Digital twin to Predict and Improve Efficiency In Egg gender sorting system

Demcon

Demcon is working on providing solutions to social challenges in the areas of various industries.

- Aerospace
- Agri & food
- Defense & security
- Energy
- High-tech systems & materials
- Life sciences & health
- Smart industry
- Water & maritime

To achieve organizational goals, the company develops, manufactures and supplies high-quality technology and innovative products







Introduction

Why they make machine- Egg gender sorting system (EGGS)?

- Every year tons of male chicks are killed in the poultry industry
- Males are removed immediately after hatching
- Categorize according to gender by hand in traditional method

Solution to minimize this issue:

- Egg gender sorting system -Machine can identify the gender of the egg before it hatches

Demcon requirement:

- Build a Digital twin for improve the machine efficiency of the EGGS as for further innovations and modifications

Present situation :

- Two egg gender sorting machines
- Machine data is not used yet for any analysis

How to achieve?

- Method 01- Overall Equipment Effectiveness (OEE) analysis
- Method 02 - Process time analysis

Method 01- Overall Equipment Effectiveness (OEE) analysis

Main research question:

How can a OEE analysis and machine learning model be used to implement a digital twin of the EGGS machine to optimize its operation efficiency?

Sub-questions:

1. What is the general characteristic of digital twin and its effectiveness in industries, How modelling the digital twin for optimize the efficiency of the machine?
2. How machine learning techniques and OEE analysis can be used to build the models for the digital twin system?
3. Which unit or process is the bottleneck of the machine?
4. Which machine learning model giver best accuracy results when predicting OEE values for specific parts of the process?
5. What parameters have the most significant impact on the OEE index?

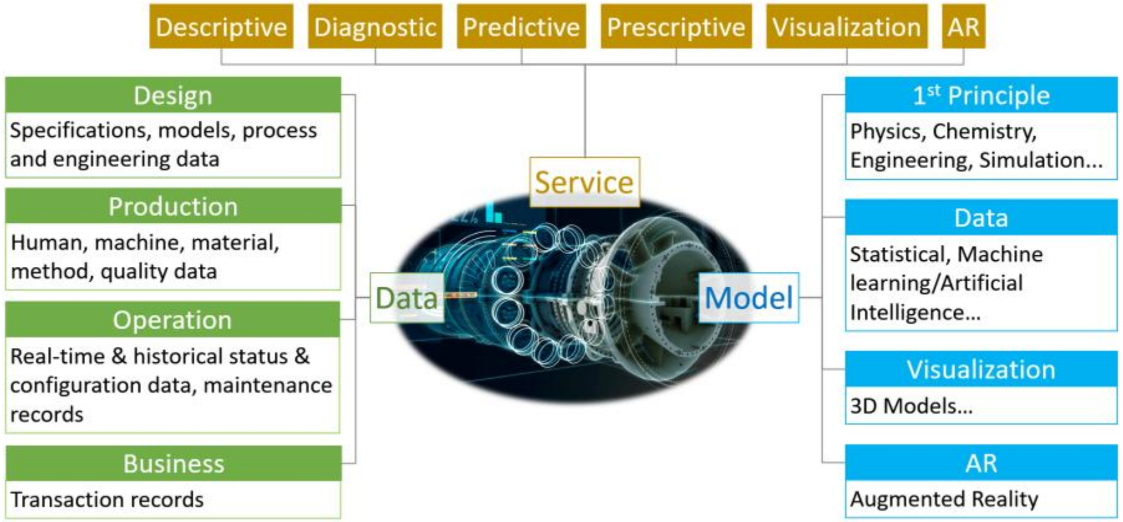
Digital Twin

What is Digital twin?

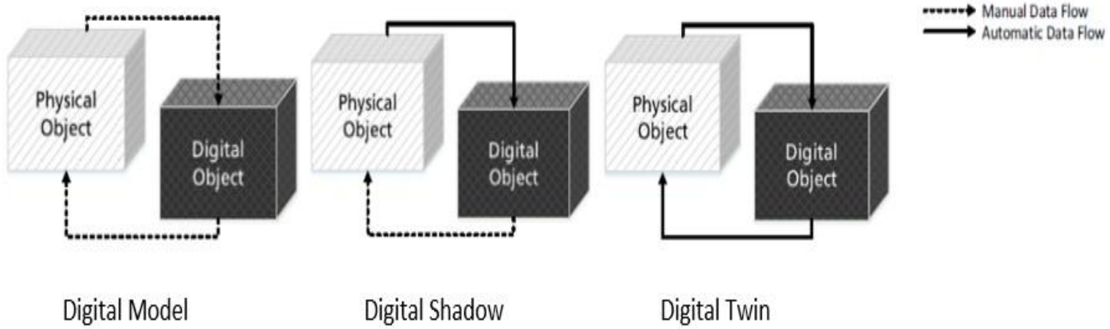
- Digitally created virtual representation of a similar physical entity
- massively using into accelerate process development and optimizing operations

Main outcomes for Industry :

- Support improve decision making
- Increase efficiency
- Reduce costs
- Better product design
- Increased innovation



Components of a digital twin



Digital twin level of data integration

Overall Equipment Effectiveness

- Useful tool for measuring manufacturing productivity

OEE score of 100% means we are manufacturing only Good Parts, as fast as possible, with no Stop Time

- Fundamental OEE definition is defined as following formula.

$$OEE = A \times P \times Q$$

Where A is availability of the machine, P is performance and Q is quality

$$A = Operating\ Time\ \div\ Planned\ Production\ Time$$

$$P = Total\ Pieces\ \div\ (Operating\ Time\ \times Ideal\ Run\ Rate)$$

$$Q = Good\ Pieces\ \div\ Total\ Pieces$$

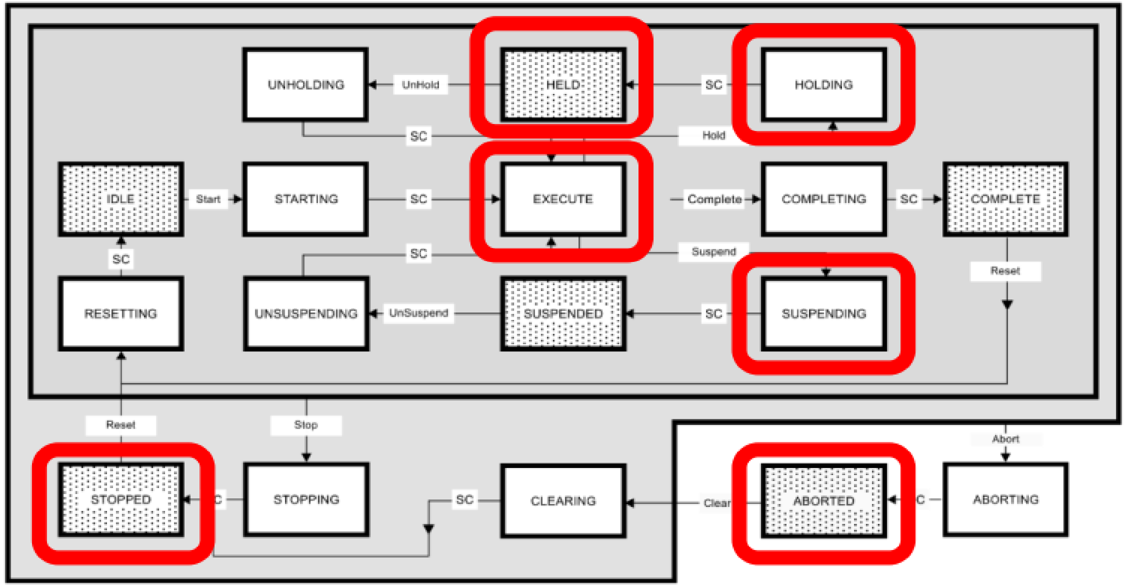
- Total pieces = Total egg production in the machine
- Good pieces = correctly identified male and female eggs

- OEE calculation assumes quality is equal to one
- This project only focuses on availability and performance
- No data available for the number of accurately identified male and female eggs and unknown eggs

- Unknown eggs - unpunched eggs, eggs damaged during sampling, punching and handling, and eggs unable to be identified

Availability Implementation

$$A = \frac{TMPP}{TMPP + TMF}$$



Packaging Machine Language (PackML) State Model - Availability

Where,

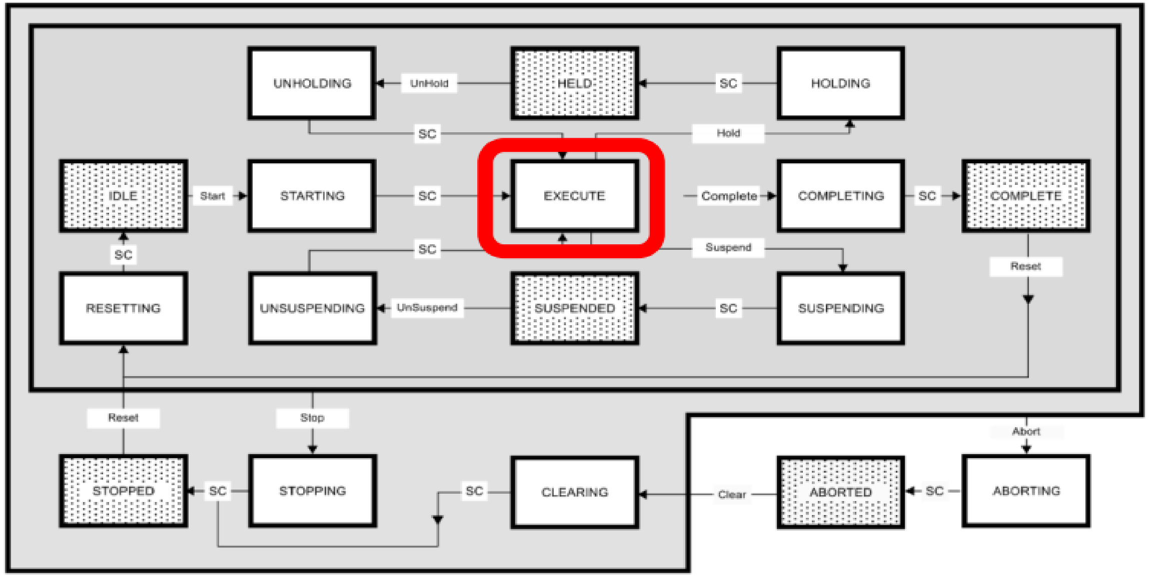
TMPP is Time machine is producing product (Execute state + Holding state + Suspending state).

TMF is Time when machine in failure (Held state + Stopped state + Aborted state).

Packaging Machine Language (PackML) is an automation standard established by the Organization for Machine Automation and Control (OMAC) and by the International Society of Automation's

Performance Implementation

$$P = \frac{TQPM}{TMPPE \times S}$$



PackML State Model - Performance

Where,

S is maximum machine speed.

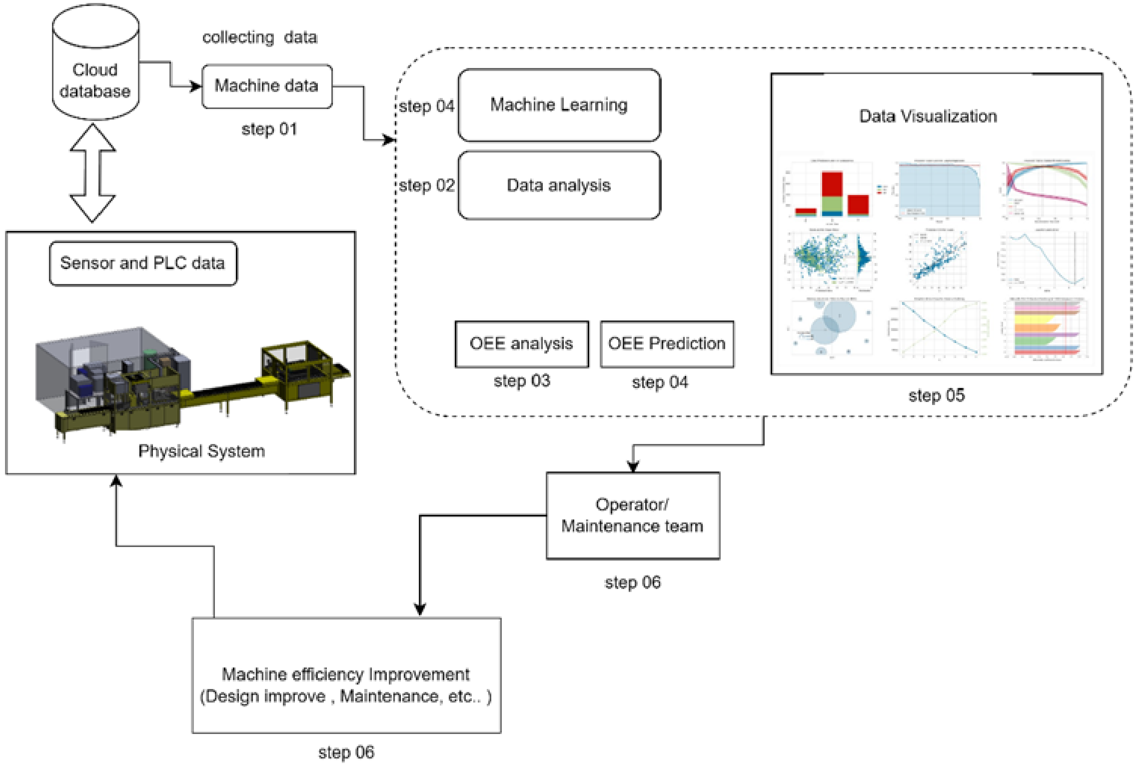
TMPPE is Time machine is producing product (Execute state)

TQPM is Total quantity produced at this machine (Total egg production in the machine)

- EGGS maximum machine speed is 7000 Eggs per hour
- Normal running speed of the machine is limited to 4000 Eggs per hour

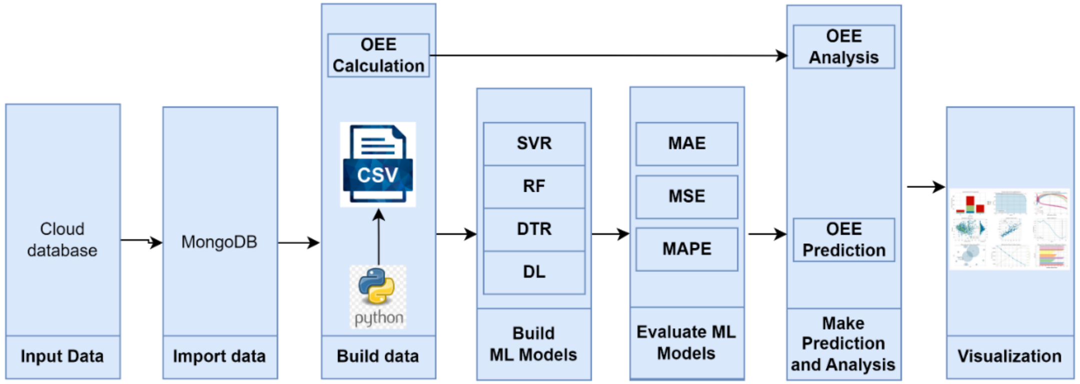
Conceptual model 01

- Step 01: Collect data
- Step 02: Data analysis and build the OEE's for Machine separate sections
- Step 03: According to OEE analysis , find the bottleneck section or process of the machine.
- Step 04: Apply machine learning for predict to future OEE for bottleneck section or process.
- Step 05: visualize the OEE analysis and prediction result
- Step 06: according to OEE analysis and prediction result maintenance team can improve the machine efficiency



Research design 01

1. First imported data to mongodb from database.
2. Calculated Egg production, according to egg tray position state.
3. Calculated Availability , Performance and OEE
4. Analysed and visualized the OEE for the individual process of sampler part.
5. Lowest OEE unit selected for the bottleneck unit of sampler part of the machine.

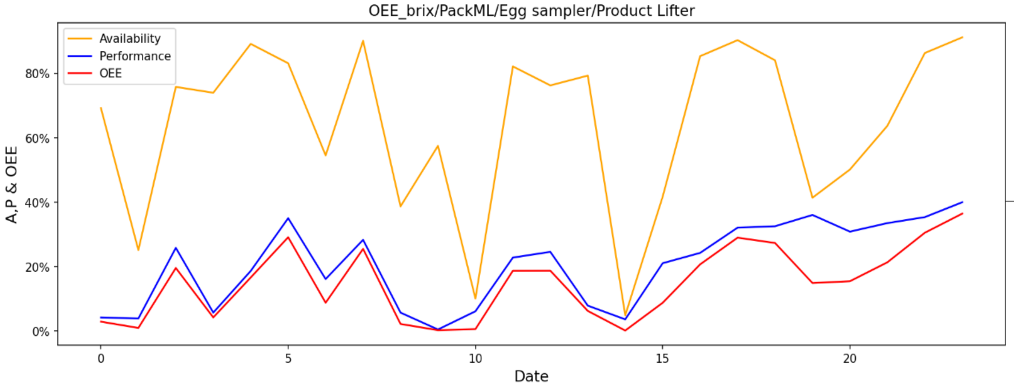


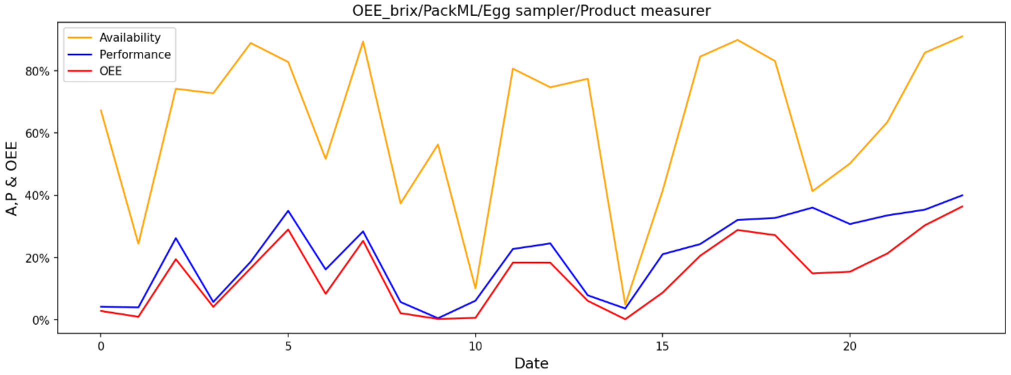
- PackML database data used for calculating the OEE
- Ella database data used for calculating the daily egg production of the machine

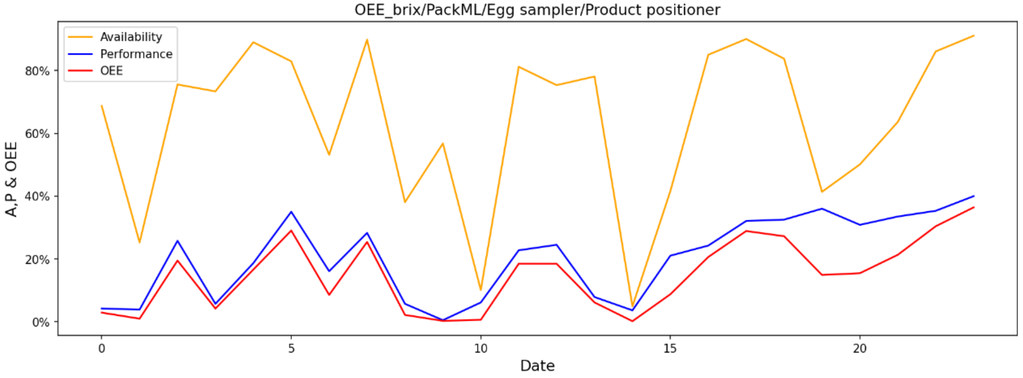
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	...	O1	O2	O3	O4	O5	O6
0	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	...	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}
1	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	...	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}
2	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	...	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}
3	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	...	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}
4	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	...	{'Status': 0}	{'Status': 0}	{'Status': 0}	{'Status': 0}	{'Status': 0}	{'Status': 0}
5	{'Status': 0}	{'Status': 0}	{'Status': 0}	{'Status': 0}	{'Status': 0}	{'Status': 0}	{'Status': 0}	{'Status': 0}	{'Status': 0}	{'Status': 0}	...	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}	{'Status': 1}

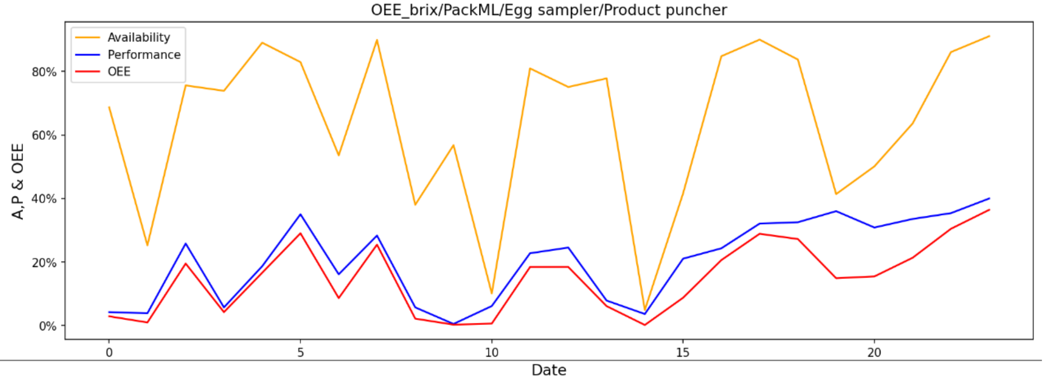
Egg position status of the each trays

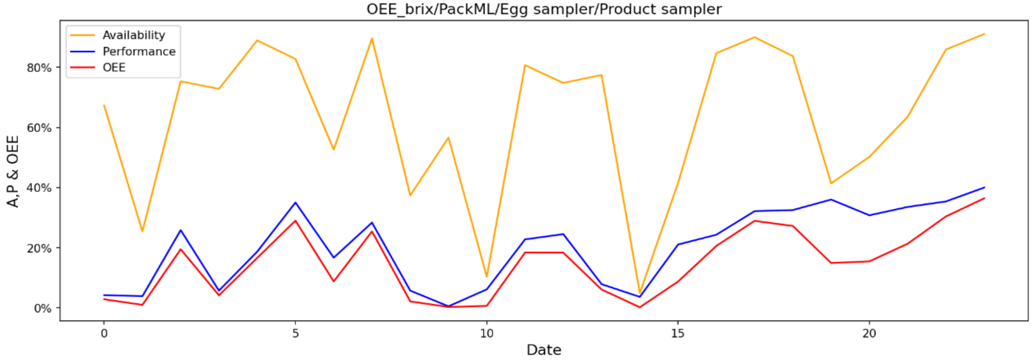
Results 01

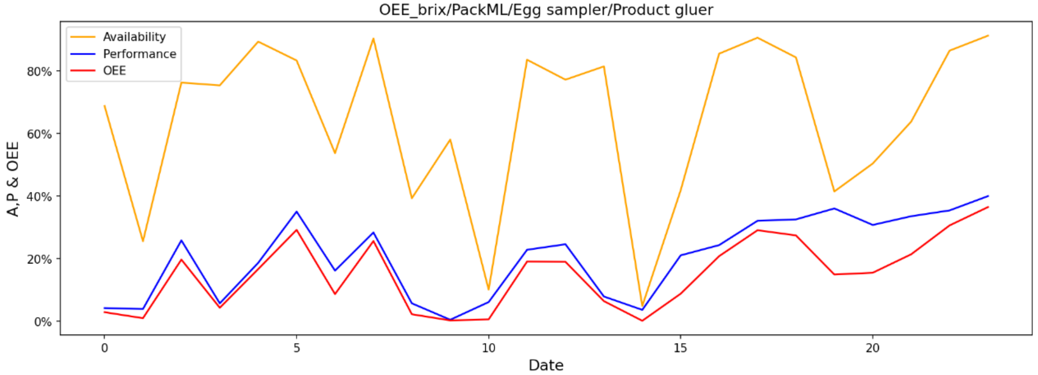






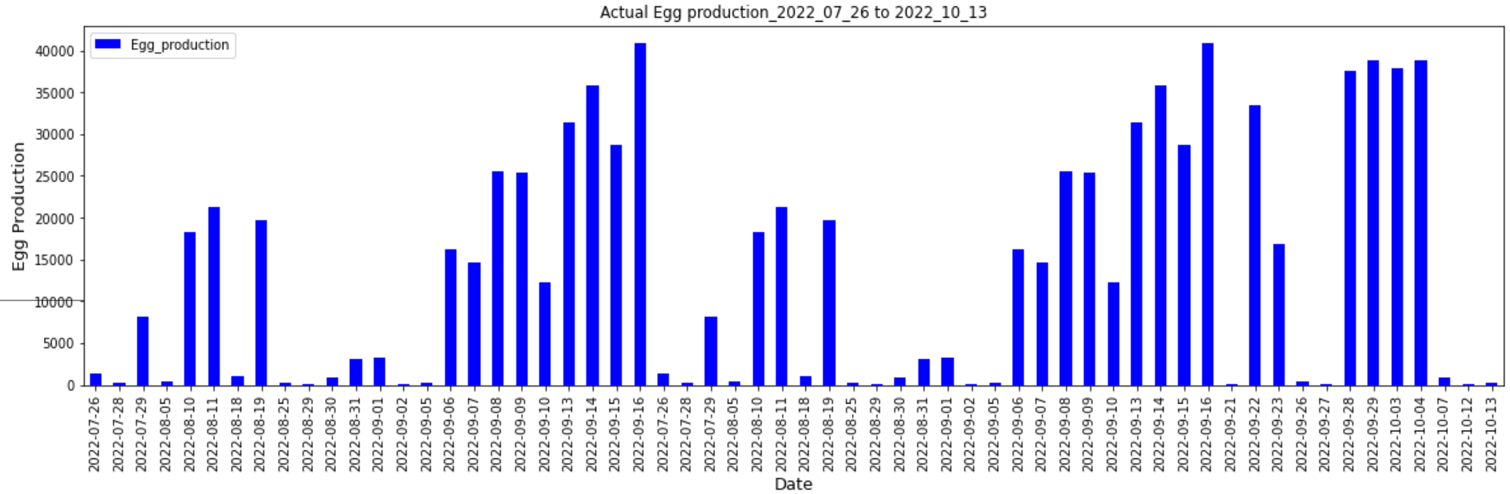






- Machine is still in the development stage
- it requires frequent maintenance and upgrades which result in downtime
- Machine operates for 16 hours a day
- Actual production time is depend on customer requirements and the need for maintenance and upgrades

Results 01



- The maximum production capacity of the machine is 7000 eggs per hour
- Normal running speed of the machine is limited to 4000 Eggs per hour
    - slower speed of the gender analyzing unit
    - slower sample transport and more cleaning time
    - frequent maintenance and upgrades
    - machine still in the development stage
- Can produce a total of 112,000 eggs in 16 hours.
- Machine availability and performance are not optimal.
- Maximum egg production is close to 40,000, fewer than 40,000 most days.
- This means that even though the machine has the potential to produce more, its actual output is limited due to lower availability and performance.

Conclusions Method 01

- Cannot identify the machine's bottleneck process.
  - Similar OEE value for every section (main machine OEE)
  - Insufficient for giving suggestions for a specific process or unit improvement

- As a result, this method not designed further steps (machine learning theories are not be implemented)

- This is the first time this machine data has been used for analysis, and the value of the data was uncertain before its gathering and analysis

PackML is designed to focus on system-level performance rather than the performance of individual components within the machine.

As a result, PackML may not be able to see the differences below the system level in the machine. This means that it may not be able to capture detailed information about the individual components within the machine.

Method 02 - Process time analysis

Main research question:

How can the process time analysis and machine learning model be used to implement a digital twin of the egg gender sorting machine to optimize its operation efficiency?

Sub-questions:

1. What is the general characteristic of digital twin and its effectiveness in industries, How modelling the digital twin for optimize the efficiency of the machine?
2. How machine learning techniques and process time analysis can be used to build the models for the digital twin system?
3. Which unit or process is the bottleneck of the machine?
4. Which machine learning model gives best accuracy results when predicting the process time for specific part of the process?
5. What is the minimum duration of a specific process, which gives the minimum possible process time?

process time and handling time implementation

Main functioning of the processEggs units

- Lift Product(LP) – lift eggs and are externally straightened and then clamped on gripper with vacuum for measure process
- Measure Product(MP) – push vision beam onto top of eggs and make photo for finding to position of the air sack in the egg
- Position Product(PP) – turn egg to right position for punching and sampling
- Punch Product(PE) – punch egg
- Glue Product(GP) – close hole
- Unlift Product(UP) – lower egg into tray

ProcessEggs units process time and handling time implementation

- Process time is the length of a particular process movement
- Handing time is the amount of time required to set up for the next process

The equations below can be used to calculate process time and handling time.

$$Process\ time\ lift\ =\ LP\ time(state = complete)\ -\ LP\ time(state = Execute)$$
$$Process\ time\ measure\ =\ MP\ time(state = complete)\ -\ MP\ time(state = Execute)$$
$$Process\ time\ position\ =\ PP\ time(state = complete)\ -\ PP\ time(state = Execute)$$
$$Process\ time\ punch\ =\ PE\ time(state = complete)\ -\ PE\ time(state = Execute)$$
$$Process\ time\ glue\ =\ GP\ time(state = complete)\ -\ GP\ time(state = Execute)$$
$$Process\ time\ unlift\ =\ UP\ time(state = complete)\ -\ UP\ time(state = Execute)$$
$$Handling\ time\ lift\ to\ measure\ = MP\ time(state = Execute)\ -\ LP\ (state = complete)$$
$$Handling\ time\ measure\ to\ position\ = PP\ time(state = Execute)\ -\ MP\ (state = complete)$$
$$Handling\ time\ position\ to\ punch\ = PE\ time(state = Execute)\ -\ PP\ (state = complete)$$
$$Handling\ time\ glue\ to\ unlift\ = UP\ time(state = Execute)\ -\ GP\ (state = complete)$$
$$Handling\ time\ lift\ =\ LP\ time(state = Execute)\ -\ LP\ time(state = complete)\ in\ previous\ step$$
$$Handling\ time\ measure\ =\ MP\ time(state = Execute)\ -\ MP\ time(state = complete)\ in\ previous\ step$$
$$Handling\ time\ position\ =\ PP\ time(state = Execute)\ -\ PP\ time(state = complete)\ in\ previous\ step$$
$$Handling\ time\ punch\ =\ PE\ time(state = Execute)\ -\ PE\ time(state = complete)\ in\ previous\ step$$
$$Handling\ time\ glue\ =\ GP\ time(state = Execute)\ -\ GP\ time(state = complete)\ in\ previous\ step$$
$$Handling\ time\ unlift\ =\ UP\ time(state = Execute) -\ UP\ time(state = complete) in\ previous\ step$$

Machine Learning & Optimization

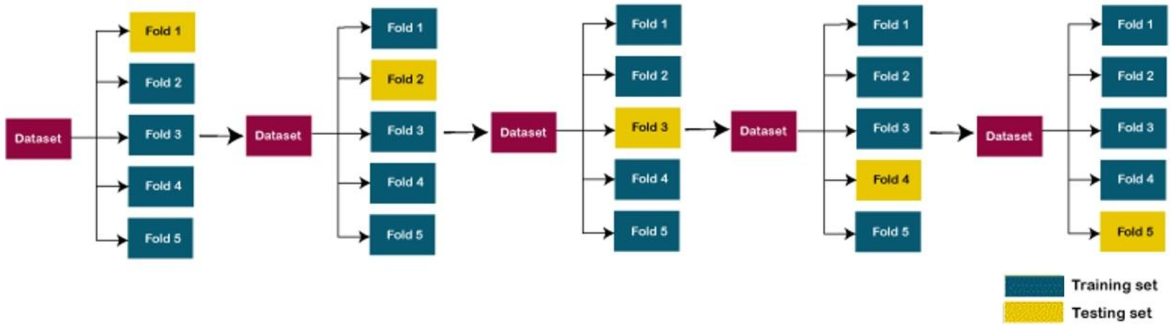
- **Machine Learning**
  - Multiple liner regression
  - Decision Tree Regression
  - Random Forest
  - Support vector Regression
- **Evaluate the performance of the model**
  - Root Mean Squared Error- RMSE
  - Mean Absolute Error - MAE

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

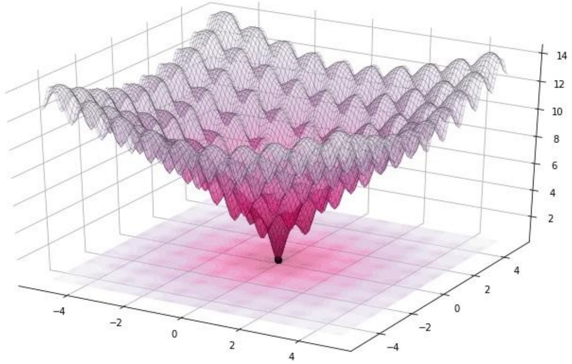
$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Where:
n: Number of observations
y_i : Actual value
$\hat{y}_i$: Predicted value

- **K-Fold Cross-Validation**



- **Optimization-Differential Evolution**



Conceptual model 02

Step 01: Collect data from cloud

Step 02: Data analysis and build the process time and handling time for Machine separate sections

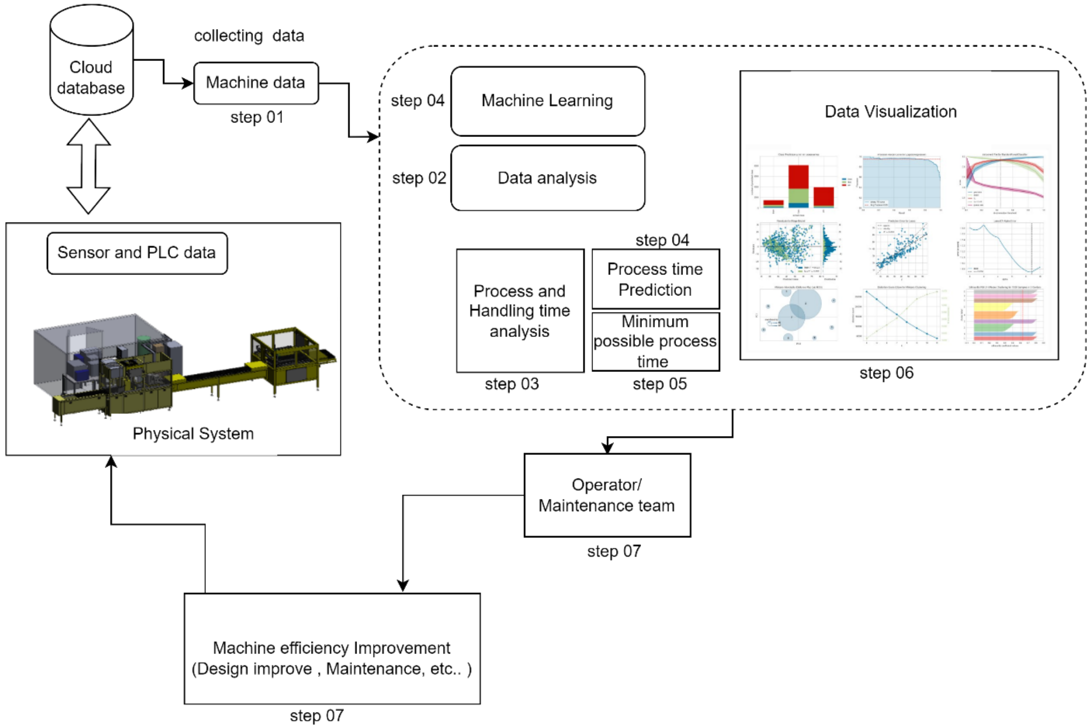
Step 03: Find the bottleneck section or process of the machine

Step 04: Apply machine learning to predict the process time for a section or process that is bottlenecked

Step 05: Apply differential evolution optimization method for calculate to minimum possible process time using prediction values

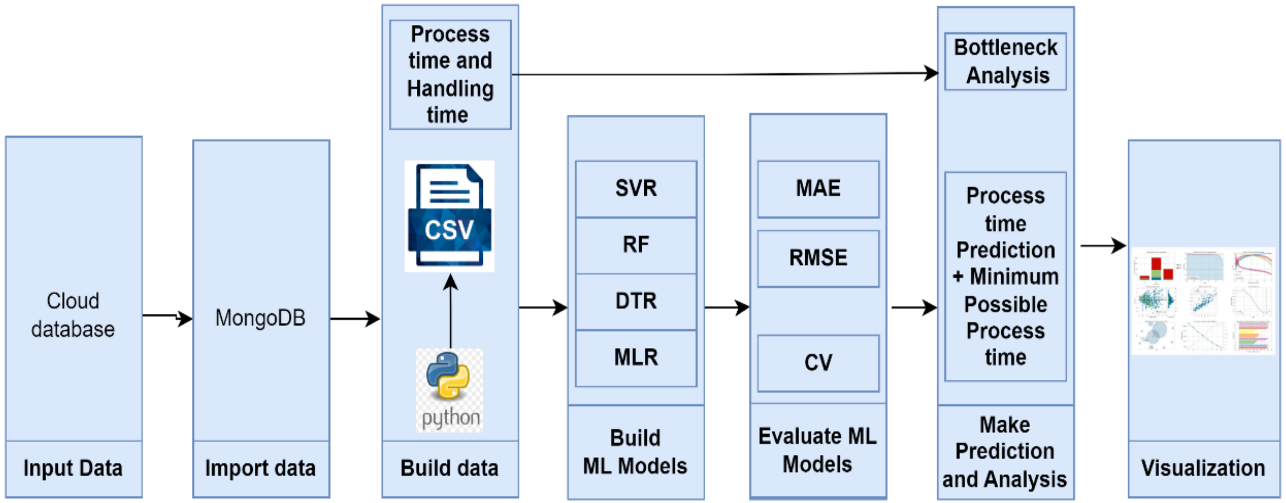
Step 06: Visualize the process time and handling time analysis, prediction and optimization result

Step 07: The efficiency of the machine can be increased by the maintenance team in according to analysis, prediction, and optimization results



Research design 02

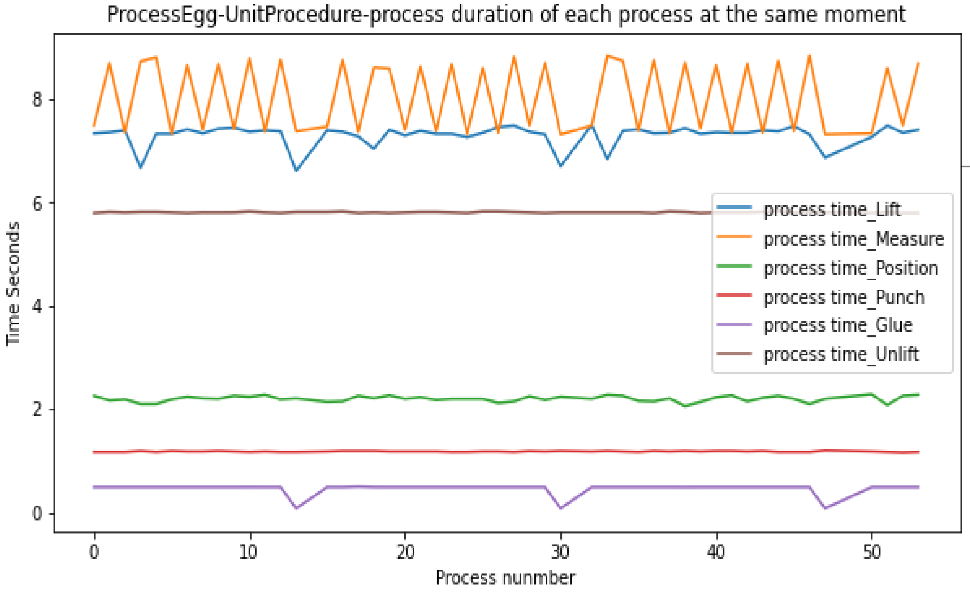
- Imported data to mongodb from cloud database
- Calculated the Process time and handling time
- Analysed and visualized for the finding bottleneck process and information. (Data from 50 process phases were chosen for this investigation)
- Bottleneck process used to determine the minimum process time
- Machine learning and optimization methods



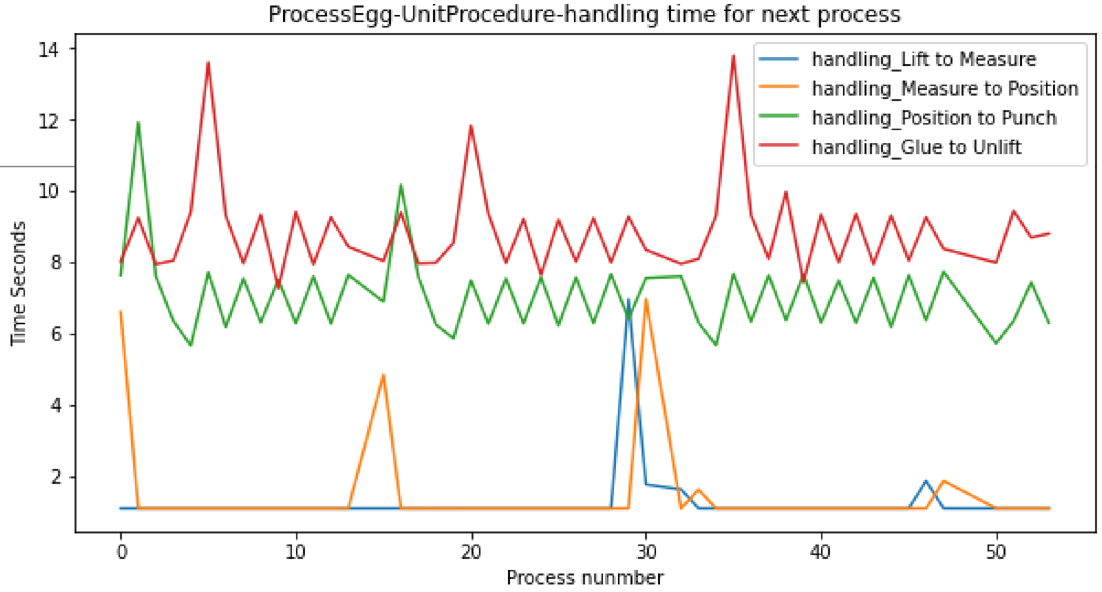
MLR-Multiple liner regression
DTR - Decision Tree Regression
RF - Random Forest
SVR - Support vector Regression
CV - Cross validation
MAE - Mean Absolute Error
RMSE - Root Mean squared error
ML- Machine Learning

- Procedures database data used for calculating the Process time and handling time

Results 02- Process time analysis & Handling time analysis



Process time analysis



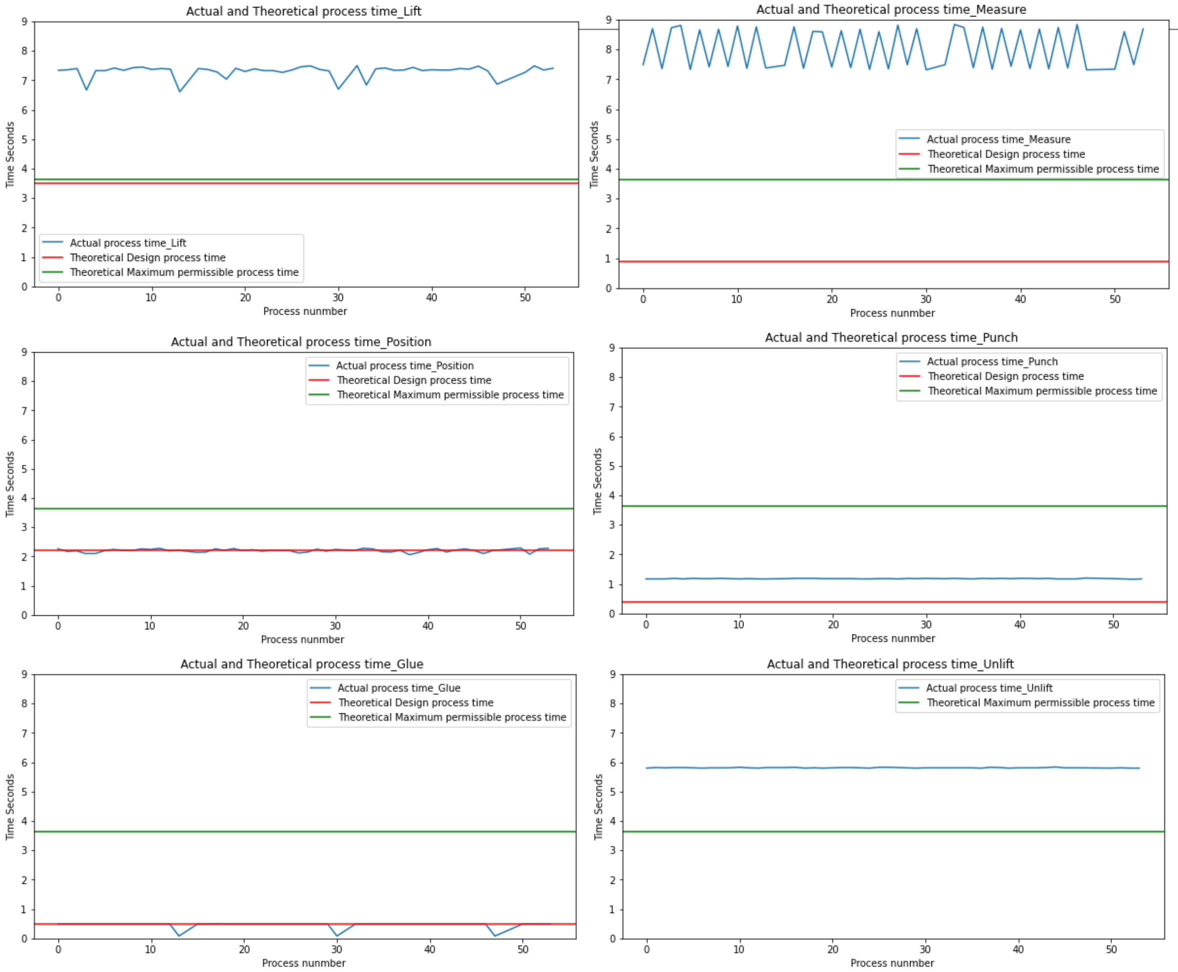
Handling time analysis

- Process time measure and process time lift take more time
- Unusual fluctuation in the measure process compared to other process units
  - Recently it was discovered that a PLC code error was the root cause of this abnormality. They now working to rectify this error and prevent any further issues
- Fluctuation in the process time glue after every 16 steps (Egg tray changing)

- Position to punch and glue to unlift steps take more time compared to other movements
- Minor fluctuations in glue to unlift and position to punch steps (due to coupling and decoupling process of the carrier)
- This show not only the handling time but also the waiting time of a unit until handling will take place

Results 02-

Actual vs theoretical value of the process time and handling time



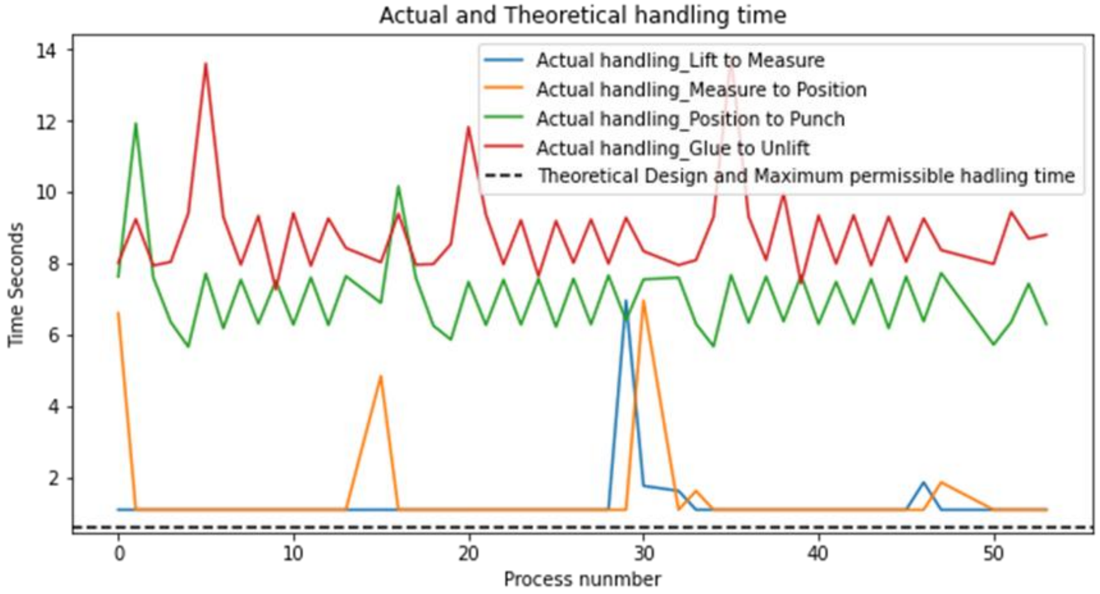
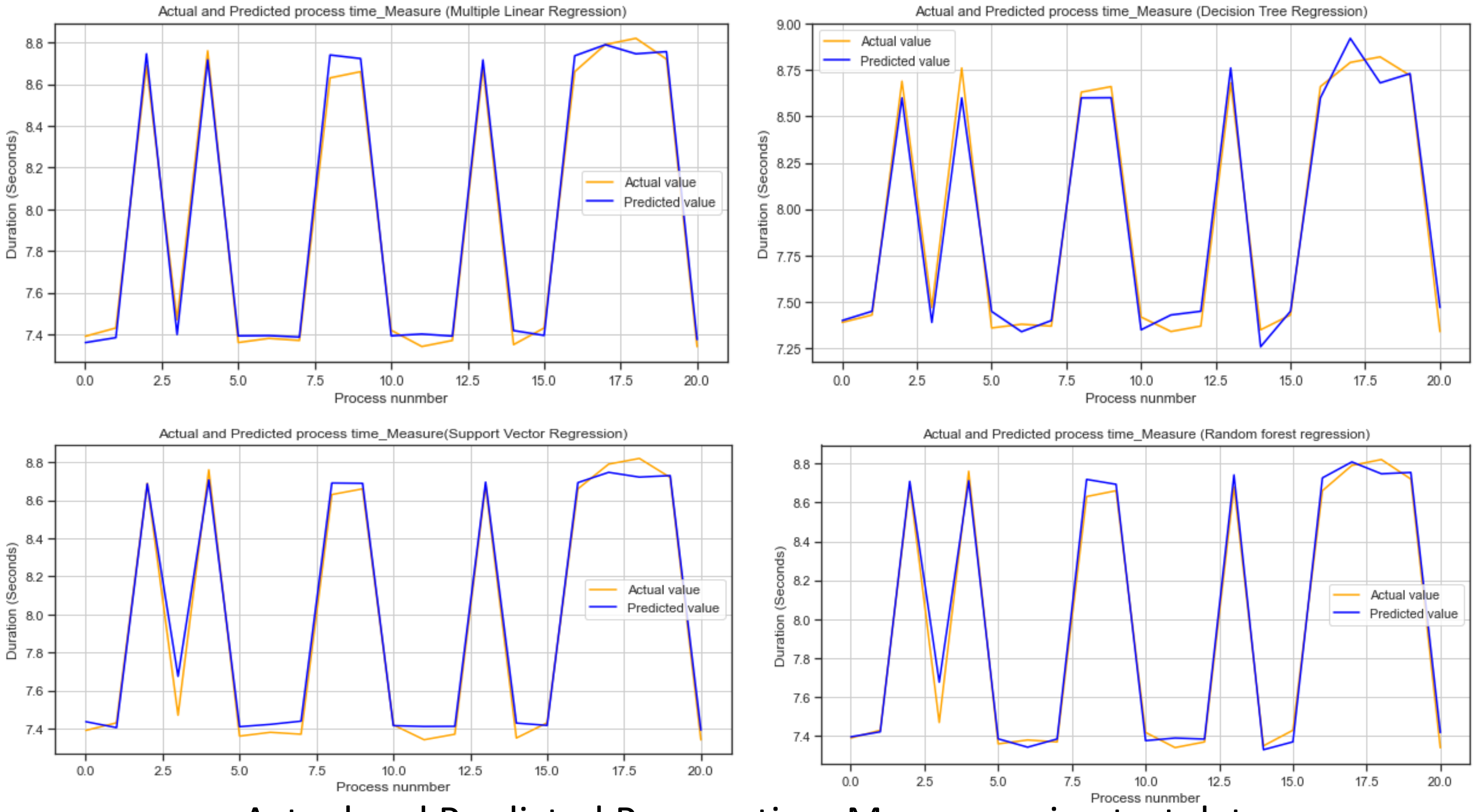


Table shows the theoretical maximum permissible and theoretical design (determined during design) value of the process time and handling time in seconds

Actual vs theoretical value of the process time

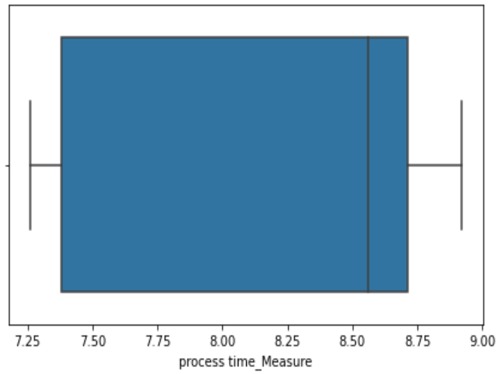
Results 02- Machine learning result

Actual and Predicted Process time Measure using test data



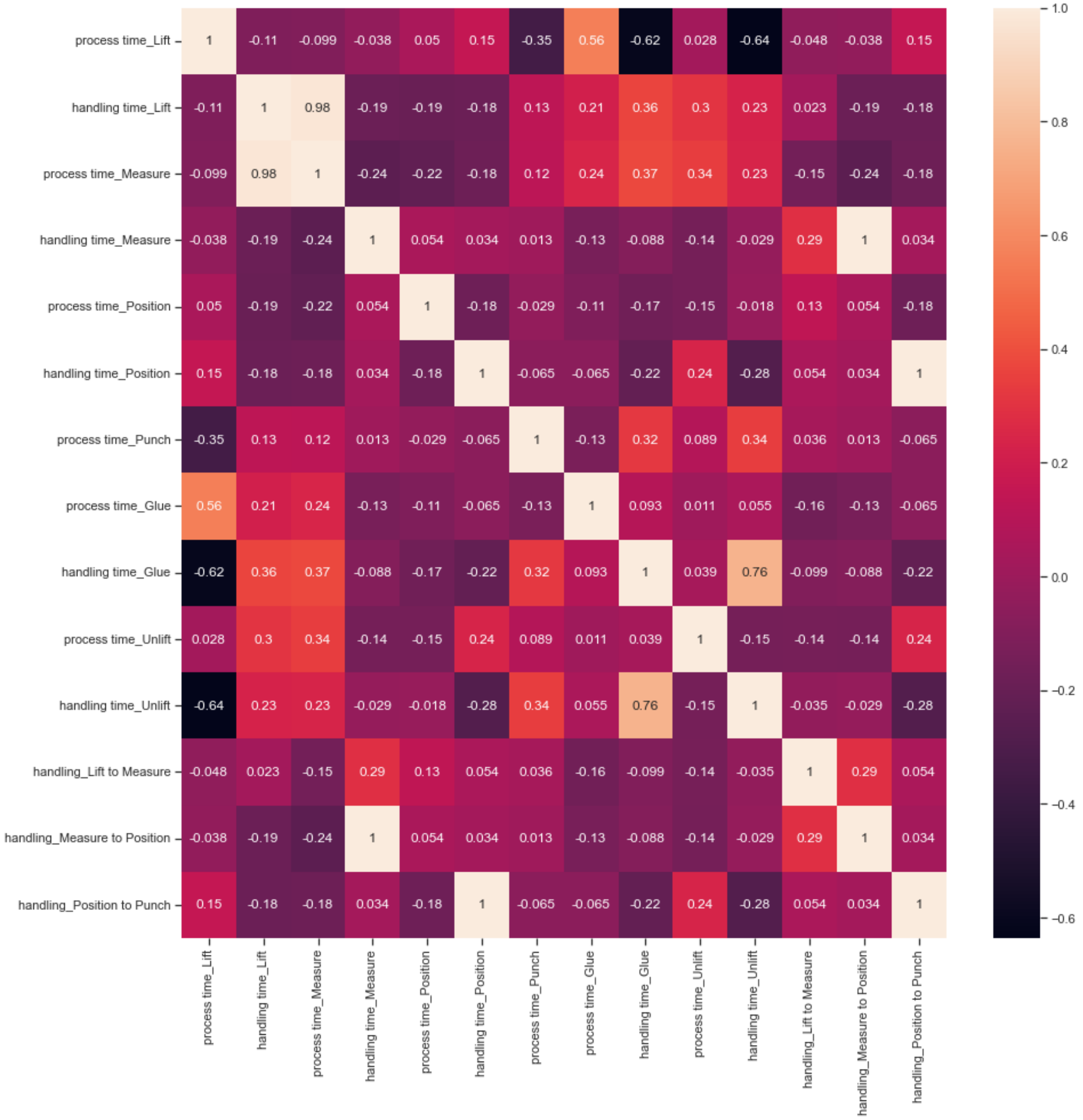
Actual and Predicted Process time Measure using test data

Machine learning result



Boxplot for process time measure

- Handling time lift and Process time measure is more positive
- Process time punch, handling time glue, process time unlift, and handling time unlift all show positive correlations with process time measure
- remaining parameters all have some negative correlations with process time measure.



Heatmap of parameters correlation

Results 02

Machine learning model selection - process time Measure

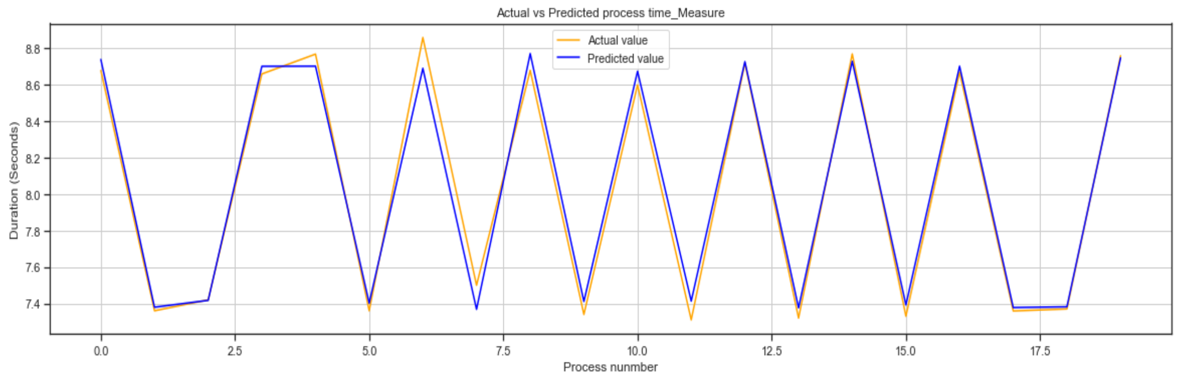
Testing data score

	ML	MAE	RMSE
0	MLR	0.0453234	0.0518738
1	SVR	0.0494875	0.0652214
2	DT	0.0719524	0.0837476
3	RF	0.0477764	0.0640288

Cross validation score

	ML	MAE	RMSE
0	MLR	0.0647519	0.0831661
1	SVR	0.0744971	0.0999033
2	DT	0.0879375	0.10661
3	RF	0.0697048	0.094825

Verifying the MLR model using new data



The Multiple Linear Regression algorithm produced the best results in this analysis, with the lowest error in MAE, RMSE of test data, and cross validation.

Results 02

Optimization for finding minimum process time - Measure Product

```
Differential evolution optimization result
Status : Optimization terminated successfully.
Total numbe of Evaluations: 2382
Total number of iterations: 11
process time_Lift: 6.699
handling time_Lift: 1.099
handling time_Measure: 1.101
process time_Position: 2.29
handling time_Position: 5.73
process time_Punch: 1.17
process time_Glue: 0.489
handling time_Glue: 17.699
process time_Unlift: 5.8
handling time_Unlift: 2.65
handling_Lift to Measure: 2.0
handling_Measure to Position: 1.101
handling_Position to Punch: 5.73
Minimum possible process time Measure:  [6.40459933]
```

The Differential evolution class from the optimize module in Scipy library was used to develop the function as part of optimization to find the minimum possible process time.

Differential evolution optimizer found an optimal minimum process time of 6.4 seconds after 2382 evaluations and 11 iterations

Conclusions

- Method 01 - Cannot be used to identify the machine's bottleneck process
- Method 02 - Measure Product is the main bottleneck in processEggs
  - Measure process takes the most time in the processEgg unit when compared to other process movements
  - Measure process is made up of several other units and processes
  - Handling movement from position to punch and glue to unlift takes more time due to long waiting times until the unit process step to be completed

- Process time measure prediction
  - Multiple Linear Regression algorithm produced the best results
  - with the lowest error in MAE, RMSE of test data, and cross validation

- Minimum possible process time - Measure Product
  - 6.4 seconds (Differential evolution optimizer discovered)

- Digital twin is referred to as digital model in this project
  - Machine history data used for input and all analysis done offline
  - No automatic link between the physical and digital objects

Recommendations

Project's suggestions for the next steps

1. Next step in Digital twin of this project is to develop a fully automatic data flow in either one way or both directions between the physical object and the digital object

2. Method 02:
    - Measure product process is made up of several other units and processes (Equipment modules and Control modules)
    - measure product process can increase efficiency by improving design, performing proper maintenance, increasing machine speed, and so on.
    - Insufficient for the giving a solution for any specific unit or module efficiency improvements

For future work on this project - Propose analysing the next level of the measure product.

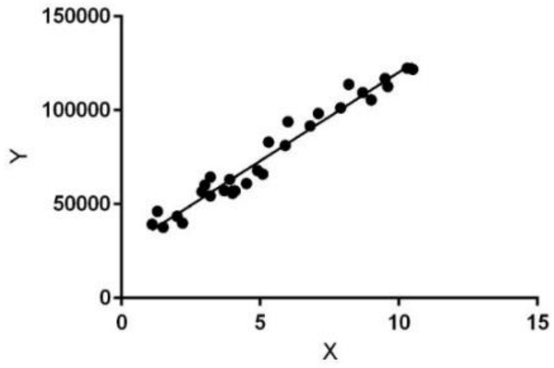
We can see specific unit or module efficiency improvements after performing the next level analysis.

Thank You.

Extra slides- losses Availability ,Performance, Quality

Availability	Performance	Quality
1. Unplanned Maintenance 2. Breakdowns ◦ Tooling Failures ◦ General Breakdowns ◦ Equipment Failure 3. Setup and Adjustments ◦ Setup/Changeover ◦ Operator Shortages ◦ Major Adjustments ◦ Warm-Up Time 4. Unable to Run Equipment ◦ Material Shortages ◦ Labour Shortages	1. Small Stops ◦ Obstructed Product Flow ◦ Component Jams ◦ Misfeeds ◦ Sensor Blocked ◦ Delivery Blocked ◦ Cleaning/Checking 2. Reduced Speed ◦ Rough Running ◦ Under Nameplate Capacity ◦ Equipment Wear ◦ Operator Inefficiency	1. Startup Rejects ◦ Scrap ◦ Rework ◦ In-Process Damage ◦ In-Process Expiration ◦ Incorrect Assembly 2. Production Rejects ◦ Scrap ◦ Rework ◦ In-Process Damage ◦ In-Process Expiration ◦ Incorrect Assembly

Machine Learning



Relationship between variables in the linear regression model

Multiple liner regression

- Two types of linear regression - Simple and Multiple liner regression
- When only one independent variable is present, simple linear regression is used to determine how linearly it relates to the dependent variable.
- MLR method, there are several independent variables for the relevant model to discover the relationship.
- Finding the best-fit linear line and the ideal intercept and coefficient values such that the error is minimized is the major goal of a linear regression model.

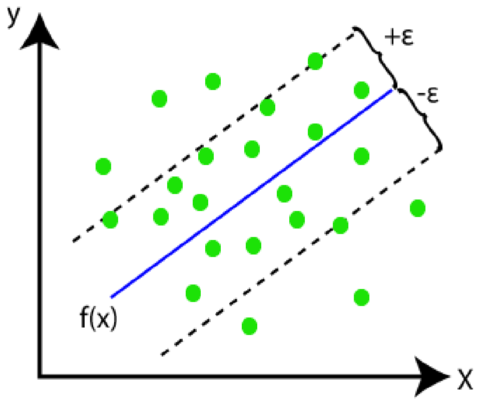
Simple linear regression equation with x as the independent variable, y as the dependent variable, b_1 as the coefficient or slope, and b_o as the intercept.

$$y = b_0 + b_1 x$$

Equation for Multiple Linear Regression, where y is the dependent variable and b_1, b_2, b_3,..., b_n are the coefficients or slopes of the independent variables x_1, x_2, x_3,..., x_n.

$$y = b_0 + b_1 x_1 + \; b_2 x_2 + b_3 x_3 \dots + b_n x_n$$

Relationship between variables in the Support Vector Regression



Support Vector Regression

The Support Vector Machine operates under the premise of this technique

In contrast to SVM, which is used to predict discrete categorical labels, SVR is a regressor that is used to forecast continuous ordered variables

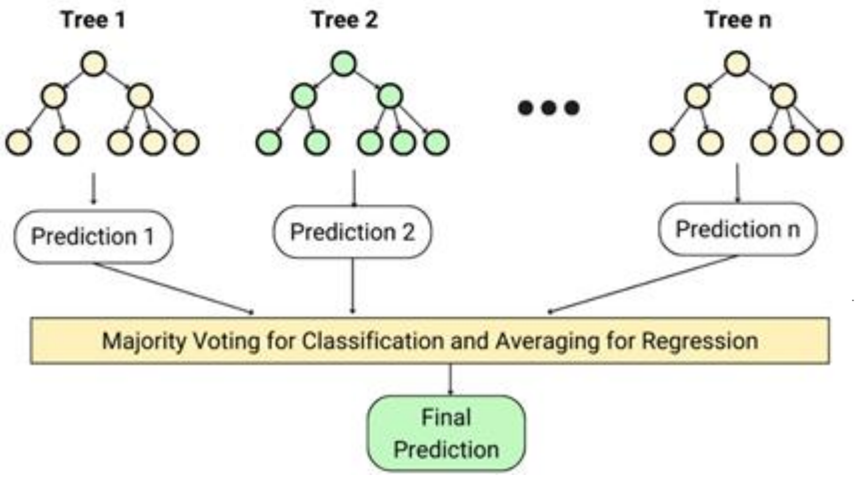
The goal of simple regression is to reduce error rates, whereas the goal of SVR is to fit the error within a predetermined threshold.

This means that the task of SVR is to approximate the best value within a predetermined margin known as the ε -tube. The goal of SVR is to find a hyperplane with a maximum margin that can accommodate the greatest number of datapoints.

The main objective of SVR is to take into account as many datapoints as possible within the boundary lines, and the hyperplane (best-fit line) must include as many datapoints as possible.

Here, the blue line is titled hyperplane, and the other two lines are identified as boundary lines

Random Forest



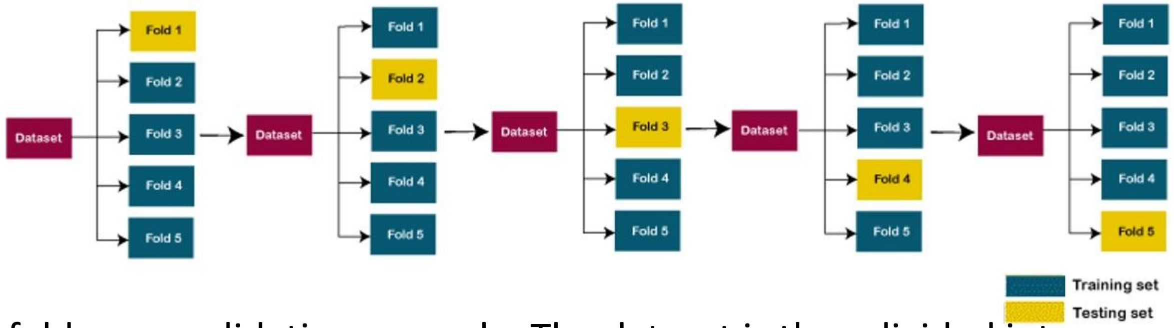
Random forest (RF) are frequently employed in classification and regression problems, and it creates decision trees on various samples and takes their most vote for classification and average in case of regression.

Step 1 : choose random samples from a given data or training set

Step 2: individual decision trees are constructed for each sample, and each decision tree will generate an output.

Finally : output is reflected based on majority voting or averaging for classification and regression respectively.

- RMSE- A lower RMSE value indicates good model performance because it shows how closely the predicted values match the actual values.
- MAE- The average differences between predicted values and actual values
- **K-Fold Cross-Validation**



5-fold cross-validation example. The dataset is then divided into five folds. The first fold is saved for testing the model on the first iteration, and the remaining folds are used to train the model. The second fold is used to test the model on the second iteration, and the remaining folds are used to train the model. Until no fold can be used for the test fold, this procedure will continue.

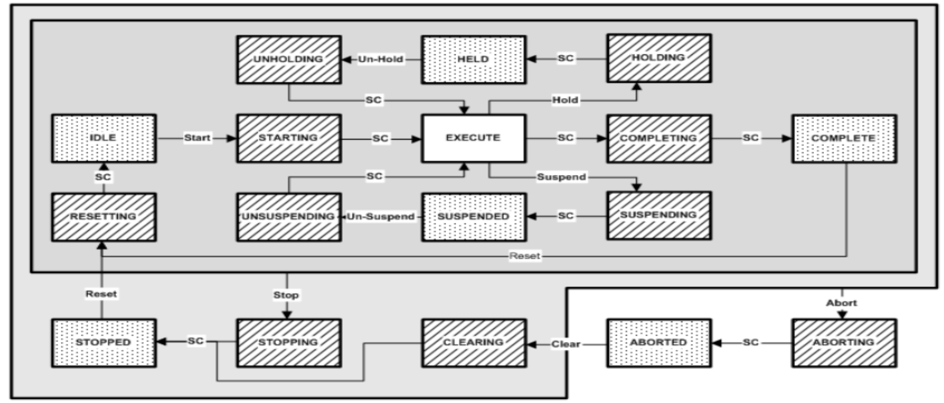
Outlier detection - Quantile rule

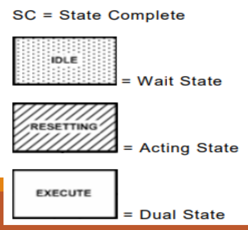
Quantile rule is means that any number that lies outside of the range (Q1-1.5*IQR to Q3+1.5*IQR) is regarded as an outlier. (Q1 stands for the first quartile, Q3 for the third, and IQR for the interquartile range.)

PackML State Model

Packaging Machine Language (PackML) is an automation standard established by the Organization for Machine Automation and Control (OMAC) and by the International Society of Automation's

The State Model defines the level control of the machine. It coordinates the operation of the machine's components to execute different actions and react to whatever is happening.





PackML State Model

Definitions of PackML states

State Name	Description
STOPPED	**State Type: Wait** The machine is powered and stationary after completing the STOPPING state. All communications with other systems are functioning (if applicable). A Reset command will cause an exit from STOPPED to the RESETTING state.
STARTING	**State Type: Acting** This state provides the steps needed to start the machine and is a result of a starting type command (local or remote). Following this command the machine will begin to Execute.
IDLE	**State Type: Wait** This is a state which indicates that RESETTING is complete. This state maintains the machine conditions which were achieved during the RESETTING state, and performs operations required when the machine is in IDLE.
SUSPENDING	**State Type: Acting** This state is a result of a change in monitored conditions due to process conditions or factors. The trigger event will cause a temporary suspension of the EXECUTE state. SUSPENDING is typically the result of starvation of upstream material in-feeds (i.e., container feed, beverage feed, crown feed, lubricant feed, etc.) that is outside the dynamic speed control range or a downstream out-feed blockage that prevents the machine from EXECUTING continued steady production. During the controlled sequence of SUSPENDING the machine will transition to a SUSPENDED state. The SUSPENDING state might be forced by the operator.

State Name	Description
SUSPENDED	**State Type: Wait** The machine may be running at a relevant set point speed, but there is no product being produced while the machine is waiting for external process conditions to return to normal. When the offending process conditions return to normal, the SUSPENDED state will transition to UNSUSPENDING and hence continue towards the normal EXECUTE state. Note: The SUSPENDED state can be reached as a result of abnormal external process conditions and differs from HELD. HELD is typically a result of an operator request or an automatically detected machine fault condition that should be corrected before an operator request to transition to the UNHOLDING state will be processed.
UNSUSPENDING	**State Type: Acting** This state is a result of a machine generated request from SUSPENDED state to go back to the EXECUTE state. The actions of this state may include ramping up speeds, turning on vacuums, and the re-engagement of clutches. This state is done prior to EXECUTE state, and prepares the machine for the EXECUTE state.
EXECUTE	**State Type: Dual** Once the machine is processing materials it is deemed to be executing or in the EXECUTE state. Different machine modes will result in specific types of EXECUTE activities. For example, if the machine is in the Production mode, the EXECUTE will result in products being produced, while in Clean Out mode the EXECUTE state refers to the action of cleaning the machine.
STOPPING	**State Type: Acting** This state executes the logic which brings the machine to a controlled stop as reflected by the STOPPED state. Normal STARTING of the machine can not be initiated unless RESETTING had taken place.
ABORTING	**State Type: Acting** The ABORTED state can be entered at any time in response to the Abort command or on the occurrence of a machine fault. The aborting logic will bring the machine to a rapid safe stop. Operation of the emergency stop will cause the machine to be tripped by its safety system. It will also provide a signal to initiate the ABORTING State.
ABORTED	**State Type: Wait** This state maintains machine status information relevant to the Abort condition. The machine can only exit the ABORTED state after an explicit Clear command, subsequently to manual intervention to correct and reset the detected machine faults.
HOLDING	**State Type: Acting** When the machine is in the EXECUTE state, the Hold command can be used to start HOLDING logic which brings the machine to a controlled stop or to a state which represents HELD for the particular unit control mode. A machine can go into this state either when an internal equipment fault is automatically detected or by an operator command. The Hold command offers the operator a safe way to intervene manually in the process (such as removing a broken bottle from the in-feed) and restarting execution when conditions are safe. To be able to restart production correctly after the HELD state, all relevant process set-points and return status of the procedures at the time of receiving the Hold command must be saved in the machine controller when executing the HOLDING procedure.

State Name	Description
HELD	**State Type: Wait** The HELD state holds the machine's operation while material blockages are cleared, or to stop throughput while a downstream problem is resolved, or enable the safe correction of an equipment fault before the production may be resumed.
UNHOLDING	**State Type: Acting** The UNHOLDING state is a response to an Operator command to resume the EXECUTE state. Issuing the Unhold command will retrieve the saved set-points and return the status conditions to prepare the machine to re-enter the normal EXECUTE state. Note: An operator Unhold command is always required and UNHOLDING can never be initiated automatically.
COMPLETING	**State Type: Acting** This state is an automatic response from the EXECUTE state. Normal operation has run to completion (i.e., processing of material at the infeed will stop).
COMPLETE	**State Type: Wait** The machine has finished the COMPLETING state and is now waiting for a Reset command before transitioning to the RESETTING state.
RESETTING	**State Type: Acting** This state is the result of a RESET command from the STOPPED or complete state. RESETTING will typically cause a machine to sound a horn and place the machine in a state where components are energized awaiting a START command.
CLEARING	**State Type: Acting** Initiated by a state command to clear faults that may have occurred when ABORTING, and are present in the ABORTED state before proceeding to a STOPPED state.

Every single unit of processEggs process time, cycle time ,handling time analysis