



Article

A Machine-Learning-Assisted Simulation Approach for Incorporating Predictive Maintenance in Dynamic Flow-Shop Scheduling

Eman Azab ^{1,*} , Mohamed Nafea ², Lamia A. Shihata ² and Maggie Mashaly ³ 

¹ Electronics Department, Faculty of Information Engineering Technology, German University in Cairo, New Cairo 11835, Egypt

² Design and Production Department, Faculty of Engineering and Materials Science, German University in Cairo, New Cairo 11835, Egypt; mohamad.nafea@guc.edu.eg (M.N.); lamia.shihata@guc.edu.eg (L.A.S.)

³ Networks Department, Faculty of Information Engineering Technology, German University in Cairo, New Cairo 11835, Egypt; maggie.ezzat@guc.edu.eg

* Correspondence: eman.azab@guc.edu.eg

Abstract: In this paper, a machine-learning-assisted simulation approach for dynamic flow-shop production scheduling is proposed. This is achieved by introducing a novel framework to include predictive maintenance constraints in the scheduling process while a discrete event simulation tool is used to generate the dynamic schedule. A case study for a pharmaceutical company by the name of Factory X is investigated to validate the proposed framework while taking into consideration the change in forecast demand. The proposed approach uses Microsoft Azure to calculate the predictive maintenance slots and include it in the scheduling process to simplify the process of applying machine-learning techniques with no need for hard coding. Several machine-learning algorithms are tested and compared to see which one provides the highest accuracy. To gather the required dataset, multiple sensors were designed and deployed across machines to collect their vitals that allow the prediction of whether and when they require maintenance. The proposed framework with discrete event simulation generates optimized schedule with minimum makespan while taking into consideration predictive maintenance parameters. Boosted Decision Tree and Neural Network algorithms showed the best results in estimating the predictive maintenance slots. Furthermore, the Earliest Due Date (EDD) model produced the minimum makespan with 76.82 h while scheduling 25 products using 18 machines.

Keywords: artificial intelligence; data acquisition system; discrete event simulation; industry 4.0; machine learning



Citation: Azab, E.; Nafea, M.; Shihata, L.A.; Mashaly, M. A Machine-Learning-Assisted Simulation Approach for Incorporating Predictive Maintenance in Dynamic Flow-Shop Scheduling. *Appl. Sci.* **2021**, *11*, 11725. <https://doi.org/10.3390/app112411725>

Academic Editor: Paolo Renna

Received: 14 November 2021

Accepted: 8 December 2021

Published: 10 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Info and Communication Technology (ICT) is currently playing a significant role in increasing the efficiency of production scheduling at industrial entities. This is made possible by integrating multiple software solutions to minimize the makespan and reach optimal resources use. Most of small to mid-sized manufacturing entities plan their production activities according to market demand. The production scheduling on the shop floor level is designed by the production planning staff in a form of static schedule. Disruptions to the planned static schedule such as: machine breakdowns, increased order priority, sudden orders, quality assurance tests failure and order cancellation require constant update to the initial static schedule. These disruptions change the nature of the scheduling task making it a highly dynamic one, and decreases the overall efficiency of the industrial entity due to repeating the scheduling process. Resolving machine breakdowns and sudden orders disruptions is not a straightforward task. Since, machine repair time is unpredictable and forecasting sudden orders arrival needs exhaustive market research.

Consequently, it is crucial that any automated scheduling tool should handle machine failures and sudden orders efficiently and as quickly as possible.

The main focus of this work is to propose a simple yet efficient framework that automates dynamic flow-shop scheduling while taking into consideration machine failures and rush orders. The proposed framework is implemented by commercial software tools used for scheduling. To handle machine failures, predictive maintenance parameters are calculated using machine-learning techniques and included in the makespan. A case study is presented for a pharmaceutical company to validate the proposed framework. The paper is organized as follows: literature review about dynamic scheduling and maintenance types is presented in Section 2. The proposed scheduling framework is given in Section 3 with detailed description of the predictive maintenance parameters calculations. A real-life case study of a pharmaceutical factory X production scheduling using the proposed framework is presented in Section 4. Results and discussion are given in Section 5 and finally, the paper is concluded in Section 6.

2. Literature Review

Researchers tackled the production scheduling problem as an optimization problem that can be solved either using exact or metaheuristic methods. Nevertheless, integrating both methods can be done to achieve better results. Example for exact method approach was used in [1,2]. The authors used Mixed Integer Linear Programming (MILP) to tackle the production scheduling problem. Their results were compared to existing commercial solver algorithms and showed better results in solving large scale scheduling problems. The work in [3] tackled the inter-related problem of process configuration, lot-sizing, and scheduling problem of the packaging industry by developing an exact branch-and-check algorithm. Mixed Integer Programming (MIP)-based heuristic was used to strengthen the optimal solutions found by the branch-and-check algorithm to reduce setup times and cost. The authors in [4] developed an exact branch and bound algorithm and a heuristic tabu search algorithm to solve a special parallel machine scheduling problem. Tests and numerical experiments were conducted to test the performance of the solution methods. The results showed that the solution methods are efficient and effective in solving problems of different sizes. Further advanced approach based on Artificial Intelligence (AI) is proposed by authors of [5]; where they proposed a Reinforcement Learning (RL) approach for solving the dynamic flexible job-shop scheduling problem with the aim of minimizing the makespan. RL relies on the availability of historical data sets (previous executed schedules) to achieve acceptable results. RL was able to generate a production schedule with minimum makespan for a pharmaceutical factory case study.

For the metaheuristic methods, the work in [6] developed a Two-phase Decoding Genetic Algorithm (TDGA) to maximize use in the TFT-LCD industry. The proposed algorithm was then compared to other metaheuristic methods as a measure of performance. Eight scenarios were tested, and the results showed the practical viability of the proposed TDGA algorithm. In [7], a fast Genetic Algorithm (GA) was developed to solve capacity planning and scheduling problems in the bio-pharmaceutical industry. The proposed algorithm was verified in two different case studies in terms of problem size. The algorithm was also compared to MILP in terms of speed of finding the global optimum. The results showed that the algorithm was able to achieve the global minimum 3.6 times faster than the MILP in case study 1, while in case study 2, which was on a longer time horizon, the GA achieved a near optimal solution as fast as the MILP. The authors in [8] designed a GA to minimize the total tardiness in a hybrid flow-shop scheduling problem. Unrelated machines and machine eligibility were the only practical assumptions considered. The proposed algorithm was compared to state of the art algorithms on 450 instances with different correlation patterns and sizes of operating processing time. The effectiveness of the proposed algorithm was validated by the results.

GA is used extensively in scheduling problems optimization. For example, the authors in [9] used two adapted versions of GA and made an extensive study in solving the flexible

scheduling problem and compared between them. Another metaheuristic framework for modeling of market demand in the semiconductor industry was developed in [10]. They assessed the performance of two metaheuristic methods; one is a GA and the other is a rule-based assignment procedure. A Discrete Event Simulation (DES) model is used to simulate one of the stages of the semiconductor manufacturing processes. The results showed that it was better to use the rule-based assignment method when stability is a concern, but if stability is not a major concern, then the GA would be better due to its superior optimization features.

Researchers have also included hybrid models to solve the scheduling optimization problem. An integrated optimization model was developed in [11] to minimize the manufacturing time of recycled parts. The model was integrated with an intelligent hybrid algorithm, a GA, neural network and simulation technique to optimize re-manufacturing planning. The model was tested on a case study and gave positive results. Another hybrid model was given in [12] that used hybrid computer simulation Artificial Neural Network (ANN) to minimize the makespan for a job-shop manufacturing system. For its ability to deal with complex, stochastic and nonlinear problems, DES was used to provide the training data for the ANN to estimate the makespan value for the dispatching rules available to find the optimal solution of the job-shop scheduling problem.

Since the scheduling problem is very challenging, many special purpose simulation tools were designed by researchers to solve this problem. Example for this approach, is the research done in [13], where the researcher developed their own simulation tool using Java script to solve the scheduling optimization problem with their proposed evolutionary computation method. Other work used preexisting software simulation tools as in [14,15]. These tools use dispatching rules and heuristic methods to generate the production planning schedule. The researchers used these tools to verify their proposed mathematical models for the scheduling optimization problem.

Although previous works investigated various approaches and techniques in solving the production scheduling problem as an optimization problem, they did not include maintenance slots in their work. Nevertheless, including maintenance slots in the production scheduling would improve the overall efficiency and reduce unplanned downtime. There are three main maintenance techniques in the literature: corrective maintenance, preventive maintenance, and predictive maintenance [16].

- Corrective Maintenance (CM) or Run-to-Failure (R2F) or Reactive Maintenance (RM) happens only when an equipment stops working. This is the simplest maintenance strategy adding a direct cost to the process, since it is necessary both the stop on the production and the repair of the parts to be replaced.
- Preventive Maintenance (PvM) or predetermined Time-Based Maintenance (TBM) is a maintenance technique performed periodically with a planned schedule in time or process iterations to avoid process/equipment failures. It is generally an effective approach to avoid failures. However, unnecessary corrective actions are taken, leading to an increase in the operating costs.
- Predictive Maintenance (PdM) uses predictive tools to determine when maintenance actions are necessary. It is based on continuous monitoring of a machine or a process integrity, allowing maintenance to be performed only when it is needed. Moreover, it allows the early detection of failures thanks to predictive tools, health monitoring data (e.g., visual aspects, wear, coloration different from original, among others), statistical inference methods and engineering approaches.

The total maintenance cost is the lowest when PdM is used [17]. Through the integration of AI and real-time Data Acquisition (DAQ), PdM time slots can be computed. Machine Learning (ML) with real-time DAQ can easily predict causes of disruptions in the manufacturing environments which allows a quick response to changes in shop floor in a timely and cost-effective manner.

Recently, researchers tried to integrate PdM in the process of scheduling using various approaches. One of the early work done with this concept is the work presented in in [18].

It proposed a new problem formulation for the scheduling optimization problem to include maintenance calculations. The method used in [18] is very complex as it needs continuous change and skilled researcher to map it with the real-life problem. A recent adaptation for this work is presented in [19], where the authors used a proposed GA model that included Condition Based Monitoring (CBM) maintenance technique—a simplified form of PdM—in the scheduling problem and verified their work with MATLAB. Applying this concept in a real-life case study is not a straightforward task. This approach needs skilled users to apply this approach efficiently. In [20], the authors developed a web-service tool to integrate scheduling with maintenance calculations. The work in [20] included PdM and PvM calculations in the scheduling process but it did not use ML technique for this task. The work done in [21] used ML algorithm to calculate PdM parameters; however, it was implemented using Python which requires a skilled programmer to adapt this approach to a new case study. Finally, with the proposal of the Digital Twin (DT) concept, a complete framework to integrate maintenance and scheduling was proposed in [22]. It relies on using multiple software tools to achieve the required task and consequently in requires a lot of software interfacing between the various tools used.

Consequently, a simplified framework that integrates PdM calculations using ML algorithms in the scheduling process is needed. Using commercial software tools for PdM calculations that does not need hard coding and be used by the scheduling simulation tools without complicated interfacing codes could be an efficient approach. In this paper, a novel framework for dynamic flow-shop scheduling is introduced. It employs only two software tools to include estimated PdM time slots in the dynamic schedule. The used software tools in the proposed framework employ ML algorithms and DES for PdM and scheduling, respectively. The proposed framework methodology is presented in the coming section.

3. Methodology

The framework proposed in this paper is shown in Figure 1, which is divided into four levels. The first level of the framework describes the physical system of the flow-shop under investigation. A complete and detailed description of the flow-shop including its process map and the used machines.

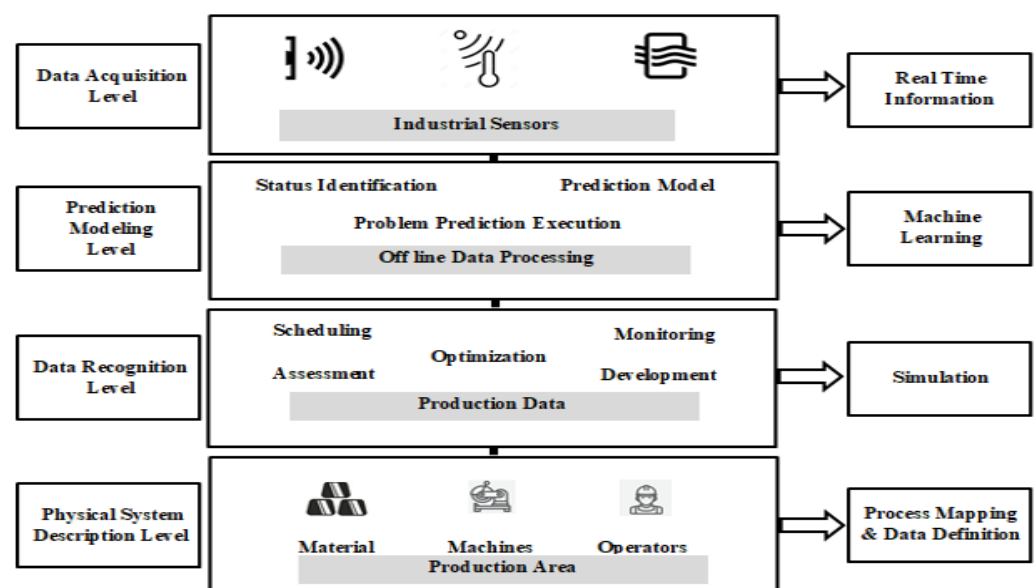


Figure 1. Proposed Framework.

In the data recognition level, DES tool is used to define the scheduling problems constraints. Before the simulation takes place, the top layers of the Data Acquisition (DAQ) using sensors and the prediction modeling level are used to monitor the machines of

the flow-shop. The nature of the data and the sensors type are selected based on the industrial process under investigation. The collected data are passed on to the prediction modeling level where machine learning is employed. At this prediction level, the data collected by the DAQ hardware is used to estimate the predictive maintenance slots for the machines existing in the production area. The predictive maintenance slots are taken into consideration by the simulator used to generate the schedule. In the following subsections, the framework is explained in detail.

3.1. Production Scheduling Using DES

DES is a very useful tool for industrial entities. It can be used to estimate the effects of possible changes in the production process on the production line performance before applying those changes in real life. These changes can be as simple as changing worker's assignment, number of machines or even the positioning of machines within the production line itself. To use DES efficiently, a simulation framework must be used to make sure that the result is consistent with the real-life case study. The proposed DES framework used in this work is an adapted version from the one presented in [23], it consists of eight phases: problem definition, production line analysis, data collection, initial layout construction, model building, model testing, experimentation, and finally results analysis and conclusion. The proposed DES framework is shown in Figure 2.

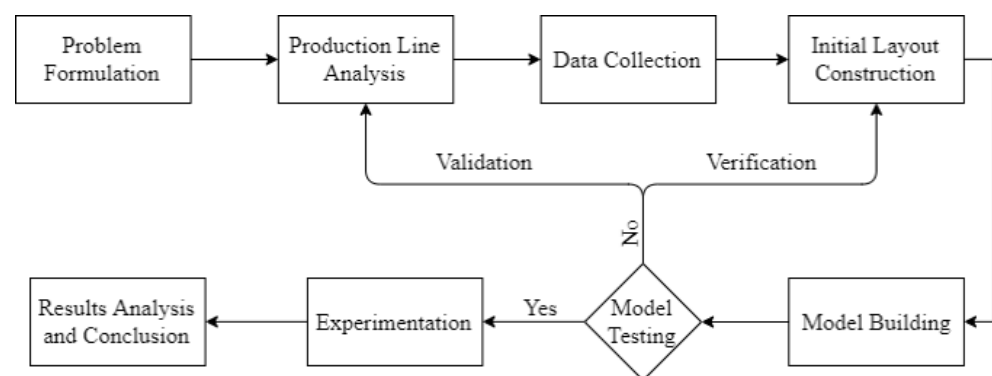


Figure 2. Proposed DES Framework.

In phase one, a clear problem definition must be determined early on to make sure that the researcher fully understands the required output. The aim of this work DES is to produce a dynamic production schedule for a flexible flow-shop production line of a specific industrial case study. This production schedule must be of the minimum makespan possible. The second phase includes analysis of the production line, after the problem is defined, a thorough analysis must be done to completely understand the production line. This must be achieved after defining the boundaries of the system, which machines to study and which products scope to work on. The production line analysis is done by creating a simple model to represent the real-life one. This can be achieved through stating the production line specs, such as:

- Machines number
- Workstations number
- Weekly work-shifts number
- Work-shifts duration
- Products type
- Production line manufacturing type (manual labor or automated)
- Workers' tasks per workstation
- Products routing
- Transportation methods per workstation (forklift, crane, trolley, etc.)
- Processing time of each product on every machine

The third phase is collecting the data based on the production line analysis. The nature of the required data depends on the production line specs mentioned above. The data required can be collected from two methods either through historical data that was already collected earlier manually or through installed sensors and observation methods. The second method is through the conduction of a time and motion study. The fourth phase includes the construction of the initial layout to show how the production line works. This is achieved using the data collected from phase three. This initial layout serves as representation of the real-life case and represents the data collected from phase three in a graphical manner. This simplifies the process of understanding how the production line works.

The fifth phase includes the building of the simulation model based on the initial layout and the data collected from the production line analysis. The simulation model is constructed using the software tool available and the data collected from the production line analysis is used as input. In this work, FLEXSIM software for this task.

The sixth phase includes validation and verification of the simulation model. To verify the simulation model, either static or dynamic testing is used. For this paper, static testing is used where every entity used through the simulation model was observed by a peer to determine whether it behaves the same as seen in the initial layout produced or not, correctness action is taken if there was an entity within the simulation model that did not follow the concept model. An initial model is constructed for the validation process, an old production schedule formerly used by the factory is used to validate that the simulation model produces the same makespan of the old schedule. This was done to test the level of accuracy of the simulation model and how close it represents the real case.

Phase seven is mainly concerned with the model experimentation, which is trying different solutions to achieve the goal set from phase one. The final phase involves analyzing the results of the simulation model, by comparing the suggested schedule by the simulation to previous executed schedules by the industrial entity to evaluate its feasibility. In this work, FLEXSIM software tool is used to perform the DES to generate the production schedule. Within the experimentation phase, two main criteria are considered: which dispatching rule is used to produce the schedule and when to add the machine breakdown time slots and breakdown times. For the dispatching rule, three main dispatching rules were considered, Earliest Due Date (EDD), Shortest Processing Time (SPT) and Longest Processing Time (LPT). These dispatching rules were considered for the fact that they are the most commonly used and that the case study considered in this paper uses EDD. For the machine breakdown time slots, PdM was explored since there is no guarantee that no extra cost would be inflicted on the manufacturing system. First, the data concerning the machines health is collected through an implemented DAQ hardware system. This serves one of the overall objectives of the ongoing research in Industrial Internet of Things (IIoT) and Industry 4.0. The machine health data are processed to be suitable as input for ML software tools to estimate the PdM time slots. The processed data are uploaded to Microsoft Azure software tool (the platform of the ML engine). In the following subsection, the two top levels of the proposed methodology framework is explained.

3.2. PdM Time Slots Estimation Using ML

To estimate the PdM time slots in the proposed framework, a ML software platform is used. Microsoft Azure software tool is used to construct the ML models where prediction takes place. Each model predicts a certain required aspect related to PdM using different ML algorithms. The flowchart of the PdM time slots estimation is shown in Figure 3.

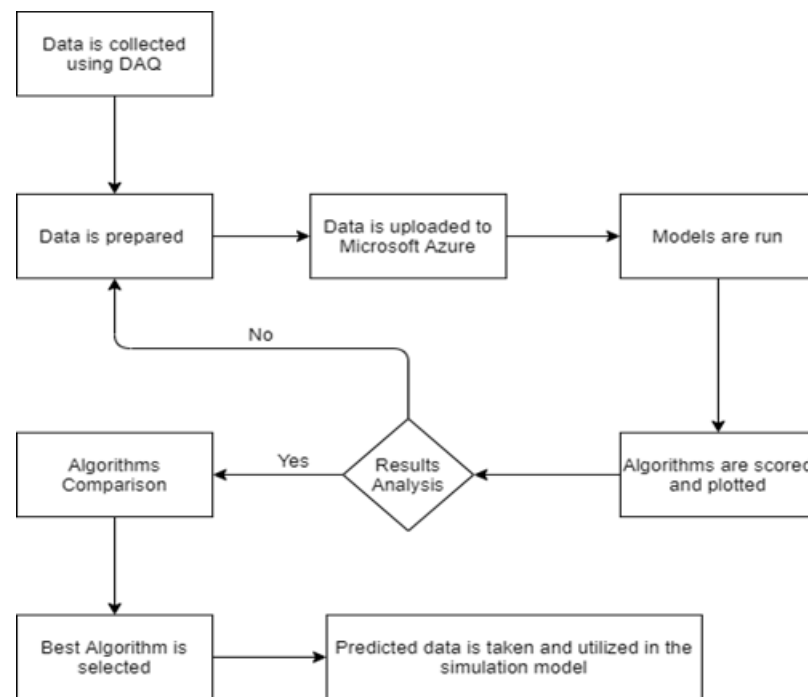


Figure 3. Flowchart of PdM time slots estimation.

The models are then run using the uploaded machine health data and the results of each model are then reviewed to see which algorithm performed well enough to consider its results. If the results of each algorithm passes a predefined threshold, then comparison between the results of each model is done to determine which algorithm would be used to predict the required PdM aspect. After the best algorithm is chosen, its predicted data of the PdM time slots is then used in the DES model constructed in Section 3.1.

Microsoft Azure is used to construct three models for PdM data. The first model is a regression model to predict the Remaining Useful Life (RUL) of a specific machine, the second model is a binary classification model to predict the time frame at which machine failure might occur in days and the final model is a multi-class classification model to predict when machine failure might occur in weeks.

According to the flowchart in Figure 3, the data are first uploaded to Microsoft Azure after being prepared. Second, the software tool runs the models multiple times using different algorithms to determine the most accurate algorithm and required parameters to produce the best results. The results of each algorithm is scored and analyzed to determine if the results are deemed acceptable or not. If results analysis are acceptable, then the algorithms results are compared with each other to determine the best performing one, else, the data preparation process starts again to achieve better results.

For each model, the software tool recommends specific set of parameters. These parameters are chosen according to the complexity of the case study and to avoid over-fitting which is the phenomenon that appears when trying to enhance the accuracy of the results by increasing or decreasing the variables. Over-fitting may cause deterioration in the accuracy of the results. One method to verify that the parameters values are optimum, it was suggested to manipulate them to see whether it would affect the results or not. Hence, the parameters for a specific algorithm have been used as is, doubled and reduced by half to generate three test cases.

Therefore, three cases are tested for the parameters, the first case is the normal case and it uses the values for the parameters as they were given, we will call this the normal case in our study. The second case is the half case, and it uses half the values of the parameters of the normal case. The third and final case is the double case and it uses double the value of the parameters of the normal case. The results are evaluated based on two main criteria:

the model run time and the values of the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE) for the regression model or accuracy and precision for the binary and multi-class classification models. The difference in the second evaluation criterion between the models is due to the difference of their output nature. For Regression models, the output is usually continuous and hence error is required to be calculated which is the deviation from the required output. In classification models, accuracy and precision are used since the output of classification models is discrete. The output of accuracy and precision values are presented between zero and one, with zero presenting zero percent and one presenting 100% while the MAE and RMSE values are presented between zero and one hundred percent.

In the proposed framework, a simple hardware DAQ system is required for machine monitoring to collect data for the PdM models. The data are stored in a cloud-based service to be exported when needed. Multiple sensors are used to collect data over long periods of time for PdM parameters calculations. This is attributed to the fact that the algorithms responsible for learning and prediction need large data sets to have optimal results. The processes of data transfer from the sensors to the cloud service and its preparation are shown in Figure 4.

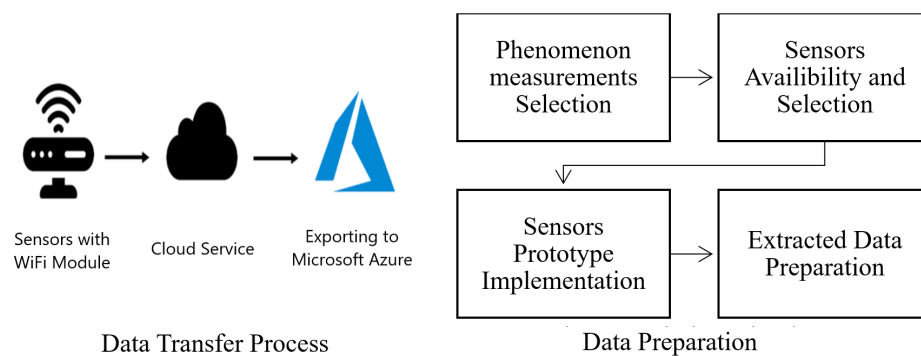


Figure 4. Data Transfer Process and Preparation flowchart.

From the industrial point of view, several steps must be followed to determine the type of sensors used, the nature of the measured data, and the sensor locations. Most of the previous research work done on PdM monitor the machine's vibration, temperature and pressure. Several factors must be considered when determining which sensors to use such as cost and the accuracy of its measured readings. For example, if the aim of the sensor is to measure high values of temperature, one must make sure that the sensor operating range can measure such high temperatures or else the collected data will be inaccurate or the sensor itself will be damaged.

To send the sensors' data to the cloud service, they must include a networking module that connects them to the Internet through a gateway. Possible connectivity options could be WiFi or Bluetooth, where both can be used by integrating the respective module onto the sensor. There are several types of commercial sensor modules that fits industrial applications such as the C series and the PXI series for temperature, pressure and vibration measurements. In our case study, the proposed sensors are small in size to ease the process of installation within the machine. The final step is determining the exact location where the sensors are going to be implemented. In this case, the sensors are going to be implemented within the body of the machine itself not within the working area of the products on the machine to obtain accurate readings. Once the data becomes available on the cloud servers, its format will be changed to be compatible with Microsoft Azure software tool. This is done over several operations to create the required format.

4. Case Study

To verify the proposed methodology, a real-life case study of a pharmaceutical company by the name of Factory X is considered. Factory X's production line consists of 7 different workstations with a total number of 18 machines. From the hundreds of products the production line can produce, it was seen fit to only select 25 of them since these products are the most demanded ones. All the data used for these products and the layout of the production line were provided by the factory.

4.1. Machine-Learning Results

The machine-learning results would first be tackled since that it is needed to determine the PdM slots for the production schedule before the use of DES. Hence, it is required to compare the results of the models and algorithms mentioned earlier. The dataset used in this research was a historical one used for similar research study cases, it describes several machines used in different workstations. The data consisted of multiple columns such as the machine id, the lifetime of the machine in cycles and multiple sensor readings to various factors such as vibration, temperature and pressure. The results are presented in units of percentage. The data were processed to be compatible with Microsoft Azure.

4.1.1. Regression Model

The regression model that predicts the RUL of a specific machine, the RMSE and MAE for the three study cases are shown in Figures 5–7, respectively. The best algorithm for the normal and double cases is the Decision Forest Regression.

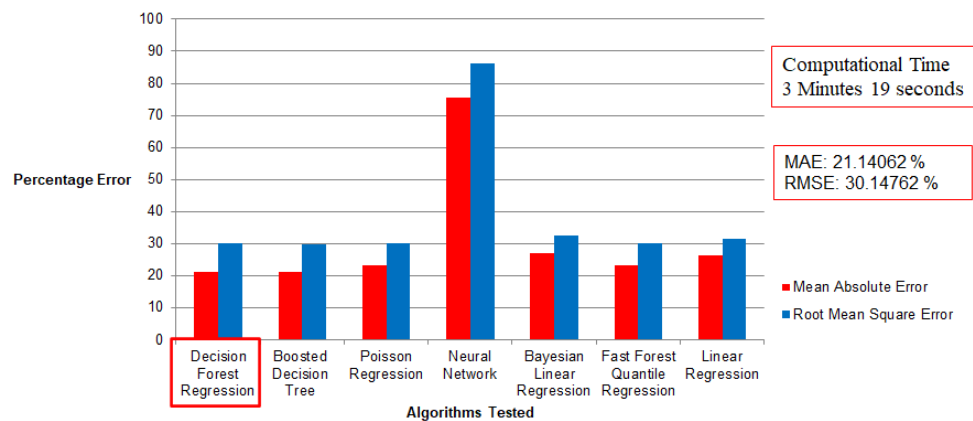


Figure 5. Regression model results for normal case.

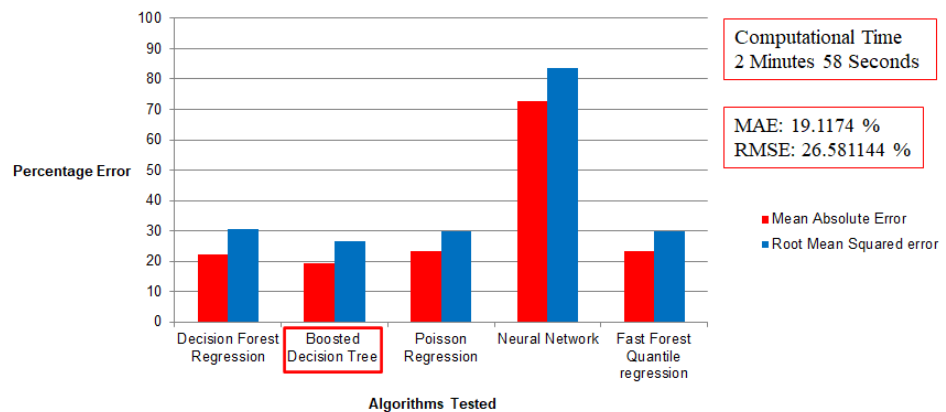


Figure 6. Regression model results for half case.

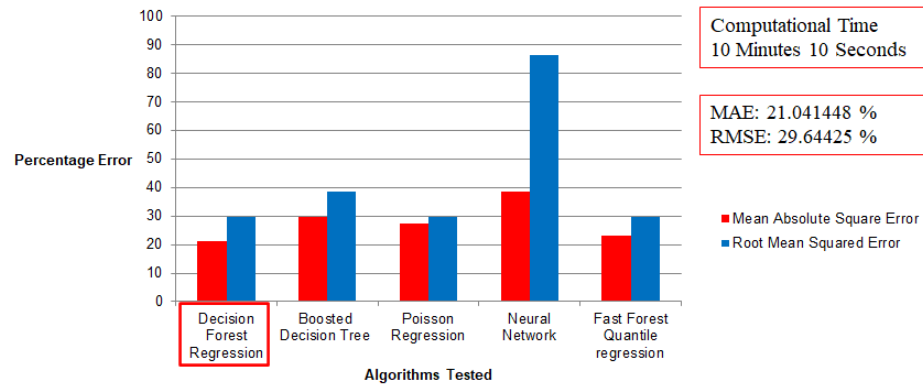


Figure 7. Regression model results for double case.

For the half case, the Boosted Decision Tree showed the best results. The Poisson Regression and Fast Forest Quantile Regression results is close to the optimum ones. The worst results are those of NN algorithm in all three cases, though the MAE is slightly improved in the double case. These results are due to one main reason which is that NNs algorithm is known to perform better with large data sizes in regression models. Some of the mentioned algorithms such as the Bayesian Linear Regression and Linear Regression are tested only in the normal case because their parameters are associated with linearization and hence it was decided not to use them to avoid over-fitting. As for the computational time, it increased significantly in the double case, while for the half case, it decreased compared to the normal case. When doubling the value of the parameters of the algorithms in question, the computational time suffered greatly with it increasing from 3 to around 10 min with no significant improvement in RMSE. In conclusion, the best parameters to select are the half case using Boosted Decision Tree algorithm.

4.1.2. Binary Classification Model

The Binary classification model results for the three cases are shown in Figures 8–10, respectively. It shows the accuracy and precision for various ML algorithms while predicting when the machine might fail in days. The best performing algorithm in all three cases is the NN followed by the Decision Forest algorithm, while the remaining ones show similar results. However, the Two-Class Support Vector Machine algorithm scored a 100% in precision which is significantly high. The worst performing algorithm is the Boosted Decision Tree algorithm. As for the computational time, there is no significant change between the normal case and the half case while the double case took more than the normal case by 12.5%.

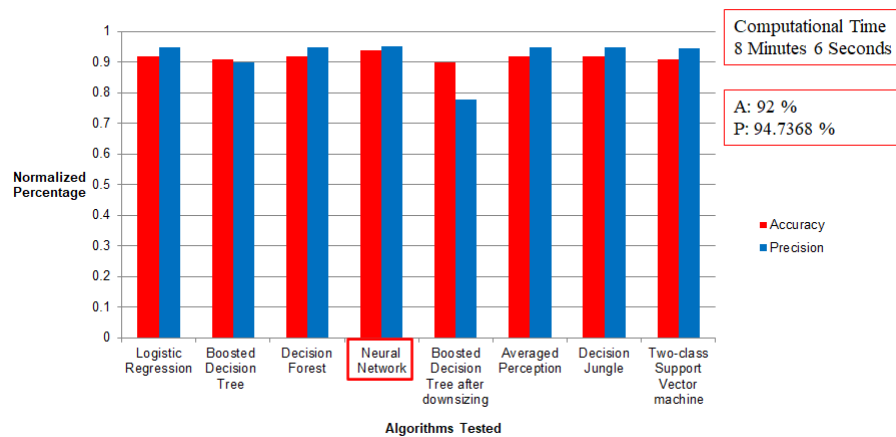


Figure 8. Binary classification model results for normal case

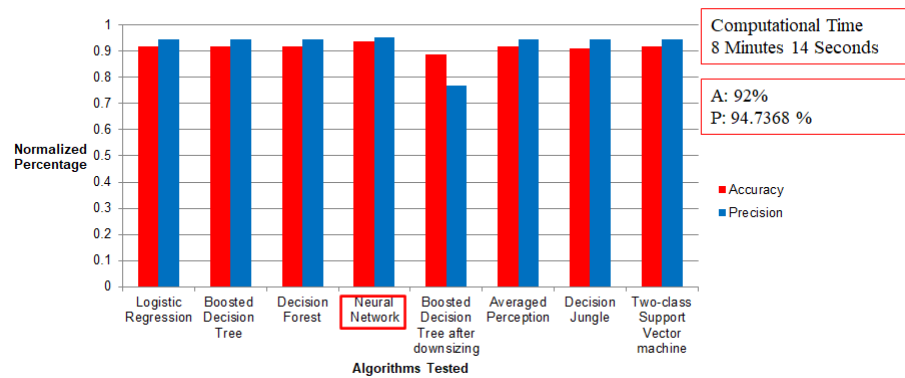


Figure 9. Binary classification model results for half case.

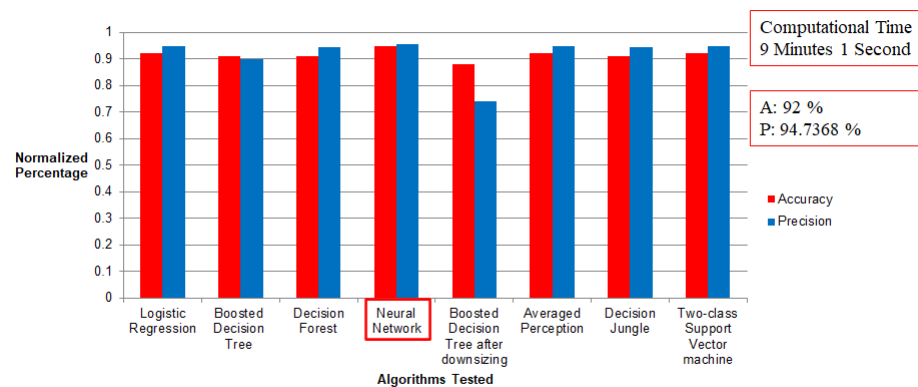


Figure 10. Binary classification model results for double case.

In conclusion, the parameters of the normal case would be the ones that will be used for binary classification model if computation time is not an issue. Since the normal case has been proven to be the best case, no further investigations would be conducted concerning the other two cases: the half case and double case.

4.1.3. Multi-Class Classification Model

For the multi-class classification model, the results are shown in Figures 11–13 for the normal, half and double case, respectively. The best case is the normal case using Multi-class NNs algorithm with respect to accuracy of results and computational time. The double case’s computational time is significantly high with almost triple the normal case time. The half case showed no significant change in computational time.

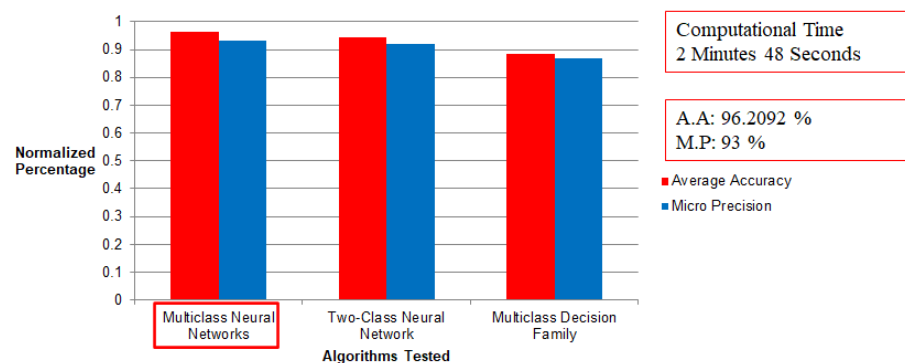


Figure 11. Multi-class classification model results for normal case.

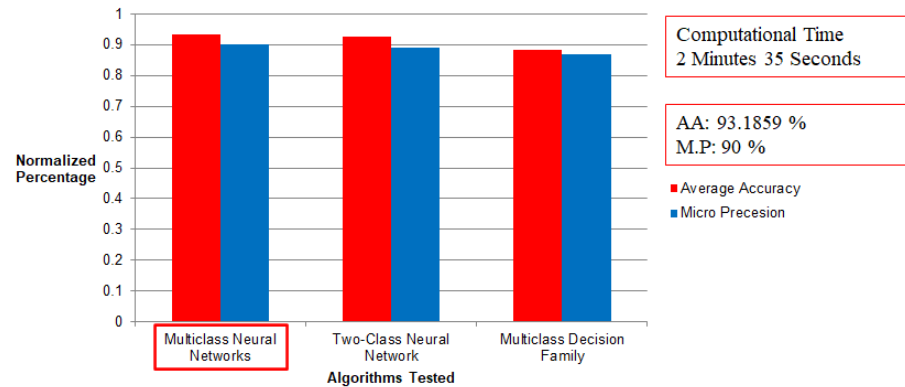


Figure 12. Multi-class classification model results for half case.

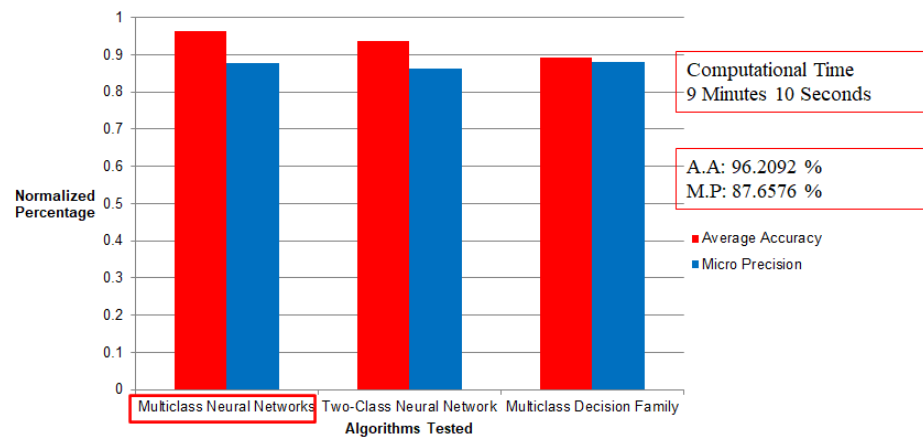


Figure 13. Multi-class classification model results for double case.

On the other hand, both the Multi-class NNs and the Two-Class NN showed similar results while the Multi-class Decision Family scored the worst results for all cases. It can be concluded that the best parameters to use to predict the machine failure time in weeks are the normal case parameters using Multi-class NNs algorithm.

4.2. DES Results

After determining the best ML algorithm and dataset case to predict the machine’s RUL and failure time, the PdM time slots will be incorporated in the schedule DES simulation. FLEXSIM is the simulation software tool used in this case study. Our case study is a pharmaceutical Factory X, Figure 14 shows the layout of the shopfloor and Table 1 contains the inputs for the production scheduling problems.

Table 1. Production Line specifications.

No. of Workstations	No. of Machines	No. of Products	Production Line Nature
7	18	25	Fully Automatic

The nature of the products on this line are tablets and capsules. The raw materials of the production line are semi-finished powders for tablets and capsules. The tablets production processes are kneading, drying, blending, compression, coating and packaging. For capsules, the processes are encapsulation, coating and packaging. The following assumptions are set in the simulation:

- The priority of the products can be set
- The path of each product through the machines is defined
- The working hours of the factory are 462 h per month

- In the fourth workstation, some products cannot be processed on certain machines and each product has different processing times on each machine.
- Products can be processed on two or more machines within the same workstation and backtracking is possible.
- Setup times differ from one product to another.

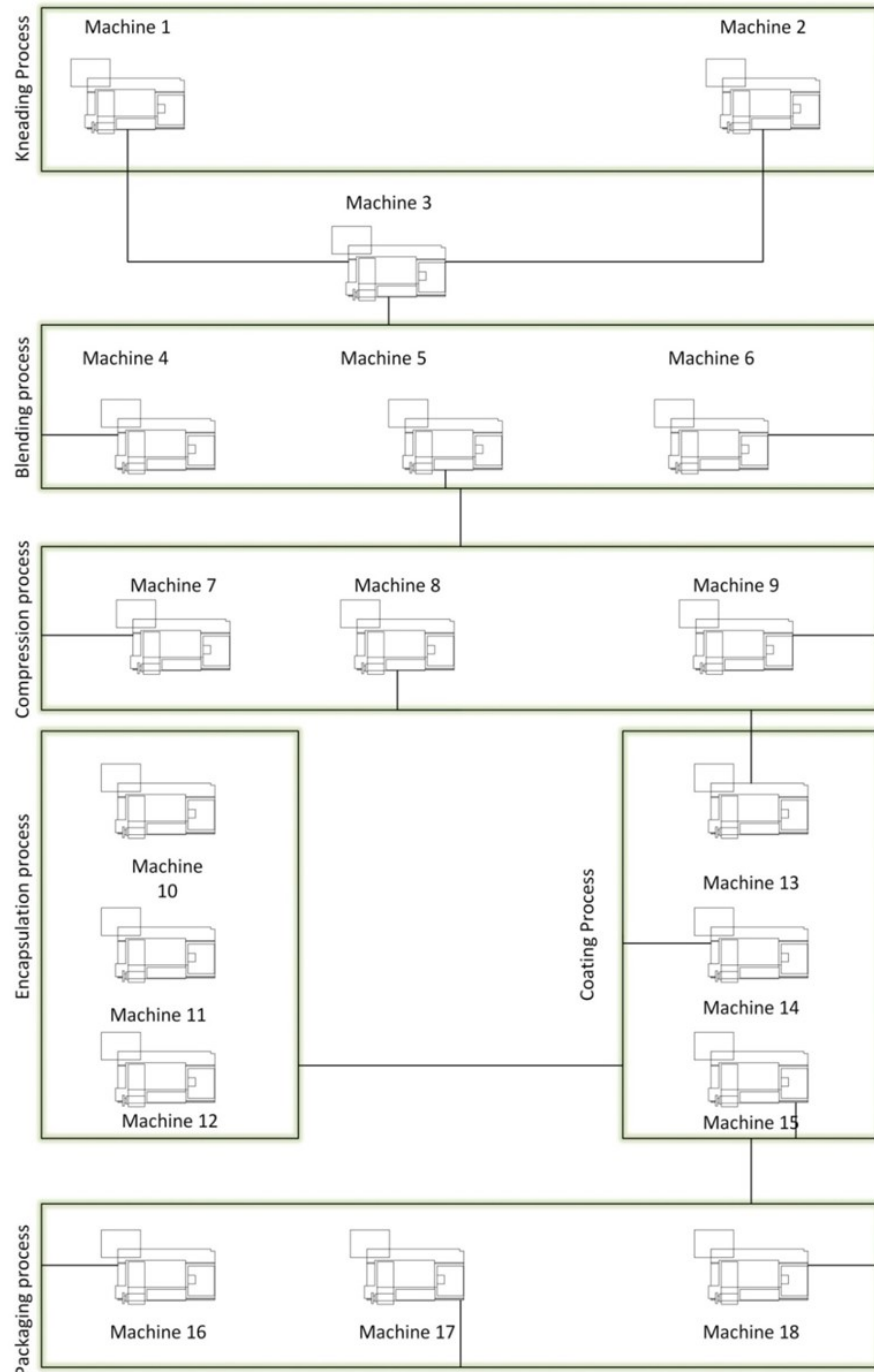


Figure 14. Shop floor layout for Factory X.

The output of the DES simulation is a Gantt chart showing the routing of the products within the production line. The simulation is made multiple times for different dispatching rules EDD, SPT and LPT. The Gantt charts showing the products on the y-axis and the time at which these products are produced on the x-axis is shown in Figure 15. For each product, the sequence of machines for which the product passed on is shown. For example,

for product 1, it was first processed on machine 1, then machine 8 and finally machine 16. The software also produces a state Gantt chart showing the used machines. The simulation model can produce such chart after using the predicted data from the ML algorithms. The state Gantt chart shows the machines used in the production line on the y-axis and the time at which they were idle, operating or in maintenance on the x-axis. Figure 16 shows the maintenance slots which were added to the simulation model marked with a red box and the machine breakdown in black. These slots are added by the ML algorithm, hence producing a complete schedule showing the sequence of products onto the machines and the PdM slots. Figure 16 shows the state chart produced for the EDD model since it is the best performing dispatching rule.

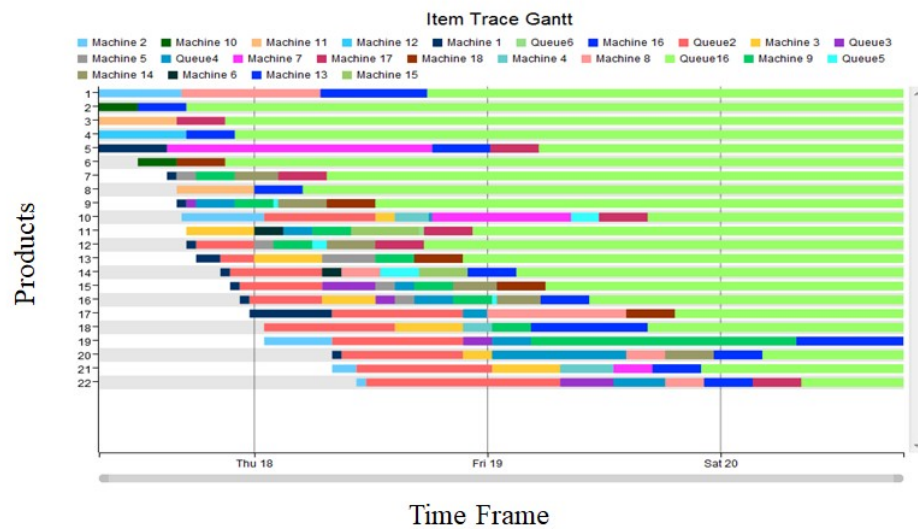


Figure 15. Gantt chart for production schedule using EDD as a dispatching rule.

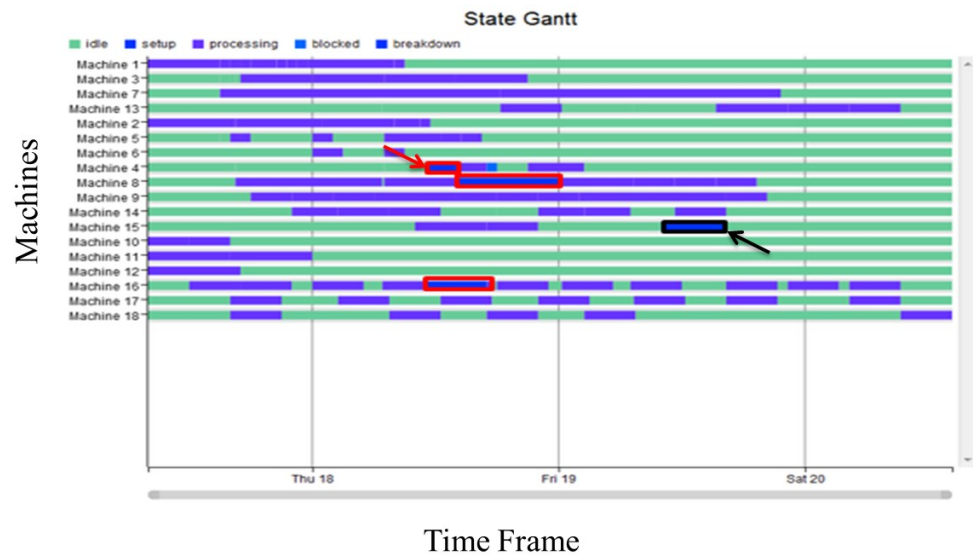


Figure 16. State chart for production schedule using EDD as a dispatching rule.

The proposed framework was used for makespan calculations for three dispatching rules: EDD, SPT and LPT while including the PdM maintenance time slots. ML algorithms was used to estimate the machine’s RUL and possible breakdowns. The EDD, SPT and LPT models produced makespan of 76.82, 104.92, and 116.42 h, respectively. The EDD model produced the minimum makespan. Hence, this case study verified the proposed methodology.

5. Discussion

DES is a commonly used approach in production scheduling to simulate various production plans and scheduling techniques with the aim of minimizing the makespan of production facilities. Using ML algorithms can enhance the simulated production plans by DES through predicting the PdM time slots and including them in the simulations of the production plans. Results testing various ML algorithms for this task have shown that Decision Forest algorithm is the best performing algorithm for the regression model used to estimate the machine RUL, while NNs algorithm showed better results in predicting the machine failure time both for the binary and multi-class classification model while using the parameters suggested as they are. Using the tested algorithms and feeding their data into the DES simulator, results testing various dispatching rules have shown that the best dispatching rule for minimizing makespan is EDD.

6. Conclusions

In this paper, a novel framework for flow-shop scheduling using DES and ML algorithms was introduced that integrates PdM into the production scheduling process. As a proof of concept, a pharmaceutical factory case study was presented to verify the proposed framework. The framework starts at the data acquisition level by integrating sensors to the machines on the production floor for collecting data indicating their vitals and statuses. These data are then aggregated on a gateway and stored on a cloud database which is updated continuously. After collecting a sufficiently large dataset, data are prepared to be fed into Microsoft Azure where various ML algorithms are tested with the aim of predicting machines' RUL which was best predicted by decision forest algorithm; and predicting machine failure times which was predicted with highest accuracy using NN algorithm. After calculating the PdM time slots, the proposed framework takes them into account in the DES scheduling process while minimizing the makespan. Along with selecting the right dispatching rule which have shown in the case study to be EDD, the proposed framework is able to enhance the efficiency and throughput of the production floor by predicting failures and preventing disruptions in the production process.

Author Contributions: Conceptualization, E.A. and L.A.S.; methodology, M.M.; software, M.N.; validation, M.N., E.A. and L.A.S.; formal analysis, M.N.; investigation, M.N.; resources, M.N.; data curation, M.M.; writing—original draft preparation, E.A.; writing—review and editing, M.M.; visualization, M.N.; supervision, L.A.S.; project administration, E.A. and M.M.; funding acquisition, E.A., M.M. and L.A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Egyptian funding agency ITAC grant number CFP-177.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
ANN	Artificial Neural Network
CM	Corrective Maintenance
DAQ	Data Acquisition
DES	Discrete Event Simulation
DT	Digital Twin
EDD	Earliest Due Date
GA	Genetic Algorithm
ICT	Info & Communication Technology

IloT	Industrial Internet of Things
LPT	Longest Processing Time
MAE	Mean Absolute Error
MIP	Mixed Integer Programming
MILP	Mixed Integer Linear Programming
ML	Machine Learning
NN	Neural Network
PdM	Predictive Maintenance
PvM	Preventive Maintenance
R2F	Run-to-Failure
RL	Reinforcement Learning
RM	Reactive Maintenance
RMSE	Root Mean Square Error
RUL	Remaining Useful Life
SPT	Shortest Processing Time
TBM	Time-Based Maintenance
TDGA	Two-phase Decoding Genetic Algorithm

References

1. Ying, K.; Lu, C.; Lin, S. Improved Exact Methods for Solving No-Wait Flowshop Scheduling Problems With Due Date Constraints. *IEEE Access* **2018**, *6*, 30702–30713. [\[CrossRef\]](#)
2. Fanjul-Peyro, L.; Ruiz, R.; Perea, F. Reformulations and an exact algorithm for unrelated parallel machine scheduling problems with setup times. *Comput. Oper. Res.* **2019**, *101*, 173–182. [\[CrossRef\]](#)
3. Martínez, K. P.; Adulyasak, Y.; Jans, R.; Morabito, R.; Toso, E. An exact optimization approach for an integrated process configuration, lot-sizing, and scheduling problem. *Comput. Oper. Res.* **2019**, *103*, 310–323. [\[CrossRef\]](#)
4. Wu, L.; Wang, S. Exact and heuristic methods to solve the parallel machine scheduling problem with multi-processor tasks. *Int. J. Prod. Econ.* **2018**, *201*, 26–40. [\[CrossRef\]](#)
5. Said, N.A.; Samaha, Y.; Azab, E.; Shihata, L.A.; Mashaly, M. An Online Reinforcement Learning Approach for Solving the Dynamic Flexible Job-Shop Scheduling Problem for Multiple Products and Constraints. In Proceedings of the International Conference on Computational Science and Computational Intelligence, Las Vegas, NV, USA, December 2021; Not Officially Started Yet.
6. Hong, T.Y.; Chen, C.F.; Wang, H.K.; Guo, H.Z. A two-phase decoding genetic algorithm for TFT-LCD array photolithography stage scheduling problem with constrained waiting time. *Comput. Ind. Eng.* **2018**, *125*, 200–211. [\[CrossRef\]](#)
7. Jankauskas, K.; Papageorgiou, L.; Farid, S. Fast genetic algorithm approaches to solving discrete-time mixed integer linear programming problems of capacity planning and scheduling of biopharmaceutical manufacture. *Comput. Chem. Eng.* **2019**, *121*, 212–223. [\[CrossRef\]](#)
8. Yu, C.; Semeraro, Q.; Matta, A. A genetic algorithm for the hybrid flow shop scheduling with unrelated machines and machine eligibility. *Comput. Chem. Eng.* **2018**, *100*, 211–229. [\[CrossRef\]](#)
9. Abdel-Basset, M.; Mohamed, R.; Abouhawwash, M.; Chakraborty, R.; Ryan, M. A Simple and Effective Approach for Tackling the Permutation Flow Shop Scheduling Problem. *Mathematics* **2021**, *9*, 270. [\[CrossRef\]](#)
10. Ponsignon, T.; Mönch, L. Simulation-based Performance Assessment of Master Planning Approaches in Semiconductor Manufacturing. *Omega* **2014**, *46*, 21–35. [\[CrossRef\]](#)
11. Wena, H.; Houb, S.; Liua, Z.; Liua, Y. An optimization algorithm for integrated remanufacturing production planning and scheduling system. *Chaos Solitons Fractals* **2017**, *105*, 69–76. [\[CrossRef\]](#)
12. Azadeh, A.; Negahban, A.; Moghaddam, M. A hybrid computer simulation-artificial neural network algorithm for optimisation of dispatching rule selection in stochastic job shop scheduling problems. *Int. J. Prod. Res.* **2012**, *50*, 551–566. [\[CrossRef\]](#)
13. Ojstersek, R.; Lalic, D.; Buchmeister, B. A new method for mathematical and simulation modelling interactivity: A case study in flexible job shop scheduling. *Adv. Prod. Eng. Manag.* **2019**, *14*, 435–448. [\[CrossRef\]](#)
14. Balog, M.; Dupláková, D.; Szilagy, E.; Mindas, M.; Knapčíková, L. Optimization of Time Structures in Manufacturing Management by using Scheduling Software Lekin. *TEM J.* **2016**, *5*, 319.
15. Dupláková, D.; Torokova, M.; Duplák, J.; Torok, J.; Hatala, M.; Steranka, J.; Radchenko, S. Determination of Optimal Production Process Using Scheduling and Simulation Software. *Int. J. Simul. Model.* **2018**, *17*, 609–622. [\[CrossRef\]](#)
16. Carvalho, T. P.; Fabrizzio, S.; Vita, R.; Francisco, R.; Basto, J.; Alcalá, S. A systematic literature review of machine learning methods applied to predictive maintenance. *Comput. Ind. Eng.* **2019**, *137*, 106024. [\[CrossRef\]](#)
17. Ran, Y.; Zhou, X.; Lin, P.; Wen, Y.; Deng, R. A Survey of Predictive Maintenance: Systems, Purposes and Approaches. *arXiv* **2019**, arXiv:1912.07383
18. Pan, E.; Liao, W.; Xi, L. A joint model of production scheduling and predictive maintenance for minimizing job tardiness. *Int. J. Adv. Manuf. Technol.* **2011**, *60*, 1049–1061. [\[CrossRef\]](#)

19. Ghaleb, M.; Taghipour, S.; Zolfagharinia, H. Real-time integrated production-scheduling and maintenance-planning in a flexible job shop with machine deterioration and condition-based maintenance. *J. Manuf. Syst.* **2021**, *61*, 423–449. [[CrossRef](#)]
20. Nikolaos, N.; Papavasileiou, A.; Dimoulas, K.; Bourmpouchakis, K.; Makris, S. On a versatile scheduling concept of maintenance activities for increased availability of production resources. *Procedia CIRP* **2018**, *78*, 172–177. [[CrossRef](#)]
21. Bampoula, X.; Siaterlis, G.; Nikolakis, N.; Alexopoulos, K.A. Deep Learning Model for Predictive Maintenance in Cyber-Physical Production Systems Using LSTM Autoencoders. *Sensors* **2021**, *21*, 972. [[CrossRef](#)]
22. Padovano, A.; Longo, F.; Nicoletti, L.; Gazzaneo, L.; Chiurco, A.; Talarico, S. A prescriptive maintenance system for intelligent production planning and control in a smart cyber-physical production line. *Procedia CIRP* **2021**, *104*, 1819–1824. [[CrossRef](#)]
23. Robinson, S.; Bhatia, V. Secrets of successful simulation projects. In Proceedings of the Winter Simulation Conference Proceedings, Arlington, VA, USA, 3–6 December 1995; pp. 61–67.