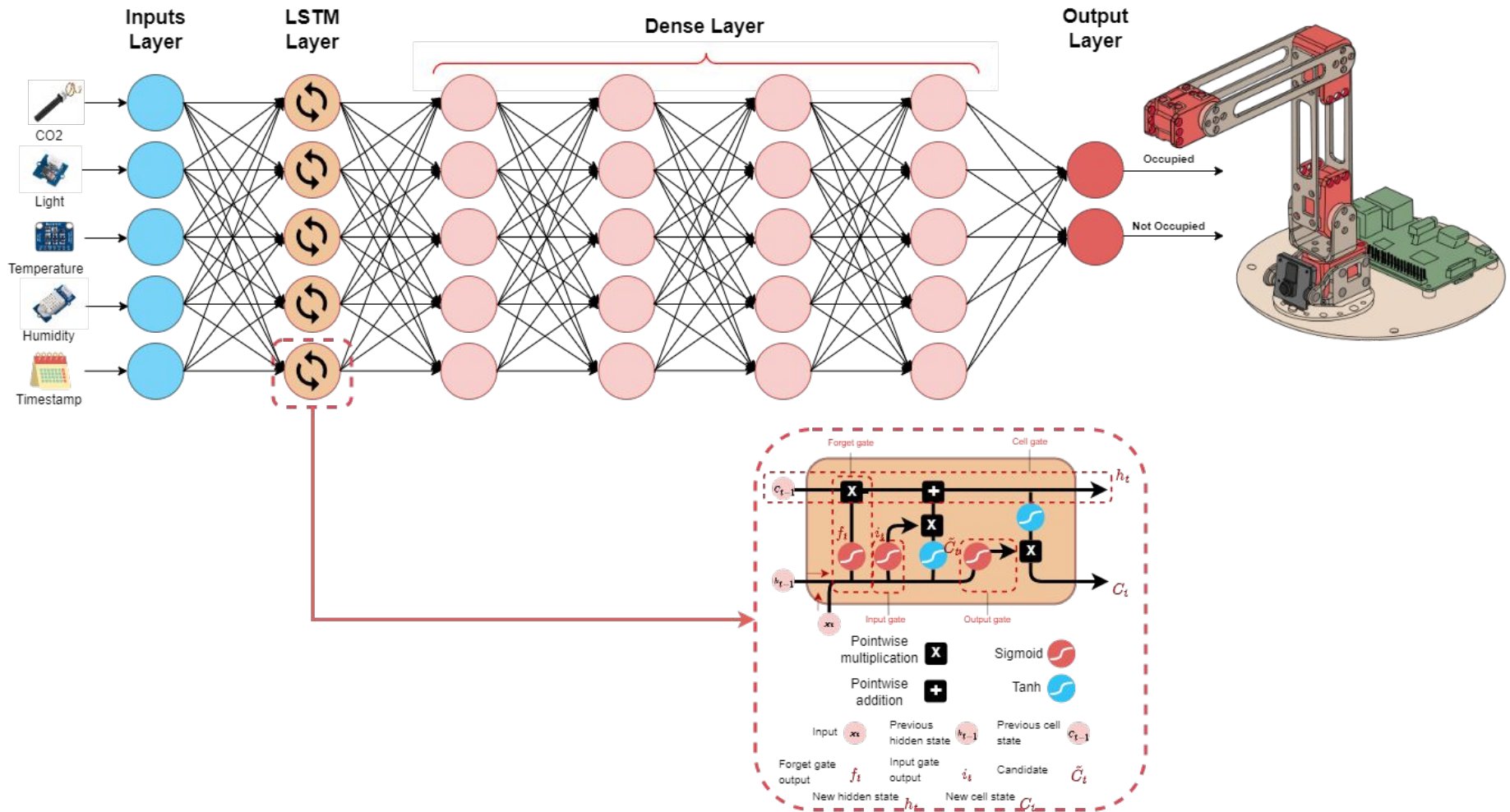
Part II.1.4: Selecting a Model for the Physical Twin

Physical Twin's Models



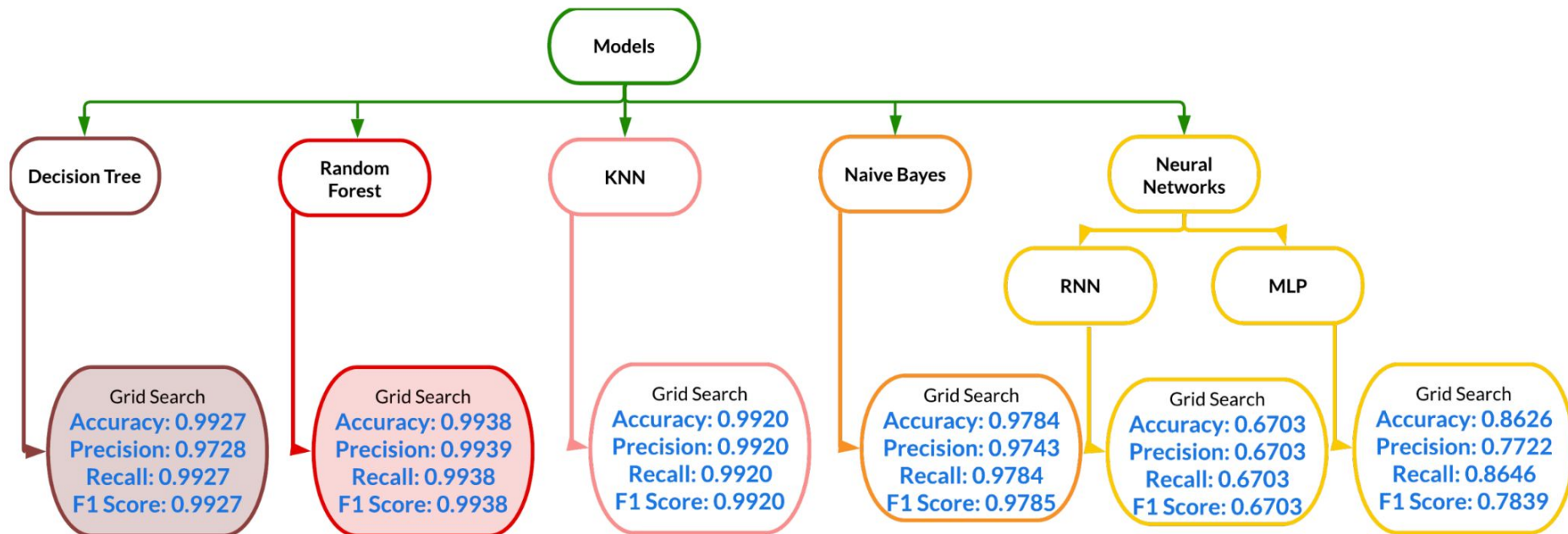
Part II.1.4: Selecting a Model for the Physical Twin

Physical Twin's Selected Model



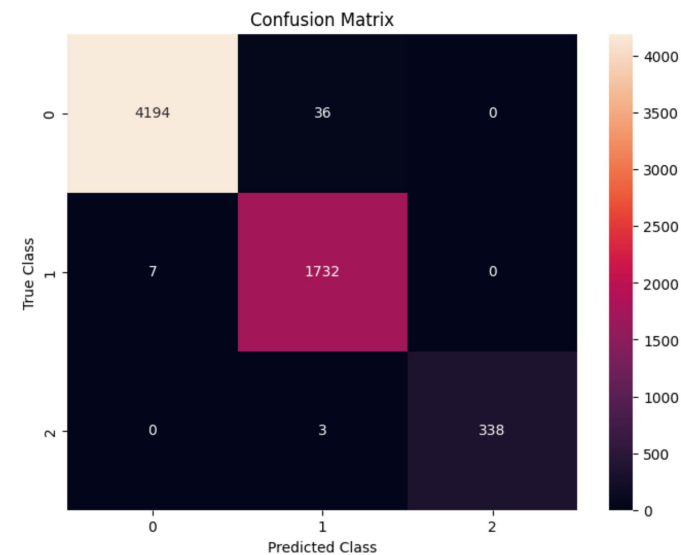
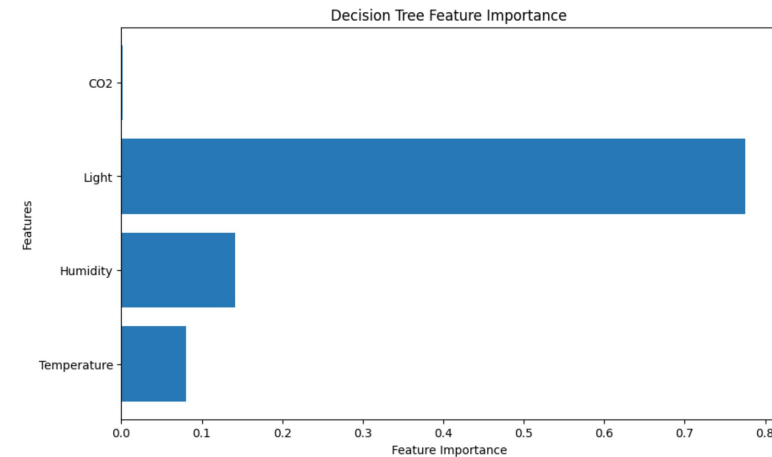
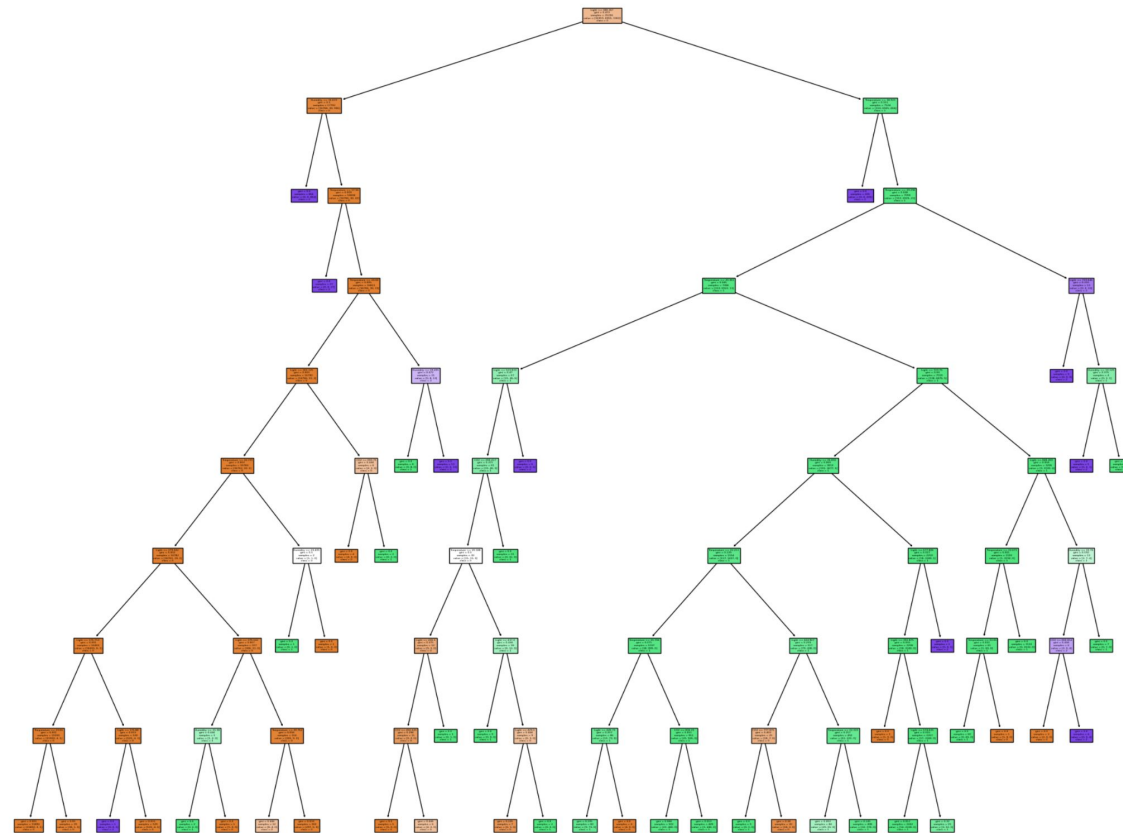
Part II.4: Selecting a Model for the Digital Twin

Digital Twin's Models



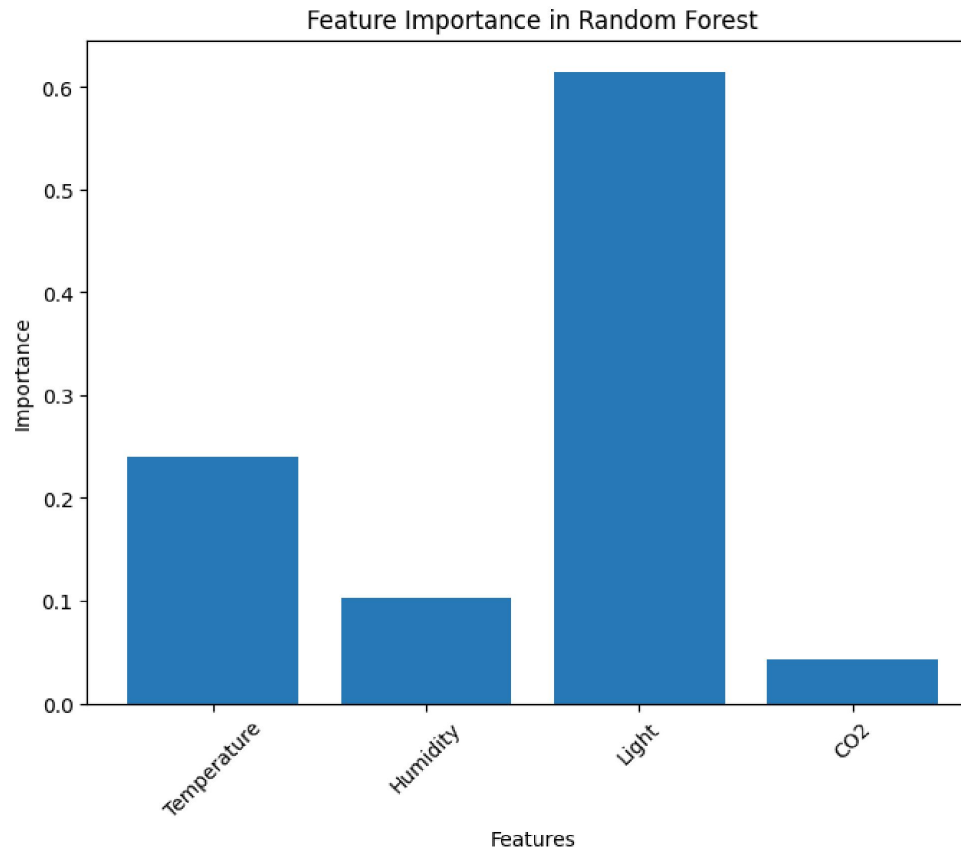
Part II.4: Selecting a Model for the Digital Twin

Physical Twin - Decision Tree



Part II.4: Selecting a Model for the Digital Twin

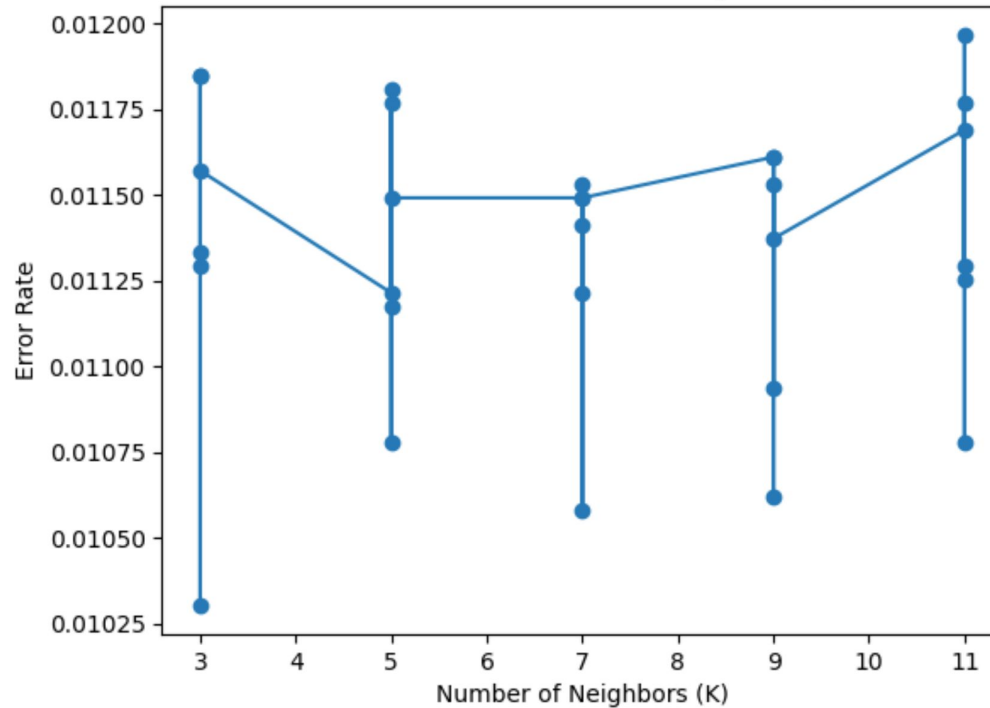
Physical Twin - Random Forest



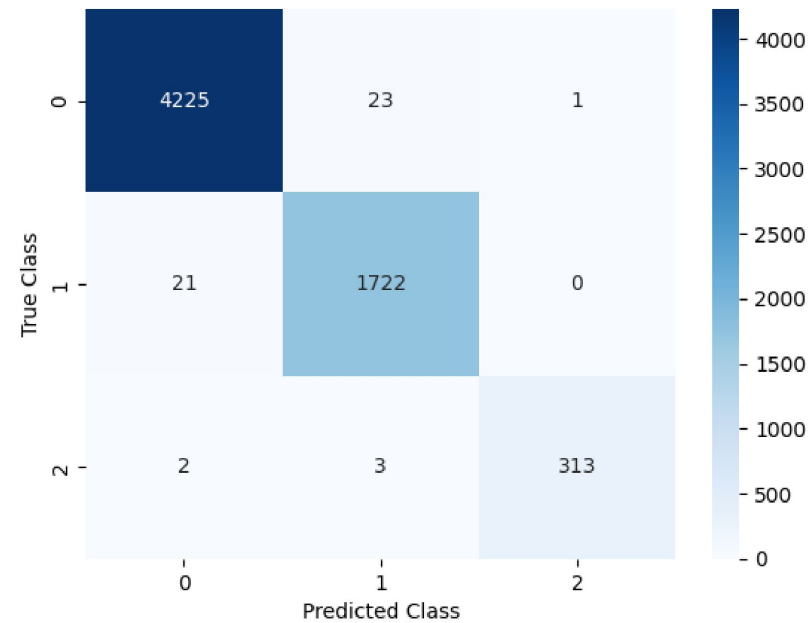
Part II.4: Selecting a Model for the Digital Twin

Physical Twin - KNN

Error Rate vs. Number of Neighbors (K)



Confusion Matrix



Part II.4: Selecting a Model for the Digital Twin

Decision Trees perform better on tabular data.

Section II.2 : Implementation

Part II.2: Implementation

Starting the Database

```
C:\Windows\System32\cmd.exe - influxd

C:\Users\msmat\Documents\InfluxData\influxdb\influxdb-1.8.10>influxd

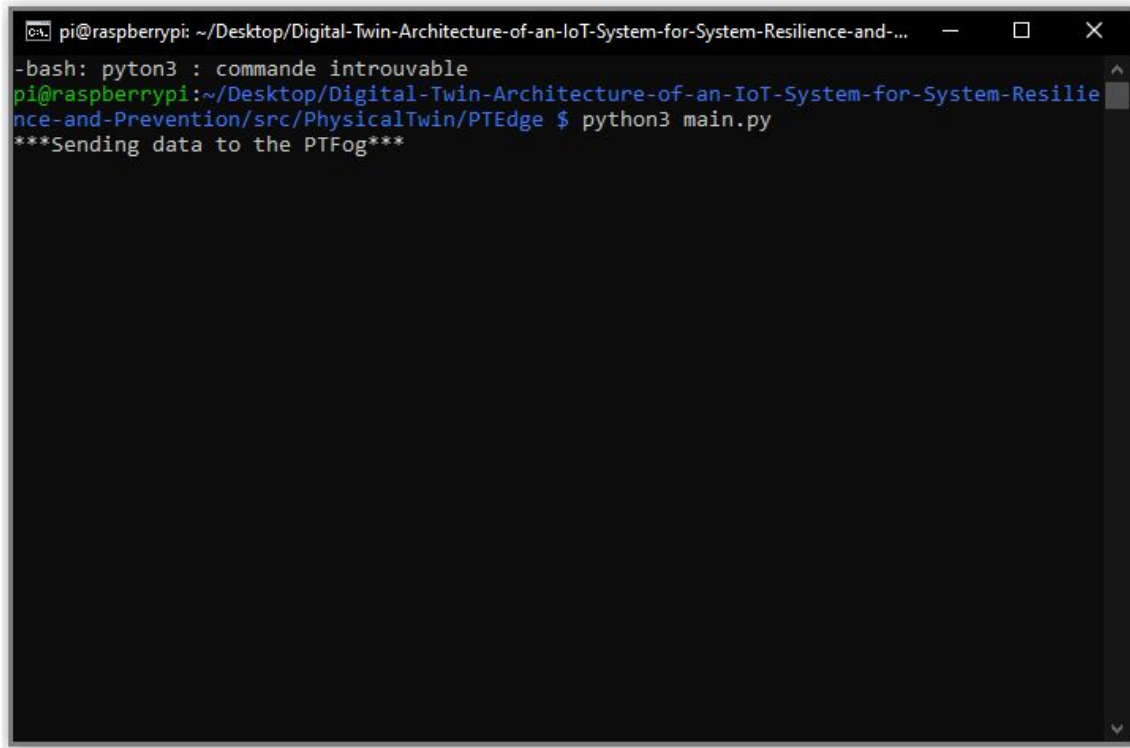
88888888      .d888 888      88888888b. 8888888b.
888      d88P" 888      888      "Y88b 888 888b
888      888 888      888      888 888 888P
888      888888b. 888888 888 888 888 888 888 88888888K.
888      888 888 888 888 888 888 Y8bd8P' 888 888 888 "Y88b
888      888 888 888 888 888 888 X88K 888 888 888 888
888      888 888 888 888 Y88b 888 .d8""8b. 888 .d88P 888 d88P
88888888 888 888 888 888 "Y88888 888 888 88888888P" 88888888P"

2023-06-12T15:32:01.898103Z info InfluxDB starting {"log_id": "0i03blvW000", "version": "1.8.10", "branch": "1.8", "commit": "688e697c51fd"}
2023-06-12T15:32:01.902091Z info Go runtime {"log_id": "0i03blvW000", "version": "go1.13.8", "maxprocs": 8}
2023-06-12T15:32:02.007876Z info Using data dir {"log_id": "0i03blvW000", "service": "store", "path": "C:\\Users\\msmat\\.influxdb\\data"}
2023-06-12T15:32:02.008370Z info Compaction settings {"log_id": "0i03blvW000", "service": "store", "max_concurrent_compactions": 4, "throughput_bytes_per_second": 50331648, "throughput_bytes_per_second_burst": 50331648}
2023-06-12T15:32:02.010066Z info Open store (start) {"log_id": "0i03blvW000", "service": "store", "trace_id": "0i03bmMW000", "op_name": "tsdb_open", "op_event": "start"}
2023-06-12T15:32:02.175985Z info Opened file {"log_id": "0i03blvW000", "engine": "tsm1", "service": "filestore", "path": "C:\\Users\\msmat\\.influxdb\\data\\PTDT_Interactions\\autogen\\21\\000000002-000000002.tsm", "id": 0, "duration": "0.850ms"}
2023-06-12T15:32:02.176481Z info Opened file {"log_id": "0i03blvW000", "engine": "tsm1", "service": "filestore", "path": "C:\\Users\\msmat\\.influxdb\\data\\PTDT_Interactions\\autogen\\23\\000000009-000000002.tsm", "id": 0, "duration": "0.495ms"}
2023-06-12T15:32:02.290286Z info Opened shard {"log_id": "0i03blvW000", "service": "store", "trace_id": "0i03bmMW000", "op_name": "tsdb_open", "index_version": "inmem", "path": "C:\\Users\\msmat\\.influxdb\\data\\PTDT_Interactions\\autogen\\8", "duration": "147.962ms"}
2023-06-12T15:32:02.290286Z info Opened shard {"log_id": "0i03blvW000", "service": "store", "trace_id": "0i03bmMW000", "op_name": "tsdb_open", "index_version": "inmem", "path": "C:\\Users\\msmat\\.influxdb\\data\\PTDT_Interactions\\autogen\\23", "duration": "148.404ms"}
2023-06-12T15:32:02.290286Z info Opened shard {"log_id": "0i03blvW000", "service": "store", "trace_id": "0i03bmMW000", "op_name": "tsdb_open", "index_version": "inmem", "path": "C:\\Users\\msmat\\.influxdb\\data\\PTDT_Interactions\\autogen\\2", "duration": "148.377ms"}
2023-06-12T15:32:02.290286Z info Opened shard {"log_id": "0i03blvW000", "service": "store", "trace_id": "0i03bmMW000", "op_name": "tsdb_open", "index_version": "inmem", "path": "C:\\Users\\msmat\\.influxdb\\data\\PTDT_Interactions\\autogen\\14", "duration": "148.377ms"}
2023-06-12T15:32:02.290286Z info Opened shard {"log_id": "0i03blvW000", "service": "store", "trace_id": "0i03bmMW000", "op_name": "tsdb_open", "index_version": "inmem", "path": "C:\\Users\\msmat\\.influxdb\\data\\PTDT_Interactions\\autogen\\21", "duration": "148.957ms"}
2023-06-12T15:32:02.290286Z info Opened shard {"log_id": "0i03blvW000", "service": "store", "trace_id": "0i03bmMW000", "op_name": "tsdb_open", "index_version": "inmem", "path": "C:\\Users\\msmat\\.influxdb\\data\\PTDT_Interactions\\autogen\\19", "duration": "149.711ms"}
2023-06-12T15:32:02.290286Z info Opened shard {"log_id": "0i03blvW000", "service": "store", "trace_id": "0i03bmMW000", "op_name": "tsdb_open", "index_version": "inmem", "path": "C:\\Users\\msmat\\.influxdb\\data\\PTDT_Interactions\\autogen\\11", "duration": "149.711ms"}
2023-06-12T15:32:02.326279Z info Opened file {"log_id": "0i03blvW000", "engine": "tsm1", "service": "filestore", "path": "C:\\Users\\msmat\\.influxdb\\data\\_internal\\monitor\\25\\000000001-000000001.tsm", "id": 0, "duration": "1.994ms"}
2023-06-12T15:32:02.326279Z info Reading file {"log_id": "0i03blvW000", "engine": "tsm1", "service": "cacheloader", "path": "C:\\Users\\msmat\\.influxdb\\wal\\_internal\\monitor\\26\\_00001.wal", "size": 5558418}
2023-06-12T15:32:02.342784Z info Opened shard {"log_id": "0i03blvW000", "service": "store", "trace_id": "0i03bmMW000", "op_name": "tsdb_open", "index_version": "inmem", "path": "C:\\Users\\msmat\\.influxdb\\data\\_internal\\monitor\\25", "duration": "52.497ms"}
```

Part II.2.1: Implementation

Sub-System 1 - Raspberry Pi

Physical Twin's Edge



```
pi@raspberrypi: ~/Desktop/Digital-Twin-Architecture-of-an-IoT-System-for-System-Resilience-and-Prevention/src/PhysicalTwin/PTEdge
-bash: python3 : commande introuvable
pi@raspberrypi:~/Desktop/Digital-Twin-Architecture-of-an-IoT-System-for-System-Resilience-and-Prevention/src/PhysicalTwin/PTEdge $ python3 main.py
***Sending data to the PTFog***
```

- Collecting the data and send it to its corresponding Fog.

Part II.2.1: Implementation

Sub-System 1 - Raspberry Pi

Physical Twin's Fog

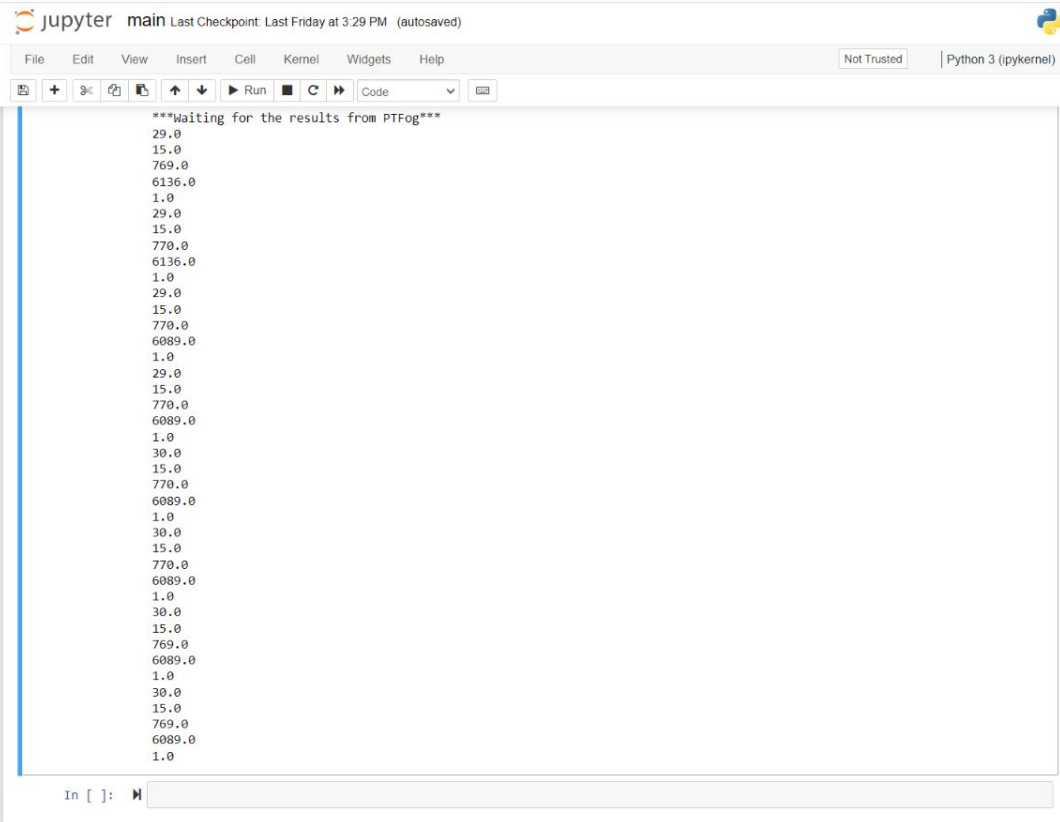
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
re names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
1
round: 1
C:\Users\msmat\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid featur
re names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
1
round: 0
***Sending results to the second Edge Poppy Ergo Jr***
Send data to Poppy Ergo Jr
C:\Users\msmat\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid featur
re names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
1
round: 10
C:\Users\msmat\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid featur
re names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
1
```

Ln 5, Col 12 Spaces: 4 UTF-8 CRLF Python 3.9.13 64-bit Go Live

Part II.2.1: Implementation

Sub-System 2 - Poppy Ergo Jr

Infos v4.0.1



The image shows a Jupyter Notebook interface. The top bar includes the Jupyter logo, the text "jupyter main", and a status message "Last Checkpoint: Last Friday at 3:29 PM (autosaved)". The right side of the top bar shows a "Not Trusted" warning and the kernel name "Python 3 (ipykernel)". Below the top bar is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". A toolbar contains icons for file operations, cell navigation, and execution. The main area is a code cell with the following output:

```
***waiting for the results from PTFog***  
29.0  
15.0  
769.0  
6136.0  
1.0  
29.0  
15.0  
770.0  
6136.0  
1.0  
29.0  
15.0  
770.0  
6089.0  
1.0  
29.0  
15.0  
770.0  
6089.0  
1.0  
30.0  
15.0  
770.0  
6089.0  
1.0  
30.0  
15.0  
770.0  
6089.0  
1.0  
30.0  
15.0  
769.0  
6089.0  
1.0  
30.0  
15.0  
769.0  
6089.0  
1.0
```

The bottom of the code cell shows the prompt "In []:" followed by a text input field.

Part II.2.1: Implementation

Digital Twin

DT Edge

DT Fog

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
247.34149064368398
Those are the generated data of DT
33.63236455861416
23.464627191302746
673.7545374925314
145.20956160516477
Those are the generated data of DT
24.1664632667842
23.81793461473127
762.7104619575682
178.84307145331368
Those are the generated data of DT
31.112292124008555
21.852497399386294
252.43738186161906
114.83651248562569
```

```
31.112292124008555
23.81793461473127
762.7104619575682
178.84307145331368
31.112292124008555
21.852497399386294
762.7104619575682
178.84307145331368
31.112292124008555
21.852497399386294
252.43738186161906
178.84307145331368
31.112292124008555
21.852497399386294
252.43738186161906
114.83651248562569
```

Ln 4, Col 24 Spaces: 4 UTF-8 CRLF Python 3.9.13 64-bit Go Live

Part II.2.52: Prospective Endeavors

- Change the type of the used DT from a static twin to a dynamic one where data and responses are done in real-time.
- Change the level of granularity from a system twin to a process twin.
- Develop the use case and include reinforcement learning.
- Find a solution to include different environments in the use case to not have the problem of different distributed data.
- Add a cloud layer to upgrade the architecture to an Edge/Fog/Cloud and deal with big data.

Thank you!