INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA DE INTELIGÊNCIA ARTIFICIAL




Otimização Multiobjetivo de Produção e Manutenção para uma Manufatura Eficaz e Participação em Programas de Resposta da Demanda

BRUNO ALEXANDRE SANTOS MOTA
Outubro de 2022

P.PORTO

Joint Optimization of Production and Maintenance for Effective Manufacturing and Demand Response Participation

Bruno Alexandre Santos Mota

Student nº: 1171198

Dissertation for obtaining the Master's Degree in Artificial Intelligence Engineering

Supervisor: Prof. Carlos Fernando da Silva Ramos

Co-supervisor: Dr. Pedro Nuno da Silva Faria

Jury:

President:

Dr. Maria Goreti Carvalho Marreiros, Coordinator Professor with Habilitation of Instituto Superior de Engenharia do Instituto Politécnico do Porto

Other members:

Dr. Paulo António da Silva Ávila, Coordinator Professor of Instituto Superior de Engenharia do Instituto Politécnico do Porto

Dr. Carlos Fernando da Silva Ramos, Full Professor of Instituto Superior de Engenharia do Instituto Politécnico do Porto

Porto, October 2022

Dedication

I would like to dedicate this dissertation to my parents, who have supported me during every hardship and ensured that I can achieve my academic and professional goals. It was through their struggles and efforts that I had the opportunity to come this far.

Resumo

Os elevados preços energéticos, principalmente nos dias de hoje, as pressões ambientais devido às mudanças climáticas, a competição elevada que leva muitas vezes as empresas a aceitarem prazos de entrega apertados ou até impossíveis, e manutenções ineficientes que recorrem a inspeções constantes e desnecessárias são os principais problemas que atormentam o setor de manufatura. Para ultrapassar estes problemas, é proposto um sistema inteligente de escalonamento de produção e atividades de manutenção, para um ambiente de manufatura em *flexible job shop*, que tem como objetivo a minimização dos custos energéticos e da deterioração do estado das máquinas. Além disso, este considera preços dinâmicos do mercado energético, a utilização de energias renováveis, a venda de energia gerada localmente em excesso, a participação em programas de *demand response*, atividades de manutenção, avarias de máquinas inesperadas, e restrições impostas sobre o plano de produção. Interligado ao sistema de escalonamento, via HTTP, encontra-se um sistema de manutenção preditiva para prever e detetar avarias de máquina antes que estas ocorram. Caso seja previsto ou detetada uma avaria de máquina, dependendo do tempo que tem até que a máquina avarie, é feito o escalonamento de atividades de manutenção ou o reescalonamento do plano de produção não tendo em conta a máquina avariada.

O sistema de escalonamento de produção e atividades de manutenção pode ser dividido em três componentes principais: o processamento de dados, caracterizado pelo balanceamento das células *job shop*; um algoritmo genético para planeamento, representado pela população inicial, cruzamento, mutação e seleção; e, finalmente, duas otimizações determinísticas que procuram diminuir ainda mais os custos e espaçamento entre tarefas sobre o resultado obtido pelo algoritmo genético. O algoritmo genético numa fase inicial faz o balanceamento entre células da linha de produção para equilibrar a energia dos produtos a manufaturar pelas diversas células. De seguida, para cada célula, é executado o algoritmo genético, destaca-se: métodos que tentam reparar o indivíduo, em termos de restrições, na criação da população inicial; um novo tipo de cruzamento mais adequado para problemas que incluem restrições, o qual combina elementos determinísticos e não-determinísticos de outros tipos de cruzamentos; na mutação é feita a troca entre tarefas e/ou troca do modo de tarefa (uma tarefa pode ter diversos perfis energéticos); e na seleção é feita uma aproximação híbrida, primeiro com a escolha dos n melhores indivíduos, sendo n definido pelo utilizador, e os restantes obtidos a partir de torneios não-elitistas. Após o algoritmo genético, também para cada célula, são executadas duas otimizações determinísticas que visam, se conseguirem: reduzir ainda mais os custos totais (custos energéticos e de manutenção), através da reafetação das tarefas para períodos livres com mais energia gerada ou menores preços energéticos; e reduzir o espaçamento vazio entre tarefas, ou seja, juntar tarefas, para a acomodação de futuros produtos ou manutenções que possam não estar planeadas. Além disso, também é proposto um sistema de reescalonamento de produção e atividades de manutenção, utilizando o mesmo algoritmo genético, para a participação em programas de *demand* *response* e avarias inesperadas de máquinas. Este sistema funciona através do reescalonamento de um plano criado anteriormente pelo escalonador.

A manutenção preditiva é feita a partir de uma rede neuronal artificial, que tem como objetivo prever se uma máquina falhou ou não. O seu processo de treino poder ser feito em batches, mini-batches ou através de um fluxo contínuo de dados. O processo de treino da rede neuronal começa com a obtenção dos dados mais recentes da máquina (temperatura do ar e do processo da máquina, velocidade de rotação, torque, desgaste da ferramenta e estado de falha da máquina) da base de dados da máquina instalada na fábrica. De seguida, antes do início do treino, é feita uma fase de pré-processamento dos dados em que: (1) é feita a agregação de todos os dados obtidos num único ficheiro (agregação de dados); (2) é feita a normalização das escalas e tipos de dados, através de uma estratégia Min-Max (normalização de dados); (3) é feito o preenchimento dos valores em falta nos dados obtidos, a partir de um método de imputação com o k-nearest neighbors (imputação de dados); (4) são removidos possíveis dados irrelevantes ou errôneos, através da deteção de outliers usando a técnica Z-score (filtragem de dados); (5) é feita a transformação de dados brutos em características que melhor representam o problema em questão (engenharia de dados); e, finalmente, (6) é feito o balanceamento das amostras de falha e não falha dos dados das máquinas, (balanceamento de dados). Depois, os dados pré-processados são enviados à rede neuronal para treino. Se o modelo já tinha sido treinado, então os pesos dos neurónios do modelo são ajustados, de acordo com os novos dados (retropropagação). É de realçar a implementação de um optimizador de hiperparâmetros automático, o qual procura os melhores valores para cada hiperparâmetro num modelo de aprendizagem automático. O processo de treino pode ser iniciado sempre que houver novos dados na base de dados das máquinas. Relativamente à aplicação em tempo real, esta funciona através de uma API REST, para a previsão/deteção de falhas numa máquina e para o ajuste dos pesos dos neurónios da rede (retreino).

Dois casos de estudo são usados para validar a solução proposta, um que representa a aplicação da solução num ambiente fabril e outro para o deslocamento de cargas numa residência. Ambos os casos de estudo utilizam dados reais. Estes destacam a robustez da metodologia proposta em reduzir a sobrecarga de tarefas em máquinas individuais, bem como a redução dos custos, através da utilização inteligente de energias geradas localmente para cobrir os gastos energéticos ou para vender a terceiros, reduzindo assim, a necessidade para recorrer a fornecedores externos. Os casos de estudo, também demonstram a integração efetiva da manutenção preditiva com a otimização de atividades de manutenção e a alta adaptabilidade do escalonador proposto para outras aplicações. Finalmente, o sistema de manutenção preditiva é comparado quantitativamente em termos de desempenho a outros sistemas de aprendizagem automática, entre os quais *random forest*, *gradient boosting* e *support vector machines*.

Palavras-chave: algoritmo genético, avaria de máquina, *demand response*, escalonamento da produção e manutenção, manutenção preditiva, rede neuronal artificial

Abstract

Production line management for cost-effective and machine longevity manufacturing, which considers production and maintenance activities is a key aspect in dealing with the ever-increasing electricity prices, rigid time commitments, maintenance costs, and environmental pressures that many manufacturing companies face nowadays. This dissertation addresses these issues by proposing a novel production line management system for joint optimization of production and maintenance for overall cost minimization and machine longevity improvement. To achieve this, it is proposed a Genetic Algorithm (GA) for production and maintenance scheduling, and an Artificial Neural Network (ANN) for predictive maintenance. The proposed GA takes into account Renewable Energy Resources (RERs), dynamic pricing, energy selling, maintenance activities, and constraints imposed on the production plan. Furthermore, it is also proposed a rescheduling system, employing the same GA, for demand response participation and unexpected machine breakdowns. Two case studies using real-production and residential data are presented. They highlight the robustness of the proposed methodology in reducing overload of single machines and utilizing RERs to cover energy expenses or for selling excess energy, reducing the demand for external suppliers. They also demonstrate the effective integration of predictive maintenance with maintenance optimization and the high adaptability of the proposed scheduler for other applications.

Keywords: artificial neural network, demand response, genetic algorithm, machine breakdown, predictive maintenance, production and maintenance scheduling

Acknowledgments

First and foremost, I would like to thank the research grant provided to me by GECAD and the experience gained by participating in the international project MUWO. This allowed me to deepen my knowledge in an area in which I intend to progress my studies, and for that, I will be eternally grateful.

To my family, a big thank you for the motivation given for the accomplishment of this dissertation, as well as for your patience and unconditional support during dark times. I am extremely grateful to my parents, José and Cristina.

I express my sincere gratitude to professor Carlos Ramos and professor Zita Vale, for all their understanding and support, sharing of knowledge, and valuable contributions to the dissertation. Above all, thank you for accompanying me on this journey and for inspiring my passion for knowledge and academia.

I thankfully acknowledge the support and inspiration that I received from my co-workers, Pedro Faria, José Araújo, and Luís Gomes for all the cooperation, understanding and availability they showed throughout this project. I especially thank Pedro Faria, which has my deepest gratitude for helping and encouraging me to go further.

Finally, I would like to offer my eternal gratitude to my grandmother Albina, who, although no longer with us, continues to inspire me to continue my studies and achieve a doctoral degree.

Index

Figure List

Table List

Acronyms and Symbols

Acronyms List

AI	Artificial Intelligence
ANN	Artificial Neural Network
BAU	Business as Usual
DR	Demand Response
DRD	Demand Response Dataset
GA	Genetic Algorithm
GB	Gradient Boosting
GDPR	General Data Protection Regulation
IDE	Integrated Development Environment
IoT	Internet of Things
MB	Machine Breakdown
ML	Machine Learning
NSGA-II	Non-Dominated Sorting Genetic Algorithm-2
PdM	Predictive Maintenance
PdMD	Predictive Maintenance Dataset
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
PV	Photovoltaic
RER	Renewable Energy Resource
RF	Random Forest
SD	Scheduling Dataset
SVM	Support Vector Machine
URL	Uniform Resource Locator

Symbols List

$\boldsymbol{E_{Buying\ Price(p)}}$	Price for buying energy in period p
$\boldsymbol{E_{Demand(p,m)}}$	Energy consumption of a machine m in period p
$\boldsymbol{E_{Generation(p)}}$	Available locally generated energy in period p
$\boldsymbol{E_{Selling\ Price(p)}}$	Price for selling energy in period p
$\boldsymbol{fit1}$	Fitness of individual 1
$\boldsymbol{fit2}$	Fitness of individual 2
$\boldsymbol{FS}$	Fitness score of an individual
$\boldsymbol{Individual^{1}_{chance}}$	Chance of individual 1 winning the non-elite tournament
$\boldsymbol{m}$	Machine index
$\boldsymbol{M}$	Total number of available machines for production
$\boldsymbol{M_{In\ Hours\ Price(p)}}$	Price of a maintenance activity done in maintenance hours in period p

$\boldsymbol{M_{Out\ Hours\ Price(p)}}$	Price of a maintenance activity done out of maintenance hours in period p
$\boldsymbol{MDC_{Factor(p,m)}}$	Classification of a factor’s contribution to the degradation of a machine m in period p
$\boldsymbol{MOR_{Factors(m)}}$	Occupation rate of factors that contribute to the degradation in machine m
$\boldsymbol{OSD}$	Occupation standard deviation of an individual
$\boldsymbol{OSD_{Norm}}$	Normalized occupation standard deviation of an individual
$\boldsymbol{p}$	Specific period
$\boldsymbol{P}$	Total number of available periods in the time window of the schedule
$\boldsymbol{P_{Machine(p,m)}}$	Priority of machine m in period p
$\boldsymbol{PEC_{Demand(p)}}$	Total energy consumed by the tasks in period p
$\boldsymbol{PEP_{Demand(p)}}$	Total energy to pay in period p
$\boldsymbol{PMP_{Maintenance(p)}}$	Maintenance costs to pay in period p
$\boldsymbol{TC}$	Total cost of an individual
$\boldsymbol{TC_{Norm}}$	Normalized total cost of an individual
$\boldsymbol{W_{OSD}}$	Optimization weight of the occupation standard deviation in an individual
$\boldsymbol{W_{TC}}$	Optimization weight of the total cost in an individual

1 Introduction

This dissertation describes the project carried out by the author in the scope of the Master's degree in Artificial Intelligence Engineering (MEIA) at the Polytechnic of Porto - School of Engineering (ISEP), in Porto, Portugal.

1.1 Contextualization

This dissertation's project was proposed by the Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, also known as GECAD, located in the Instituto of Engineering of the Polytechnic of Porto (ISEP) in Porto, Portugal. GECAD has the mission of carrying out and promoting scientific research for the development and evolution of intelligent systems in engineering and other fields. Currently, GECAD develops projects in the areas of Smart Grids and Electricity Markets, Internet of Things (IoT), Cyberphysical Systems, Smart Cities, Optimization, Knowledge Management, and Decision Support Systems [1]. The proposed project implements and explores a system capable of scheduling production lines, while also taking into account energy costs, renewable energy usage, energy selling, Demand Response (DR) participation, maintenance activities, Machine Breakdown (MB) events, ensuring machine longevity, and complying with constraints imposed on the production plan. The optimization of energy and maintenance costs with Artificial Intelligence (AI) in the manufacturing field are areas increasingly sought after, due to their importance to the development of production lines more effective and efficient.

1.1.1 Topic Relevancy

Europe's rising electricity prices are hitting new records, causing inflation and increasing expenses for millions of homes and industries throughout the continent. It is estimated that energy prices in 2022 will surge up to 14% in France and 15% in Germany, leaving many experts uneasy about the future [2]. Due to this crisis, many member states that have limited electricity interconnection capacity have seen astronomical electricity market prices increase. As a result, many industries, mainly in the manufacturing field, are facing problems regarding competitiveness, because of the high energy costs [3]. However, member states that had a high share of Renewable Energy Resources (RER)s have seen fewer spikes in electricity prices, primarily during the COVID-19 pandemic [4]. RERs have an important impact on energy grid stability, while also providing environmental advantages in the face of climate change and global warming [5]. RERs contributed for 25% of 2018's power generation and are predicted to account for 40% by 2040, according to [6]. Manufacturing companies can participate in minimizing their energy consumption through intelligent production scheduling systems capable of reducing costs while also maximizing RERs usage [7],[8]. In addition, participation in DR programs can also be a great way to further minimize monetary costs [9]. These programs provide incentives for manufacturers when there are changes in energy demand, either through adjusting energy prices or by paying companies who participate in DR events [10].

Nevertheless, high electricity prices and low RERs usage are not the only problems plaguing the manufacturing field. Maintenance scheduling is also one of the most prevalent and important problems confronting the manufacturing field [11]. According to [12], it is estimated that maintenance cost is between 15% and 70% of the cost of products sold. Between production and maintenance scheduling, there are trade-offs and conflicts to consider, since delaying maintenance for production may raise the likelihood of machine failure [13]. Machine maintenance can be classified into preventive maintenance, which is a schedule of planned maintenance procedures designed to prevent system malfunctions and failures, and corrective maintenance, that is, the replacement or repair of faulty or broken down components in machines [14],[15]. One form of preventive maintenance is the use of intelligent production scheduling systems that aim to optimize machine longevity, through the balancing of tasks by taking into account machines' failure-rate probability [16]. Another form of preventive maintenance is the usage of Machine Learning (ML) to anticipate machine failures ahead of time, allowing for a better decision-making process, helping to avoid downtime [17]. This later form, also known as Predictive Maintenance (PdM), is a historic data-based model that uses statistical or ML models to predict behavior patterns, correlations, and trends [18],[19].

1.1.2 MUWO Project

The present dissertation is also associated with an international project from ITEA4 Programme [20], designated as project MUWO (Multi-method workspace for highly scalable production lines) [21], [22]. This project aims to "create an opportunity to use production systems more effectively through flexible scaling. Scalability is achieved by the development of smart hardware interfaces. This will allow workstations to advance to multi-method workstations that support both manual and automated processes. Additionally, workstations can combine different processes. A transmutable simulation validates the workstation configuration and a process combiner optimizes the production configuration using AI/ML methods. Through this, MUWO improves the design and operation of production systems" [21]. The project counts with nine partners from three different countries, Portugal, Spain, and Turkey. This dissertation partnership with the MUWO project was only possible due to GECAD's high international recognition for its research excellence.

Also noteworthy is that the MUWO project has some continuity from a previous international project, in which the dissertation's author also participated, the SPEAR project [23], funded by the European Union and also from the ITEA4 Programme [24].

1.1.3 Motivation

One of the biggest motivators is the desire to develop a system capable of delivering fast solutions and able to adapt to unexpected situations in a manufacturing environment. It would be not only interesting for the research community, but also for the manufacturing field, one of the sectors to be growing exponentially mainly during the COVID-19 pandemic where orders skyrocketed and innovation accelerated [25].

Another big incentive that led to the acceptance of this project was the desire to further explore and acquire knowledge of the ever-growing manufacturing field. Moreover, there is an enormous amount of applications in the real world in the energy and manufacturing fields, which can be used to improve the quality of life of millions of people.

Being GECAD a research group of excellence, it also facilitates the ambition of the author to progress in my studies for a doctoral degree.

1.2 Problem Statement

This dissertation's purpose is to solve some of the problems that many manufacturing companies face when dealing with uncertainties in scheduling production goods, by using AI techniques to develop a robust and efficient system. The unexpected problems that need to be taken into consideration are DR programs, volatile energy prices, efficient RERs usage, energy selling, MB events, and preventive as well as predictive machine maintenance. Consequently, the proposed system needs to be capable of adapting to unexpected situations that might occur at any minute, thus the ability to reschedule an already scheduled production plan is mandatory.

Furthermore, the system should take into account constraints that might be imposed on the production plan, for example, machine priority, product deadlines, task setups, task order, and collision between tasks. Additionally, production lines follow a flexible job shop configuration, and excess remaining RERs from scheduling could be sold to other entities.

The system through preventive machine maintenance, which efficiently schedules products to balance machine tasks and thus reduce overload and usage of single machines, should increase machines' lifespan. Nevertheless, the system also needs to predict eventual machine failures (i.e., MB events), in order to not compromise the production plans' execution.

In brief, the proposed project tackles two crucial problems that hunt the manufacturing field:

- The capacity to efficiently reduce energy costs and increase sustainability through RERs usage and participation in DR programs;
- Increase machine longevity and reduce unexpected MBs events during working hours.

From the project, the following tasks are expected: solution design and implementation, tests and demonstrations, and system documentation.

The use of real production data, provided by a real company, for tests and demonstrations is crucial to validate the proposed methodology in a real-world scenario.

1.2.1 Aims and Objectives

To achieve a system capable of overcoming unexpected scenarios in a manufacturing environment regarding the scheduling of production lines, the following methodology represented in Figure 1 can be employed, which is divided into three fundamental components:

- Scheduler/Rescheduler, through the usage of a Genetic Algorithm (GA);
- Demand Response Participation, by also utilizing the same GA;
- Predictive Maintenance, achieved by using an Artificial Neural Network (ANN) to predict a machine's failure status.

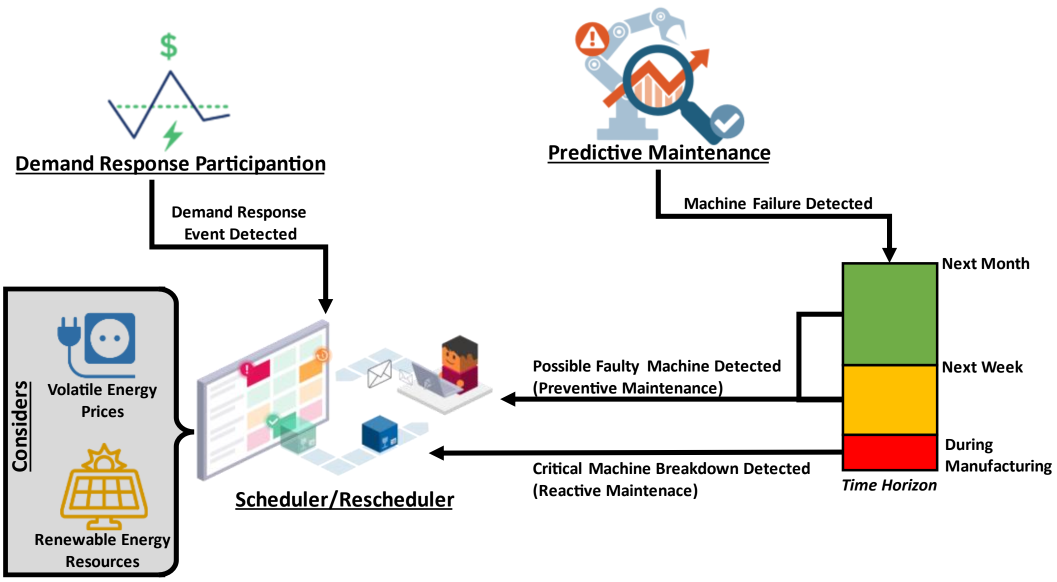

Figure 1 – Proposed methodology for production and maintenance scheduling optimization

1.2.1.1 Scheduler/Rescheduler and Demand Response Participation

A production and maintenance scheduler system is proposed to achieve an energy cost-effective and preventive maintenance optimization, capable of taking into account a wide variety of constraints, such as machine priority, product deadlines, task setups, task order, task collision, etc. This is done through intelligent production scheduling using a GA. To minimize energy costs the scheduler takes advantage of the volatile energy prices, RERs availability, DR programs, and energy buyers. To reduce machine degradation (i.e., preventive maintenance), tasks are balanced in order to reduce overload in single machines.

Furthermore, in this component, it is crucial the ability to reschedule production plans so that the system is prepared to adapt, in real-time, to DR programs and MBs events (i.e., reactive maintenance). It is worth noting that DR events include sudden changes in energy prices and energy consumption restrictions, which, if they are respected, can award the user with monetary incentives.

1.2.1.2 Predictive Maintenance

To predict whenever a machine is going to become faulty in some manner or even breakdown, an ANN is employed. It predicts machines' failure status by assuming a value of 0 or 1 for non-failure and failure, respectively. To achieve this, five variables are taken into account, obtained from sensors located in machines, these are: air and process temperature, rotational speed, torque, and tool wear. These variables can be obtained from third-party systems that provide forecasts of what these variables might be in a foreseeable time horizon.

From the obtained results of the PdM model, if one or more machines are detected to have a failure, the scheduler/reschedule is used to consider an MB event and/or to schedule maintenance activities. According to the failure status of machines and forecast time horizon, the production plan is rescheduled to consider these changes. On one hand, a reactive maintenance approach is followed if a failure is predicted or occurs during the manufacturing process (i.e., no third-party system was used to obtain forecasted variables), in which the affected machines are removed from the schedule (i.e., MB event). On the other hand, if a third-party system is used to get predicted machine data, and thus the prediction made by the PdM system is for a specified time horizon (e.g., next week, or next month), a scheduling of maintenance activities is done, in order to follow a preventive maintenance approach.

1.2.1.3 Other Objectives

Other key objectives for the success of the project are:

- Evolution of the system with new features – The specific objectives of the features to be implemented are ambiguous since they have an evolutionary characteristic, and thus depend on the need for them. Implementation of mandatory and important functionalities is prioritized in relation to other secondary ones;

- System documentation – Rich and extensive documentation of the system should be carried out in order to make its use and understanding easier and more accessible;

- Tests on the system – The system must undergo several tests in order to determine its effectiveness and efficiency. Reporting the results is essential to evaluate the system's progress as well as simulate scenarios that use real production and machine data.

1.2.2 Implementation Approach

The implementation approach adopted for this project is one of prototyping, which ensures the evolution of the system and is widely used for research. Due to the rising market response, design has gradually sought new methods to ensure greater success. Prototyping, both real and virtual, from manual to rapid, is an essential aspect of the product development process. The emergence of virtual designed goods ushered in a new era, allowing for more formal complexity in less time [26]. When users do not know the specific project requirements in advance the Prototyping model is employed. This model, consists in the creation, testing, and improvement of a system prototype, based on users' input until an acceptable final prototype is reached, which serves as the foundation for the development of the final system [27]. Prototyping advantages include high involvement of field experts in the design, early design visualization, flexibility for modifications, and fast requirement completion [28].

In addition, the implementation of the proposed system is done using the Python [29] programming language. The GA scheduler/rescheduler was developed without using any libraries related to GAs since no library is capable of solving the complexity of the problem presented in this dissertation. For the ANN model, it was used the Keras library [30] for its classifier implementation, the Scikit-learn [31] library to implement an automatic hyperparameter optimizer as well as data normalization and imputation methods, and the SciPy library [32] for a data filtration method which uses a Z-score function.

To validate the proposed scheduler/rescheduler real production data provided by a textile company was used. Some maintenance data was obtained from the paper [33]. For the PdM model, a public dataset was used instead, available in [34]. Two methodologies are adopted to validate the proposed system, a scenario-based analysis approach for the proposed scheduler/rescheduler (i.e., GA), and another more oriented for model performance comparison of the proposed PdM model (i.e., ANN).

1.2.3 Contributions

From the present project a variety of state-of-the-art innovations emerged that should be highlighted, of which:

- An innovative GA crossover approach, which allows getting more consistent individuals that respect imposed constraints in the schedule;
- A flexible GA scheduler/rescheduler that allows a wide variety of considerations to be applied at the same time (e.g., volatile energy prices, RER usage, energy selling, maintenance activities, constraints, DR programs, and MB events) while at the same time minimizing total costs and maximizing machine longevity;
- A novel ML training approach for PdM that improves model performance while also taking into account imbalanced and irrelevant/erroneous data;
- An automatic hyperparameter optimization strategy, used to determine the optimal hyperparameters for the ANN, hence enhancing the models' performance even further;
- A new methodology to integrate PdM for machines and production scheduling, ensuring reliability and reducing unexpected situations during the manufacturing process.

The present dissertation’s solution is directly associated with the international MUWO project [21], as part of an ITEA4 project [20], counting with nine partners from three different countries. Furthermore, the work developed within the scope of the dissertation provides GECAD with a new solution that responds to the needs of energy variations in the network and imposed restrictions presented in a manufacturing environment, as well as a new method to predict a machine failure status. Additionally, due to the AI techniques being explored, such as GAs and ANNs, new paths are open to research that aims at combining these techniques.

In addition, from the present dissertation several papers were published:

Production Line Optimization to Minimize Energy Cost and Participate in Demand Response Events [35] – This journal article highlights how the proposed scheduler is able to participate in DR events, through the rescheduling of a production plan. Also, it demonstrates how the scheduler is able to handle multiple constraints on a heavy load (i.e., a large number of tasks to schedule);

Scheduling of a Textile Production Line Integrating PV Generation using a Genetic Algorithm [36] – This journal article demonstrates the scheduler's capability in reducing energy costs by efficiently utilizing RERs, in this case, Photovoltaic (PV) energy, and electricity prices. Furthermore, it shows how the proposed scheduler adapts to dynamic and flat electricity tariffs;

Residential Load Shifting in Demand Response Events for Bill Reduction using a Genetic Algorithm [37] – This journal article validates the high adaptability of the proposed scheduler for other applications besides industrial production scheduling as it can also be used for residential load shifting. In addition, it demonstrates the scheduler’s ability to adapt to different levels of flexibility as well as a constraint that enables the user to choose when a load can start and finish (in the industrial context it can be used to indicate when stock becomes available, or a deadline for a specific task).

Predictive Maintenance for Maintenance-effective Manufacturing using Machine Learning Approaches [38] – This paper was presented at the 17th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2022), Salamanca, Spain [39]. The paper validates the proposed predictive maintenance system in dealing with unbalanced and irrelevant/erroneous data, continuous improvement of the models due to their ability of retraining, easy implementation of the models due to an automatic hyperparameter optimization approach, and the high flexibility to apply to other contexts since it requires a low number of common features from machine data. Furthermore, it explores the proposed predictive maintenance system when implemented with Gradient Boosting (GB) or Support Vector Machine (SVM) models.

Machine Learning applied to Industrial Machines for an Efficient Maintenance Strategy: a Predictive Maintenance Approach – This paper was presented at the 9^{th} International Conference on Energy and Environment Research (ICEER 2022), Porto, Portugal [40]. It will be published in the open-access journal Energy Reports, ScienceDirect. The paper validates the proposed predictive maintenance system in utilizing an automatic hyperparameter optimizer to further enhance a machine learning model's performance, and the application in real-time of implemented machine learning models by considering model retraining and user application. Moreover, it explores the proposed predictive maintenance system when implemented with Random Forest (RF) or ANN models.

1.3 Dissertation Structure

This dissertation is divided into six main sections. The first section presents an introductory segment that contextualizes the reader regarding the topic relevancy, author's motivations, and state the problem at hand. Section two addresses the current state-of-the-art techniques for production and maintenance scheduling, and PdM, as well as different methodologies for the combination of these components. An overview of the major decisions made for the development of the proposed methodology, such as adopted architecture, models, methods, technologies, and datasets, is made in section three. Then, in section four, the proposed solution implementation is shown in detail. Section five provides two case studies to validate the proposed solution and a discussion of the obtained results is done. The final section sums up the main conclusions of the dissertation, accomplishments, limitations, ethical questions, and future work.

2 State-of-the-art

A literature review in production line management techniques applied to AI models was performed to identify how can we deal with uncertainties (e.g., DR and MB events) when scheduling production goods in an industrial environment. This literature review was conducted following some elements of the (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) PRISMA 2020 [41] guidelines for reporting systematic reviews. This section presents the main research question as well as the submain questions formulated, the search sources, search queries and keywords, inclusion and exclusion criteria used, and the research selection process defined, in Sub-section 2.1. Then, in sub-section 2.2 the obtained results from the literature review are presented, and how these studies help answer the main research question. Finally, in sub-section 2.3 a discussion is made about the findings and how the proposed dissertation's solution compares to other works in the literature.

It is worth noting that the present literature review only represents a portion of a literature review made by the author, which also includes the topic of product quality, the subsequent joint optimization of production, maintenance, and product quality, the use of deep learning for scheduling, and a much higher quantity of literature references (e.g., reviews and papers).

2.1 Research Methodology

The present literature review answers the main research question: "What are the current state-of-the-art production line management techniques applied to AI models?". To achieve this, the literature review is divided into two submain questions, that explore different perspectives. The first submain question proposed is: "What are the best techniques for each production line management component (i.e., production/maintenance scheduling, and predictive maintenance) to be applied?" The cited question is important to answer the main research question since it focuses on finding the most efficient, reliable, and robust techniques in dealing with uncertainties in the manufacturing environment when managing production lines. The second submain question proposed is: "How can the different production line management components (i.e., production/maintenance scheduling, and predictive maintenance) be effectively combined/interact?". This submain question explores how the different components for production line management can be combined for effective production and maintenance scheduling, as well as how these components' interactions can further help optimize the solution obtained (e.g., PdM helps to more reliably and efficiently schedule maintenances). Regarding the present literature review search strategy it is defined into three crucial phases: the definition of the search sources and queries, as well as how the research selection process is done. The first phase consists of identifying and defining which search sources are going to be used in order to search and review the works in the literature for the literature review. In the present literature review, four electronic databases were considered, ArXiv, B-on, IEEE Xplore, and Science Direct, their respective Uniform Resource Locator (URL) is represented in Table 1, respectively. It is noteworthy that, even though B-on aggregates a wide variety of sources, including from ArXiv, IEEE Xplore, and Science Direct some works were still not indexed on B-on. Therefore, to maximize the quality and quantity of studies found, the databases from ArXiv, IEEE Xplore, and Science Direct were also explored, even if there was a large number of duplicate studies that later needed to be removed. Also, in some rare instances, some works were considered from a complementary index, Google Scholar, since they were not indexed on the above-cited databases.

Table 1 — Main electronic databases used in the literature review

Identifier	Database	URL
ED1	ArXiv	https://arxiv.org/
ED2	B-on	https://www.b-on.pt/
ED3	IEEE Xplore	https://ieeexplore.ieee.org/Xplore/home.jsp
ED4	Science Direct	https://www.sciencedirect.com/

In the second phase, the definition of the search query, which reflects the submain research questions, is essential to achieve the answers for the main research question. The main search query can be divided into five sub-queries that can represent the different submain research questions, these are presented in Table 2. It is worth noting that the terms present in the search queries are considered to be keywords of the present literature review. In addition, even though the terms “preventive maintenance”, “opportunistic maintenance”, and “reactive maintenance” are not included in the search queries since they are already included in the term “maintenance”, their concept as a whole is still very important to this field of research.

As a result, the main raw search query could be described as the following: “((("production") OR ("manufacturing") OR ("job shop") OR ("flowshop") OR ("load")) AND ("maintenance") AND (("scheduling") OR ("shifting") OR ("planning"))) AND (("demand response") OR ("DR") OR ("demand-side management") OR ("DSM") OR ("peak demand") OR ("dynamic price")) AND (("predictive maintenance") OR ("fault diagnosis") OR ("failure prognostics") OR ("remaining useful lifetime prediction")) AND (("artificial intelligence") OR ("AI") OR ("machine learning") OR ("ML"))”. The last phase focuses on the definition of the selection and refinement methodology to filter undesirable works in the literature as well as revise and choose the most relevant works that could answer the main research question. Regarding the exclusion of undesirable works, Table 3 and Table 4 show the inclusion and exclusion criteria of the present systematic literature review, respectively.

Table 2 — Search queries used in the literature review

Scope	Query
Production/Maintenance Scheduling	((Production OR Manufacturing OR Job Shop OR Flowshop OR Load) AND Maintenance AND (Scheduling OR Shifting OR Planning)) AND
Demand Response	(Demand Response OR DR OR Demand-Side Management OR DSM OR Peak Demand OR Dynamic Price) AND
Predictive Maintenance	(Predictive Maintenance OR Fault Diagnosis OR Failure Prognostics OR Remaining Useful Lifetime Prediction) AND
Artificial Intelligence	(Artificial Intelligence OR AI OR Machine Learning OR ML)

Studies that meet one or more inclusion criteria were chosen, while studies that infringed at least one exclusion criteria, were not considered. To save the chosen studies, a reference manager was used, namely Mendeley [20].

Table 3 — Inclusion criteria applied in the literature review

Identifier	Inclusion criteria
IC1	The source focuses on the use of production line management (i.e., production/maintenance scheduling, and predictive maintenance)
IC2	The source belongs to the field of artificial intelligence in computer science
IC3	The source is peer-reviewed, except in cases that the source shows a relevant contribution to the area
IC4	The source describes a significant contribution to the fields of study

Table 4 — Exclusion criteria applied in the literature review

Identifier	Exclusion criteria
EC1	The source is a duplicate
EC2	The source is published before 2017
EC3	The source is not written in English
EC4	The source is either a book chapter, dissertation, review, or thesis
EC5	The source does not present studies related to production line management (i.e., production/maintenance scheduling, and predictive maintenance)
EC6	The source focuses on a very specific domain, lacking adaptability from the used technique/model to different domains

After the definition of the selection methodology to filter undesirable studies, the study selection and data extraction phases, in order to choose the most relevant works, were done using some elements of PRISMA guidelines, available in [41].

2.2 Literature Review

In this section, the results obtained through the literature review are presented in four subsections. The first two subsections (i.e., sections 2.2.1, and 2.2.2) answer the first submain question, that is, “What are the best techniques for each production line management component (i.e., production/maintenance scheduling, and predictive maintenance) to be applied?”. On the other hand, section 2.2.3 answers the second submain question, “How can the different production line management components (i.e., production/maintenance scheduling, and predictive maintenance) be effectively combined/interact?”.

2.2.1 Production/Maintenance Scheduling

Production or maintenance scheduling is one of the most important challenges in the manufacturing field and has been extensively researched in the literature. However, a large part of these works are directed towards a more simplistic view of the problem, which can bring great advantages regarding system processing time, but they have low flexibility and are hard to apply in real environments. These works focus on the flow-shop concept, in which it is assumed that a product only goes through the same machine once and must follow a sequential order of machines, similar to a production line. On the other hand, one of the most famous problems in production/maintenance scheduling is the job shop, which is NP-hard. It can be characterized by allowing products to transfer between machines without any imposed sequential machine order, nevertheless, just like the flow-shop, products can only go through a machine once. However, the present dissertation's solution is based on an extension of the job-shop problem, which allows for greater flexibility but being much more complex and NP-hard, the flexible job shop problem. Unlike the classical job shop, in a flexible job shop there is a group of unrestricted machines available to the product, where there is no machine order imposed and a product can be processed in the same machine more than once [42]. Therefore, flexible job shop is a more complex problem to solve than the classical job shop problem.

It is also worth noting that, in the present work, DR is described as a change in energy consumption such that the energy demand compares to the energy that can be provided during that time. It can consider a series of stimuli, such as a change in the price of energy or the payment of incentives to manufacturers who engage in DR events [10]. A specific type of DR is load shifting, which can be described as shifting tasks (i.e., load) from peak demand periods to off-peak periods in order to reduce peak energy demand, thus influencing the load curve and reducing energy costs while also improving reliability [43].

2.2.1.1 Available Techniques

The use of computational algorithms for production line scheduling is essential for solving complex production scheduling problems that aim to minimize energy consumption and costs. As a result, there is a big room for improvement regarding possible energy and monetary savings, when taking into account DR participation, volatility of electricity prices, RERs usage, and energy selling. There is a wide range of methods that focus on optimizing the above-mentioned problems, such as the usage of reinforcement learning [44], [45], linear programming [46], [47], particle swarm optimization [48], [49], ant-lion optimization [50], [51], grasshopper optimization [52], [53], Non-Dominated Sorting Genetic Algorithm-2 (NSGA-II) [54], sub-gradient descent [55], grey wolf optimization [56], and improved whale optimization [57].

However, preventive and reactive maintenance in production scheduling is another topic that this dissertation explores. They are required in order to prevent equipment failures and guarantee product quality. Machine maintenance must be planned to not interfere with manufacturing but at the same time needs to be scheduled as soon as possible, because postponing the next maintenance on a machine could mean an unexpected breakdown during the next manufacturing hours. There are fewer works in the literature regarding this issue, nevertheless, some of the most used techniques are reinforcement learning [58], [59], particle swarm optimization [60], [61], simulated annealing [62], [63], linear programming [64], [65], ant colony optimization [66], artificial bee colony algorithm [67], tabu search [68], and NSGA-II [69].

2.2.1.2 Genetic Algorithm

One of the biggest drawbacks of the above-cited techniques, either for energy cost savings or preventive/reactive maintenance is their lack of ability to easily integrate and solve both of these complex problems. Therefore, a GA approach was chosen for the production scheduler for the present dissertation's solution. It was chosen because it is a robust metaheuristic search-based optimization algorithm that provides semi-consistent good solutions, and is thoroughly explored in the literature mainly solving traveling salesman analogous problems, such as the scheduling of tasks/products. Furthermore, it is ideal for unpredictable manufacturing environments, since it is remarkably adaptable regarding execution time, for instance, it could be executed for 5 hours, 1 hour, or even 10 minutes, with lower times having the disadvantage of providing worse results. Another advantage of GAs is their ability to avoid local optimum, which is a major problem in other search techniques [70], [71].

Genetic algorithms are evolutionary search-based optimization algorithms based on natural selection and genetic principles. They apply an organized and parallel yet random search strategy aimed at increasing the number of high fitness points. Even though they are random, they are not equal to random, untargeted searches, since they explore prior data to find new search points, where it is likely to have better results (i.e., higher fitness scores) [72]. GAs can be divided into two crucial phases, the creation of the initial population, and the iteration of the aforementioned population through various generations. Each generation is composed of three fundamental sequential subphases: (1) crossover, to spread the genes across the population; (2) mutation, which aims to maintain genetic diversity; and (3) selection, which inherits the next individuals for the next population [73], [74]. The GA was first proposed by John Holland in 1992 [75] and is based on the findings of Charles Darwin in his Theory of the Evolution of Species [76].

Regarding energy cost savings (i.e., intelligent use of volatile energy market prices, RERs, and DR participation) works in the literature that employ a GA, [77] proposes a demand-side management strategy that provides the optimal solution based on the optimization of load shaping in demand-side management, by using a GA. An energy management system that employs a GA is proposed in [78] that minimizes energy costs through load shifting, by responding to electricity price changes. Reference [79] utilizes a GA to obtain the optimal schedule that minimizes energy costs. It takes into account the usage of PV energy (i.e., RER), energy storage, and different electricity tariffs to achieve its objective. Also, in [80] a GA is proposed to schedule loads with the objective of reducing costs and peak-to-average ratio, as well as improving user comfort. It considers PV energy, real-time electricity prices, energy demand, and user preferences. In [81] it is proposed a multi-objective GA that reduces energy costs, makespan, and tardiness, in a flexible job shop environment. In addition, the proposed solution allows for the use of multiple types of constraints, such as cost constraints, resource constraints, and sequential constraints. Other noteworthy works are presented in [82]–[84], and reviews/surveys in [85]–[87].

Concerning preventive or reactive maintenance scheduling by using a GA there are fewer works in the literature when compared to production scheduling, nevertheless, there is still some research that stands out. For instance, in [88] it is proposed a maintenance scheduling system that uses a GA to schedule not only maintenance but also repairs and operations. In addition, it minimizes excessive human resources (i.e., unnecessary maintenances/repairs) and customer waiting time, while also improving the cost performance index of the resources. The work proposed in [89] focuses on a GA to schedule maintenance activities in a flexible flow-shop environment, in order to find the optimal schedule for processing tasks, reduce delays, and minimize equipment repair costs. Moreover, it also takes into account the uncertainties of the flexible flow-shop scheduling of waste-to-energy. A preventive maintenance scheduling system is proposed in [90] that employs a GA to achieve a maintenance schedule that minimizes the life-cycle cost rate of machines. It considers different failure modes, classifying them into two types: (1) degraded, which indicates that there is some degradation; (2) functional failure, which signals that there is an imminent failure. Reference [91] describes a proactive maintenance scheduling system for the optimal strategy to reduce maintenance costs, by using a GA. Furthermore, it considers imperfect repairs (i.e., repairs not well done or not enough) and postponed replacements (i.e., repair parts not available). In [92] it is proposed a bi-objective maintenance scheduling system to reduce maintenance costs and total tardiness. It achieves these results by using a GA to schedule as few maintenance activities as soon as possible in order to fulfill the bare minimum requirements, as well as, by scheduling tasks and organizing maintenance activities such that processing times are kept to a minimum and excessive maintenance time is avoided. Other noteworthy works are described in [93]–[95], and reviews/surveys in [96], [97].

It is also worth noting that GAs are also commonly used in other fields such as healthcare [98], education [99], smart buildings [100], network security [101], robotics [102], finance [103], agriculture [104], and much more.

2.2.2 Predictive Maintenance

Nowadays, due to the emergence of new and improved technologies such as in Big Data, IoT, data analytics, augmented reality, and cloud computing, there has been a shift in the industrial maintenance strategy to systems capable of predicting machine longevity [105], [106]. This has led mainly to the research of two new maintenance concepts to identify irregularities in the manufacturing environment, condition-based maintenance, and prognostic and health management [107]. Predictive maintenance, which uses prior data to predict behavior patterns, is often employed with these two concepts in mind, either with condition-based maintenance or prognostic and health management and in some cases, with both of them [108].

2.2.2.1 Available Techniques

In a manufacturing environment, the use of predictive systems to detect when maintenance activities are required is crucial not only to reduce unnecessary costs but also to improve product quality. Predictive maintenance allows for continuous monitoring of the machine's integrity, allowing maintenance to be conducted only when it's really necessary. Furthermore, prediction systems based on statistical inference methods, historical data, integrity variables, and engineering approaches enable the early detection of problems [109]. There are a variety of techniques for predicting systems health and/or condition, for instance, by using a RF [110], [111], Deep Learning [112], [113], SVM [114], [115], k-means

[116], Naive Bayes [117], GB [118], Decision Tree [119], k-nearest neighbors [120], Gaussian Process Regression [121], Generalized Linear Model [122], Fuzzy unordered rule induction algorithm [123], and Principal Component Analysis [124].

2.2.2.2 Artificial Neural Network

While the usage of a DNN might be tempting for PdM, the present dissertation's project focuses more on the integration between PdM and production scheduling. Therefore, there is no need for a complex preventive maintenance system based on deep learning, capable of classifying multiple types of problems, but instead, a simple system that has the ability to predict a machine's failure status. Furthermore, due to the simplistic nature of the problem at hand, a deep learning approach would probably lead to an overfit model, and thus an ANN was chosen instead. The use of an RF was also considered, because of its popularity in PdM, however, ANNs provide a much better performance, both in accuracy and context adaptation. In addition, contrarily to RFs, ANNs have the advantage of backpropagation (i.e., fine-tuning of the weights in the network based on the error rate), allowing for a current model to be constantly fed with data and improving over time, without the need to recreate the model every time there is new training data, which is ideal for a manufacturing environment [125], [126].

ANNs are a ML technique inspired by the human nervous system that enables learning by example from representative data that depicts a decision process or a physical occurrence. It has the unique ability to build empirical correlations between dependent and independent variables, as well as extract complex knowledge and information from datasets. They differ from DNNs due to their much lower number of layers and consequently worse capacity to handle very large datasets. The first perceptron, that is, the first true ANN, was proposed in 1958 by Frank Rosenblatt in the work [127], it consisted of binary classifiers with supervised learning. The perceptron was only possible due to the research done in the first artificial neuron [128] and in cell assembly theory [129].

The use of ANNs for PdM is a rather common approach in the literature due to its robustness and high performance, as such there is a wide range of works regarding this topic. For example, in [130] it is proposed a PdM system that takes into account machines' past breakdown history and mean time to failure to predict when these are going to fail, through the usage of an ANN model. It's worth noting that the proposed approach does not consider IoT technology, which can be an advantage in less developed manufacturing environments. A feed-forward ANN is employed in [131] to predict and classify faults in induction motors, by considering voltage and current changes. Furthermore, it incorporates a monitoring system to detect faults early and act accordingly. The work presented in [132] predicts two crucial machine tools condition, the cutting tool and the spindle motor. Three classification models are explored, SVM and two ANNs, a recurrent neural network, and a convolutional neural network. Reference [133] proposes a PdM system that predicts air booster compressor motor failure, by using an ANN. The highlight of the present work is the usage of a particle swarm optimization algorithm to train the bias and weights of the ANN. A preventive maintenance system to identify defects in power substation equipment is proposed in [134]. It uses a multi-layered perceptron to process eleven different features, in order to classify the thermal conditions of components in power substation equipment. Other also noteworthy works are detailed in [135], [136], as well as a review in [137].

Artificial neural networks can also be applied to other fields, such as healthcare [138], education [139], network security [140], robotics [141], finance [142], and agriculture [143].

2.2.3 Related Works

The work proposed in [144] is very similar, in some components, to the present dissertation's methodology since it incorporates real-time rescheduling of production and maintenance activities in a flexible job shop with the objective of reducing the overall monetary costs. In addition, it also considers the minimization of maintenance costs and product tardiness. To achieve this, the proposed solution employs a real-time GA optimization approach for scheduling/rescheduling, and an integrated proactive-reactive model (i.e., dynamic and stochastic) for machine condition monitoring. There are six problems that the proposed methodology takes into account, machine degradation, condition-based maintenance, random breakdowns, minimal repairs, unexpected deadline changes, and new incoming products to manufacture. To validate the solution several real-time event scenarios are used, and results demonstrate cost reductions averaging 27% when rescheduling was used. Also, there were additional costs savings, of up to 30%, when the executing time was increased from 30 to 90 seconds.

Also identical to the problem at hand, [145] proposes a joint production and maintenance scheduling, as well as quality control optimization for a serial-parallel multistage production system. It considers machines' conditions (e.g., age and usage) in order to determine product quality. In addition, a real-time monitoring product quality system is employed during the work-in-process manufacturing environment to provide data to a preventive maintenance system, that schedules maintenance activities. To accomplish these results, a constrained stochastic mathematical model that reduces the overall costs, by taking into account the length of a manufacturing run, maintenance and quality control thresholds, is proposed and solved by a simulation-based approach using a GA and Monte Carlo simulation. Results reveal that the proposed solution outperforms conventional maintenance strategies in terms of monetary performance.

The paper in [146] presents a GA approach for production and maintenance scheduling, while also satisfying product demands, that is, product quality control. The proposed approach is applied to batch manufacturing systems susceptible to changing operational conditions, in order to minimize the total costs. It integrates, following each batch production, an imperfect preventative maintenance methodology to define the best maintenance strategy that limits system deterioration. Furthermore, product inspections are also taken into consideration to sort nonconforming items from the final products. The GA was compared to the simulated annealing technique, and while results are close, the GA reached better monetary solutions. Moreover, the overall methodology was also compared to other more traditional methods, and results show that the proposed solution outperforms traditional methods in all cost-saving scenarios.

A DR-oriented production and maintenance scheduling system for cost minimization is presented in [147]. It balances production capability degradation, maintenance activities, and time-of-use electricity prices, through a multi-objective function. The case study carried out demonstrates that, when compared to just lowering electricity costs, the suggested methodology can save up to 19% in monetary costs. Furthermore, when compared to approaches that just reduce maintenance costs, the suggested model may save 14% in costs.

In [148] a multi-objective GA optimization is proposed to minimize energy costs and at the same time maximize the value of machine unavailability. The work considers time-of-use pricing (i.e., electricity prices are adjusted according to customer demand) and machine failure. Furthermore, two innovative methods are proposed, α-improvement and P-improvement, to create a better Pareto frontier, so as to improve the trade-offs in the multi-objective function. The solution was also compared to the recent NSGA-II, and results show that the algorithm proposed not only provides better solutions but is also faster.

A dual production scheduling and PdM optimization are proposed in [13], which searches for the optimal maintenance schedule, through the usage of a GA. The paper's main objective is to minimize machine failure rate, maximize utilization, and reduce downtime costs. It proposes an integrated decision model that combines PdM choices based on prognostics data with a single machine scheduling decision to reduce the overall costs. Furthermore, the health condition and dummy age subjected to machine degradation are taken into account in the integrated model. Results show that the proposed solution when compared to other strategies, such as scheduling production and maintenance independently, is more effective and efficient at solving the problem at hand, with cost reductions of up to 8.25%.

In [149] it is proposed an opportunistic maintenance scheduling methodology that considers stochastic opportunity duration in a PdM schedule. Prognostic data is utilized to identify opportunities that will occur before failure. The proposed maintenance scheduling system is founded on the Bruss algorithm, which solves optimal stopping problems. This paper's innovation is that it takes into account the stochastic nature of opportunities duration, by using a Monte-Carlo simulation. A numerical study is conducted to validate the solution, while it has some drawbacks, the solution presents promising results.

Reference [150] focuses on a maintenance scheduling approach by using predictive scheduling in a job shop environment. The proposed methodology can be divided into four fundamental parts: scheduling, predictive maintenance, maintenance schedule, and assessment. In the first part, the generation of a basic schedule using an immune algorithm is accomplished. Then, based on probability theory, machine failure-free times are predicted. Afterward, maintenance tasks and shiftable operations are scheduled. Finally, in the event of bottleneck failure, prediction accuracy and predictive scheduling efficiency are evaluated, in order to better handle these situations in future schedules. The proposed methodology allowed to achieve dependable values of Maxwell characteristics, which are crucial for achieving a reliable schedule. Maximum likelihood and empirical moments were used to estimate the Maxwell parameters. Moreover, four criteria were taken into account to find the optimal predictive schedule: flow time, makespan, total delay, and machine idle time. Also noteworthy is that the proposed methodology considers two secondary criteria to further improve the resulting schedule, quality and solution robustness. The case study in the paper showed that the proposed solution has good scheduling results and promising prediction capabilities.

For machine prognosis through PdM applied to production scheduling, [151] proposes a holistic framework that integrates an ML model for PdM with production scheduling. To achieve this, a conditional variational autoencoder based on a generative deep learning model that can extract an operation-specific health indicator from machine condition monitoring data is employed. Due to the unsupervised learning nature, the system is capable of o dealing with the scarcity of labeled machine failure data. The proposed methodology for health prognostics presents its results in a quantitative measure of degradation allowing for easy integration between PdM and production scheduling.

Therefore, the present paper only describes how the PdM component is done and how it could be integrated into a production scheduling system, it does not describe directly the deployment of the production scheduler. A case study using real industrial data was used to validate the solution, and the obtained results demonstrate that the proposed methodology is able to capture and quantify variations in machine condition.

More noteworthy works in the field, relating to the present dissertation's problem, are represented in Table 5.

Table 5 — Summary of other existing work similar to the present dissertation's approach

Article	Objective[1]	Method[2]	Application[3]
[152]	CM, DR, PS, MS	MIP, GA	HFS, M
[153]	CM, DR, PS, MS	GA	JSP, M
[154]	CM, PS, MS	ANSGA-III	FJSP, M
[155]	CM, PS, MS	GA, LP	M
[156]	PS, MS	GA	M
[157]	PS, MS	PSO	JSP, M

1 CM: Cost Minimization, PS: Product Scheduling, MS: Maintenance Scheduling.

2 MIP: Mixed Integer Programming, ANSGA-III: Approximate Nondominated Sorting Genetic Algorithm-3, LP: Linear Programming, PSO: Particle Swarm Optimization.

3 HFS: Hybrid Flow Shop, M: Manufacturing, JSP: Job Shop, FJSP: Flexible Job Shop.

2.3 Final Remarks

In this section, several works and techniques in the literature have been reviewed that address the problem of dealing with uncertainties in the manufacturing environment, from efficient production and maintenance schedulers to PdM systems. Furthermore, various works that focus on minimizing monetary costs by considering DR participation, volatile energy prices, and RERs, in a manufacturing environment, were also taken into account.

Regarding the overall scheduling of production tasks or maintenance activities, there are a wide variety of available techniques to solve this type of problem, being the most popular, reinforcement learning, linear programming, particle swarm optimization, and simulated annealing. However, there is one technique that stands out among the others due to its robustness, performance, and flexibility, the GA. Because of these characteristics and others mentioned in section 2.2.1.2, the GA is ideal to be employed in the problem at hand.

For PdM, either to predict machine health, part condition, or failure, a wide range of ML and statistical techniques were explored, with the most used being, RFs, deep learning, SVMs, and k-means. Nevertheless, the chosen technique for the current dissertation's system was ANNs, due to its performance, simple implementation, backpropagation capabilities, as well as other benefits already cited in section 2.2.2.2 when compared to other techniques.

Finally, a comprehensive literature review concerning works that are very similar or complement the current dissertation's system was done. Its objective is to determine how the different components in the proposed system can be effectively combined. These components are DR participation, MB events, volatile electricity prices adaptation, RERs usage, energy selling, production and maintenance scheduling, constraints imposed on the production plan, and PdM. While there are works that consider production and maintenance scheduling such as in the cited [145], [146], these approaches are simplified too much by not considering the volatility in electricity prices, constraints imposed in manufacturing, or even reactive maintenance. The number of relevant works that consider DR participation and maintenance scheduling is very low, even the works presented in [147], [148] are nothing special since they don't consider RERs usage, something crucial in minimizing energy costs. On the other hand, there is an abundance of works that only consider production scheduling and preventive maintenance, for instance, in [149], [150] and with special attention to [151]. However, even if these works are more simple and better at what they aim to achieve, there are still some problems, for example in [13] the problem is simplified too much to the point of only considering the production and maintenance scheduling of a single machine. Nevertheless, from all the research done, the work proposed in [144] is without a doubt the most similar to the dissertation's solution. The paper considers scheduling and real-time rescheduling of a production line, maintenance scheduling, through preventive and reactive maintenance, product deadline, and the overall minimization of costs. Furthermore, it also employes a GA for optimization and is applied to a flexible job shop manufacturing environment. Its drawbacks are not taking into account DR participation, MB events, volatile electricity prices, RER usage, and energy selling.

From the above-cited works, none of them implement a system that handles all of the most relevant real-world production scheduling problems, such as considering the existence of possible reschedules, flexible job shop layouts, DR programs, MB events, volatile electricity prices, RERs, energy selling, and machine failure/degradation, while also focusing on the most important factor that keeps companies afloat, monetary cost reductions. This is a big gap in the literature, and it would be interesting to know how all of these problems would balance themselves in order to minimize costs. Therefore, the objective of the current dissertation is to solve completely or partially these problems, and if possible, inspire other researchers to also branch out in this research field.

It is noteworthy that the exploration of related works in other fields was also considered in the literature review, however, because the culmination of production/maintenance scheduling and PdM is a very specific problem in manufacturing, there were little to no works in other fields that were published in 2017 or afterward.

3 Adopted Methodology and Implemented System

This section presents the major decisions made for the development of the proposed methodology. It encompasses the proposed architecture, which AI techniques were chosen to be implemented, the selected programming language as well as libraries, and the datasets used to train and validate the proposed solution.

3.1 Architecture

The proposed methodology achieves cost reduction optimization capable of participating in DR programs and MB events while also lowering maintenance costs, as well as improve machine longevity. It uses an intelligent scheduler/rescheduler based on a search-based algorithm, a genetic algorithm, to find a low-cost production plan. The scheduler/rescheduler proposed in this paper considers locally generated energy, dynamic pricing, excess energy selling to third parties, maintenance activities, DR programs, MB events, and constraints imposed on the production plan to minimize the overall costs and machine occupancy rate standard deviation (i.e., improve machine longevity) of a production line. Furthermore, it uses an ML approach based on ANNs to predict a machine failure status in order to redirect such information to the scheduler for a more effective cost optimization. Figure 2 represents the proposed methodology, where users must first provide which products they want and how many (i.e., product requests), as well as constraints that are imposed on the production plan. Retailer energy (i.e., electricity prices and available retailers) and generated energy (e.g., PV availability) must also be provided to the scheduler. In addition, production data (e.g., product task list, available machines and their respective energy profile, setup information, and maintenance activities duration) is also required to be provided to the scheduler. Machines' failure status are also provided to the scheduler, by the PdM system. After obtaining all the aforementioned data, a production plan, that reduces the overall costs and machine occupancy rate standard deviation of a production line, is provided by the scheduler. However, if a DR program occurs, a new production plan, using the same data, is scheduled, but with the added energy limit constraint in order to comply with the DR program.

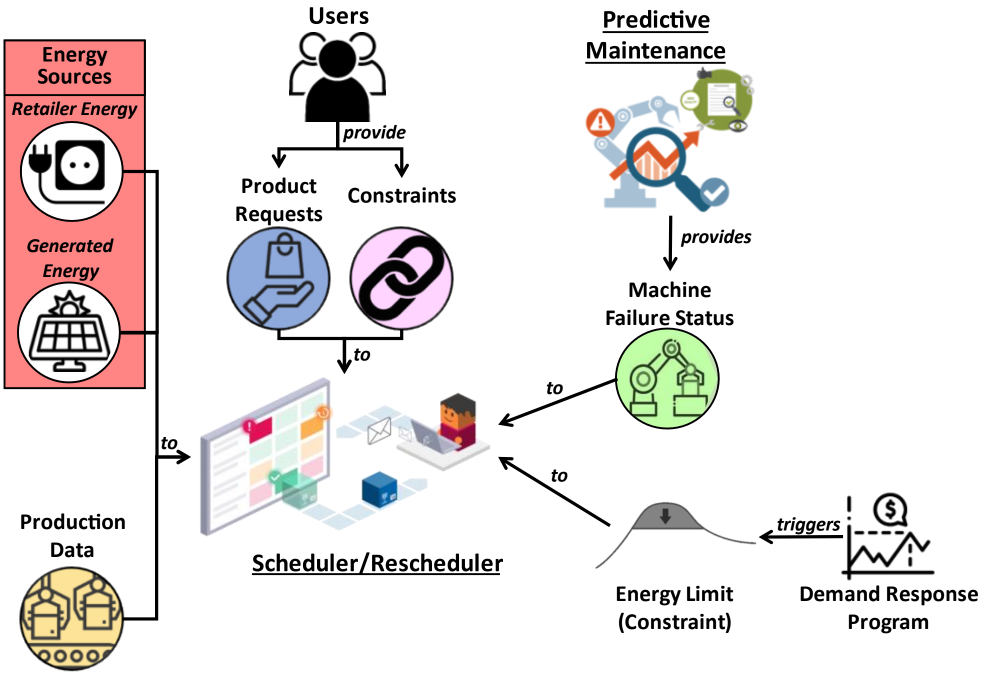

Figure 2 – Proposed architecture for production and maintenance optimization in production lines

3.1.1 Production/Maintenance Scheduler Domain Model

The proposed scheduler to achieve energy cost minimization uses a combination of AI techniques, such as genetic algorithms, and deterministic-based optimizations. It is developed in the programming language Python, without using any libraries related to GAs, since no library is capable of solving the complexity of the problem presented in this dissertation. The proposed scheduler not only allows energy cost optimization by scheduling tasks according to the availability of energy sources and the energy prices, but it also enables the consideration of DR programs and maintenance activities, by shifting tasks away from the DR event period and by shifting maintenance activities to more cost-effective periods, respectively.

The proposed scheduler is not dependent on time units. There is no definition regarding time units, leaving this definition to the user during execution time. The methodology uses the concept of periods without actually knowing what it represents (e.g., one hour, fifteen minutes, or one second). Therefore, all the inputs must respect the same period. For instance, if the duration of tasks is defined in a one-minute period, then the energy market prices and forecasts must be given also in one-minute periods, maintaining data consistency. The energy units must be provided as Wh, but they can vary in their unit (e.g., kWh or MWh), using the same logic as periods, the user must use the same prefix for all energy units. Energy usage is always represented as energy unit per period (e.g., Wh/period, kWh/period, or MWh/period).

In this dissertation, production lines follow a flexible job shop configuration. They are characterized by a set of concepts that allow solving the cost and machine occupancy deviation minimization problem of task and maintenance activity scheduling. The proposed domain model for production and maintenance scheduling optimization, shown in Figure 3, can be divided into twelve sections:

- **Task** — an activity that needs to be executed to create a product;
- **Task mode** — a combination of energy and time profiles regarding the execution of a given task, where each task can have multiple task modes associated, which means that a single task can be executed in different ways;
- **Machine** — a representation of a machine that has a list of compatible task modes that it can execute, thus being indirectly associated with tasks;
- **Maintenance** — an activity that needs to be done to repair a machine, can have stipulated maintenance hours with corresponding prices, depending on whether or not the maintenance activity was done during the stipulated time;

- **Cell** — portrays a collection of machines where a product can be built;
- **Product** — a representation of a product describing the necessary tasks to be completed before the product is considered finished;
- **Product request** — the request of a new product to be produced along with the quantity needed;
- **Energy source** — the energy source availability amount and price, an energy source can represent external providers (e.g., aggregator or retailer) or local generation (e.g., photovoltaic), the energy prices of local renewable sources can be set to zero;
- **Energy buyer** — the price a buyer is able to pay for each generated energy (i.e., energy with no cost) in excess;
- **Demand response** — identifies the DR program that the production line will participate in;
- **Machine breakdown** — identifies the MB event that the production line will have to adapt to;
- **Constraint** — a condition that will be respected by the algorithm.

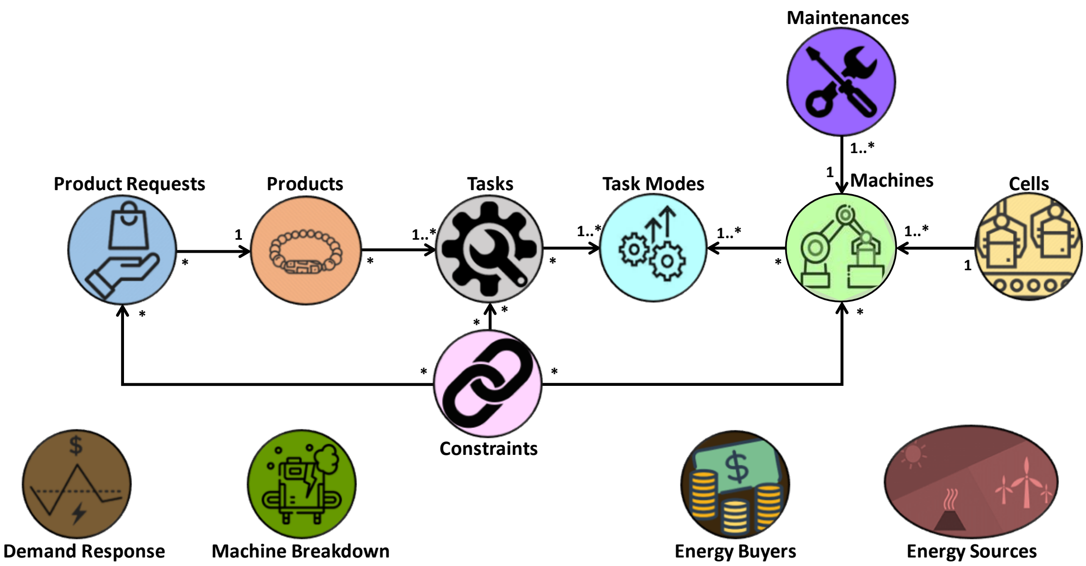

Figure 3 – Domain model of the proposed production/maintenance scheduler methodology

It is worth noting that, in this dissertation, the concept of a production line is characterized as being the culmination of all the above-mentioned concepts. It describes a collection of job shop cells, thus while not identical, the concepts of production line and cells are very similar. A production line represents a part of the factory, which has a specific function for the manufacturing of products.

The use of constraints is not mandatory, but it is needed to represent realistic cases where physical restrictions are in place. Currently, the proposed algorithm has the following constraints available to use:

- **Task order** — defines a sequence between two tasks, for example, task B can only be executed after task A is completed;

- **Product task order** — defines a sequence of tasks for a given product, allowing not only to have chains of repeated tasks but also have order constraints unique to a product. It is noteworthy that tasks that integrate this constraint do not need to comply with the task order constraint, due to logical fallacies;

- **Task collision** — defines two tasks that cannot be executed at the same time (i.e., same period);

- **Task setup** — defines a setup action that needs to occur before task execution, this setup is defined by time and energy;

- **Time leap** — can define breaks in time, the algorithm can schedule an entire week, but in cases where the production line does not operate 24 h a day, the time leap is used to indicate the last period before the break. This is necessary for tasks that cannot be stopped and must completely end before the break;

- **Interruptible task** — defines tasks that can be paused and resumed any time, thus bypassing the time leap constraint;

- **Machine available frames** — defines a maximum number of periods a machine is able to operate, for instance, it can be useful to prevent a machine from overloading or to indicate that a machine is no longer operable (if the constraint is set to zero), useful for MB events;

- **Machine priority** — defines, for each period, if a specified machine is to be benefited, penalized, or neutral regarding its energy costs;

- **Product request deadline** — defines a deadline to which production of a given product request must be completed;

- **Product request task period range** — defines when a task is able to start and must be finished (e.g., task A can only start at 7:00 and must be completed by 15:00);

- **Product request cell choosing** — defines the cell where the product request must be produced, bypassing the cell balancing optimization;

- **Energy limit** — defines an energy limit, within a given interval, to be applied to energy sources with prices above zero. This constraint may also have monetary compensation for the full compliance with its limit;

- **Shift margin** — the margin on energy price that the algorithm can use to allow the task to be close to each other, avoiding the existence of empty periods even if this increases the energy costs.

3.1.2 Predictive Maintenance Domain Model

For PdM, the corresponding domain model is dependent on the features of the dataset used, in this case, the dataset described in section 3.4.1.3. Therefore, it is not as flexible as the production/maintenance scheduler domain model. The domain model for the proposed PdM model, represented in Figure 4, is characterized by:

- **Database sample** – represents a unique sample from a machine database;

- **Temperature difference** – describes the difference between the temperature inside the machine and the exterior temperature of a machine;

- **Rotational speed** – describes the rotational speed of the tools of a machine;

- **Torque** – describes the produced force to rotate the tools of a machine;

- **Tool wear** – describes the degradation level of the tools of a machine;

- **Machine failure status** – describes whether a machine has failed or not. It assumes the value of 0 or 1 for non-failure and failure, respectively;

- **Machine** — portrays a machine for production.

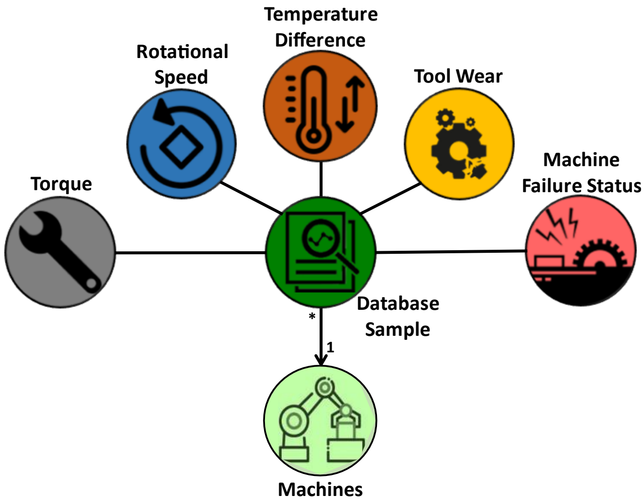

Figure 4 – Domain model of the proposed predictive maintenance methodology

3.2 Adopted Models and Methods

This section describes the selected AI models to be employed and methods used in the proposed solution as well as their overall implementation.

3.2.1 Production/Maintenance Scheduling

The scheduler/rescheduler of the proposed solution schedules both production and maintenance activities in order to minimize both the total costs and machine occupancy deviation, while also taking into account constraints imposed on the production plan. To achieve this, a GA was selected, since, as already cited in section 2.2.1.2, they provide semi-consistent good solutions, are a robust search-based optimization algorithm ideal for task scheduling problems, are thoroughly explored in the literature, are highly adaptable due to their optimized control parameters (e.g., population size, elite size, mutation rate, and stopping criteria), and if implemented with the most novel approaches it can easily avoid local optimum.

The overall implementation of the GA can be divided into four main phases: the creation of the initial population, the breeding of individuals (i.e., crossover), maintaining genetic diversity (i.e., mutation), and the selection of the individuals for the next generation, as shown in Figure 5.

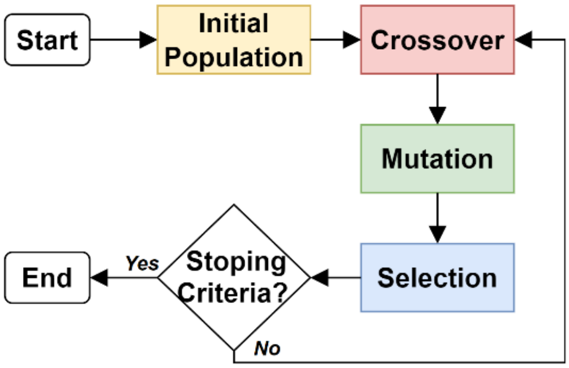

Figure 5 – Flowchart of the proposed genetic algorithm [37]

In the first phase, the GA starts by initially creating an initial random population, able to comply with all the imposed constraints on the production plan. It is noteworthy that duplicated individuals in the population are removed and replaced with another random individual that is not a duplicate. Then, a genetic generation begins with the crossover phase applied to the initial population, however, in subsequent genetic generations, the crossover is applied to the previous generation. In the crossover phase, individuals are randomly crossed in pairs from the population pool, for each pair, composed of parent 1 (i.e., individual 1) and parent 2 (i.e., individual 2), with each pair of parents producing two children. If an invalid child (i.e., crossed individual that does not respect all the imposed constraints) is produced it is never added to the population pool. Afterward, using the population obtained from the crossover, a mutation is performed. In this phase, the algorithm takes each individual, and according to the percentage of mutation defined in the input data, it decides whether or not the mutation will be applied to a given individual. If a mutation is to be applied it can consist of a simple task swapping procedure. In the last phase, the selection begins with the union of the new and the old populations, that is, the crossed and mutated population with the initial population of the previous generation. It is worth noting that in this phase duplicated individuals are eliminated. Then, each individual is evaluated using a fitness score equation that minimizes the overall total costs, maximizes profit from selling energy, and minimizes machine occupancy deviation. After each individual is evaluated, the algorithm selects the n best individuals (i.e., elite selection), based on the input data defined by the user, to be inherited to the next generation. Additionally, any remaining individuals (i.e., population size – elite size) are obtained from non-elite tournaments. After the selection phase, a new genetic generation begins with the population obtained from this selection, and all the phases mentioned above are repeated. Finally, when a stop condition is met (e.g., by the number of generations, algorithm execution time, fitness stagnation, or cost reached), the individual with the lowest energy cost production plan generated by the GA is extracted and used as the production plan found by the algorithm.

In addition, other heuristic methods are also added to the GA to further reduce energy costs, improve the chances of creating genetic individuals that respect all the imposed constraints on the production plan, help balance products between cells, and increase space for maintenance activities.

The optimization control parameters considered in the proposed GA are:

- **Population size** — describes the maximum number of individuals considered in the genetic population. The population size influences both the quality of the solution and the processing time. On one hand, a small population can have performance drops because it provides a small coverage of the search space of the problem. On the other hand, a large population generally provides representative coverage of the problem domain, as well as preventing premature convergences to local rather than global solutions. However, to work with large populations, greater computational resources are required, or the algorithm must run for considerably longer periods of time. Therefore, in general, the ideal population size for the proposed GA, which tends to be more computationally heavy than other GA approaches, is 10 to 20;

- **Elite size** — represents the number of highest fitness individuals that are guaranteed to inherit to the next genetic generation. The higher the number of elites, the more elitist the GA gets, which can lead prematurely to local optimum. However, in the case of a very low elite population, the convergence of the GA tends to be slower, as there is a large number of high fitness individuals being discarded. Therefore, the ideal elite size is about 15% to 40% of the population size (e.g., with a population size of 20 and 15% of elites means an elite size of 3);

- **Mutation rate** — defines the probability of an individual being mutated. A low mutation rate prevents a given position from stagnating at a value, in addition to making it possible to arrive at any point in the search space. When the rate is exceedingly high, the search becomes essentially random. The general rule of thumb for the mutation rate, for the proposed GA, is 1% to 5%;

- **Stopping criteria** — portrays criteria that are used to stop the GA. The most commonly used are by execution time, number of generations run, generations since last improvement (i.e., stagnation), and target value reached. It is recommended to define more than one stopping criteria, for example, to avoid wasting time when the fitness has already stagnated or to not wait too much time for a solution. While stopping criteria are more dependent on the type and complexity of the problem, it is usually recommended to add a maximum execution time of no more than 30 minutes to 1 hour along with a stagnation stopping criteria of 15 to 30 generations since the last improvement. Nevertheless, this criteria is highly dependent on the complexity of the problem at hand, with more complex problems requiring more computing time.

3.2.2 Predictive Maintenance

To predict a machine failure status, an ANN model was employed because, as already stated in section 2.2.2.2, it has really good performance when compared to other ML models, is able to be continuously trained due to the backpropagation feature, and avoids overfitting in this type of simplistic problem due to not being overly complex like a deep learning model. It is worth noting that in section 5.2.5 the proposed ANN is compared to other ML models (GB, SVM, and RF) to validate the proposed model's performance.

The training process begins by obtaining the newest machine data (i.e., air temperature, machines' process temperature, rotational speed, torque, tool wear, and machine failure status information) from the machine database employed in the facility. Then, before training starts, a data preprocessing phase begins in which: (1) aggregate all the data collected into a single data file (i.e., data aggregator); (2) normalize data scales and types, using a Min-Max strategy with the MinMaxScaler method [158] (i.e., data normalization); (3) fill in missing values on the gathered data, by using a k-Nearest Neighbors imputation technique from the KNNImputer method [159] (i.e., data imputation); (4) remove possible irrelevant and erroneous data, by detecting outliers using the Z-score technique with the SciPy library stats Z-score function [160] (i.e., data filtering); (5) transform raw data into features that better represent the underlying problem (i.e., data engineering); and finally, (6) balance machine data failure and non-failure points, by using the imbalanced-learn [161] library (i.e., data balancing).

Afterward, the preprocessed data is fed to the ANN for training. If the model has already been trained, then the model's neuron weights are adjusted, according to the new data, due to the back-propagation feature. This training process can start every time there is new data in the machine database. It is also worth noting that the initial ANN model is constructed using the Holdout method, with a split of 80% of the dataset for training and 20% for testing.

The application in real-time of the proposed ANN model is similar to the training process. It begins by obtaining the necessary machine data, which can represent the present data of a machine or forecasted data, within a predefined time horizon, provided by a third-party system. Then, data normalization, imputation, filtering, and engineering are applied to the obtained machine data. Finally, the ANN is used to predict the machine failure status (0 for non-failure and 1 for failure) of the machine.

3.3 Technologies

In this section, it is described the available programming languages that could be used to implement the proposed solution and which one was chosen, in section 3.3.1. In addition, section 3.3.2 explores several libraries that can be used to implement the proposed GA and ANN, and consequently which ones were selected.

3.3.1 Programming Language

There are a wide variety o programming languages that could be used to implement the proposed solution, such as Java [162], C [163], C++ [164], Prolog [165], and Python [29]. Some are more appropriate for scheduling problems (e.g., Prolog, C, and C++) while others are more suited for ML models (e.g., Python, and Java). Furthermore, programming languages such as Java, C, and C++ have better runtime performance when compared to Prolog or Python [166].

However, due to the rigid syntax of the Java, C, and C++ programming languages their median hours to solve a problem tend to be much higher. In fact, Python presents itself as one of the fastest languages to solve a problem, on average it takes 1.2 to 1.6 times shorter to solve a problem than other programming languages [167].

It is noteworthy that Python also offers a wide range of ML libraries that have great performance and are robust. Moreover, when compared to Prolog, Python can be easily implemented to scheduling and prediction problems, whereas Prolog is not so good regarding problems that incorporate ML. Therefore, because Python has lower solving times, has a variety of ML libraries, and is adaptable to different kinds of problems it was chosen to be the main programming language both for the GA scheduler as well as for the ANN for PdM.

In addition, for the development of the GA and ANN model the Anaconda software [168] was used since it provides a wide range of useful libraries already installed. Also, the Integrated Development Environment (IDE) and source-code editor used for implementation was the Visual Studio Code [169]. The ANN model was initially developed using Google Colab [170], due to its ease of use, available ML libraries, and the ability to run models remotely using Google's servers. However, due to processing power limitations, the model had to be moved to Anaconda/Visual Code Studio, in order to be executed on a local machine.

3.3.2 Libraries/Tools

This section presents what Python technologies and libraries exist to implement the GA, in section 3.3.2.1, and the ANN in section 3.3.2.2. In addition, it indicates which technologies and libraries were chosen and why.

3.3.2.1 Genetic Algorithm

There is a huge variety of technologies and libraries in Python that allow easy and fast implementation of a GA. For instance, MATLAB [171], DEAP [172], PyGAD [173], and pyeasyga [174]. However, none of the above-mentioned libraries is capable of solving the complexity of the problem presented in the dissertation, mainly regarding flexible job shop, the possibility of having task modes, and the application of complex constraints in the production plan. Therefore, the GA scheduler/rescheduler was developed in the Python programming language without the use of any libraries related to GAs. It is noteworthy that in order to deploy the GA as a REST API server the Flask [175] library is used.

3.3.2.2 Artificial Neural Network

Technologies and libraries for ML models in Python are widely available, as it is one of the main appeals for the usage of Python in the current days. The most relevant technologies and libraries are MATLAB [171], Keras [30] TensorFlow [176], PyTorch [177], Scikit-learn [31], and SciPy [32]. It is also worth noting that there are other libraries that help in dealing with datasets, such as Pandas [178] and NumPy [179]. The Keras library was chosen for the ANN model since it provides good performance, is robust, allows easy implementation of ML models, has a wide range of other ML techniques which can be used to compare it to the proposed solution's model, but primarily because it allows changing almost every hyperparameter of the model (e.g., criterion, learning rate, maximum depth, maximum features, minimum samples in a leaf, minimum samples to split, and number of estimators. It is also worth noting that the Scikit-learn library was used to implement an automatic hyperparameter optimizer for the ANN, which determines the optimal hyperparameter values to create a high-performing model, as well as a data normalization method using a Min-Max strategy and an imputation method using a k-Nearest Neighbors imputation technique. Also, to detect outliers it was used the Z-score function from the SciPy library. It is worth noting that the PdM system is also deployed as a REST API server, using the Flask [175] library.

3.4 Datasets

To validate the proposed solution of the present dissertation three different datasets are considered. The first dataset contains production, generated energy and maintenance data from a manufacturing environment, being characterized as the Scheduling Dataset (SD), since all the data described in the dataset is used to validate the scheduling component. The second dataset provides de electricity buying and selling prices as well as DR events that might occur, it is defined as the Demand Response Dataset (DRD). The last dataset describes machine data for PdM, defined as the Predictive Maintenance Dataset (PdMD). It is noteworthy that, in section 3.4.1 a more generalistic description of the SD and DRD will be presented since both of the datasets are not disclosed publicly, while the PdMD will have a more detailed description of its features due to being publically available. Section 3.4.2 presents the steps taken for data protection.

3.4.1 Dataset Description

This section describes in detail the origin and contents of the datasets used in the present dissertation, as well as indicate problems that they have and solutions to solve them.

3.4.1.1 Scheduling Dataset

The SD uses real production data, provided by a textile company that manufactures hang tags. The data represents working schedules from 7h00 to 23h00 each working day, from Monday to Saturday. The acquisition of the dataset was only possible due to the MUWO project [21] and GECAD's reputation. Furthermore, its contents are restricted to the public.

While there is a wide range of data to choose from, the main data used to validate the proposed solution is the following:

- Machine data (i.e., compatible tasks, available modes for each task, machine energy profile, and setup information);
- Product data (i.e., product task list, product orders, available stock, and deadlines);
- Generation data (i.e., available generated energy, for example, PV);
- Maintenance data (i.e., how often a machine has maintenance, business as usual maintenance hours, labor cost, and maintenance activities duration).

The SD dataset presents some problems such as inconsistent timestamps, for instance, some machines have a sensor that obtains energy consumption every 5 seconds, while there are others at 10 to 15 seconds. Furthermore, this problem also extends to different scales, with machines reading in kW and others at W. Both of these problems were solved through data normalization, nevertheless, it is still a time-consuming and grueling task, mainly to normalize the timestamps. There are also problems regarding missing values in the generation data, for example, there are months where there is no data, which is something critical when all of these months coincide with the hot or cold season. To solve this issue, feature imputation is applied (i.e., techniques to fill missing values) by using the months of the other years. Moreover, concerning generation data, there are also rare energy spikes, even during the night, which might be a sign of sensor degradation. This issue was solved by detecting the outliners and applying data normalization based on the neighborhood's values.

It is also worth mentioning that some maintenance data was obtained from the paper available in [33], such as labor costs and maintenance activities duration.

3.4.1.2 Demand Response Dataset

The DRD includes real electricity market prices from MIBEL (Iberian Electricity Market) [180], from the Portuguese market. In addition, DR program data is provided by an external demand-side management system, which cannot be disclosed to the public.

From the DRD the information used to validate the proposed solution was:

- Electricity data (i.e., electricity prices, and available retailers as well as energy buyers);
- Demand response data (i.e., DR predicted dates, maximum consumption allowed in a given moment of a DR event).

Regarding problems in the DRD, there were basically none. There are some instances where data needed to be adapted to be used as input in the scheduler, however, there are no major issues regarding, for example, missing data, outliers, or even different data scales or types. Most of the data provided was already pre-processed and was easily applied to the present proposed solution.

3.4.1.3 Predictive Maintenance Dataset

The PdMD was obtained in the Machine Learning Repository from the University of California, Irvine [181]. The PdMD, titled "AI4I 2020 Predictive Maintenance Dataset Data Set" from 2020, is publicly available in [34]. It is a synthetic dataset that represents a specific type of manufacturing machine, compiled of 10,000 data points where 339 represent failures and 9661 non-failures data points (i.e., a ratio of 1:28), as presented in Figure 6, that attempts to replicate industrial machine data for PdM. The most relevant features from the PdMD used in the validation of the proposed solution, with the exception of the last feature, are:

- **Air temperature** – Represents the air temperature outside the machine, in Kelvin (K);
- **Process temperature** − Describes the temperature generated inside the machine, in Kelvin (K);
- **Rotational speed** – Depicts the rotational speed of the tools inside the machine, in Revolutions per minute (rpm);
- **Torque** – Describes the force produced to rotate the tools inside the machine, in Newton-meters (Nm);

- **Tool wear** – Represents the level of degradation of the tools inside the machine, in minutes (min);

- **Machine failure** – Defines whether a machine has failed or not. It assumes the value of 0 or 1 for non-failure and failure, respectively;

- **Failure mode data** – Provides what type of failure(s) occurred. There are five different failure modes: tool wear failure, heat dissipation failure, power failure, overstrain failure, and other random failures. While this feature is not used in the proposed methodology, it is still worth mentioning since it could be implemented in future work.

There are some problems concerning the PdMD, from too many features to data imbalances. Since the dataset is going to be used in an ML model to predict machine failure, there is a greater need to reduce the number of inputs. To achieve this, feature engineering is crucial because it allows the combination of multiple features, for example, combining the air and process temperature to obtain the difference in temperature between the machine and the air. Also noteworthy, even though it is not used as a feature in the proposed methodology, each type of failure mode data has a corresponding column (i.e., feature), creating unnecessary inputs. This issue could be also solved through label encoding, by combining all of the failure modes into one column and classifying them numerically from 1 to 5. Regarding data imbalances, there is a big gap of around 1:28 ratio (339 failures and 9661 non-failures) of instances where a machine failed to when it did not fail, respectively. This rather common problem in machine failure datasets can be solved by oversampling failures and/or undersampling non-failures. Nevertheless, apart from the issues mentioned above, the dataset has no problems regarding outliers, missing data, or different data scales.

The correlation heatmap between the used dataset features is described in Figure 7. It demonstrates that there is a medium correlation between machine failure and the features torque and tool wear. On the other hand, the lowest correlation found to machine failure was the rotational speed.

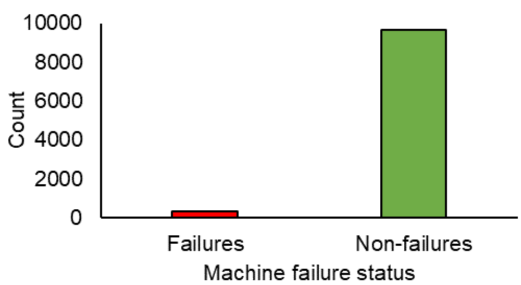

Figure 6 – Machine failure status bar chart of the used dataset

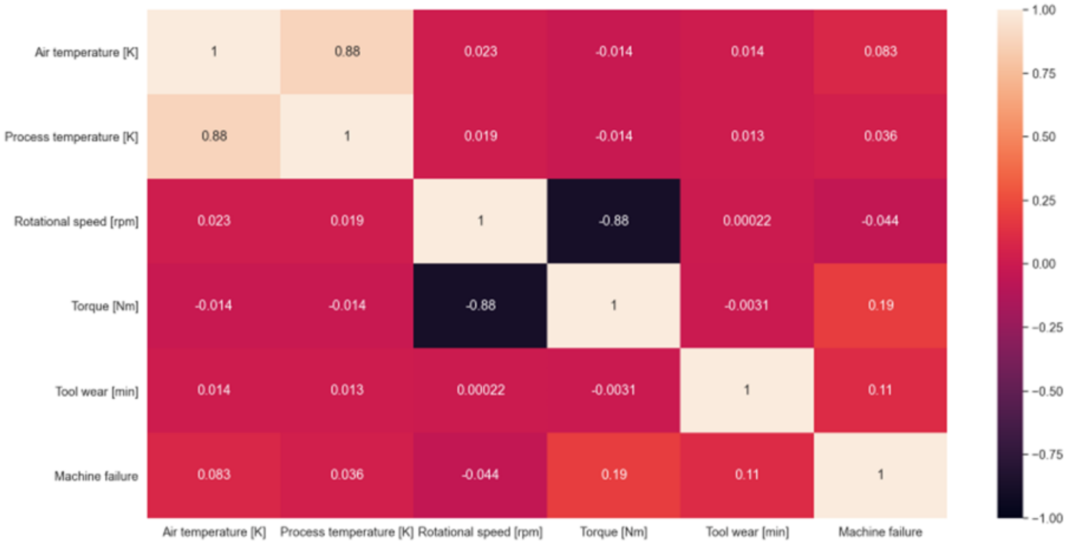

Figure 7 – Correlation heatmap between the used dataset features

3.4.2 Data Protection

Data-driven industrial monitoring systems have rapidly evolved and considerably enhanced performance on industrial monitoring jobs, which when coupled with production line scheduling systems improve productivity substantially [182]. However, the commonly used data-driven models expose industrial data to increasingly unprotected threats such as Phishing (i.e. attacks to steal user data in order to gain access to the internal systems) [183]. Even with data protections in Europe from the General Data Protection Regulation (GDPR) companies are still not safe from anonymous attackers.

In order to address the GDPR the following questions were formulated:

- What data is collected? – machine, product, energy generation and maintenance data are collected for the scheduler/rescheduler while for the PdM system machine condition data is collected;

- How is data collected? – Data is collected via sensors installed on machines and PV panels. Further data is collected from records regarding product and maintenance data;

- How is the data used? – In the scheduler/rescheduler data is used to validate the proposed solution, via performance tests. On the other hand, PdM data is collected to improve the performance of the ML model;

- Is the data used for marketing? – No data is used for marketing;

- What are the data protection rights for the company? – The right to access, rectification, erasure, restrict processing and data portability;

- Who is responsible in case of a data breach? − In the event of a breach of security leading to the accidental or unlawful destruction, loss, alteration, unauthorized disclosure of, or access to, personal data, the author of the proposed scheduler/rescheduler and PdM systems is responsible for such events.

To address these issues when the system is employed in the real world, the proposed scheduler is designed with a Service-Oriented Architecture (SOA) where its services are stateless (i.e., no records of previous interactions) and there is no direct access to databases, for instance, all the information needed for scheduling is sent when a request is made. Therefore, even if an attacker got the chance to access the proposed scheduler, there was no way to steal data, since there was none.

Regarding the PdM system, the training process of the ML model would be done outside of the main network system, and only after the model was trained, it would replace the old PdM model. With this approach, and keeping in mind that the proposed model is an ANN, if the attacker got access to the PdM system they would only have a black-box model, where no data concerning machine data could be extracted.

Finally, the dissemination of the current dissertation will always comply with GDPR, by not sharing data that the companies have not authorized to be distributed publically.

4 Implementation

In this section, a detailed description of the scheduling and PdM systems is made. The different components (i.e., scheduler/rescheduler and PdM) are connected internally in the manufacturing facilities via an HTTP protocol. Both systems are deployed as a REST API server, using the Flask library [175]. These systems are not connected automatically to each other, meaning that if a machine is detected to fail, the MB event or maintenance activity has to be manually added to the scheduler/rescheduler.

4.1.1 Production Line Optimization to Minimize Total Cost and Maximize Machine Longevity

The production line for total cost and machine longevity optimization is divided into three main components: data processing characterized by the cell balancing; genetic algorithm represented by the initial population, crossover, mutation, and selection; and finally, the deterministic optimizations defined by the cost and shift optimizations. Figure 8 represents the flowchart of the production line energy cost optimization.

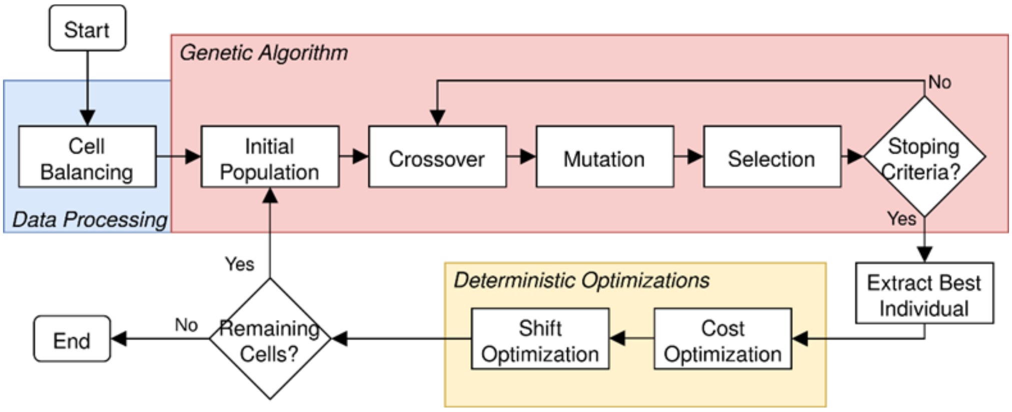

Figure 8 – Flowchart of the production line energy cost optimization [35]

4.1.1.1 Cell Balancing

The proposed algorithm starts by balancing the requests by the different cells available for manufacturing. This phase begins with the creation of a product request list, sorted by descending order of energy consumption needed to complete a product request. Because the tasks needed to complete a product request can be executed using different task modes with different energy profiles, an average is made considering all the possible task modes. An average of all the possible task modes is considered, above any other type of measure, since it better represents the overall processing time that a task can take.

Afterward, following the product request list, each request is assigned to the cell that has the lowest overall energy consumption, being the product request consumption added to the cell's overall energy consumption. Moreover, the cell assignment process also takes into account if a cell is able to produce the requested product and if it has enough periods to execute the product tasks.

Finally, after each request is assigned to a cell, each cell is executed by a genetic algorithm, followed by deterministic-based optimizations. However, the cell optimizations are connected among them by the general data and comply with general constraints, such as the availability of renewable energy sources that are shared among cells, and energy limit constraints that are applied to the plant floor (i.e., the sum of cells).

The cell balancing optimization focuses on maximizing energy distribution among cells, thus allowing more flexible request management, which in turn reduces the overall energy cost.

4.1.1.2 Initial Population

The GA starts by initially creating an initial random population, respecting all the imposed constraints. To facilitate the creation of an individual and decrease the rate of not feasible individuals, the algorithm takes into account the compatible machines for each task and prioritizes tasks with a lower number of compatible machines and with a higher processing time. It is noteworthy that, in this stage of the algorithm, maintenance activities and machine setups are treated as tasks with imposed constraints, such as when a maintenance activity could begin and end, and that setups must precede a certain task. After knowing the list of tasks associated with each machine, a two-dimensional matrix is created, which represents the work plan of the job shop, where each line represents the plan of a machine, and each column represents a period. Figure 9 shows an example of such a matrix, where tasks, which some of them can represent a maintenance activity or a machine setup, are identified by colors and their identifiers.

Machine/Period	1	2	3	4	5	6	7	8	9	10
Machine 1	t1	t1	t1	t7	t7	t8	t8		t9	
Machine 2		t5	t5				t2	t2	t2	t2
Machine 3	t4	t6	t6	t6	t6	t6	t6	t3	t3	t3

Figure 9 – Example of an individual matrix (machine/period), from the genetic algorithm, where tasks are identified by colors and their identifiers [35]

After an individual matrix is generated and at least one constraint is not respected, there are implemented methods that try to repair the generated individual (i.e., by shifting tasks left or right, swapping tasks). In case the repair is not successful, another individual is generated.

Since the number of columns (i.e., time period) corresponds to the time window of the schedule, every matrix from each cell will have the same number of columns but may differ in the number of rows, as cells can have a different number of machines associated.

4.1.1.3 Crossover

A new generation begins with the crossover of the population of the previous generation. The crossover performed is done between two individuals randomly chosen, which have not yet been crossed; this is guaranteed through a permutation of the population before the crossover is done. The crossover is a two-dimensional type, which will try to balance the tasks coming from each parent (individual). The crossover between two individuals (parents) begins with the formation of the list of tasks for the entire plan, ordered in decreasing order of processing time, and if they have equal times, it is ordered in ascending order of the number of machines compatible with the task. This, therefore, allows reducing the rate of invalid crosses between two individuals. In order of the task list, each task is taken and inserted in the child according to the parents' coordinates; first, it starts with parent 1, then it changes to parent 2, then parent 1 again, and in general, follows this sequence. Another child is made from the same parents but starts with parent 2 coordinates. Figure 10a represents the first four steps in a crossover, starting from parent 1.

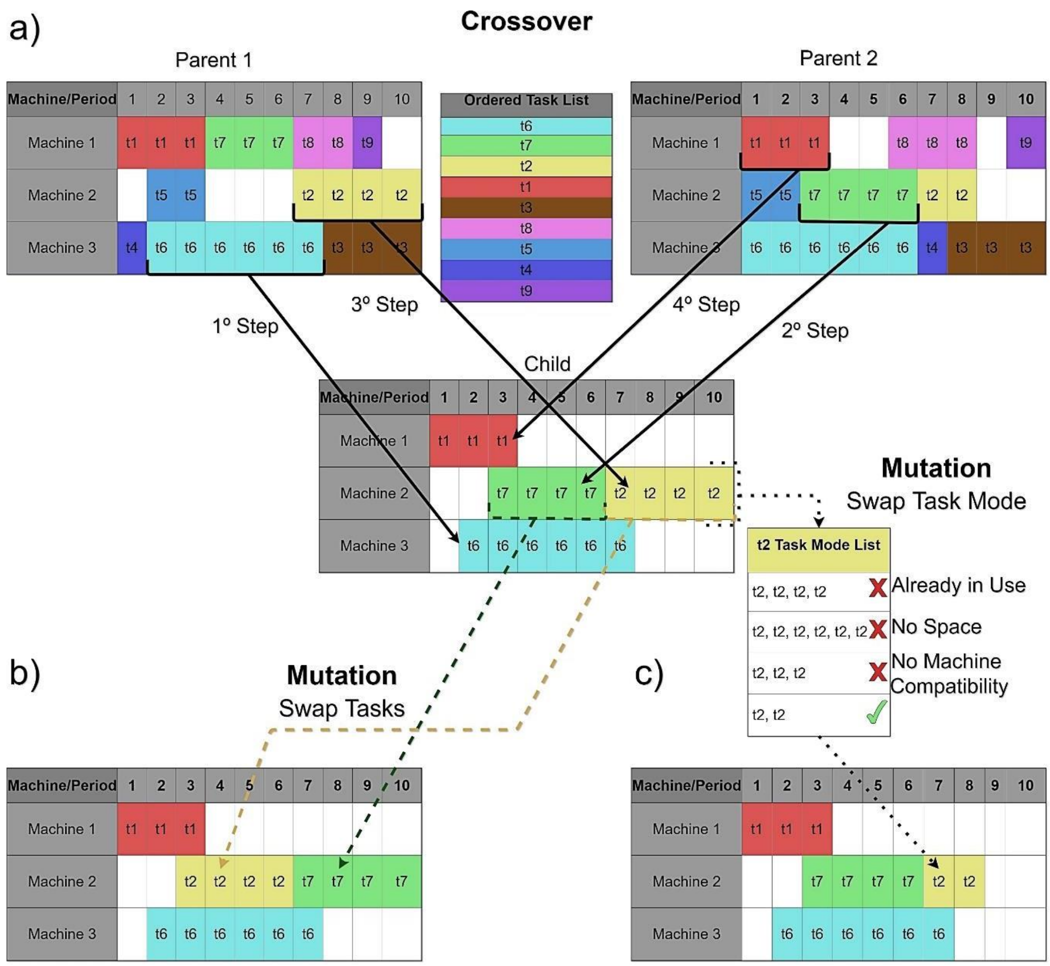

Figure 10 – Individuals' crossover and mutation. (a) Example of the first 4 steps in a crossover, starting from parent 1; (b) example of a swapping tasks mutation; (c) example of a swapping the task mode on a task mutation [35]

After each crossover, where there are no incompatibilities between parent 1 coordinates and parent 2 coordinates, the resulted child is evaluated in terms of respecting all the imposed constraints. If at least one of the imposed constraints is not respected, the crossover is considered invalid.

In case of an invalid crossover, a perfect copy of either parent 1 or parent 2 is done to the child. This is done since afterward, in the selection phase, all the repeating individuals are eliminated, thus removing the child from the population pool. In short, if an invalid crossover occurs the child is never added to the population.

The most common approach to a crossover in a matrix is separating the parents based on a cutting point(s), in either a column or row manner, in other words, a probabilistic strategy. However, due to the problem at hand, many problems would arise such as task splitting, a high overlap of tasks, between the parents, and inconsistencies (i.e., tasks can have different processing times on each parent, due to task modes). Therefore, it was necessary to use a more deterministic approach to solve these issues.

4.1.1.4 Mutation

Using the population obtained from the crossover, a mutation is performed. In this procedure, the algorithm takes each individual, and according to the percentage of mutation defined in the input data, it decides whether or not the mutation will be applied to a given individual. However, if the mutation is applied, one of three different types, using equal probabilities, is randomly chosen:

- Swapping tasks, changing two tasks in order of execution and/or machine. Figure 10b represents an example of a swapping tasks mutation;

- Swapping the task mode on a task, affecting energy consumption and processing time. Figure 10c represents an example of swapping the task mode on a task mutation;

- The combination of both swapping tasks and task mode.

If a mutation leads to an invalid individual (i.e., not respecting a certain constraint) then the mutation is reversed and another mutation, affecting different tasks in the schedule, is tried. Nevertheless, if no valid individual can be reached with a mutation, after a short number of tries, no mutation is done.

4.1.1.5 Selection

The selection phase begins with the union of the new and the old populations, that is, the crossed and mutated population with the initial population of the previous generation. Additionally, repetitions of individuals are eliminated.

Afterward, each individual is evaluated according to their fitness score, which follows a multi-objective function that minimizes the overall total costs, maximizes profit from selling energy, and minimizes machine occupancy deviation.

The first two objectives (i.e., minimize costs and maximize profit) can be joined into a single function which is divided into four fundamental equations: period energy consumption, period energy to pay, period maintenance to pay, and total cost.

The Period Energy Consumption (PEC), represented by $PEC_{Demand(p)}$, gives the total energy consumed by the tasks in a given period p, it can be described by eq. (1).

$$PEC_{Demand(p)} = \sum_{m=1}^{M} E_{Demand(p,m)} \times P_{Machine(p,m)} \tag{1}$$

The variable p portrays a specific period, m describes a machine index, and M the total number of available machines for production. Variables p and m can be compared to the x (i.e., column) and y (i.e., row) cartesian coordinates, respectively, in order to navigate in the individual matrix. The energy consumption of a machine m in period p is described by $E_{Demand(p,m)}$. Furthermore, if a machine priority constraint is applied, variable $P_{Machine(p,m)}$ represents the priority of machine m in period p. For $P_{Machine}$ values above 1, the priority is decreased, while below 1 it is increased, as a result, neutral priority (i.e., no priority associated) is represented by the value 1.

Regarding the Period Energy to Pay (PEP), portrayed as $PEP_{Demand(p)}$, it represents the energy to pay (i.e., energy cost) in a given period p, it is represented by eq. (2).

$$PEP_{Demand(p)} = \begin{cases} PEP_{Demand(p)} = 0, \quad if\ E_{Generation(p)} = PEC_{Demand(p)} \\ PEP_{Demand(p)} = \left(E_{Generation(p)} - PEC_{Demand(p)}\right) \times E_{Selling\ Price(p)}, \\ \quad if\ E_{Generation(p)} > PEC_{Demand(p)} \\ PEP_{Demand(p)} = \left(PEC_{Demand(p)} - E_{Generation(p)}\right) \times E_{Buying\ Price(p)} \end{cases} \tag{2}$$

Variable $E_{Generation(p)}$ portrays available locally generated energy that is free of charge (e.g., PV generation) in period p, $E_{Selling\ Price(p)}$ describes the price for selling energy in period p, and $E_{Buying\ Price(p)}$ represents the price for buying energy in period p. In case PEP_{Demand} results in a positive value (i.e., above zero), it indicates that there are energy costs to be paid, while negative values (i.e., below zero) indicate that profit was made by selling generated energy in excess (i.e., all the PEC_{Demand} was covered by $E_{Generation}$ and the rest sold to energy buyers). Subsequently, zero indicates that there are no energy costs to be paid and no profit was obtained from generated energy in excess.

To calculate the maintenance costs, it is used the Period Maintenance to Pay (PMP), represented by $PMP_{Maintenance(p)}$ it portrays the maintenance costs to pay in a given period p, it is represented by eq. (3).

$$PMP_{Maintenance(p)} = \begin{cases} PMP_{Maintenance(p)} = 0, \quad if\ there\ is\ no\ maintenace\ scheduled \\ PMP_{Maintenance(p)} = M_{In\ Hours\ Price(p)}, \\ if\ there\ is\ a\ maintenace\ scheduled\ in\ maintenance\ hours \\ PMP_{Maintenance(p)} = M_{Out\ Hours\ Price(p)} \end{cases} \quad (3)$$

Maintenances can be scheduled either in maintenance hours (i.e., the interval of periods in which the maintenance must/can be done) or out of maintenance hours (i.e., not in the stipuled interval of periods, normally has a monetary penalty). Accordingly, variable $M_{In\ Hours\ Price(p)}$ describes the price of a maintenance activity done in maintenance hours in period p, while $M_{Out\ Hours\ Price(p)}$ represents the maintenance price of a maintenance activity done out of maintenance hours in period p.

Finally, the Total Cost (TC) of an individual can be obtained through eq. (4).

$$TC = \sum_{p=1}^{P} PEP_{Demand(p)} + \left(\sum_{m=1}^{M} PMP_{Maintenance(p)}\right) \quad (4)$$

The total number of available periods in the time window of the schedule is represented by the variable P. Also, eq. (4) can be seen as the sum of the energy cost of each individual, determined as a result of the energy balance ($consumption - generation$) multiplied by the respective energy price, and the maintenances cost according to their respective maintenance hours price.

It is noteworthy that, variable $PEP_{Demand(p)}$ already includes the energy costs from all the machines in a given period p. However, variable $PMP_{Maintenance(p)}$ does not include all the maintenance costs from period p, thus the need to incorporate, in eq. (4), the sum of all maintenance costs from all the machines in period p.

To minimize machine occupancy deviation (i.e., machine occupation rates standard deviation), and thus maximize machine longevity by reducing overload and usage of single machines, it is employed a function that can be divided into the following three equations: machine degradation classifier, machine occupation rate, and occupation standard deviation.

To calculate the machine occupancy deviation, only tasks and setups are considered to influence the degradation of a machine, since they require the machine to be working. Therefore, the classification of factors that contribute to the degradation of a machine is done using the Machine Degradation Classifier (MDC), represented in eq. (5) as $MDC_{Factor(p,m)}$, which classifies a factor contributing to the degradation of a machine m in period p with the value 1, otherwise, it classifies it with 0.

$$MDC_{Factor(p,m)} = \begin{cases} MDC_{Factor(p,m)} = 1, & if\ there\ is\ a\ task\ or\ setup\ scheduled \\ & MDC_{Factor(p,m)} = 0 \end{cases} \tag{5}$$

The Machine Occupation Rate (MOR), portrayed by $MOR_{Factors(m)}$, gives the occupation rate of factors that contribute to the degradation in a given machine m, it can be described by eq. (6).

$$MOR_{Factors(m)} = \frac{\sum_{p=1}^{P} MDC_{Factor(p,m)}}{P} \tag{6}$$

Lastly, the Occupation Standard Deviation (OSD) of an individual can be determined by calculating the population standard deviation (i.e., not the sample standard deviation), as represented in eq. (7).

$$OSD = \sqrt{\frac{\sum_{m=1}^{M} \left(MOR_{Factors(m)} - \left(\frac{\sum_{m=1}^{M} MOR_{Factors(m)}}{M} \right)^2 \right)}{M}} \tag{7}$$

After obtaining both the TC and OSD for each individual a Min-Max normalization approach is taken, using the results obtained from the individuals in the population, to normalize the TC and OSD values of each individual in the population. Then, each individual is evaluated according to the Fitness Score (FS), described by eq. (8).

$$FS = TC_{Norm} \times W_{TC} + OSD_{Norm} \times W_{OSD} \tag{8}$$

Variables TC_{Norm} and OSD_{Norm} describe the normalized TC and OSD values, respectively, of an individual. The optimization weights, defined by the user in the input data, for the overall costs (i.e., TC) and machine occupancy deviation (i.e., OSD) are represented by variables W_{TC} and W_{OSD}, respectively. The sum of variables W_{TC} and W_{OSD} must always be 1, and they must assume a value from 0 to 1, inclusive.

The selection of the n best individuals is made according to the input parameter of the algorithm, chosen by the user. The remaining individuals (i.e., population size less n) are obtained from non-elite tournaments. Each tournament consists of two individuals randomly chosen, where they compete based on their fitness scores (i.e., FS). The algorithm calculates the chance of individual 1 winning the tournament using eq. (9).

$$Individual_{chance}^{1} = 1 - \frac{fit1}{fit1 + fit2} \tag{9}$$

where $fit1$ and $fit2$ represent the fitness of individual 1 and individual 2, respectively. Then, a random decimal number between 0 and 1 is generated. If the generated decimal is lower than the chance of individual 1 winning, eq. (9), then individual 1 is declared the winner. Otherwise, individual 2 leaves victorious. Therefore, the individual with the lowest fitness, which in turn has the lowest combination of overall cost and machine occupancy deviation, is the one most likely to be chosen.

4.1.1.6 Extract Best Individual

At the end of the selection phase, the next generation begins with the population obtained from this selection, and all the procedures mentioned above are repeated. Finally, after a stop condition is met, the best individual is extracted from the last population of the genetic, that is, the lowest balance of cost and machine occupancy deviation job shop schedule, generated by the GA. Thus, in general, the GA, through eq. (8), which represents the multi-objective function, tries to minimize the overall cost and machine occupancy deviation of the elaborated job shop schedule as much as possible, taking into consideration constraints, such as task orders, task collisions, energy limits, request deadlines, PV generation, etc.

4.1.1.7 Cost Optimization

After the execution of the GA, a post-processing phase is required to adjust/further optimize the result given by the GA through a deterministic approach.

The cost optimization function will analyze the result of the GA, and it will identify periods with lower energy costs that do not have an assigned task, then it will try to place the tasks with higher energy consumption to the lower energy cost time slots. If tasks can be exchanged while resulting in a cost reduction and respecting all the constraints, the exchange is carried out. This optimization function uses the energy cost resulted from the prices of all energy sources available, including local renewable energy sources.

It is worth noting that this deterministic optimization only affects tasks scheduled in the same machine, and as such, tasks are never swamped between machines. Therefore, this optimization does not interfere with the machine occupancy deviation optimization done by the GA.

4.1.1.8 Shift Optimization

Shift optimization is made by sliding tasks and/or maintenance activities in time to reduce the time between them. This optimization differs from the cost optimization because no task/maintenance will be changed, only slide to near periods. Its main purpose is to join tasks and maintenances, reducing the time between them, and increase space for future unexpected product requests and maintenance activities. For each isolated task/maintenance (i.e., with empty periods on the left or on the right), the shift optimization will try to shift it to the left, and if all constraints are valid and the overall costs decrease or are maintained within the shift margin constraint level, the shift is performed. If the left shifting is not possible, then the shift optimization tries to shift the task/maintenance to the right, applying the same conditions described before. Figure 11 shows an example of the shift optimization performed to a machine work plan.

This optimization also only swaps scheduled tasks/maintenances in the same machine, thus it does not interfere with the machine occupancy deviation optimization done by the GA.

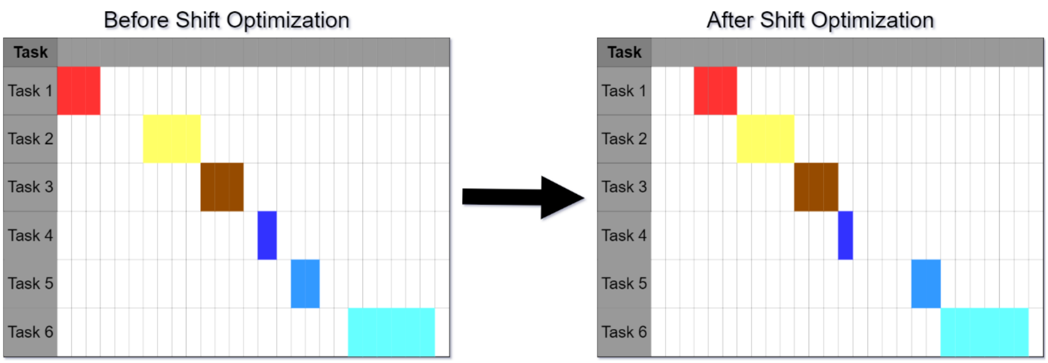

Figure 11 – Example of the shift optimization for a machine work plan [35]

4.1.2 Demand Response Participation and Machine Breakdown Events Using Production Line Flexibility

The proposed algorithm enables the use of flexibility in production lines for DR participation and MB events. To use the flexibility of the production line, three main parts are needed: data processing, genetic algorithm, and determinist optimizations. This process is similar to the one presented in Figure 8, with the addition of the block of flexible data processing introduced to enable the use of flexibility in the scheduling optimization. Figure 12 shows how flexibility is used in the proposed methodology.

The flowchart of Figure 12 is only applied to schedule plans that were already resulted from the proposed algorithm of section 4.1.1. For plans that did not already start, the DR event can be integrated directly into the optimization using the energy limit constraint, or in the case of an MB event, the machine available frames constraint. For plans that had already started being executed, then we need to apply the flowchart of Figure 12.

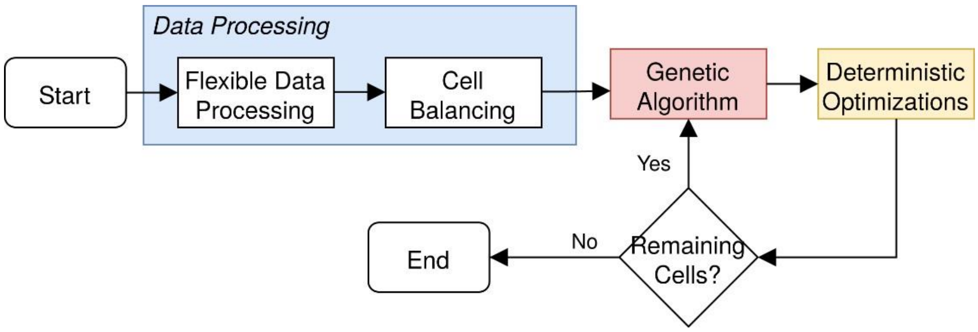

Figure 12 – Flowchart for flexibility usage [35]

In the flexible data processing module, the already-made schedule is divided into two segments: one that represents the schedule with the tasks that were already started or executed, and a second segment with the schedule of tasks that are planned but not yet started. The second segment is then parsed to create an input for the GA, taking into account the tasks and maintenance activities already executed in the first schedule segment and updating all the constraints. However, before the GA execution, the proposed solution applies the cell balancing method described in section 4.1.1.1.

The GA is executed with an additional constraint: energy limit for DR participation, or machine available frames for MB events. The first constraint defines a limit within a given interval of periods of energy we can get from the energy provider that had launched the DR event. Considering this constraint, the GA is able to produce a result compliant with the DR event. The other constraint, machine available frames, is able to limit the number of periods a machine can operate, and if set to zero it indicates that the corresponding machine is no longer available for production. Accordingly, with this constraint, the GA is able to produce a schedule that complies with the MB event.

The same cost and shift optimizations, described in sections 4.1.1.7 and 4.1.1.8, are applied in the result of the GA. Finally, the new schedule is attached to the schedule of the first segment, with the tasks and maintenance activities already started or executed, and the final schedule is presented to the user.

4.1.3 Predictive Maintenance to Minimize Machine Breakdowns

The training process for the PdM system, either to predict or detect MBs, is divided into three fundamental components: data acquisition, data preprocessing, and machine failure status training. Figure 13 represents the flowchart of the proposed PdM training process. Also, the training process can be done in batches, mini-batches, or continuous data streaming. In an initial phase, the batch approach is taken since it allows to have a model ready to be applied in the real world. Nevertheless, after the initial model is constructed the training process is made through mini-batches or data streaming, in real-time.

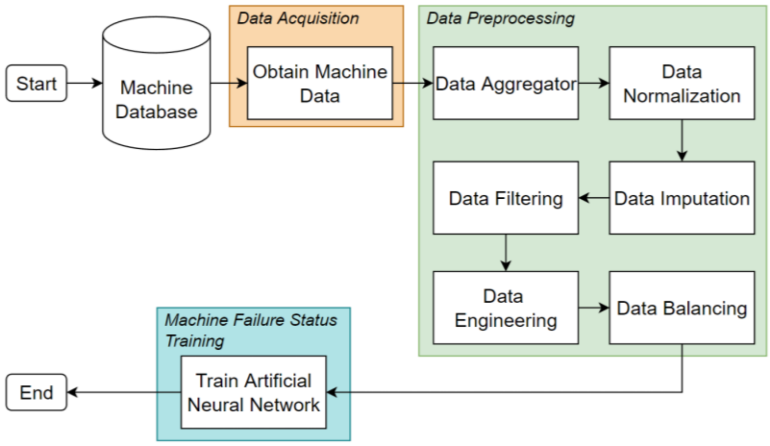

Figure 13 – Flowchart of the proposed predictive maintenance training process [38]

4.1.3.1 Data Acquisition

For real-time training, the proposed solution starts by obtaining the newest machine data (i.e., air temperature, machines' process temperature, rotational speed, torque, tool wear, and machine failure status information) from the machine database employed in the facility. The database can describe either the culmination of all machine data from all the machines in the facility, or the data of a single machine.

4.1.3.2 Data Preprocessing

Before training starts, a data preprocessing phase begins in which:

1. **Data aggregator** — aggregate all the data collected into a single data file, if the data is separated into different files;

2. **Data normalization** — normalize data scales and types, primarily between data from two different machines, using a Min-Max strategy with the MinMaxScaler method [158] from the Scikit-Learn library;

3. **Data imputation** — fill in missing values on the gathered data, by using a k-Nearest Neighbors imputation technique with the KNNImputer method [159] from the Scikit-Learn library;

4. **Data filtering** — remove possible irrelevant and erroneous data, by detecting outliers using the Z-score technique through the SciPy stats Z-score function [160] from the SciPy library [32];

5. **Data engineering** — transform raw data into features that better represent the underlying problem;

6. **Data balancing** — balance machine data failure and non-failure points, by using the imbalanced-learn [161] library.

Afterward, the preprocessed data is fed to an ANN for training, where the model's neuron weights are adjusted due to the back-propagation feature. This training process can start every time there is new data in the machine database.

The initial model for the ANN is constructed using:

- The dataset described in section 3.4.1.3;

- The Holdout method, 80% for training and 20% for testing;

- A newly added dataset feature, machine temperature difference (i.e., process temperature − air temperature);

- A data balancing method, 5% oversampling on failure data and a majority undersampling strategy on non-failure data. To achieve this, it is used the imbalanced-learn [161] library.

4.1.3.3 Machine Failure Status Training

For machine failure status training, an ANN was used due to its robustness, performance, and backpropagation feature. It was trained using an automatic hyperparameter optimizer, which finds the optimal hyperparameter values to obtain a high-performing model. This is achieved by using the GridSearchCV [184] method available in the Scikit-Learn [31] library. The automatic hyperparameter optimizer works by exploring each hyperparameter's possible values, at random, in order to find a high-performing ANN model, which contains the optimal values for each hyperparameter. A 5-fold cross-validation splitting strategy to search for the best hyperparameters is used. Table 6 presents the possible and found optimal hyperparameter values for the ANN model.

Table 6 — Artificial neural network hyperparameters possible and optimal values using the automatic hyperparameter optimizer method

Hyperparameter	Possible Values	Optimal Value
Batch Size	10, 20, 40, 60, 80, 100, 200, 500, 1000, 2000, or 5000	5000
Dropout Regularization on Hidden Layers	0%, 5%, 10%, 20%, 30%, 35%, 40%, 50%, 60%, 70%, 80%, or 90%	35%
Dropout Regularization on Input Layer	0%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, or 90%	0%
Epochs	10, 50, 100, 150, 200, 300, 500, 1000, 2000, 5000, 8000, or 10000	5000
Neuron Activation Function	Softmax, Softplus, Softsign, Relu, Tanh, Sigmoid, Hard Sigmoid, or Linear	Relu
Neurons per Hidden Layer	1 to 4 layers 5, 10, 15, 20, 25, or 30 neurons per layer	Layer 1 – 25 neurons Layer 2 – 20 neurons Layer 3 – 15 neurons Layer 4 – 15 neurons
Optimizer	SGD, RMSprop, Adagrad, Adadelta, Adam, Adamax, or Nadam	Nadam
Weight Initialization in Input Layer	Uniform, Lecum Uniform, Normal, Zero, Glorot Normal, Glorot Uniform, He Normal, or He Uniform	Glorot Uniform

However, some hyperparameters were predefined, as there was no need to find the optimal value, such as:

- **Loss function** – Binary Crossentropy;
- **Metrics** – Binary Accuracy;
- **Number of input neurons** – 4 (temperature difference, rotational speed, torque, tool wear);
- **Number of output neurons** – 1 (machine failure);
- **Output layer activation function** – Sigmoid;
- **Weight initialization in hidden layers** – Normal.

For the ANN classifier it was used the KerasClassifier [185] from the Keras [30] library. The classifier works by creating rules during the training phase to achieve the lowest possible accuracy error in contrast to the training classes. The model is ready to generate predictions once it has been properly fitted using training data.

4.1.3.4 Real-time Application

The application in real-time of the proposed PdM model, which is represented in Figure 14, can be divided into three crucial sequential phases:

1. **Data acquisition** — obtains the necessary machine data from the machine to be evaluated. The necessary data can be obtained either directly from the machine (i.e., present data) or from a third-party system that provides forecasted machine data (i.e., forecasted data from a predefined time horizon);
2. **Data preprocessing** — applies data normalization, imputation, filtering, and engineering on the obtained machine data;

3. **Machine failure status prediction** — uses the ANN to predict the machine failure status (0 for non-failure and 1 for failure).

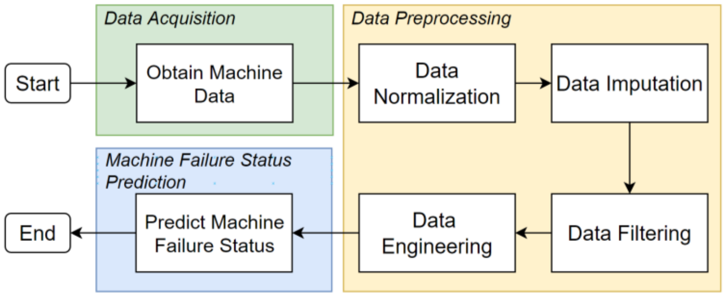

Figure 14 – Flowchart of the proposed artificial neural network application in real-time [38]

4.1.4 Security

The proposed system is interconnected with others systems in the manufacturing environment through IoT by using protocols such as HTTP. In fact, the scheduler/rescheduler acts as a REST API server whose function is to receive scheduling requests and provide an optimized production plan. As a consequence, there are a variety of vulnerabilities that could compromise not only the proposed system but also other systems to which the proposed system has direct access. The most common attacks in manufacturing systems are "Logic Bomb", Denial-of-Service (DoS), Distributed Denial of Service (DDoS), Man-in-the-Middle (MitM), Phishing, and Password attacks [186]–[188]. Therefore, cybersecurity in manufacturing systems is crucial to mitigate not only cyber-attacks but also safety threats [189]. For instance, in case the proposed system was to be connected directly to machines and had full access to controlling them, there are critical issues that could occur if an attacker gained access, such as:

- Making machines run at maximum speeds, leading to early machines breakdown;
- Making machines do different tasks every second or unexpectedly changing tasks, leading to possible worker accidents;
- Manufacture random/defective products, wasting stock and energy;
- Stop production lines, wasting time;
- Create unnecessary maintenance activities, wasting time and money.

To address security vulnerabilities in manufacturing systems, there are a wide range of available approaches for detecting and preventing cyber-attacks. In [190] it is proposed a kinetic cyber-attack detection system for manufacturing systems that operate in physical environments. It uses a novel attack detection method that is able to reach accuracies of 77.45%. Furthermore, the system in [191] proposes a cyber-physical vulnerability assessment system for intelligent manufacturing systems. It provides manufacturers with a comprehensive report of identified cyber-physical vulnerabilities and their respective impacts. Also, in [192] it is proposed a unified framework approach to manage both cybersecurity issues as well as safety in manufacturing environments. It provides a high-performance system for detecting and mitigating cyber-attacks and safety threats.

5 Results and Discussion

This section presents the results and discussion of two case studies that highlight the benefits of the proposed solution. From the first case study, eight different scenarios are explored. The first demonstrates the energy cost optimization capabilities of the scheduler while the second adds an energy buyer in order to validate how the scheduler adapts to energy selling. The third one shows the benefits of effectively considering machine occupancy deviation in scheduling to improve machine longevity. Then, in the fourth scenario, the capabilities of participation in DR events by the scheduler are explored. Scenario five validates the performance of the PdM system for industrial usage. The validation of maintenance and production optimization is done in scenario six, while scenario seven simulates an MB event. The final scenario combines all the features shown in the previous scenarios, from production and maintenance scheduling to imposed constraints. Regarding the second case study, it validates the high adaptability of the proposed scheduler for other applications, in this case, for residential load shifting. It is composed of three scenarios, each with a different level of flexibility, demonstrating how the proposed scheduler is able to adapt to these contexts.

5.1 Validation Methodology

To validate the proposed solution, two methodologies are considered, one to validate the proposed scheduler/rescheduler (i.e., GA) capabilities, and another to explore the performance of the proposed ML model (i.e., ANN).

Since there are no available systems capable of achieving the objectives of the proposed GA scheduler/rescheduler, and thus no comparable systems, a scenario-based analysis approach was instead considered. This approach allows validating the proposed solution without the need to compare it to other approaches/methods. Furthermore, to better simulate the uncertainties of the real world and provide a more robust validation, all scheduling scenarios use real production data, provided by a textile company that manufactures hang tags. The main methods used to validate the proposed scheduler are:

- Overall energy cost reductions, through the comparison of business as usual scenarios to optimized scenarios;
- Energy consumption throughout the production plan, by analyzing graphs that provide the energy consumed from the retailer, energy consumed from generated energy sources (e.g., PV), energy not consumed and thus sold, and the overall energy consumption;
- Energy cost throughout the production plan, by analyzing graphs that provide the energy cost paid for each retailer and the difference in energy costs from business as usual and optimized scenarios;
- Valid imposed constraints, by analyzing Gantt diagrams of the production plan for constraints that are not being complied with;
- Machine occupation balance, through graphs that provide the machine occupation rates;
- Intelligent maintenance activities scheduling, by analyzing how the proposed scheduler was able to minimize the impact of maintenance activities in the manufacturing process;
- Fitness function evolution, through the usage graphs that show the fitness convergence per generation of the genetic generation.

It is noteworthy that, to further facilitate the evaluation of the system, Excel [193] templates were developed that automatically, from the output data of a production plan, generate graphs, charts, and tables that facilitate the reading and analysis of the production plan.

Regarding the proposed ANN, input data can be easily adapted to other ML or statistical models. Therefore, a model performance comparison approach was considered to validate the proposed ANN. The ML models used to compare were GB, SVM, and RF. To evaluate the performance of the models, which are for classification, the following metrics and methods are used: accuracy, F1 score (including precision and recall), and confusion matrix.

All the validation was done on a computer with an AMD® Ryzen 7 3700X processor running at 4.05 GHz using 32 GB of RAM, running Windows 10 Home version 21H1. The only exceptions are scenarios 1.1 and 1.4 from case study 1 (i.e., sections 5.2.1 and 5.2.4), which were executed on a computer with an Intel® Core™ i3-7020U processor running at 2.30 GHz using 8 GB of RAM, running Windows 10 Home version 2004.

5.2 Case Study 1 – Industrial Production and Maintenance Optimization

The first case study of this dissertation uses real production data, provided by a textile company that manufactures hang tags. Their working schedule is from 7:00 to 23:00 each working day, from Monday to Saturday. The case study uses a period of 5 min for all tasks' energy profiles and maintenance durations, thus an overall time window of 1152 (i.e., each day is composed of 192 periods). The case study considers six working days, Monday to Saturday, from 7:00 to 23:00 each day. The scheduling algorithm was used for three machines that share the same cell, thus each individual matrix, in the GA, is 3 rows by 1152 columns. Table 7 describes the machines according to the tasks they are able to execute.

Table 7 — Tasks that each machine executes [35]

Task	Machine		
	118	119	120
Anti-Shrinkage			X
Harden [0.5]	X	X	
Harden [1]	X	X	
Harden [1.5]	X	X	
Harden [2]	X		
Ironing	X	X	X
Sublimation			X

The case study used four constraints: time leap, task order, task collision, and product request deadline. The time leap constraint was applied in every period that represents a day transition, time periods 192, 384, 576, 768, and 960, which corresponds to the final time period in each day (23:00 of Monday, Tuesday, Wednesday, Thursday, and Friday). A task order constraint where every task "Harden [1.5]" must be preceded by a task "Harden [2]", a task collision constraint between tasks "Sublimation" and "Harden [2]", and a product request deadline, in a product, of time period 960, which corresponds to a deadline of 23:00 of Friday.

During the week, 132 units of 14 products were requested, amounting to 275 tasks to be scheduled by the proposed GA.

The market prices considered are MIBEL (Iberian Electricity Market) [180] prices from 7 January 2019 to 12 January 2019. The local renewable generation profile was obtained from 6 June 2020 to 11 June 2020 in a 3 kW peak PV installation in Portugal.

For all the scheduling scenarios of the present case study, the GA was executed using a population size of 20, an elite size of 3, and a mutation rate of 5%.

It is worth noting that scenarios 1.1 and 1.4 (i.e., sections 5.2.1 and 5.2.4) are from a published paper belonging to the author of the dissertation, which is available at [35]. In addition, scenario 1.5 (i.e., section 5.2.5) is based on two papers presented at conferences, one at the SOCO 2022 conference [39], "Predictive Maintenance for Maintenance-effective Manufacturing using Machine Learning Approaches" [38], and another at the ICEER 2022 conference [40], “Machine Learning applied to Industrial Machines for an Efficient Maintenance Strategy: a Predictive Maintenance Approach”.[1]

5.2.1 Scenario 1.1 – Energy Cost Optimization

To validate the scheduler energy cost optimization capabilities, the algorithm was executed in 2 hours, which for this scenario averages around 1500 generations, with a total cost optimization weight of 1 and 0 for machine occupancy deviation. Its main results can be seen in Figure 15 where tasks are distributed by the available machines and along the time periods available. Due to size limitations, it was not possible to discriminate all the tasks names; nonetheless, the complete open data is published in [194]. Machine occupation rates of 59.72%, 81.77%, and 79.86% were reached for machines 118, 119, and 120, respectively.

[1] The paper “Machine Learning applied to Industrial Machines for an Efficient Maintenance Strategy: a Predictive Maintenance Approach” has been presented and will be published in a journal, however, at the time the present dissertation was concluded the paper has not yet been published in a journal.

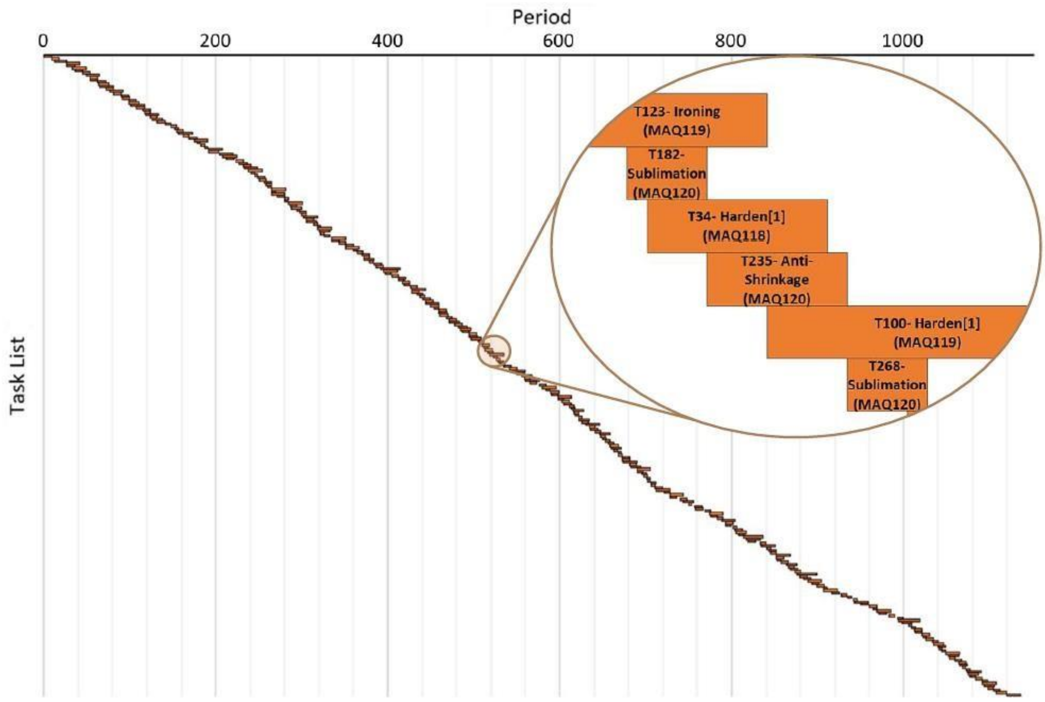

Figure 15 – Gantt diagram of the scheduled production plan for the energy cost optimization scenario [35]

Figure 16 shows the consumption aggregation of production profiles considering the result of the scheduling optimization. Energy consumed is obtained through the algorithm, which provides stats. It is noteworthy that the higher the energy price in a given period the lower the energy consumed, indicating that the algorithm was able to shift the tasks to lower energy price periods. The local energy generation, provided by PV panels, was used as much as possible by the algorithm, reducing the demand for external supplies with higher energy prices.

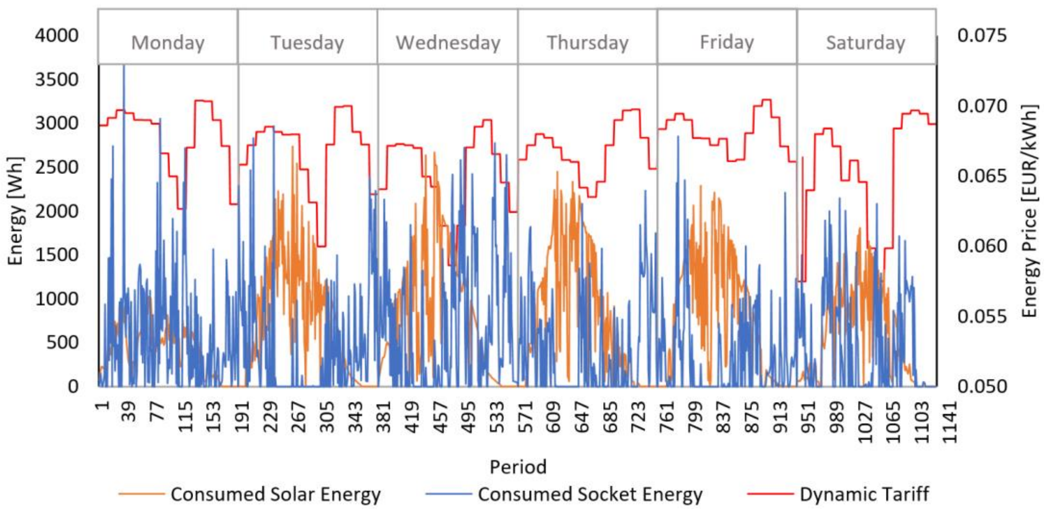

Figure 16 – Energy consumption with energy costs for the energy cost optimization scenario [35]

The overall performance of the algorithm can be seen in Figure 17 where the best individual of each generation is presented according to its overall cost. The results follow a decreasing logarithmic function, showing good performance. The total energy costs for the scenario, including the deterministic optimizations, is 36.42 EUR.

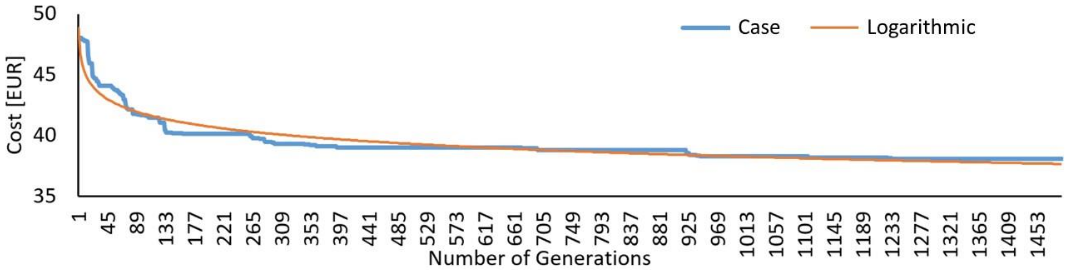

Figure 17 – Cost per generation of the genetic algorithm for the energy cost optimization scenario [35]

It is noteworthy that, this scenario is available in the published paper [35].

5.2.2 Scenario 1.2 – Energy Cost Optimization with Energy Selling

An energy cost optimization considering the presence of an energy buyer is proposed to validate the scheduler's ability to maximize profits while also minimizing energy costs. The scenario uses the same data as the scenario described in section 5.2.1, but with an added sales value corresponding to 50% of the buying. For this scenario, the GA was executed for 2 hours, with 1 and 0 for the optimization weights total cost and machine occupancy deviation, respectively. All the data is available at [195].

The energy consumed per source, energy sold, and the respective buying and selling tariffs, are presented in Figure 18. The obtained results show that the local energy generation, provided by PV panels, was utilized not only to cover the energy consumed by tasks but also to turn some profit by selling PV energy in excess. Furthermore, during periods of high energy prices, which in turn makes selling energy more compelling, and with at least some PV availability, tasks were shifted away from these periods to periods with lower energy costs. As a result, the algorithm was able to increase profits, by selling energy, and also reduce energy costs.

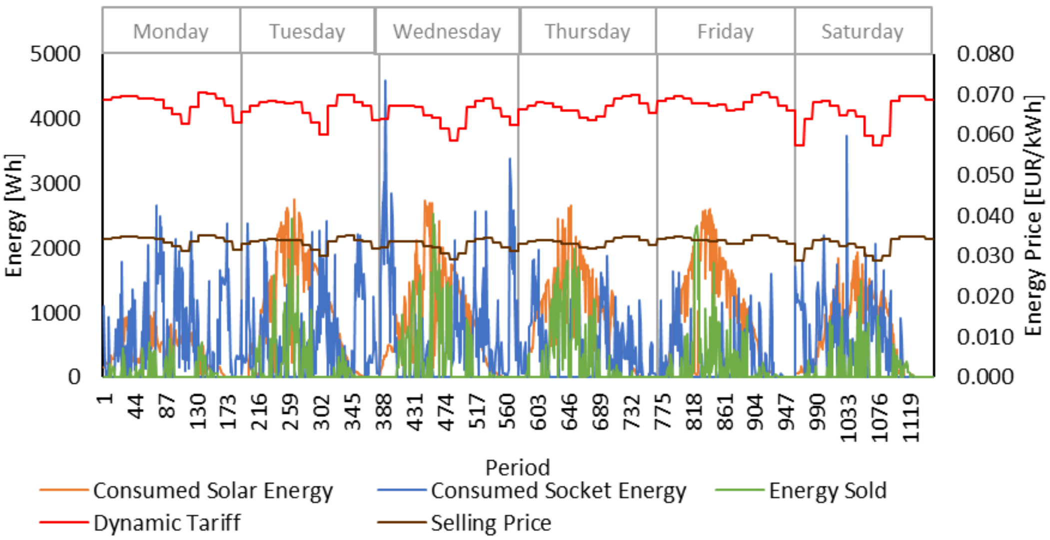

Figure 18 – Energy consumption, energy sold, and energy buying and selling prices for the energy cost optimization with energy selling scenario

The energy cost of the best individual per generation of the GA is shown in Figure 19. The present scenario demonstrates very good evolution performance since it almost resembles a decreasing logarithmic function. Compared to the first scenario, with the addition of an energy buyer, it was possible to decrease the energy costs by 24.6%, from 36.42 EUR to 27.46 EUR from the previous and present scenarios, respectively.

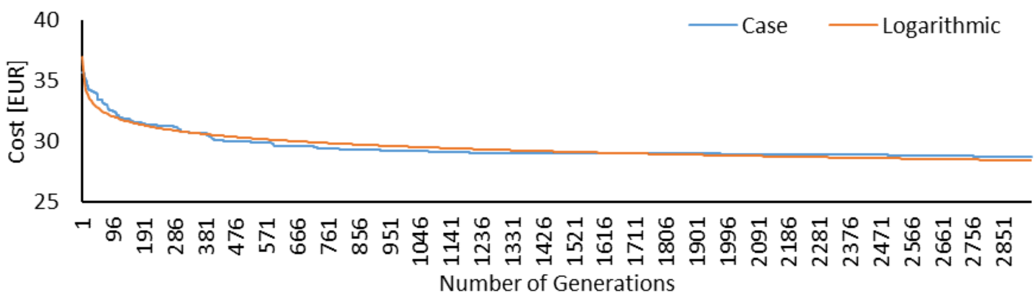

Figure 19 – Cost per generation of the genetic algorithm for the energy cost optimization with energy selling scenario

5.2.3 Scenario 1.3 – Total Cost and Machine Occupancy Deviation Optimization

The present scenario replicates the first scenario (i.e., from section 5.2.1) with the exception of the optimization weights. It demonstrates how the proposed scheduler is able to balance tasks between machines while also reducing overall costs. For this scenario, it was considered an optimization weight of 0.5 for both the total costs and machine occupancy deviation objectives and the GA was executed for 2 hours. The complete data is available at [196].

The machine occupation rates without the machine occupancy deviation optimization (i.e., from the scenario in section 5.2.1) and with such optimization (i.e., from the present scenario) are presented in Figure 20. As a result, without the machine occupancy deviation optimization, there is a machine occupancy rate standard deviation of around 0.0997% while with the optimization it achieves 0.0122%, a decrease of 87.8%.

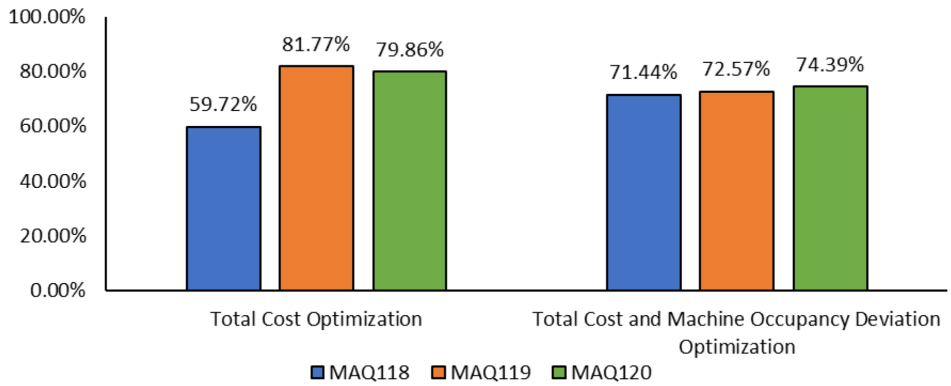

Figure 20 – Machine occupation rates per machine without and with the machine occupancy deviation optimization

On the other hand, the total cost went up by 12.8%, from 36.42 EUR to 41.07 EUR. This clearly demonstrates that the proposed scheduler is able to improve machine longevity, by reducing overload and usage of single machines, while also balancing it with the total cost, by not allowing a drastic increase in the overall costs.

The total cost and machine occupancy rate deviation of the best individual of each genetic generation, for this scenario, are represented in Figure 21. It shows an average performance regarding the total cost evolution since it follows a decreasing logarithmic function for the first 1000 generations, but it also shows how the scheduler is able to intelligently balance these objectives, by decreasing or increasing the total costs and machine occupancy rate deviation.

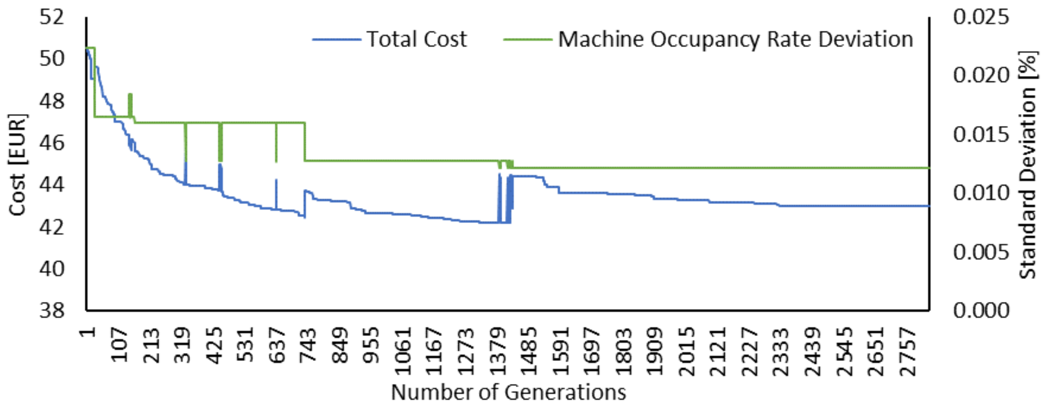

Figure 21 – Total cost and machine occupancy deviation per generation of the genetic algorithm for the total cost and machine occupancy deviation optimization scenario

5.2.4 Scenario 1.4 – Demand Response Participation

Using the first scenario's data as the starting point (i.e., from section 5.2.1), an announcement of a DR program was simulated on Friday at 6:00, describing a DR event from Friday at 21:00 (i.e., period 937) to Friday at 23:00 (i.e., period 960), where each period represents five minutes. The DR program imposed a limit consumption, during its event, of 2.5 kWh. The announcement of the DR allowed the use of the proposed solution to limit energy consumption. For that, the proposed GA was executed at period 769 (i.e., Friday at 7:00) for 1 hour, and with the optimization weights of 1 for total cost and 0 for machine occupancy deviation. Figure 22 shows the machines' energy consumption before and after the DR participation rescheduling of tasks. The full dataset can be seen in [197]. The DR event is represented with an orange rectangle, while the rescheduling consumption is represented by an orange line.

Figure 23 shows the machines' energy consumption difference, considering as base the previously scheduled plan, and it is compared to the new scheduled plan. It is possible to see that during the DR event (i.e., orange rectangle) the consumption decreased to not go above 2.5 kWh.

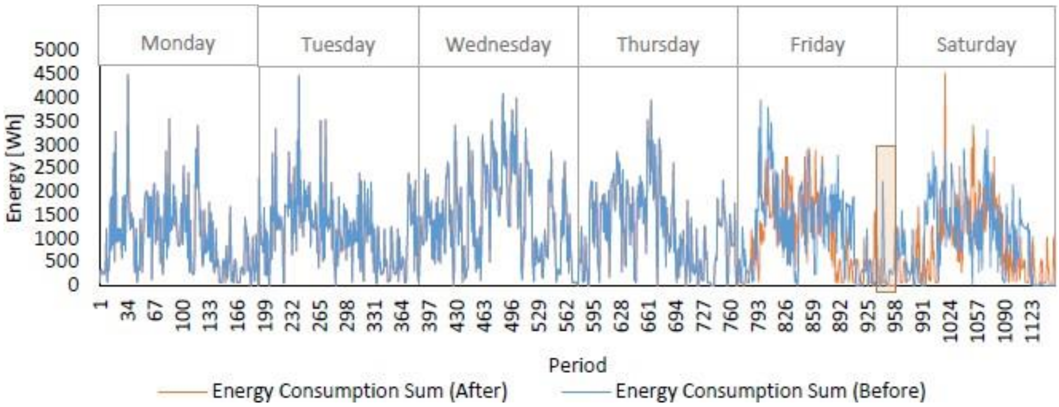

Figure 22 – Energy consumption of the demand response event [35]

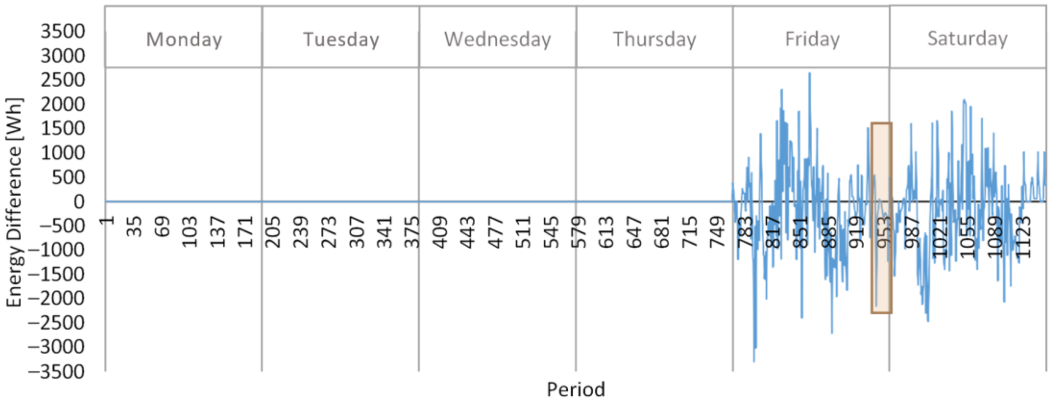

Figure 23 – Energy consumption difference (after – before) of the demand response event [35]

The tasks were rescheduled away from the DR event period. Figure 24 shows a magnification of before and after the DR, where the full before scenario is shown in Figure 15. As can be seen, the algorithm started the DR event with scheduled tasks, but when it reached a total energy of near 2.5 kWh, no more tasks were executed.

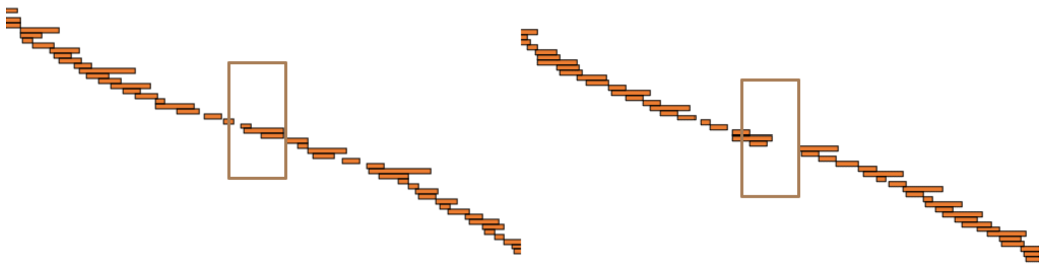

Figure 24 – Before and after Gantt diagrams of the demand response event, respectively, from period 848 to period 1058 [35]

As with the previous scenario, this scenario was also obtained from the published paper available in [35].

5.2.5 Scenario 1.5 – Predictive Maintenance Application

To validate the proposed ANN predictive maintenance capabilities, the three most relevant ML models from the literature were considered for comparison: GB, SVM, and RF. This scenario is based on the presented papers "Predictive Maintenance for Maintenance-effective Manufacturing using Machine Learning Approaches" [38] during the SOCO 2022 conference [39], which contains the GB and SVM models, and the paper "Machine Learning applied to Industrial Machines for an Efficient Maintenance Strategy: a Predictive Maintenance Approach" at the ICEER 2022 conference [40], where the RF and ANN models were implemented and explored.

The implementation of these models for comparison was done using the same methodology, presented in section 4.1.3, and an automatic hyperparameter optimizer, described in section 4.1.3.3, as the proposed ANN. Accordingly, the possible values and corresponding optimal value for each hyperparameter of the GB, SVM, and RF, using the automatic hyperparameter optimizer method, are presented in Table 8, Table 9, and Table 10, respectively.

Table 8 — Gradient boosting hyperparameters possible and optimal values using the automatic hyperparameter optimizer method [38]

Hyperparameter	Possible Values	Optimal Value
Criterion	Friedman MSE, Squared Error, MSE, or MAE	Friedman MSE
Learning Rate	5%, 7.5%, 1%, 25%, 50%, 75%, or 100%	100%
Maximum Depth	10 to 32	12
Maximum Features	Auto, Sqrt, or Log2	Log2
Minimum Samples in a Leaf	1, 2, 4, 6, 8, or 10	4
Minimum Samples to Split	2, 5, 10, 20, or 30	2
Number of Estimators	200 to 3000	2000

Table 9 — Support vector machine hyperparameters possible and optimal values using the automatic hyperparameter optimizer method [38]

Hyperparameter	Possible Values	Optimal Value
C (Regularization Parameter)	0.1, 1, 10, 100, or 1000	100
Gamma	Scale, Auto, 1, 0.1, 0.01, 0.001, or 0.0001	Scale
Kernel	Linear, Poly, RBF, or Sigmoid	RBF
Probability Estimates	True or False	False
Shrinking Heuristic	True or False	True

Table 10 — Random forest hyperparameters possible and optimal values using the automatic hyperparameter optimizer method

Hyperparameter	Possible Values	Optimal Value
Bootstrap	True or False	True
Criterion	Gini or Entropy	Gini
Maximum Depth	10 to 32	10
Maximum Features	Auto, Sqrt, or Log2	Log2
Minimum Samples in a Leaf	1, 2, 4, 6, 8, or 10	1
Minimum Samples to Split	2, 5, 10, 20, or 30	2
Number of Trees	200 to 3000	511

To implement the GB classifier, it was employed the GradientBoostingClassifier [198] from the Scikit-Learn [31] library. For the SVM model, it was used the SVC [199] classifier from the Scikit-Learn library. Lastly, the RandomForestClassifier [200] from the Scikit-Learn library was used as the RF classifier. Furthermore, all three models (i.e., GB, SVM, and RF) used the RandomizedSearchCV [201] technique to achieve an automatic hyperparameter optimization approach.

To validate the proposed ML models' performance four metrics were taken into account: recall, precision, f1-score, and accuracy.

In PdM, the recall, precision, and f1-score metrics are essential to validate the proposed models, since in this type of problem classes are almost always not balanced. As a result, recall allows to have an idea of how many correct predictions (i.e., true positives) to incorrect predictions (i.e., false negatives) are being made by the proposed models. False negatives may have major repercussions until the machine is not detected to have a failure, such as the manufacturing of defective items, resulting in a loss of money, resources, and time. Furthermore, false alarms (i.e, false positives) also need to be considered in order to minimize unnecessary maintenance activities, thus why the precision metric is also taken into account. To consolidate both recall and precision, the f1-score is also used to validate the proposed ML models. The last metric, accuracy, describes the overall ratio of the truly predicted test cases to all the test cases.

The performance metrics for the best GB, SVM, RF, and ANN models, using the optimal hyperparameters found in Table 8, Table 9, Table 10, and Table 6, respectively, and using the dataset described in section 3.4.1.3, are presented in Table 11.

Table 11 — Performance metrics for the gradient boosting, support vector machine, random forest, and artificial neural network models using the optimal hyperparameters

Model	Recall	Precision	F1-score	Accuracy
Gradient Boosting	0.89	0.27	0.41	92.35%
Support Vector Machine	0.92	0.21	0.34	89.25%
Random Forest	0.95	0.27	0.42	91.95%
Artificial Neural Network	0.98	0.18	0.30	86.05%

According to the results presented in Table 11, each model has its own benefits and drawbacks, with the ANN being slightly better at predicting when there is about to be an MB event and the GB excelling at lowering the number of false alarms (i.e., false positives). As a result, on one hand, if maintenance costs are inexpensive and undetected MBs can lead to dire consequences, the ANN is the preferred model to be employed. On the other hand, the GB model is better at reducing the number of false alarms. In case a balanced approach for predicting MBs and reducing false alarms is preferred, either the SVM or RF models can be chosen. Nevertheless, all the models have good accuracy scores, mainly the GB model with 92.35%, for this type of problem, where imbalanced PdM datasets are common and affect negatively accuracy scores.

Table 12 and Figure 25 present the GB, SVM, RF, and ANN confusion matrixes and actual/predicted values, respectively. Figure 25 only presents 200 samples out of the 2000 of the testing data due to space limitations and to better demonstrate the obtained results. Furthermore, failure data points are represented with the value 1, and non-failure data points as 0. Also, in Figure 25, the GB, SVM, RF, ANN, and actual values are represented by the colors green, blue, yellow, brown, and red, respectively, each with their specific marker, to better clarify the reader. It is noteworthy that there is a trade-off between correctly predicting a machine failure and giving false alarms, with the GB having 6 more unsuccessful machine failure predictions than the ANN, but having 132 fewer false alarms (i.e., a trade-off of 1 unsuccessful prediction to 22 false alarms). For the present work, the author favors the prioritization of successful machine failure prediction above the reduction of false alarms. Therefore, the ANN is the preferred model to be used in the present work, not only due to its capabilities to predict successful machine failures but also for its backpropagation feature, allowing for continuous training without having to recreate a model from the beginning.

Table 12 — Gradient boosting, support vector machine, random forest, and artificial neural network confusion matrix

Model		Actual	
		Failure	Non-failure
Gradient Boosting	Failure	54	146
	Non-failure	7	1793
Support Vector Machine	Failure	56	210
	Non-failure	5	1729
Random Forest	Failure	58	158
	Non-failure	3	1781
Artificial Neural Network	Failure	60	278
	Non-failure	1	1661

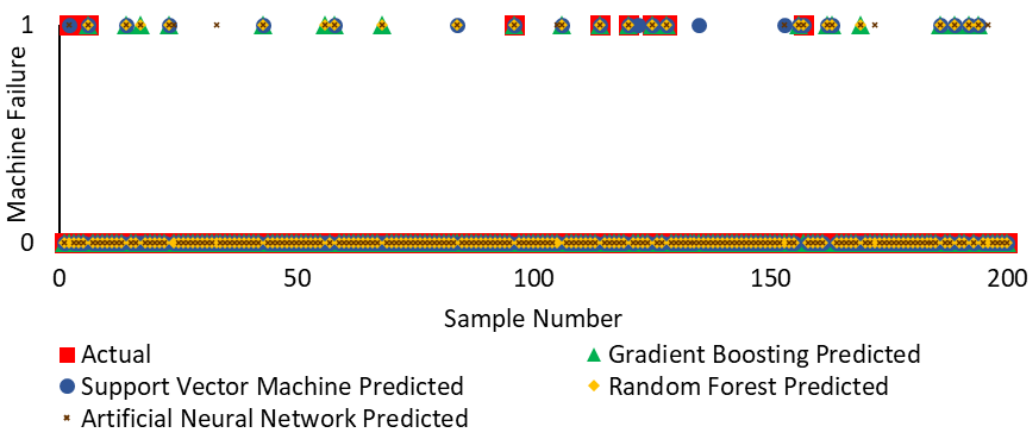

Figure 25 – Actual values and their respective predicted values from the gradient boosting, support vector machine, random forest, and artificial neural network models (200 samples out of 2000 of the testing data)

It is worth noting that another work, available in [202], used the same dataset to validate the application of a bagged trees ensemble classifier. However, due to a lack of good practices, such as using the whole dataset for training and testing, instead of splitting a part of the dataset for training and another for testing, the referred work was not used to compare to the present models. Nevertheless, it is worth noting that even though the other work inflated their obtained results due to overfitting, the present methodology still achieved a better recall score of 0.98 when compared to 0.71 obtained in [202].

5.2.6 Scenario 1.6 – Maintenance Optimization

A maintenance optimization scenario was formulated to validate the scheduler's ability to schedule tasks as well as maintenance activities while also minimizing the total costs. Accordingly, it was considered an optimization weight of 1 for the total cost and 0 for machine occupancy deviation, as well as a 2-hour execution time for the GA. Each maintenance activity, for every machine, has a duration of 6 hours and 10 minutes, an associated labor cost of 3,22 EUR/hour during the stipulated maintenance hours, and a monetary penalty, that doubles the cost (i.e., 6.44 EUR/hour) if done out of maintenance hours. The maintenance data obtained is available in [33]. Furthermore, the input and output data used in the scheduler are published in [203].

It was simulated that the PdM model, validated in section 5.2.5, detected that machines MAQ118 and MAQ120 could breakdown in the next week (i.e., a forecasted time horizon of a week). Therefore, it was imposed that the scheduler needed to include maintenance activities for these machines. One of the maintenance activities needed to be done on MAQ118 from Monday at 07:00 (i.e., period 1) to Monday at 17:00 (i.e., period 120), and another, with no stipulated maintenance hours (i.e., more flexibility), needed to be done on MAQ120. The results of the simulated scenario are presented in Figure 26 where the two maintenance activities are marked as green.

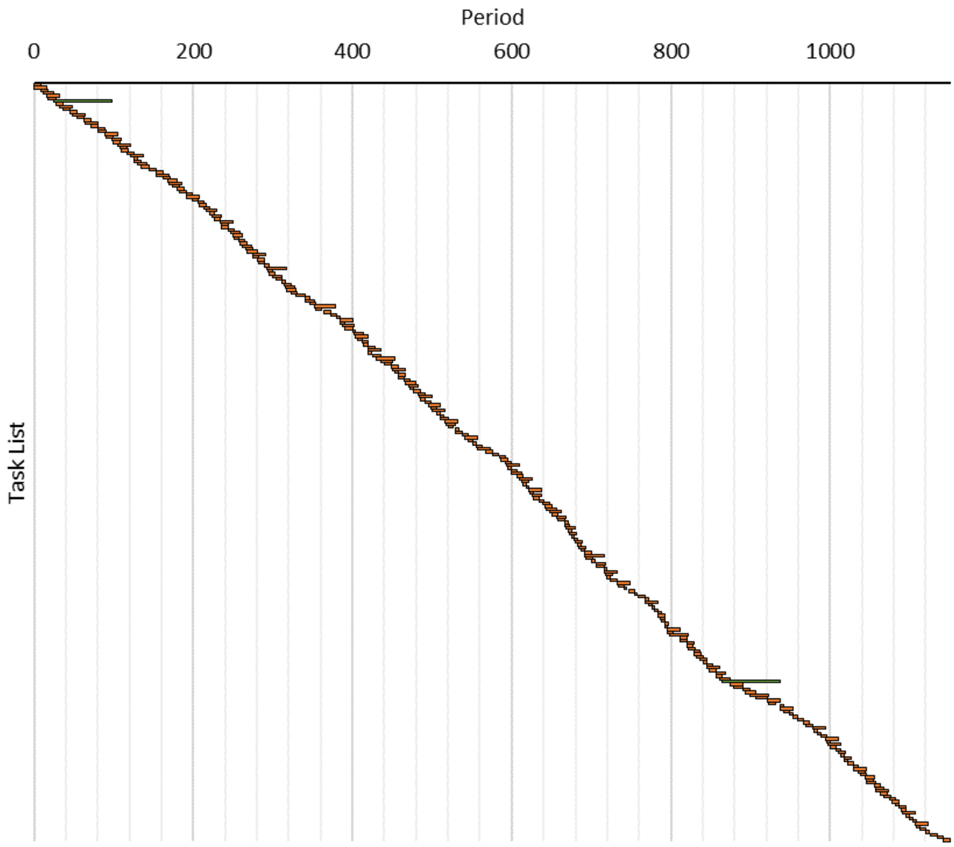

Figure 26 – Gantt diagram of the scheduled production plan with maintenance activities

From the result of the production and maintenance optimization, tasks were shifted in order to accommodate the two maintenance activities. The maintenance for machine MAQ118 was successfully scheduled for Monday at 09:00 (i.e., period 25) to Monday at 15:10 (i.e., period 98), thus complying with the stipulated maintenance hours. The maintenance activity for machine MAQ120 was scheduled for Friday at 15:00 (i.e., period 865) to Monday at 21:10 (i.e., period 938). It is worth noting that the later maintenance activity, which had much more flexibility, was done during periods with low local energy generation and high energy prices, as shown in Figure 27, demonstrating that the proposed scheduler is able to optimize maintenance activities not only by complying their stipulated maintenance hours, if imposed, but also consider where tasks and maintenance activities should be allocated in order to minimize the overall total costs. In short, the scheduler minimizes energy costs by shifting tasks to high local energy generation and lower energy price periods while the opposite is done to maintenance activities, and at the same time, it minimizes maintenance costs by complying with stipulated maintenance hours.

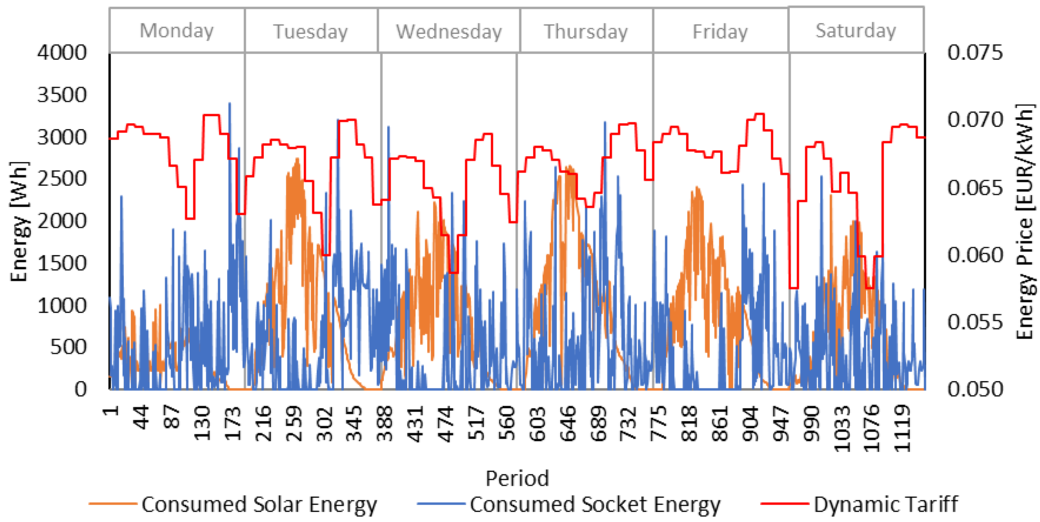

Figure 27 – Energy consumption with energy costs for the maintenance optimization scenario

The overall performance of the scheduler for the maintenance optimization scenario is described in Figure 28, which depicts the total cost (i.e., energy costs plus maintenance costs) of the best individual of each generation of the GA. While the results do not follow a straight decreasing logarithmic function in the first 100 generations, they still show good genetic generation evolution performance in the later generations. The final total cost achieved by the scheduler was 76.60 EUR.

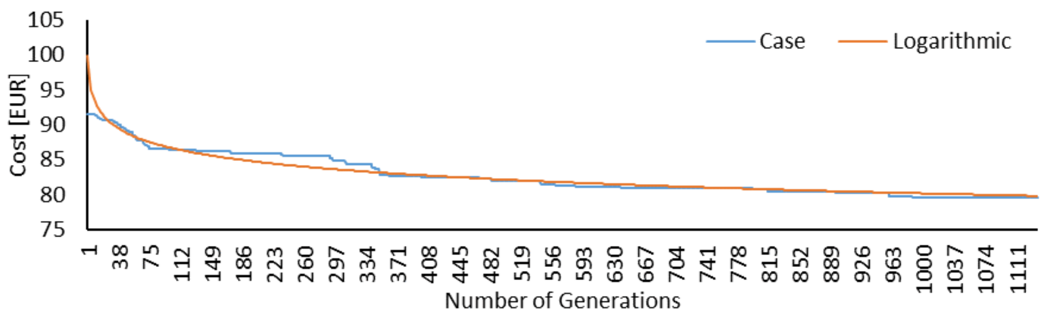

Figure 28 – Cost per generation of the genetic algorithm for the maintenance optimization scenario

5.2.7 Scenario 1.7 – Machine Breakdown Event

An announcement of an MB event was simulated on Friday at 6:00, describing that machine MAQ119 could breakdown at any moment, detected using the predictive maintenance ANN model validated in section 5.2.5. Accordingly, the proposed scheduler imposed a machine available frames constraint, during its event, of 0 usable frames, thus removing the machine from production. The proposed GA, using the energy cost scenario data as the starting point (i.e., from section 5.2.1) was executed for 1 hour at period 769 (i.e., Friday at 7:00) until the remainder of the schedule's time window. Also, the predefined optimization weights were 1 for total cost and 0 for machine occupancy deviation. The complete data is available at [204].

The Gantt diagram of the scheduled production plan, before and after the machine MAQ119 breakdown event for machines MAQ118, MAQ119, and MAQ120, is represented in Figure 29, Figure 30, and Figure 31, respectively. Due to space limitations, the mentioned figures only describe the production plan from period 577 (i.e., Thursday at 07:00) to period 1152 (i.e., Saturday at 23:00). Figure 29 shows that the majority of tasks were shifted away from the machine MAQ119 into MAQ118. Nevertheless, from Figure 31 it is also clear that the rest of the tasks were shifted to machine MAQ120, since after the MB event tasks got more concentrated, with less empty space between each other. Furthermore, the imposed constraint was complied seeing that after period 769 machine MAQ119 does not have any scheduled tasks, thus demonstrating that the algorithm has the capability to adapt to unexpected MBs, even during the production process.

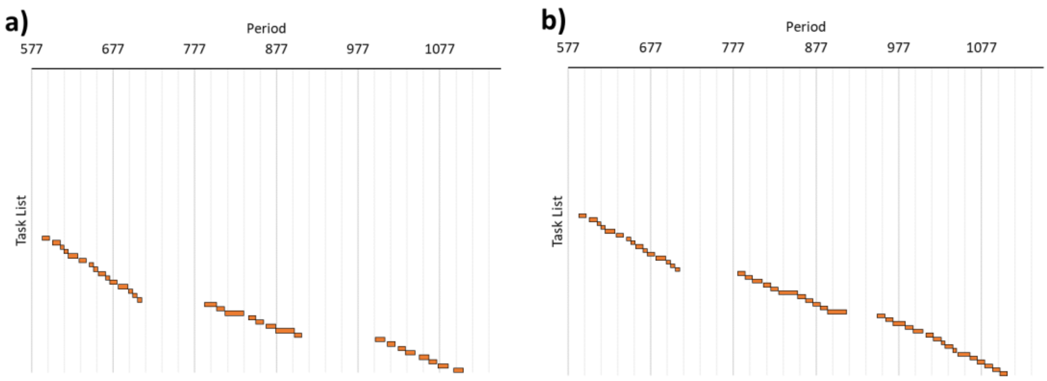

Figure 29 – Gantt diagram of the scheduled production plan for machine MAQ118, from period 577 to period 1152. (a) Before machine MAQ119 breakdown event; (b) after machine MAQ119 breakdown event

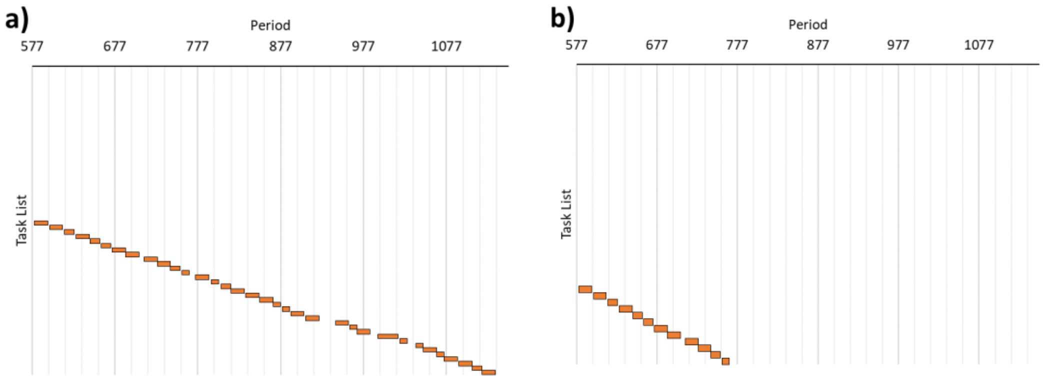

Figure 30 – Gantt diagram of the scheduled production plan for machine MAQ119, from period 577 to period 1152. (a) Before machine MAQ119 breakdown event; (b) after machine MAQ119 breakdown event

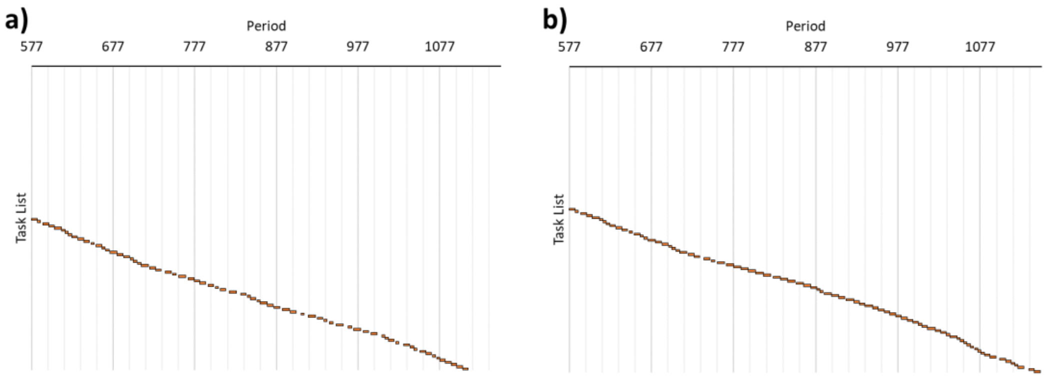

Figure 31 – Gantt diagram of the scheduled production plan for machine MAQ120, from period 577 to period 1152. (a) Before machine MAQ119 breakdown event; (b) after machine MAQ119 breakdown event

For this scenario, the total cost, after including the deterministic optimizations, was 37.95 EUR, up by 4.2%, from 36.42 EUR in the scenario without the MB event (i.e., from section 5.2.1).

5.2.8 Scenario 1.8 – Joint Optimization of Production and Maintenance

The present scenario combines all the features shown in scenarios 1.1, 1.2, 1.3, and 1.6 (i.e., sections 5.2.1, 5.2.2, 5.2.3, and 5.2.7, respectively). The other scenarios are not considered since they either represent events (e.g., demand response or a machine breakdown) that require rescheduling or are for a different application (e.g., predictive maintenance). For energy selling, it was considered an added sales value corresponding to 50% of the buying, the same as presented in scenario 1.2 (i.e., section 5.2.2). Regarding maintenance activities, it was used the same data as in scenario 1.6 (i.e., 5.2.6), that is, a maintenance activity for MAQ118 from Monday at 07:00 (i.e., period 1) to Monday at 17:00 (i.e., period 120), and another for MAQ120 which can be done at any time. These maintenance activities take 6 hours and 10 minutes to complete and have an associated labor cost of 3,22 EUR/hour in maintenance hours, and a double cost penalty (i.e., 6.44 EUR/hour) if done out of maintenance hours. The scenario was executed in 2 hours, with the corresponding optimization weights of 0.8 and 0.2 for the total cost and machine occupancy deviation, respectively. Data for this scenario is available at [205].

The obtained results are shown in Figure 32, which describes the energy consumption, energy sold, and energy buying and selling prices of the present scenario. The machine occupation rates for MAQ118, MAQ119, and MAQ120 are 62.41%, 79.17%, and 77.52%, which shows some level of machine task balancing (i.e., machine occupancy rate standard deviation) even with an imposed weight of 0.2 for machine occupancy deviation. It is also noteworthy that, only about 62,2% of the maintenance activity for MAQ118 was done during the stipulated maintenance hours, the rest had to be done outside of maintenance hours. One of the main reasons could be that tasks were more favorable to be in these periods, primarily when also taking into account energy selling and machine occupancy deviation.

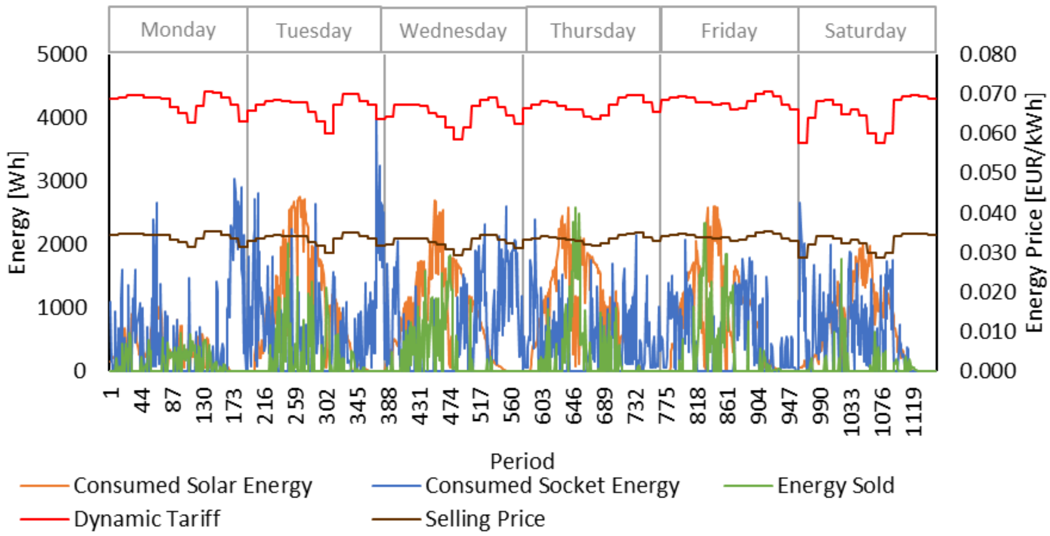

Figure 32 – Energy consumption, energy sold, and energy buying and selling prices for the joint optimization of production and maintenance scenario

The balance between the total cost and machine occupancy deviation is represented in Figure 33, which depicts the total Cost and machine occupancy deviation per generation of the genetic algorithm. It demonstrates good total cost performance while also balancing it with the machine occupancy rate, even when there is a big difference in optimization weights. The total cost, including deterministic optimizations, and machine occupancy rate standard deviation for this scenario are 79.44 EUR and 0.0754%, respectively.

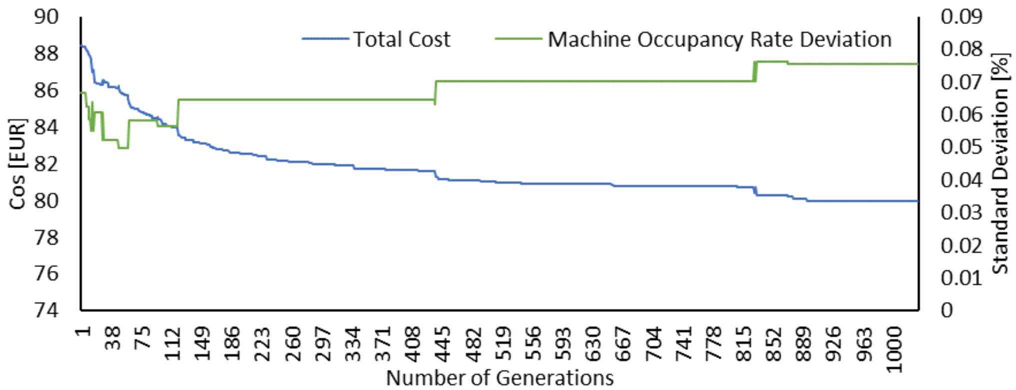

Figure 33 – Total cost and machine occupancy deviation per generation of the genetic algorithm for the joint optimization of production and maintenance scenario

5.2.9 Discussion on Results of Case Study 1

The scenarios presented for the first case study demonstrate the primary capabilities of the proposed scheduler in a manufacturing environment. It is worth noting that other capabilities were not described in the present dissertation due to space limitations. For instance the usage of constraints such as task order in a specific product, task setups, interruptible tasks, machine priority, product request cell choosing, and shift margin. In addition, it was not demonstrated the usage of cells and task modes. Nevertheless, these features are implemented in the system and are more suitable for other types of problems (e.g., smart grids and households in a resource-sharing community).

The total cost and machine occupancy rate standard deviation for scenarios 1.1, 1.2, 1.3, 1.4, 1.6, 1.7, and 1.8, and their overall total cost and machine occupancy rate standard deviation reduction/increase percentage are represented in Table 13. Scenario 1.5 was excluded from Table 13 since it describes the predictive maintenance system performance and not the capabilities of the production/maintenance scheduler, thus a different type of problem. Furthermore, scenario 1.1 is used as a baseline for all other scenarios since they all include scenario 1.1 data. The obtained results demonstrate, on one hand, that the proposed scheduler is able to reduce the total cost of manufacturing even further when allowed to sell PV energy in excess, as shown when comparing the total cost reduction of 24.6% from scenarios 1.1 to 1.2. On the other hand, if machine longevity is a priority, the proposed schedule is able to improve machine occupancy rate standard deviation up to 87.8% while not penalizing too much the total cost (i.e., only an increase of 12.8%), as demonstrated by comparing scenarios 1.1 and 1.3. Moreover, by comparing scenarios 1.1 and 1.4 we can conclude that the scheduler is able to not only comply with DR events but also achieve a 4.2% total cost reduction. Scenarios 1.1 and 1.6 validate the scheduler's ability to incorporate maintenance activities, and while a direct total cost or machine occupancy rate standard deviation comparison is unfair since maintenance activities add up a lot of the costs and time (i.e., an increase of 110.3% in total costs and 69.3% for machine occupancy rate standard deviation), the system was still able to schedule maintenance activities in an already almost full production schedule without having the need to remove tasks from the schedule. Regarding scenarios 1.1 and 1.7 they demonstrate the scheduler's ability in adapting to unexpected MBs during the production process, by scheduling all the tasks and maintenance activities away from the machine that broke down into the other machines. In addition, it did not remove tasks from production and it had an increase of 4.2% and 37.3% for the total cost and machine occupancy rate standard deviation, respectively. Finally, by comparing scenarios 1.1 and 1.8 it is shown that the scheduler is able to incorporate production, maintenance activities, and constraints and still improve machine occupancy rate standard deviation by 24.4%. It is worth noting that for scenario 1.8 the total cost increased by 118.1% from scenario 1.1 because of the added maintenance costs, however, when comparing scenario 1.8 with 1.6 (i.e., both include maintenance activities) there was only an increase of 3.7% in the total costs while improving the machine occupancy rate standard deviation by 55.3%.

Table 13 — Scenarios' total cost and machine occupancy rate standard deviation table

Scenario	Total Cost (EUR)	Machine Occupancy Rate Standard Deviation (%)	Total Cost Reduction/Increase from Scenario 1.1 (%)	Machine Occupancy Rate Standard Deviation Reduction/Increase from Scenario 1.1 (%)
Scenario 1.1	36.42	0.0997	0%	0%
Scenario 1.2	27.46	0.1192	-24.6%	19.6%
Scenario 1.3	41.07	0.0122	12.8%	-87.8%
Scenario 1.4	31.82	0.1337	-12.6%	34.1%
Scenario 1.6	76.60	0.1688	110.3%	69.3%
Scenario 1.7	37.95	0.1369	4.2%	37.3%
Scenario 1.8	79.44	0.0754	118.1%	-24.4%

To validate the proposed predictive maintenance system in a manufacturing environment scenario 1.5 was formulated, which compares the proposed ANN model to GB, SVM and RF models. Its results show that the ANN has the worst accuracy among all models, with

86.05% compared to 89.25%, 91.95%, and 92.35% for the SVM, RF, and GB, respectively. However, it has the best recall score, with an almost perfect 0.98, when compared to 0.95, 0.92, and 0.89 for the RF, SVM, and GB. On top of that, the recall metric is considered to be the most important metric to measure machine learning models' performance in predictive maintenance problems, since in this type of problem classes are almost always not balanced. Accordingly, the ANN was chosen to be the preferred model to be employed not only because of its recall score while also maintaining acceptable levels of accuracy but also for its ability for continuous training, due to its backpropagation feature.

5.3 Case Study 2 – Residential Load Shifting

The proposed scheduler is highly adaptable for other applications besides industrial production scheduling as it can be used for residential load shifting. To demonstrate this, a case study involving load shifting in appliances is proposed, that utilizes the proposed GA scheduler of the present dissertation. This case study was obtained from a published paper [37] in the Energy Journal, Elsevier, by the dissertation's author. The case study used is based on a household with four appliances to be scheduled for five days that has available PV energy. While the proposed solution is able to provide the schedule for far more days (e.g., for a whole week or a longer period), only five days were considered to better demonstrate the obtained results. Also, the five days represent concrete consumption days, with specific days for flexibility.

The case study uses real household workload data, locally generated PV data, and energy market prices provided by [206] and [207]. The house case study is presented in Figure 34. Energy prices reflect a double tariff: on-peak hours, between 8:15 and 22:00, 0.1879 EUR/kWh; off-peak prices are 0.101 EUR/kWh. The workload data represents five full days (i.e., from 0:00 to 23:59) of Business as Usual (BAU) workload, which is to be optimized energy cost-wise (i.e., reduce electricity bill) by the proposed algorithm. Three load shifting optimization scenarios are explored: scenario "2.1", which represents standard load flexibility; scenario "2.2", which describes low load flexibility; and scenario "2.3", which is characterized by having high load flexibility. All scenarios were executed with the respective optimization weights of 1 and 0 for the total cost and appliance occupancy deviation. The case study uses 15 min for all loads and energy data, thus having a total number of available periods of 480. The case study considers four different appliances and a culmination of them (e.g., HI-FI, microwave, television, and fridge) consuming energy and are not considered to be able to shift, denominated as "Other". The four appliances considered are an air conditioner, a dishwasher, a washing machine, and a clothes dryer. Figure 35 represents the energy profiles of one load, from start to finish, for each appliance. Therefore, the case study uses four different loads: an air conditioner load with a duration of five and a half hours, the dishwasher with an execution time of one and a half hours, a washing machine with a profile of one and a half hours, and the clothes dryer with 1 hour. Also, for this case study, each load for each appliance is executed three times. Non-shiftable appliances, denominated as the load "Other", it has the energy profile represented in Figure 36, for each day considered in the case study.

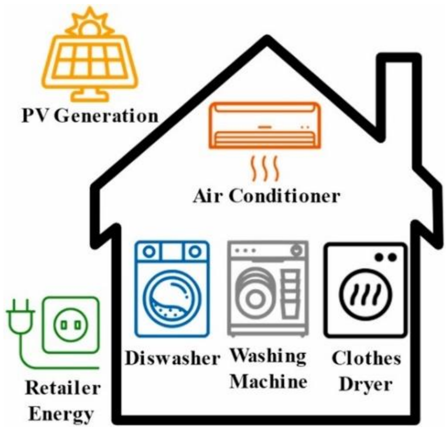

Figure 34 – Case study house scenario [37]

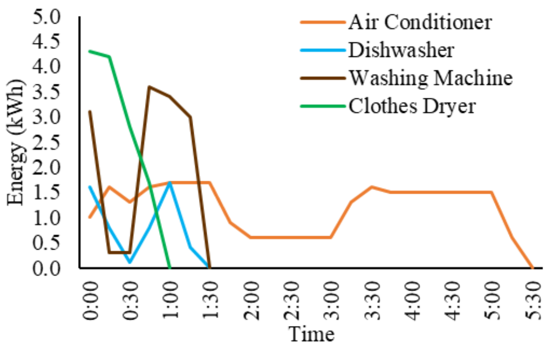

Figure 35 – Appliances energy profiles [37]

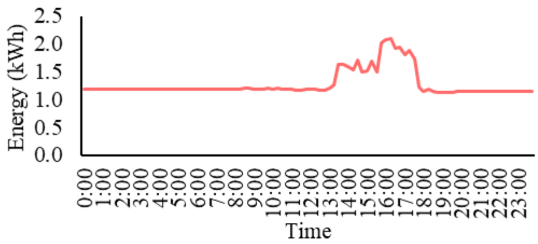

Figure 36 – Non-shiftable appliances energy profile in a day [37]

The case study uses two constraints: load order (i.e., task order constraint) and load operating time window (i.e., product request task period range constraint). The order constraint was applied so that a completed washing machine load must precede every clothes dryer load. The operation schedule execution time ranges considering both the BAU and load shifting scenarios for the load operating time window constraint are represented in Table 14.

Table 14 — Load operating time window [37]

Load	BAU	Scenario 2.1 (Standard Flexibility)		Scenario 2.2 (Low Flexibility)		Scenario 2.3 (High Flexibility)	
		Lower bound	Upper bound	Lower bound	Upper bound	Lower bound	Upper bound
Dishwasher “1”	7:00 Day 1	9:00 Day 1	17:00 Day 1	9:00 Day 1	14:00 Day 1	9:00 Day 1	7:00 Day 2
Air Conditioner “1”	16:00 Day 1	14:00 Day 1	22:30 Day 1	14:00 Day 1	20:00 Day 1	13:00 Day 1	00:00 Day 2
Washing Machine “1”	19:00 Day 1	0:00 Day 2	17:00 Day 2	0:00 Day 2	8:00 Day 2	9:00 Day 1	7:00 Day 2
Clothes Dryer “1”	21:30 Day 1	19:00 Day 2	17:00 Day 3	10:00 Day 2	7:00 Day 3	9:00 Day 2	23:59 Day 3
Dishwasher “2”	7:00 Day 2	9:00 Day 2	17:00 Day 2	9:00 Day 2	14:00 Day 2	9:00 Day 2	7:00 Day 3
Air Conditioner “2”	16:00 Day 2	14:00 Day 2	22:30 Day 2	14:00 Day 2	20:00 Day 2	13:00 Day 2	00:00 Day 3
Washing Machine “2”	19:00 Day 2	0:00 Day 3	17:00 Day 3	0:00 Day 3	8:00 Day 3	9:00 Day 2	7:00 Day 3
Clothes Dryer “2”	21:30 Day 2	19:00 Day 3	17:00 Day 4	10:00 Day 3	7:00 Day 4	9:00 Day 3	23:59 Day 4
Dishwasher “3”	7:00 Day 3	9:00 Day 3	17:00 Day 3	9:00 Day 3	14:00 Day 3	9:00 Day 3	7:00 Day 4
Air Conditioner “3”	16:00 Day 3	14:00 Day 3	22:30 Day 3	14:00 Day 3	20:00 Day 3	13:00 Day 3	00:00 Day 4
Washing Machine “3”	19:00 Day 3	0:00 Day 4	17:00 Day 4	0:00 Day 4	8:00 Day 4	9:00 Day 3	7:00 Day 4
Clothes Dryer “3”	21:30 Day 3	19:00 Day 4	17:00 Day 5	10:00 Day 4	7:00 Day 5	9:00 Day 4	23:59 Day 5
Other	0:00 Day 1	0:00 Day 1	23:59 Day 5	0:00 Day 1	23:59 Day 5	0:00 Day 1	23:59 Day 5

The BAU scenario aims to simulate the regular user consumption when no electricity bill reduction is taken into account. Therefore, it does not prioritize PV usage and low energy price periods. The operation schedule of appliances for the BAU scenario is shown in Figure 37. In this scenario, generated PV, provided by photovoltaic panels, is poorly used, as most appliances execute in low PV availability periods. Furthermore, as complemented by Figure 38, some loads are positioned in high demand periods, where energy price is highest. Therefore, there is a significant margin of improvement for the BAU scenario, as loads on both low PV availability and high energy price periods could be shifted for electricity bill savings. For this scenario, the overall total electricity bill is 119.26 EUR.

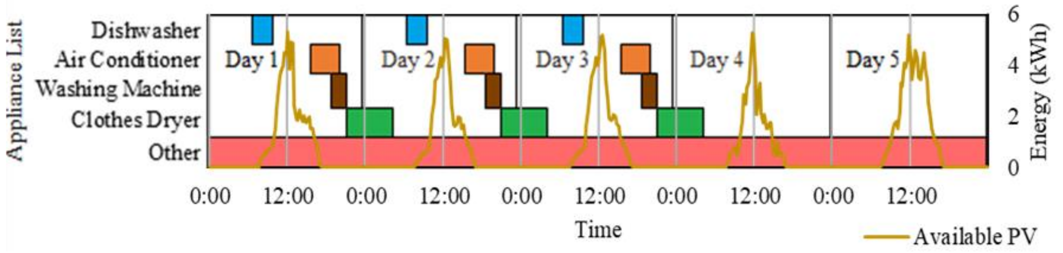

Figure 37 – Business as usual operation schedule of appliances [37]

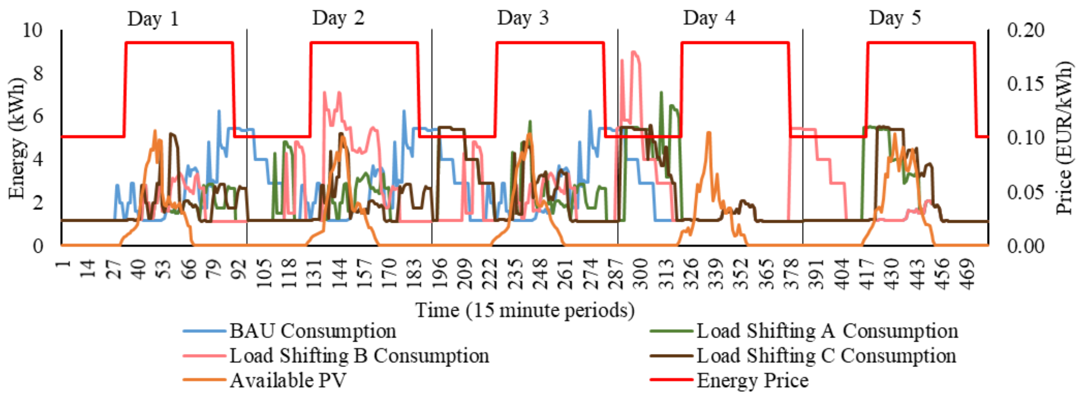

Figure 38 – Energy consumption, energy price, and available PV [37]

5.3.1 Scenario 2.1 – Standard Flexibility

The load shifting scenario with standard load flexibility, also denominated as scenario "2.1", contrarily to the BAU scenario, clearly demonstrates, through Figure 39, that the proposed solution utilizes generated PV as much as possible, thus reducing the demand for external energy suppliers. Also, the black arrows represent the imposed order constraint; thus, all constraints are respected. For this case study, the overall total electricity bill from the load shifting scenario "2.1" is 104.05 EUR, thus improving 12.7% from the BAU scenario.

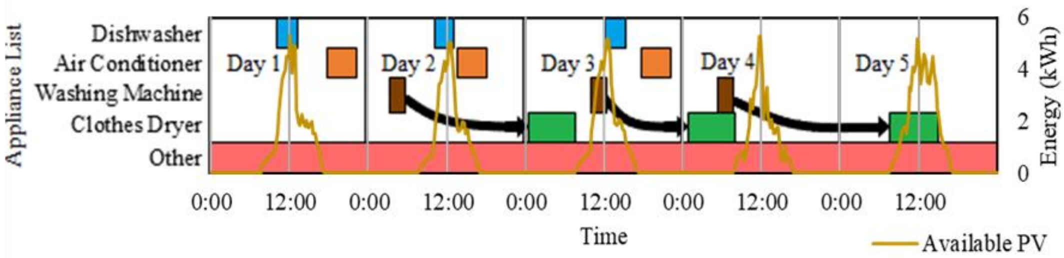

Figure 39 – Load shifting operation schedule of appliances for scenario 2.1, black arrows represent the order constraint "washing machine" then "clothes dryer" [37]

Figure 38 represents the energy consumption from both scenarios (i.e., BAU and scenario "2.1"), available PV, and energy market prices. Apart from high generated PV usage from the algorithm in the load shifting scenario, the higher the energy prices, the lower the energy consumed, thus loads are being shifted to reduce electricity bill. Also, it is noteworthy that on day 4, even though there is available PV, the algorithm chose instead to shift loads to lower energy price periods. Due to high energy prices and high consumption, the available PV would not cover the high costs. The cost (i.e., electricity bill) difference between both scenarios (load shifting "2.1" − BAU) is represented in Figure 40. It demonstrates bill reductions mainly at the end of days 1, 2, and 3. The overall electricity bill reduced is 15.21 EUR.

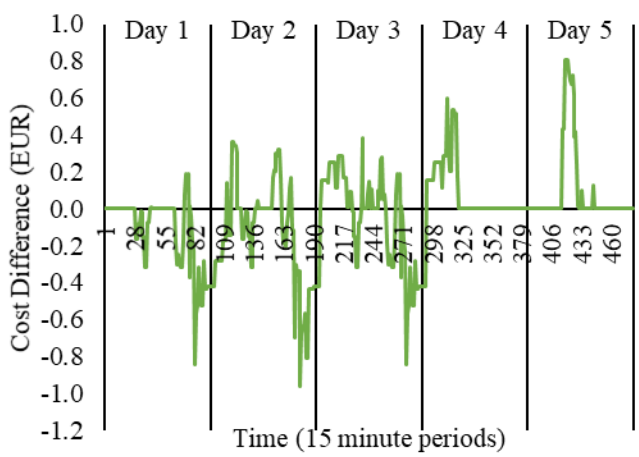

Figure 40 – Scenario 2.1 cost difference (load shifting "2.1" − BAU) [37]

5.3.2 Scenario 2.2 – Low Flexibility

Scenario "2.2" describes a load shifting scenario with lower load flexibility than scenario "2.1". The operation schedule of appliances and available PV for scenario "2.2" are represented in Figure 41, showing less generated PV utilization when compared to scenario

“2.1”. This could be related to having lower flexibility and thus not shifting loads to periods with available PV. Nonetheless, when complemented with Figure 38, it is clear that even though the algorithm could not shift loads to available PV periods, it instead shifted them to lower energy price periods. For example, there are huge energy consumptions around the dawn of days 4 and 5, where energy price is at its lowest. Furthermore, there is also some consumption in the mornings of days 2 and 3, just before on-peak hours. The overall total electricity bill of scenario “2.2” is 108.68 EUR, thus improving 8.9% from the BAU scenario.

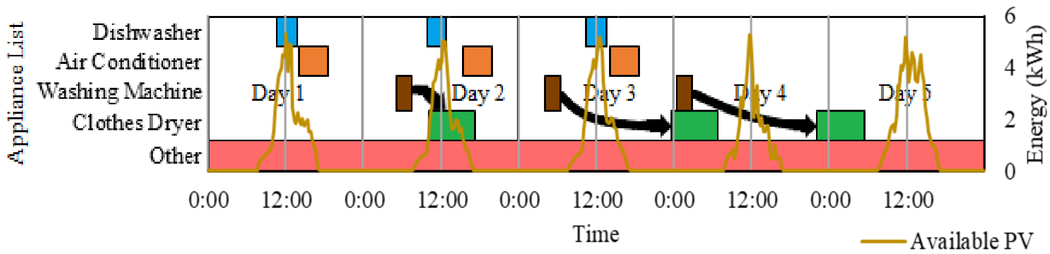

Figure 41 – Load shifting operation schedule of appliances for scenario 2.2, black arrows represent the order constraint “washing machine” then “clothes dryer" [37]

The cost difference between the BAU and “2.2” scenarios is shown in Figure 42. Similar to scenario “2.1”, it demonstrates electricity bill savings at the end of days 1, 2, and 3. However, it is noteworthy that there is a huge bill increase on the noon of day 2, meaning that loads were shifted to these periods instead. The overall electricity bill reduced is 10.58 EUR.

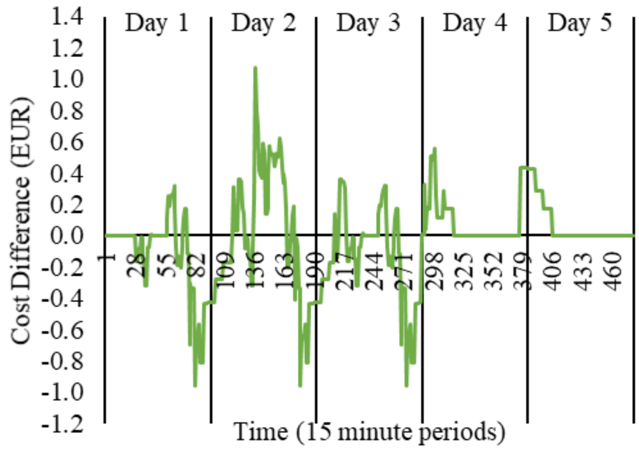

Figure 42 – Scenario 2.2 cost difference (load shifting “2.2” – BAU) [37]

5.3.3 Scenario 2.3 – High Flexibility

The load shifting scenario “2.3” aims at simulating scenario “2.1” but with much higher load flexibility, allowing loads to be shifted to available PV periods that previously could not. Figure 43 represents the operation schedule of appliances for scenario “2.3”, which shows when compared to scenario “2.1”, that scenario “2.3” utilizes PV generation much more efficiently, having the majority of its loads in these periods. Moreover, through Figure 38, we can conclude that almost every load not using PV generation is in off-peak hours. Therefore, due to PV availability and energy price optimization coupled with very flexible loads, scenario “2.3” can further minimize electricity bills than scenario “2.1”, reaching an overall total electricity bill of 101.70 EUR. Compared to the BAU scenario, scenario “2.3” has an improved bill reduction of 14.7%.

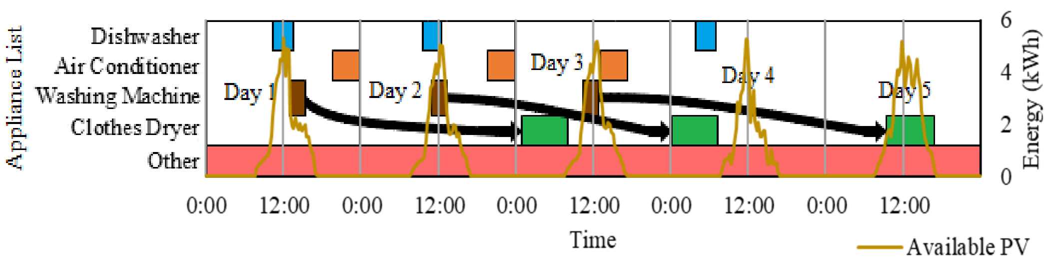

Figure 43 – Load shifting operation schedule of appliances for scenario 2.3, black arrows represent the order constraint “washing machine” then “clothes dryer" [37]

Figure 44 represents the cost difference between scenarios BAU and “2.3”. As already stated in scenarios “2.1” and “2.2”, scenario “2.3” also has electricity bill reductions at the end of days 1, 2, and 3. This similar occurrence in the three scenarios implies that, when taking into account Figure 37, the clothes dryers are highly inefficient at 21:30 of days 1, 2, and 3. Also noteworthy from Figure 44 is that electricity bill increases are more or less balanced between the five days, except for bill spikes at noon of days 1 and 5. Therefore, when given more flexibility, the proposed solution can efficiently balance the loads' electricity bills among its available scheduling days. The overall electricity bill reduced is 17.56 EUR.

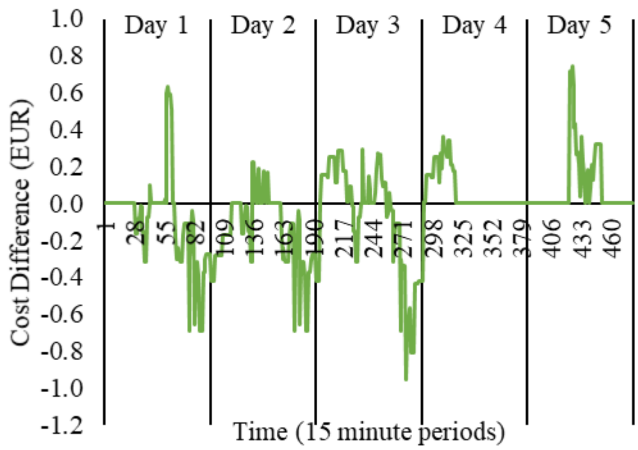

Figure 44 – Scenario 2.3 cost difference (load shifting “2.3” − BAU) [37]

5.3.4 Discussion on Results of Case Study 2

The electricity bill of each scenario and their bill difference with the BAU scenario and overall bill reduction percentage are represented in Table 15. Results show that the proposed solution effectively reduces electricity bills, allowing household consumers to save up 10.58 to 17.56 EUR on five days. Subsequently, if applied to a whole month, it can save 65.60 to 108.87 EUR on bills. Furthermore, even though scenario “2.2” is the least flexible when comparing bill reductions, it still pulls an overall electricity bill reduction of 8.9%. Also, it is noteworthy that there are diminishing returns when increasing load flexibility. For instance, from scenario “2.2” to scenario “2.1”, there is an improvement of 3.8%, while from scenario “2.1” to scenario “2.3”, there is only a gain of 2.0%. Therefore, the algorithm is already effective enough for consumers to use it with their standard loads' schedules without sparing more time to save more money since there are diminishing returns.

Table 15 — Scenario bill table [37]

Scenario	Bill (EUR)	Bill Difference from BAU	Bill Reduction (%)
BAU	119.26	0	0%
Scenario 2.1	104.05	15.21	12.7%
Scenario 2.2	108.68	10.58	8.9%
Scenario 2.3	101.70	17.56	14.7%

For this case study, load shifting scenario “2.1”, the GA was executed for 30 min, equivalent to around 634 generations. The performance of the proposed GA can be seen through the cost of the best individual of each genetic generation, as shown in Figure 45. The results imply a good performance since they follow a decreasing logarithmic function. However, they do not represent a perfect logarithmic function because we also have to consider the number of constraints applied since they limit the number of available solutions. It is noteworthy that, for this case study, only 187 genetic generations were needed to achieve a good solution, thus demonstrating that the algorithm only needed about 9 min to find a good solution, lower than the stipulated 30 min of execution time.

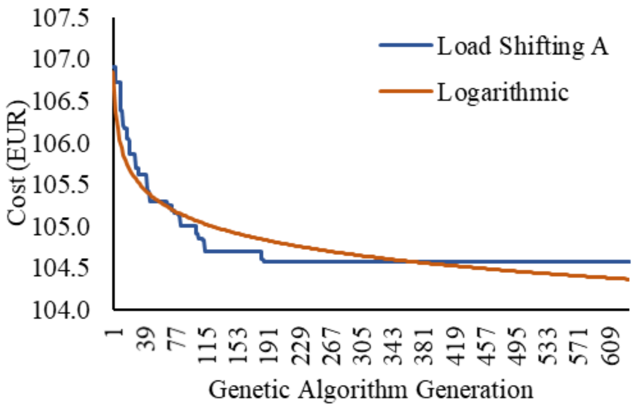

Figure 45 – Cost per generation of the genetic algorithm in scenario 2.1 [37]

The GA in scenario “2.2” was also executed for 30 min, having 783 genetic generations. The cost evolution of the best individual in the GA can be seen in Figure 46. Results show that the GA has bad performance since the cost does not follow a decreasing logarithmic function. However, we must consider that this scenario has the lowest load flexibility compared to the other scenarios, making it very difficult for the algorithm to find a reasonable solution (i.e., a solution that respects all imposed constraints) and one with better fitness values. As a result, from Figure 46, it is unclear if the GA had a good performance. Nonetheless, let's only consider the difference between the first and last best individuals, which amounts to 6.21, an improvement of 5.41%, and compare it to 2.67% from scenario “2.1” and 7.27% from scenario “2.3”. We can conclude that scenario “2.2” has a similar bill reduction as the other scenarios, thus implying that it has a good GA performance. Furthermore, the best solution found by the GA was at generation 54; therefore, the algorithm only needed around 2 min of execution time to find a good solution for scenario “2.2”.

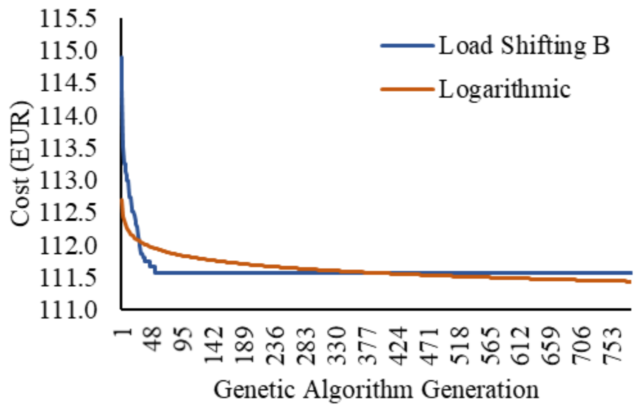

Figure 46 – Cost per generation of the genetic algorithm in scenario 2.2 [37]

Scenario “2.3” has a total of 1149 genetic generations from 30 min of GA execution time. Compared to the other scenarios, the high number of genetic generations is that, due to the high load flexibility in the scenario, there is a wider variety of reasonable solutions, thus reducing the need for repair or exclusion of invalid solutions. The cost of the best individual per generation of the GA, shown in Figure 47, clearly demonstrates a high performance from the GA, with the cost evolution of scenario “2.3” almost being a decreasing logarithmic function. To achieve such a good solution, the GA used 588 generations, equivalent to around 15 min of execution time.

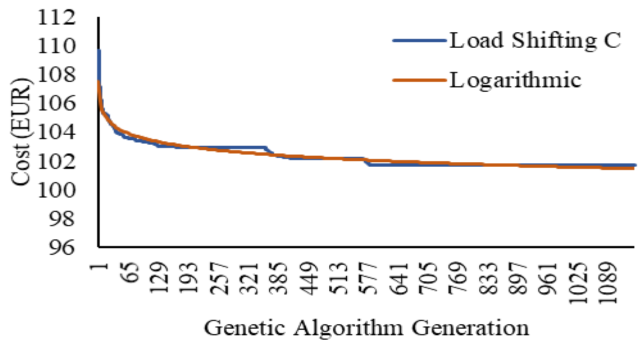

Figure 47 – Cost per generation of the genetic algorithm in scenario 2.3 [37]

Through the convergence analysis of the GA, by using different scenarios, we can conclude that the proposed solution, when confronted with lower load flexibility scenarios, has worse fitness evolution. In contrast, higher load flexibility scenarios reach an almost perfect convergence function.

6 Conclusions

Manufacturing companies nowadays operate in a challenging dynamic and uncertain environment where time commitments are very rigid and electricity prices are skyrocketing. Consequently, all the costs should be considered and minimized as much as possible. This includes the energy costs from production, maintenance costs, and the opportunities coming from the participation in DR events. Furthermore, due to the increasing threat of climate change, manufacturing companies should also start using RERs (e.g., PV energy) as an alternative energy source to retailer energy.

Accordingly, the present dissertation proposes and implements a production line management system for joint optimization of production and maintenance for cost-effective and machine longevity manufacturing while also considering DR participation and MB events. The proposed system can be divided into two different components: the scheduler/reschedule and the PdM component. For the scheduler/rescheduler an approach based on a GA to minimize total costs and machine occupancy rates standard deviation of a production line by considering the usage of RERs, dynamic energy market pricing, excess energy for selling, maintenance activities, and constraints imposed on the production plan is employed. The implemented approach was able to handle complex optimization problems in a time frame fast enough to receive DR programs, MB events, and energy prices in real-time and update the production plan taking into account such inputs. For PdM, an ANN was used to predict machine failure status (i.e., health prognosis). These different components of the proposed system are connected internally in the manufacturing facilities via an HTTP protocol.

The obtained results highlight the robustness of the implemented methodology to handle a set of three machines operating to deliver products in a one-week time frame. Given the flexibility of time available to perform each task, it was possible to allocate each task considering its specific consumption looking at the prices in real-time, DR program to participate in, and MB events that can occur. In this way, the production and maintenance costs were reduced, while also reducing machine degradation.

6.1.1 Accomplishments

Table 16 shows the objectives, presented in the introductory section, and, for each of them, a description of their level of accomplishment.

Table 16 — Dissertation accomplished objectives

Objective	Accomplishments
Implementation of a production and maintenance scheduling/rescheduling system	The development of an intelligent scheduling and maintenance system for cost minimization and machine longevity maximization was accomplished. All the main entities are implemented and the system shows good performance and adaptability, considering the case studies presented in sections 5.2, and 5.3, as well as other tests made during the course of the project. Furthermore, both the input and output data formats are suitable for the problem at hand and their structure is coherent and simple. It should be noted that the system is also implemented in a REST API application, allowing communication with third-party applications.
Implementation of a system capable of participating in demand response programs and adapting to machine breakdown events	For participation in DR programs and MB events an intelligent rescheduling system was developed successfully. The rescheduling system supports all types of normal scheduling constraints and takes into account all the nuances that exist when rescheduling part of a production plan. As a result, the system is able to respond/adapt to DR programs and MB events, through energy limit constraints, or machine available frames constraints (i.e., removing machines from production), respectively.
Implementation of a predictive maintenance system	The development of a PdM system for machine failure prediction was successful. The system presents really good performance when compared to more conventional ML and statistical models, as shown in sections 5.2.5.
Evolution of the system with new features	The evolution of the production line management system with new features was successful. It presents several features such as task order, product task order, task collision, task setup, time leap, interruptible task, machine available frames, machine priority,

	product request deadline, product request task period range, product request cell choosing, energy limit, shift margin, energy peaks in energy sources, multiple retailers and buyers, input and output data validation, cost and repair heuristics, among others.
System documentation	Throughout the project, the production line management system documentation was always produced and updated with the latest functionalities. In fact, almost every two weeks there were MUWO project meetings to assess the current state of the system, which lead to the documentation being always up-to-date. Furthermore, the documentation also covers the parameterization of input and output data, as well as the structure of the various entities in the system. Therefore, the production of system documentation was successful.
Tests on the system	From an early stage, the system was tested in its various components. To facilitate the evaluation of the system, Excel templates were developed that automatically, from the output data of a production plan, generate graphs, charts, and tables that facilitate the reading and analysis of the production plan. Furthermore, real-production and -residential data was used to better simulate the uncertainties of the real world and provide a more robust validation. In short, the validation of the proposed system was very successful.

Therefore, since all the proposed objectives were accomplished, it is safe to assume that the project was accomplished with great success.

6.1.2 Limitations

While there were some limitations in the project, most of them were of minor importance, for example, report delays, little knowledge concerning common terms in the energy and manufacturing fields (e.g., PV, DR, Time of Use), and language barriers during MUWO project meetings. However, one of the main limitations of the current project was the ability to gain access to real machine data for predictive maintenance from a manufacturing company. In the end, data was obtained but much of the project's time had already passed leaving little time to validate the proposed solution with real data. In addition, implementation time was also a concern since there was a complex production line management system to be developed with numerous features.

6.1.3 Ethical Questions

While AI can have huge benefits for manufacturing companies, mainly providing very-high levels of productivity and reducing workplace accidents, many of these technologies come with ethical problems [208]. Most ethical issues that come with AI concern the loss of jobs, privacy invasion, worker exploitation, environmental impact, accountability, and AI bias [209], [210]. Therefore, it is crucial to identify what are the main ethical questions that could arise from a system based on AI. Subsequently, the main ethical questions identified in the proposed system are:

- Extendending the current scheduler to also take into account worker's performance, which could lead to unethical firings;
- Using the PdM system to evaluate workers' performance regarding their maintenance inspections and repairs, which could also lead to unethical firings;
- The level of automation given by the proposed solution can substitute workers' jobs, which can lead to mass unemployment;
- The system's ability to consider maintenance activities out of regular maintenance hours could be used to exploit the workers for unscheduled hours;
- The cost-oriented optimization of the scheduler can lead to wasteful behaviors regarding natural resources (e.g., disregarding generated energy for cheaper production);
- If the proposed system had full control over machines, decisions could arise regarding whether a machine should wait or not for a worker to leave the manufacturing space, which could lead to AI morality dilemmas and possible worker accidents.

6.1.4 Future Work

The success of the present project has brought many opportunities to further improve the system, not only by extending features but also by adding new functionalities. Accordingly, there are a variety of planned features to be implemented, such as:

- Improve the creation of valid genetic individuals that respect all the imposed constraints in the production plan, by adding new heuristics at the creation of the initial population that repair individuals;

- Reduce task tardiness, either through multi-objective optimization or heuristics;
- Maintenance priority scheduling based on machines' downtime impact, type, and age;
- Client priority on product requests, based on the client's importance, premium payments, and ties with the company.
- GA multi-objective optimization based on not only minimizing the overall costs and machine occupation deviation but also maximizing product quality;
- Consider minimum product quality standards as a constraint imposed on a product request;
- Reduce the GA scheduler/rescheduler processing time through multiprocessing (i.e., process-based parallelism);
- Predict machine failure type in the PdM system, which would allow having different types of maintenance activities for the same machine, each with a different duration;
- Develop an alternative scheduler/reschedule, based on the principles of ML, for when there is little to no time to execute the GA;
- Develop a production and maintenance management system that would automatically manage the proposed scheduler/reschedule and predictive maintenance systems through notifications (e.g., if one or more machines are detected to have a failure, a warning notification is sent to the scheduler/reschedule).

References

[1] "Gecad - Home." http://www.gecad.isep.ipp.pt/GECAD/Pages/Presentation/Home.aspx (accessed Dec. 17, 2021).

[2] "Europe's Electricity Prices Hit Record High, Setting Energy Crisis to Last - Bloomberg." https://www.bloomberg.com/news/articles/2021-12-08/energy-crunch-sends-benchmark-european-power-prices-to-record (accessed Dec. 19, 2021).

[3] J. Serrano González and C. Álvarez Alonso, "Industrial electricity prices in Spain: A discussion in the context of the European internal energy market," *Energy Policy*, vol. 148, p. 111930, Jan. 2021, doi: 10.1016/j.enpol.2020.111930.

[4] S. Halbrügge, P. Schott, M. Weibelzahl, H. U. Buhl, G. Fridgen, and M. Schöpf, "How did the German and other European electricity systems react to the COVID-19 pandemic?," *Appl. Energy*, vol. 285, p. 116370, Mar. 2021, doi: 10.1016/j.apenergy.2020.116370.

[5] O. Abrishambaf, F. Lezama, P. Faria, and Z. Vale, "Towards transactive energy systems: An analysis on current trends," *Energy Strateg. Rev.*, vol. 26, p. 100418, Nov. 2019, doi: 10.1016/j.esr.2019.100418.

[6] G. Vural, "Renewable and non-renewable energy-growth nexus: A panel data application for the selected Sub-Saharan African countries," *Resour. Policy*, vol. 65, p. 101568, Mar. 2020, doi: 10.1016/j.resourpol.2019.101568.

[7] Y. Fernando and W. L. Hor, "Impacts of energy management practices on energy efficiency and carbon emissions reduction: A survey of malaysian manufacturing firms," *Resour. Conserv. Recycl.*, vol. 126, pp. 62–73, Nov. 2017, doi: 10.1016/j.resconrec.2017.07.023.

[8] P. Saxena, P. Stavropoulos, J. Kechagias, and K. Salonitis, "Sustainability assessment for manufacturing operations," *Energies*, vol. 13, no. 11, p. 2730, May 2020, doi: 10.3390/en13112730.

[9] B. Parrish, P. Heptonstall, R. Gross, and B. K. Sovacool, "A systematic review of motivations, enablers and barriers for consumer engagement with residential demand response," *Energy Policy*, vol. 138, p. 111221, Mar. 2020, doi: 10.1016/j.enpol.2019.111221.

[10] P. Faria, R. Barreto, and Z. Vale, “Demand Response in Energy Communities Considering the Share of Photovoltaic Generation from Public Buildings,” Sep. 2019, doi: 10.1109/SEST.2019.8849082.

[11] L. Xiao, S. Song, X. Chen, and D. W. Coit, “Joint optimization of production scheduling and machine group preventive maintenance,” *Reliab. Eng. Syst. Saf.*, vol. 146, pp. 68–78, Feb. 2016, doi: 10.1016/j.ress.2015.10.013.

[12] D. S. Thomas, “The Costs and Benefits of Advanced Maintenance in Manufacturing,” *Natl. Inst. Stand. Technol.*, no. NIST AMS 100-18, pp. 1–45, 2018, doi: 10.6028/nist.ams.100-18.

[13] Q. Liu, M. Dong, and F. F. Chen, “Single-machine-based joint optimization of predictive maintenance planning and production scheduling,” *Robot. Comput. Integr. Manuf.*, vol. 51, pp. 238–247, Jun. 2018, doi: 10.1016/j.rcim.2018.01.002.

[14] P. Baraldi, M. Compare, and E. Zio, “Maintenance policy performance assessment in presence of imprecision based on Dempster-Shafer Theory of Evidence,” *Inf. Sci. (Ny).*, vol. 245, pp. 112–131, Oct. 2013, doi: 10.1016/j.ins.2012.11.003.

[15] J. P. Kenne and L. J. Nkeungoue, “Simultaneous control of production, preventive and corrective maintenance rates of a failure-prone manufacturing system,” *Appl. Numer. Math.*, vol. 58, no. 2, pp. 180–194, Feb. 2008, doi: 10.1016/j.apnum.2006.11.010.

[16] D. Belyi, E. Popova, D. P. Morton, and P. Damien, “Bayesian failure-rate modeling and preventive maintenance optimization,” *Eur. J. Oper. Res.*, vol. 262, no. 3, pp. 1085–1093, Nov. 2017, doi: 10.1016/j.ejor.2017.04.019.

[17] E. Sezer, D. Romero, F. Guedea, M. MacChi, and C. Emmanouilidis, “An Industry 4.0-Enabled Low Cost Predictive Maintenance Approach for SMEs,” Aug. 2018, doi: 10.1109/ICE.2018.8436307.

[18] J. Lee, J. Ni, D. Djurdjanovic, H. Qiu, and H. Liao, “Intelligent prognostics tools and e-maintenance,” *Comput. Ind.*, vol. 57, no. 6, pp. 476–489, Aug. 2006, doi: 10.1016/j.compind.2006.02.014.

[19] K. S. Kiangala and Z. Wang, “Initiating predictive maintenance for a conveyor motor in a bottling plant using industry 4.0 concepts,” *Int. J. Adv. Manuf. Technol.*, vol. 97, no. 9–12, pp. 3251–3271, May 2018, doi: 10.1007/s00170-018-2093-8.

[20] “ITEA 4 · Welcome.” https://itea4.org/ (accessed Dec. 18, 2021).

[21] “ITEA 4 · Project · 19022 Muwo.” https://itea4.org/project/muwo.html (accessed Dec. 17, 2021).

[22] “Muwo.” https://muwo.accuro.es/ (accessed Oct. 22, 2022).

[23] “SPEAR - Smart Prognosis of Energy with Allocation of Resources.” https://spear-project.eu/ (accessed Dec. 18, 2021).

[24] “ITEA 4 · Project · 16001 SPEAR.” https://itea4.org/project/spear.html (accessed Oct. 22, 2022).

[25] A. Zimmerling and X. Chen, “Innovation and possible long-term impact driven by COVID-19: Manufacturing, personal protective equipment and digital technologies,” *Technol. Soc.*, vol. 65, p. 101541, May 2021, doi: 10.1016/j.techsoc.2021.101541.

[26] C. Palhais, “PROTOTIPAGEM: Uma abordagemao processo de desenvolvimento de um produto,” 2015.

[27] M. Madhav, “Software Engineering | Prototyping Model - GeeksforGeeks,” *Geeks for Geeks*, 2020. https://www.geeksforgeeks.org/software-engineering-prototyping-model/ (accessed Dec. 18, 2021).

[28] S. D. Nelson, G. Del Fiol, H. Hanseler, B. I. Crouch, and M. R. Cummins, “A case report of refining user requirements for a health information exchange dashboard,” *Appl. Clin. Inform.*, vol. 7, no. 1, pp. 22–32, 2016, doi: 10.4338/ACI-2015-07-CR-0091.

[29] G. van Rossum and T. Python development team, “Python Tutorial Release 3.7.0 Guido van Rossum and the Python development team,” 2018.

[30] F. Chollet and & O., “Keras: the Python deep learning API,” *Keras: the Python deep learning API*, 2020. https://keras.io/ (accessed Jan. 25, 2022).

[31] “scikit-learn: machine learning in Python — scikit-learn 1.0.2 documentation.” https://scikit-learn.org/stable/ (accessed Jan. 25, 2022).

[32] “SciPy documentation — SciPy v1.9.2 Manual.” https://docs.scipy.org/doc/scipy/index.html (accessed Oct. 11, 2022).

[33] I. H. Afefy, “Reliability-Centered Maintenance Methodology and Application: A Case Study,” *Engineering*, vol. 02, no. 11, pp. 863–873, Nov. 2010, doi: 10.4236/eng.2010.211109.

[34] S. Matzka, “UCI Machine Learning Repository: AI4I 2020 Predictive Maintenance Dataset Data Set,” 2020. https://archive.ics.uci.edu/ml/datasets/AI4I+2020+Predictive+Maintenance+Dataset (accessed Jan. 23, 2022).

[35] B. Mota, L. Gomes, P. Faria, C. Ramos, Z. Vale, and R. Correia, “Production line optimization to minimize energy cost and participate in demand response events,” *Energies*, vol. 14, no. 2, p. 462, Jan. 2021, doi: 10.3390/en14020462.

[36] C. Ramos, R. Barreto, B. Mota, L. Gomes, P. Faria, and Z. Vale, “Scheduling of a textile production line integrating PV generation using a genetic algorithm,” *Energy Reports*, vol. 6, pp. 148–154, Dec. 2020, doi: 10.1016/j.egyr.2020.11.093.

[37] B. Mota, P. Faria, and Z. Vale, “Residential load shifting in demand response events for bill

reduction using a genetic algorithm,” *Energy*, vol. 260, p. 124978, Dec. 2022, doi: 10.1016/J.ENERGY.2022.124978.

[38] B. Mota, P. Faria, and C. Ramos, “Predictive Maintenance for Maintenance-Effective Manufacturing Using Machine Learning Approaches,” Springer, Cham, 2023, pp. 13–22.

[39] “The 17th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2022) – Salamanca, Spain.” http://2022.sococonference.eu/ (accessed Oct. 10, 2022).

[40] “The 9th International Conference on Energy and Environment Research (ICEER 2022): ‘Greening Energy to Shape a Sustainable Future’ – Porto, Portugal.” http://iceer.net/ (accessed Oct. 10, 2022).

[41] M. J. Page *et al.*, “PRISMA 2020 explanation and elaboration: Updated guidance and exemplars for reporting systematic reviews,” *The BMJ*, vol. 372. British Medical Journal Publishing Group, Mar. 29, 2021, doi: 10.1136/bmj.n160.

[42] J. Xie, L. Gao, K. Peng, X. Li, and H. Li, “Review on flexible job shop scheduling,” *IET Collab. Intell. Manuf.*, vol. 1, no. 3, pp. 67–77, Sep. 2019, doi: 10.1049/IET-CIM.2018.0009.

[43] H. J. Jabir, J. Teh, D. Ishak, and H. Abunima, “Impacts of Demand-Side Management on Electrical Power Systems: A Review,” *Energies 2018, Vol. 11, Page 1050*, vol. 11, no. 5, p. 1050, Apr. 2018, doi: 10.3390/EN11051050.

[44] T. Zhou, D. Tang, H. Zhu, and L. Wang, “Reinforcement Learning with Composite Rewards for Production Scheduling in a Smart Factory,” *IEEE Access*, vol. 9, pp. 752–766, 2021, doi: 10.1109/ACCESS.2020.3046784.

[45] X. Huang, S. H. Hong, M. Yu, Y. DIng, and J. Jiang, “Demand Response Management for Industrial Facilities: A Deep Reinforcement Learning Approach,” *IEEE Access*, vol. 7, pp. 82194–82205, 2019, doi: 10.1109/ACCESS.2019.2924030.

[46] J. Wang, Y. Shi, and Y. Zhou, “Intelligent demand response for industrial energy management considering thermostatically controlled loads and evs,” *IEEE Trans. Ind. Informatics*, vol. 15, no. 6, pp. 3432–3442, Jun. 2019, doi: 10.1109/TII.2018.2875866.

[47] I. Djordjevic, D. Petrovic, and G. Stojic, “A fuzzy linear programming model for aggregated production planning (APP)in the automotive industry,” *Comput. Ind.*, vol. 110, pp. 48–63, Sep. 2019, doi: 10.1016/j.compind.2019.05.004.

[48] M. A. Shehzad Hassan, M. Chen, H. Lin, M. H. Ahmed, M. Z. Khan, and G. R. Chughtai, “Optimization modeling for dynamic price based demand response in microgrids,” *J. Clean. Prod.*, vol. 222, pp. 231–241, Jun. 2019, doi: 10.1016/j.jclepro.2019.03.082.

[49] J. Ding, S. Schulz, L. Shen, U. Buscher, and Z. Lü, “Energy aware scheduling in flexible flow shops with hybrid particle swarm optimization,” *Comput. Oper. Res.*, vol. 125, p. 105088, Jan. 2021, doi: 10.1016/j.cor.2020.105088.

[50] K. Roy, K. K. Mandal, and A. C. Mandal, “Ant-Lion Optimizer algorithm and recurrent neural network for energy management of micro grid connected system,” *Energy*, vol. 167, pp. 402–416, Jan. 2019, doi: 10.1016/j.energy.2018.10.153.

[51] A. A. Abdelsalam, H. A. Zedan, and A. A. ElDesouky, “Energy Management of Microgrids Using Load Shifting and Multi-agent System,” *J. Control. Autom. Electr. Syst.*, vol. 31, no. 4, pp. 1015–1036, Aug. 2020, doi: 10.1007/s40313-020-00593-w.

[52] I. Ullah, I. Hussain, and M. Singh, “Exploiting grasshopper and cuckoo search bio-inspired optimization algorithms for industrial energy management system: Smart industries,” *Electron.*, vol. 9, no. 1, p. 105, Jan. 2020, doi: 10.3390/electronics9010105.

[53] A. Ziadeh, L. Abualigah, M. A. Elaziz, C. B. Şahin, A. A. Almazroi, and M. Omari, “Augmented grasshopper optimization algorithm by differential evolution: a power scheduling application in smart homes,” *Multimed. Tools Appl.*, vol. 80, no. 21–23, pp. 31569–31597, Sep. 2021, doi: 10.1007/s11042-021-11099-1.

[54] Z. Zhang, L. Wu, T. Peng, and S. Jia, “An improved scheduling approach for minimizing total energy consumption and makespan in a flexible job shop environment,” *Sustain.*, vol. 11, no. 1, p. 179, Dec. 2019, doi: 10.3390/su11010179.

[55] W. Cui, L. Li, and Z. Lu, “Energy-efficient scheduling for sustainable manufacturing systems with renewable energy resources,” *Nav. Res. Logist.*, vol. 66, no. 2, pp. 154–173, Mar. 2019, doi: 10.1002/nav.21830.

[56] T. Jiang, C. Zhang, H. Zhu, and G. Deng, “Energy-efficient scheduling for a job shop using grey wolf optimization algorithm with double-searching mode,” *Math. Probl. Eng.*, vol. 2018, 2018, doi: 10.1155/2018/8574892.

[57] T. Jiang, C. Zhang, H. Zhu, J. Gu, and G. Deng, “Energy-efficient scheduling for a job shop using an improved whale optimization algorithm,” *Mathematics*, vol. 6, no. 11, p. 220, Oct. 2018, doi: 10.3390/math6110220.

[58] J. Huang, Q. Chang, and J. Arinez, “Deep reinforcement learning based preventive maintenance policy for serial production lines,” *Expert Syst. Appl.*, vol. 160, p. 113701, Dec. 2020, doi: 10.1016/j.eswa.2020.113701.

[59] H. Yang, W. Li, and B. Wang, “Joint optimization of preventive maintenance and production scheduling for multi-state production systems based on reinforcement learning,” *Reliab. Eng. Syst. Saf.*, vol. 214, p. 107713, Oct. 2021, doi: 10.1016/j.ress.2021.107713.

[60] D. Lin, B. Jin, and D. Chang, “A PSO approach for the integrated maintenance model,” *Reliab.*

Eng. Syst. Saf., vol. 193, p. 106625, Jan. 2020, doi: 10.1016/j.ress.2019.106625.

[61] A. K. Mishra, D. Shrivastava, D. Tarasia, and A. Rahim, "Joint optimization of production scheduling and group preventive maintenance planning in multi-machine systems," *Ann. Oper. Res.*, pp. 1–44, Dec. 2021, doi: 10.1007/S10479-021-04362-Z/TABLES/14.

[62] H. Feng, L. Xi, L. Xiao, T. Xia, and E. Pan, "Imperfect preventive maintenance optimization for flexible flowshop manufacturing cells considering sequence-dependent group scheduling," *Reliab. Eng. Syst. Saf.*, vol. 176, pp. 218–229, Aug. 2018, doi: 10.1016/j.ress.2018.04.004.

[63] C. M. La Fata and G. Passannanti, "A simulated annealing-based approach for the joint optimization of production/inventory and preventive maintenance policies," *Int. J. Adv. Manuf. Technol.*, vol. 91, no. 9–12, pp. 3899–3909, Feb. 2017, doi: 10.1007/s00170-017-0053-3.

[64] Z. Zhang and Q. Tang, "Integrating flexible preventive maintenance activities into two-stage assembly flow shop scheduling with multiple assembly machines," *Comput. Ind. Eng.*, vol. 159, p. 107493, Sep. 2021, doi: 10.1016/j.cie.2021.107493.

[65] A. Moritz, S. Dauzere-Peres, O. Ben-Ammar, and P. Vialletelle, "Maintenance with Production Planning Constraints in Semiconductor Manufacturing," in *Proceedings - Winter Simulation Conference*, Dec. 2020, vol. 2020-Decem, pp. 1921–1930, doi: 10.1109/WSC48552.2020.9383976.

[66] L. Li, Y. Wang, and K. Y. Lin, "Preventive maintenance scheduling optimization based on opportunistic production-maintenance synchronization," *J. Intell. Manuf.*, vol. 32, no. 2, pp. 545–558, Feb. 2021, doi: 10.1007/s10845-020-01588-9.

[67] S. Ozcan and F. Simsir, "A new model based on Artificial Bee Colony algorithm for preventive maintenance with replacement scheduling in continuous production lines," *Eng. Sci. Technol. an Int. J.*, vol. 22, no. 6, pp. 1175–1186, Dec. 2019, doi: 10.1016/j.jestch.2019.08.003.

[68] H. Krim, N. Zufferey, J.-Y. Potvin, R. Benmansour, and D. Duvivier, "Tabu search for a parallel-machine scheduling problem with periodic maintenance, job rejection and weighted sum of completion times," *J. Sched.*, pp. 1–17, Oct. 2021, doi: 10.1007/s10951-021-00711-9.

[69] L. S. Tavassoli, R. Massah, A. Montazeri, M. Mirmozaffari, G. J. Jiang, and H. X. Chen, "A New Multiobjective Time-Cost Trade-Off for Scheduling Maintenance Problem in a Series-Parallel System," *Math. Probl. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/5583125.

[70] A. Lambora, K. Gupta, and K. Chopra, "Genetic Algorithm- A Literature Review," in *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Prespectives and Prospects, COMITCon 2019*, Feb. 2019, pp. 380–384, doi: 10.1109/COMITCon.2019.8862255.

[71] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimed. Tools Appl.*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021, doi: 10.1007/s11042-020-10139-6.

[72] A. Kumar, "Genetic Algorithms - GeeksforGeeks." https://www.geeksforgeeks.org/genetic-algorithms/ (accessed Dec. 26, 2021).

[73] V. Mallawaarachchi, "Introduction to Genetic Algorithms — Including Example Code," *Towards*

Data Science, 2017. https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3 (accessed Dec. 26, 2021).

[74] D. Whitley, “A genetic algorithm tutorial,” *Stat. Comput.*, vol. 4, no. 2, pp. 65–85, Jun. 1994, doi: 10.1007/BF00175354.

[75] J. H. Holland, “Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence,” *Adapt. Nat. Artif. Syst.*, Jan. 1992, doi: 10.7551/MITPRESS/1090.001.0001.

[76] “Theory of Evolution | National Geographic Society.” https://www.nationalgeographic.org/encyclopedia/theory-evolution/ (accessed Dec. 26, 2021).

[77] C. Bharathi, D. Rekha, and V. Vijayakumar, “Genetic Algorithm Based Demand Side Management for Smart Grid,” *Wirel. Pers. Commun.*, vol. 93, no. 2, pp. 481–502, 2017, doi: 10.1007/s11277-017-3959-z.

[78] Y. Wang *et al.*, “Energy management of smart micro-grid with response loads and distributed generation considering demand response,” *J. Clean. Prod.*, vol. 197, pp. 1069–1083, Oct. 2018, doi: 10.1016/j.jclepro.2018.06.271.

[79] S. Zhang and Y. Tang, “Optimal schedule of grid-connected residential PV generation systems with battery storages under time-of-use and step tariffs,” *J. Energy Storage*, vol. 23, pp. 175–182, Jun. 2019, doi: 10.1016/j.est.2019.01.030.

[80] U. Asgher *et al.*, “Smart energy optimization using heuristic algorithm in smart grid with integration of solar energy sources,” *Energies*, vol. 11, no. 12, p. 3494, Dec. 2018, doi: 10.3390/en11123494.

[81] Y. Wang, L. Fu, Y. Su, Q. Yang, and L. Wu, “Genetic algorithm in flexible work shop scheduling based on multi-objective optimization,” *J. Interdiscip. Math.*, vol. 21, no. 5, pp. 1249–1254, Jul. 2018, doi: 10.1080/09720502.2018.1495398.

[82] W. Xu, Y. Hu, W. Luo, L. Wang, and R. Wu, “A multi-objective scheduling method for distributed and flexible job shop based on hybrid genetic algorithm and tabu search considering operation outsourcing and carbon emission,” *Comput. Ind. Eng.*, vol. 157, p. 107318, Jul. 2021, doi: 10.1016/j.cie.2021.107318.

[83] G. Zhang, Y. Hu, J. Sun, and W. Zhang, “An improved genetic algorithm for the flexible job shop scheduling problem with multiple time constraints,” *Swarm Evol. Comput.*, vol. 54, p. 100664, May 2020, doi: 10.1016/j.swevo.2020.100664.

[84] H. Mokhtari and A. Hasani, “An energy-efficient multi-objective optimization for flexible job-shop scheduling problem,” *Comput. Chem. Eng.*, vol. 104, pp. 339–352, Sep. 2017, doi: 10.1016/j.compchemeng.2017.05.004.

[85] S. Nguyen, Y. Mei, and M. Zhang, “Genetic programming for production scheduling: a survey with a unified framework,” *Complex Intell. Syst.*, vol. 3, no. 1, pp. 41–66, Feb. 2017, doi: 10.1007/s40747-017-0036-x.

[86] M. Akbar and T. Irohara, “Scheduling for sustainable manufacturing: A review,” *Journal of Cleaner Production*, vol. 205. Elsevier, pp. 866–883, Dec. 20, 2018, doi: 10.1016/j.jclepro.2018.09.100.

[87] M. K. Amjad *et al.*, “Recent Research Trends in Genetic Algorithm Based Flexible Job Shop Scheduling Problems,” *Mathematical Problems in Engineering*, vol. 2018. Hindawi Limited, 2018, doi: 10.1155/2018/9270802.

[88] H. Li, S. Mi, Q. Li, X. Wen, D. Qiao, and G. Luo, “A scheduling optimization method for maintenance, repair and operations service resources of complex products,” *J. Intell. Manuf.*, vol. 31, no. 7, pp. 1673–1691, Oct. 2020, doi: 10.1007/s10845-018-1400-4.

[89] H. Gholizadeh, H. Fazlollahtabar, A. M. Fathollahi-Fard, and M. A. Dulebenets, “Preventive maintenance for the flexible flowshop scheduling under uncertainty: a waste-to-energy system,” *Environ. Sci. Pollut. Res.*, pp. 1–20, Sep. 2021, doi: 10.1007/s11356-021-16234-x.

[90] C. Duan, C. Deng, Q. Gong, and Y. Wang, “Optimal failure mode-based preventive maintenance scheduling for a complex mechanical device,” *Int. J. Adv. Manuf. Technol.*, vol. 95, no. 5–8, pp. 2717–2728, Nov. 2018, doi: 10.1007/s00170-017-1419-2.

[91] T. Wu, X. Ma, L. Yang, and Y. Zhao, “Proactive maintenance scheduling in consideration of imperfect repairs and production wait time,” *J. Manuf. Syst.*, vol. 53, pp. 183–194, Oct. 2019, doi: 10.1016/j.jmsy.2019.09.011.

[92] J. Seif, A. J. Yu, and F. Rahmanniyay, “Modelling and optimization of a bi-objective flow shop scheduling with diverse maintenance requirements,” *Int. J. Prod. Res.*, vol. 56, no. 9, pp. 3204–3225, May 2018, doi: 10.1080/00207543.2017.1403660.

[93] Y. Bensmain, M. Dahane, M. Bennekrouf, and Z. Sari, “Preventive remanufacturing planning of production equipment under operational and imperfect maintenance constraints: A hybrid genetic algorithm based approach,” *Reliab. Eng. Syst. Saf.*, vol. 185, pp. 546–566, May 2019, doi: 10.1016/j.ress.2018.09.001.

[94] M. Sheikhalishahi, N. Eskandari, A. Mashayekhi, and A. Azadeh, “Multi-objective open shop scheduling by considering human error and preventive maintenance,” *Appl. Math. Model.*, vol. 67, pp. 573–587, Mar. 2019, doi: 10.1016/j.apm.2018.11.015.

[95] G. Levitin, L. Xing, and Y. Dai, “Joint optimal mission aborting and replacement and maintenance scheduling in dual-unit standby systems,” *Reliab. Eng. Syst. Saf.*, vol. 216, p. 107921, Dec. 2021, doi:

10.1016/j.ress.2021.107921.

[96] E. I. Basri, I. H. A. Razak, H. Ab-Samat, and S. Kamaruddin, “Preventive maintenance (PM) planning: A review,” *J. Qual. Maint. Eng.*, vol. 23, no. 2, pp. 114–143, 2017, doi: 10.1108/JQME-04-2016-0014.

[97] A. Boudjelida, “On the robustness of joint production and maintenance scheduling in presence of uncertainties,” *J. Intell. Manuf.*, vol. 30, no. 4, pp. 1515–1530, Apr. 2019, doi: 10.1007/s10845-017-1303-9.

[98] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, D. S. Rajput, R. Kaluri, and G. Srivastava, “Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis,” *Evol. Intell.*, vol. 13, no. 2, pp. 185–196, Jun. 2020, doi: 10.1007/s12065-019-00327-1.

[99] P. Dwivedi, V. Kant, and K. K. Bharadwaj, “Learning path recommendation based on modified variable length genetic algorithm,” *Educ. Inf. Technol.*, vol. 23, no. 2, pp. 819–836, Mar. 2018, doi: 10.1007/s10639-017-9637-7.

[100] B. Mota *et al.*, “Climatization and luminosity optimization of buildings using genetic algorithm, random forest, and regression models,” *Energy Informatics*, vol. 4, no. 2, pp. 1–18, Sep. 2021, doi: 10.1186/s42162-021-00151-x.

[101] S. Ijaz, F. A. Hashmi, S. Asghar, and M. Alam, “Vector Based Genetic Algorithm to optimize predictive analysis in network security,” *Appl. Intell.*, vol. 48, no. 5, pp. 1086–1096, May 2018, doi: 10.1007/s10489-017-1026-9.

[102] C. Lamini, S. Benhlima, and A. Elbekri, “Genetic algorithm based approach for autonomous mobile robot path planning,” in *Procedia Computer Science*, Jan. 2018, vol. 127, pp. 180–189, doi: 10.1016/j.procs.2018.01.113.

[103] Y. Wu, X. Li, Q. Liu, and G. Tong, “The Analysis of Credit Risks in Agricultural Supply Chain Finance Assessment Model Based on Genetic Algorithm and Backpropagation Neural Network,” *Comput. Econ.*, pp. 1–24, Jun. 2021, doi: 10.1007/s10614-021-10137-2.

[104] S. K. Roy and D. De, “Genetic Algorithm based Internet of Precision Agricultural Things (IopaT) for Agriculture 4.0,” *Internet of Things*, p. 100201, Mar. 2020, doi: 10.1016/j.iot.2020.100201.

[105] A. Sharma, G. S. Yadava, and S. G. Deshmukh, “A literature review and future

perspectives on maintenance optimization," *Journal of Quality in Maintenance Engineering*, vol. 17, no. 1. Emerald Group Publishing Limited, pp. 5–25, Mar. 29, 2011, doi: 10.1108/13552511111116222.

[106] M. Faccio, A. Persona, F. Sgarbossa, and G. Zanin, "Industrial maintenance policy development: A quantitative framework," *Int. J. Prod. Econ.*, vol. 147, no. PART A, pp. 85–93, Jan. 2014, doi: 10.1016/j.ijpe.2012.08.018.

[107] A. Garg and S. G. Deshmukh, "Maintenance management: Literature review and directions," *Journal of Quality in Maintenance Engineering*, vol. 12, no. 3. Emerald Group Publishing Limited, pp. 205–238, Jul. 01, 2006, doi: 10.1108/13552510610685075.

[108] J. H. Shin and H. B. Jun, "On condition based maintenance policy," *J. Comput. Des. Eng.*, vol. 2, no. 2, pp. 119–127, Apr. 2015, doi: 10.1016/j.jcde.2014.12.006.

[109] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. da P. Francisco, J. P. Basto, and S. G. S. Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," *Comput. Ind. Eng.*, vol. 137, p. 106024, Nov. 2019, doi: 10.1016/j.cie.2019.106024.

[110] S. Ayvaz and K. Alpay, "Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time," *Expert Syst. Appl.*, vol. 173, p. 114598, Jul. 2021, doi: 10.1016/j.eswa.2021.114598.

[111] M. Paolanti, L. Romeo, A. Felicetti, A. Mancini, E. Frontoni, and J. Loncarski, "Machine Learning approach for Predictive Maintenance in Industry 4.0," Aug. 2018, doi: 10.1109/MESA.2018.8449150.

[112] R. M. Souza, E. G. S. Nascimento, U. A. Miranda, W. J. D. Silva, and H. A. Lepikson, "Deep learning for diagnosis and classification of faults in industrial rotating machinery," *Comput. Ind. Eng.*, vol. 153, p. 107060, Mar. 2021, doi: 10.1016/j.cie.2020.107060.

[113] X. Bampoula, G. Siaterlis, N. Nikolakis, and K. Alexopoulos, "A deep learning model for predictive maintenance in cyber-physical production systems using LSTM autoencoders," *Sensors (Switzerland)*, vol. 21, no. 3, pp. 1–14, Feb. 2021, doi: 10.3390/s21030972.

[114] M. Baptista, S. Sankararaman, I. P. de Medeiros, C. Nascimento, H. Prendinger, and E. M. P. Henriques, "Forecasting fault events for predictive maintenance using data-driven techniques and ARMA modeling," *Comput. Ind. Eng.*, vol. 115, pp. 41–53, Jan. 2018, doi: 10.1016/j.cie.2017.10.033.

[115] A. Lasisi and N. Attoh-Okine, "Principal components analysis and track quality index: A machine learning approach," *Transp. Res. Part C Emerg. Technol.*, vol. 91, pp. 230–248, Jun. 2018, doi: 10.1016/j.trc.2018.04.001.

[116] N. Amruthnath and T. Gupta, "A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance," in *2018 5th International Conference on Industrial Engineering and Applications, ICIEA 2018*, Jun. 2018, pp. 355–361, doi: 10.1109/IEA.2018.8387124.

[117] B. Ahmad, B. K. Mishra, M. Ghufran, Z. Pervez, and N. Ramzan, "Intelligent Predictive Maintenance Model for Rolling Components of a Machine based on Speed and Vibration," in *3rd*

International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2021, Apr. 2021, pp. 459–464, doi: 10.1109/ICAIIC51459.2021.9415249.

[118] W. Udo and Y. Muhammad, "Data-Driven Predictive Maintenance of Wind Turbine Based on SCADA Data," *IEEE Access*, vol. 9, pp. 162370–162388, Dec. 2021, doi: 10.1109/ACCESS.2021.3132684.

[119] S. Trivedi, S. Bhola, A. Talegaonkar, P. Gaur, and S. Sharma, "Predictive Maintenance of Air Conditioning Systems Using Supervised Machine Learning," Dec. 2019, doi: 10.1109/ISAP48318.2019.9065995.

[120] G. Mazzuto, S. Antomarioni, F. E. Ciarapica, and M. Bevilacqua, "Health Indicator for Predictive Maintenance Based on Fuzzy Cognitive Maps, Grey Wolf, and K-Nearest Neighbors Algorithms," *Math. Probl. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/8832011.

[121] M. Benker, A. Bliznyuk, and M. F. Zaeh, "A Gaussian Process Based Method for Data-Efficient Remaining Useful Life Estimation," *IEEE Access*, vol. 9, pp. 137470–137482, 2021, doi: 10.1109/ACCESS.2021.3116813.

[122] S. Butte, A. R. Prashanth, and S. Patil, "Machine Learning Based Predictive Maintenance Strategy: A Super Learning Approach with Deep Neural Networks," in *2018 IEEE Workshop on Microelectronics and Electron Devices, WMED 2018*, May 2018, pp. 1–5, doi: 10.1109/WMED.2018.8360836.

[123] A. Kumar, R. Shankar, and L. S. Thakur, "A big data driven sustainable manufacturing framework for condition-based maintenance prediction," *J. Comput. Sci.*, vol. 27, pp. 428–439, Jul. 2018, doi: 10.1016/j.jocs.2017.06.006.

[124] H. Chen, J. Y. Hsu, J. Y. Hsieh, H. Y. Hsu, C. H. Chang, and Y. J. Lin, "Predictive maintenance of abnormal wind turbine events by using machine learning based on condition monitoring for anomaly detection," *J. Mech. Sci. Technol.*, vol. 35, no. 12, pp. 5323–5333, Dec. 2021, doi: 10.1007/s12206-021-1105-z.

[125] O. I. Abiodun *et al.*, "Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition," *IEEE Access*, vol. 7. Institute of Electrical and Electronics Engineers Inc., pp. 158820–158846, 2019, doi: 10.1109/ACCESS.2019.2945545.

[126] R. Bala and D. Kumar, "Classification Using ANN: A Review," 2017. Accessed: Dec. 28,

2021. [Online]. Available: http://www.ripublication.com.

[127] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, Nov. 1958, doi: 10.1037/h0042519.

[128] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, Dec. 1943, doi: 10.1007/BF02478259.

[129] D.O. Hebb's, "The Organization of Behavior," *Wiley, New York*, vol. 50, no. 5–6. Elsevier, p. 437, Nov. 01, 1949, doi: 10.1016/S0361-9230(99)00182-3.

[130] O. Koca, O. T. Kaymakci, and M. Mercimek, "Advanced Predictive Maintenance with Machine Learning Failure Estimation in Industrial Packaging Robots," in *2020 15th International Conference on Development and Application Systems, DAS 2020 - Proceedings*, May 2020, pp. 1–6, doi: 10.1109/DAS49615.2020.9108913.

[131] V. Kavana and M. Neethi, "Fault Analysis and Predictive Maintenance of Induction Motor Using Machine Learning," in *3rd International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques, ICEECCOT 2018*, Dec. 2018, pp. 963–966, doi: 10.1109/ICEECCOT43722.2018.9001543.

[132] W. J. Lee, H. Wu, H. Yun, H. Kim, M. B. G. Jun, and J. W. Sutherland, "Predictive maintenance of machine tool systems using artificial intelligence techniques applied to machine condition data," in *Procedia CIRP*, Jan. 2019, vol. 80, pp. 506–511, doi: 10.1016/j.procir.2018.12.019.

[133] N. S. Rosli, N. R. Ain Burhani, and R. Ibrahim, "Predictive Maintenance of Air Booster Compressor (ABC) Motor Failure using Artificial Neural Network trained by Particle Swarm Optimization," in *2019 IEEE Student Conference on Research and Development, SCOReD 2019*, Oct. 2019, pp. 11–16, doi: 10.1109/SCORED.2019.8896330.

[134] I. Ullah *et al.*, "Predictive maintenance of power substation equipment by infrared thermography using a machine-learning approach," *Energies*, vol. 10, no. 12, p. 1987, Dec. 2017, doi: 10.3390/en10121987.

[135] U. R. Muduli, K. Al Jaafari, and R. K. Behera, "Optimized Neural Network Based Predictive Maintenance for Five-Phase Induction Motor Failure," in *Proceedings of the Energy Conversion Congress and Exposition - Asia, ECCE Asia 2021*, May 2021, pp. 1537–1541, doi: 10.1109/ECCE-Asia49820.2021.9478989.

[136] M. De Benedetti, F. Leonardi, F. Messina, C. Santoro, and A. Vasilakos, "Anomaly detection and predictive maintenance for photovoltaic systems," *Neurocomputing*, vol. 310, pp. 59–68, Oct. 2018, doi: 10.1016/j.neucom.2018.05.017.

[137] A. Cachada *et al.*, "Maintenance 4.0: Intelligent and Predictive Maintenance System Architecture," in *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, Oct. 2018, vol. 2018-Septe, pp. 139–146, doi: 10.1109/ETFA.2018.8502489.

[138] M. S. Roy, R. Gupta, J. K. Chandra, K. Das Sharma, and A. Talukdar, "Improving Photoplethysmographic Measurements under Motion Artifacts Using Artificial Neural Network for Personal Healthcare," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 12, pp. 2820–2829, Dec. 2018, doi:

10.1109/TIM.2018.2829488.

[139] F. Widyahastuti and V. U. Tjhin, “Predicting students performance in final examination using linear regression and multilayer perceptron,” in *Proceedings - 2017 10th International Conference on Human System Interactions, HSI 2017*, Aug. 2017, pp. 188–192, doi: 10.1109/HSI.2017.8005026.

[140] A. Iqbal and S. Aftab, “A Feed-Forward and Pattern Recognition ANN Model for Network Intrusion Detection,” *Int. J. Comput. Netw. Inf. Secur.*, vol. 11, no. 4, pp. 19–25, 2019, doi: 10.5815/ijcnis.2019.04.03.

[141] G. Iannace, G. Ciaburro, and A. Trematerra, “Fault diagnosis for UAV blades using artificial neural network,” *Robotics*, vol. 8, no. 3, p. 59, Jul. 2019, doi: 10.3390/robotics8030059.

[142] W. Heo, J. M. Lee, N. Park, and J. E. Grable, “Using Artificial Neural Network techniques to improve the description and prediction of household financial ratios,” *J. Behav. Exp. Financ.*, vol. 25, p. 100273, Mar. 2020, doi: 10.1016/j.jbef.2020.100273.

[143] A. Etminan, A. Pour-Aboughadareh, R. Mohammadi, L. Shooshtari, M. Yousefiazarkhanian, and H. Moradkhani, “Determining the Best Drought Tolerance Indices using Artificial Neural Network (ANN): Insight into Application of Intelligent Agriculture in Agronomy and Plant breeding,” *Cereal Res. Commun.*, vol. 47, no. 1, pp. 170–181, Mar. 2019, doi: 10.1556/0806.46.2018.057.

[144] M. Ghaleb, S. Taghipour, and H. Zolfagharinia, “Real-time integrated production-scheduling and maintenance-planning in a flexible job shop with machine deterioration and condition-based maintenance,” *J. Manuf. Syst.*, vol. 61, pp. 423–449, Oct. 2021, doi: 10.1016/j.jmsy.2021.09.018.

[145] G. Cheng and L. Li, “Joint optimization of production, quality control and maintenance for serial-parallel multistage production systems,” *Reliab. Eng. Syst. Saf.*, vol. 204, p. 107146, Dec. 2020, doi: 10.1016/j.ress.2020.107146.

[146] L. Wang, Z. Lu, and X. Han, “Joint optimisation of production, maintenance and quality for batch production system subject to varying operational conditions,” *Int. J. Prod. Res.*, vol. 57, no. 24, pp. 7552–7566, Dec. 2019, doi: 10.1080/00207543.2019.1581956.

[147] F. Dababneh, L. Li, R. Shah, and C. Haefke, “Demand response-driven production and

maintenance decision-making for cost-effective manufacturing," *J. Manuf. Sci. Eng. Trans. ASME*, vol. 140, no. 6, Jun. 2018, doi: 10.1115/1.4039197/366768.

[148] I. H. Sin and B. Do Chung, "Bi-objective optimization approach for energy aware scheduling considering electricity cost and preventive maintenance using genetic algorithm," *J. Clean. Prod.*, vol. 244, p. 118869, Jan. 2020, doi: 10.1016/j.jclepro.2019.118869.

[149] C. Nzukam, A. Voisin, E. Levrat, D. Sauter, and B. Iung, "Opportunistic maintenance scheduling with stochastic opportunities duration in a predictive maintenance strategy," in *IFAC-PapersOnLine*, Jan. 2018, vol. 51, no. 11, pp. 453–458, doi: 10.1016/j.ifacol.2018.08.348.

[150] I. Paprocka, W. M. Kempa, and B. Skołud, "Predictive maintenance scheduling with reliability characteristics depending on the phase of the machine life cycle," *Eng. Optim.*, vol. 53, no. 1, pp. 165–183, 2021, doi: 10.1080/0305215X.2020.1714041.

[151] S. Zhai, B. Gehring, and G. Reinhart, "Enabling predictive maintenance integrated production scheduling by operation-specific health prognostics with generative deep learning," *J. Manuf. Syst.*, vol. 61, pp. 830–855, Oct. 2021, doi: 10.1016/j.jmsy.2021.02.006.

[152] J. Wang, H. Du, J. Xing, F. Qiao, and Y. Ma, "Novel Energy- And Maintenance-Aware Collaborative Scheduling for A Hybrid Flow Shop Based on Dual Memetic Algorithms," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5613–5620, Oct. 2020, doi: 10.1109/LRA.2020.3005626.

[153] K. Shen, T. De Pessemier, X. Gong, L. Martens, and W. Joseph, "Genetic optimization of energy-and failure-aware continuous production scheduling in pasta manufacturing," *Sensors (Switzerland)*, vol. 19, no. 2, p. 297, Jan. 2019, doi: 10.3390/s19020297.

[154] X. Chen, Y. An, Z. Zhang, and Y. Li, "An approximate nondominated sorting genetic algorithm to integrate optimization of production scheduling and accurate maintenance based on reliability intervals," *J. Manuf. Syst.*, vol. 54, pp. 227–241, Jan. 2020, doi: 10.1016/j.jmsy.2019.12.004.

[155] G. Ettaye, A. El Barkany, A. Jabri, and A. El Khalfi, "Optimizing the integrated production and maintenance planning using genetic algorithm," *Int. J. Eng. Bus. Manag.*, vol. 10, May 2018, doi: 10.1177/1847979018773260.

[156] C. Gu, Y. He, Z. Chen, X. Han, D. Zhou, and A. Zhang, "Mission reliability-oriented integrated scheduling optimization model by combining production planning and maintenance activities," *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.*, vol. 235, no. 5, pp. 839–849, Nov. 2021, doi: 10.1177/0954405420971080.

[157] R. Li and H. Ma, "Integrating Preventive Maintenance Planning and Production Scheduling under Reentrant Job Shop," *Math. Probl. Eng.*, vol. 2017, 2017, doi: 10.1155/2017/6758147.

[158] "sklearn.preprocessing.MinMaxScaler — scikit-learn 1.1.2 documentation." https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html (accessed Sep. 08, 2022).

[159] "sklearn.impute.KNNImputer — scikit-learn 1.1.2 documentation." https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html (accessed Sep. 08, 2022).

[160] “scipy.stats.zscore — SciPy v1.9.2 Manual.” https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.zscore.html#scipy-stats-zscore (accessed Oct. 11, 2022).

[161] “imbalanced-learn documentation — Version 0.9.0.” https://imbalanced-learn.org/stable/ (accessed Apr. 20, 2022).

[162] B. Ken Arnold, J. Gosling, D. Holmes Publisher, and A. Wesley Professional, “THE Java™ Programming Language, Fourth Edition,” 2005.

[163] B. W. Kernighan and D. M. Ritchie, “The C programming language,” 2006, [Online]. Available: http://citebay.com/how-to-cite/c/.

[164] B. Stroustrup, “The C ++ Programming Language Third Edition,” 1997.

[165] “SWI-Prolog.” https://www.swi-prolog.org/ (accessed Jan. 25, 2022).

[166] “Programming Languages Benchmarks.” https://attractivechaos.github.io/plb/plb-original.html (accessed Aug. 25, 2022).

[167] S. Nanz and C. A. Furia, “A comparative study of programming languages in rosetta code,” in *Proceedings - International Conference on Software Engineering*, Aug. 2015, vol. 1, pp. 778–788, doi: 10.1109/ICSE.2015.90.

[168] Anaconda Inc., “Anaconda | The World’s Most Popular Data Science Platform,” *Anaconda*, 2021. https://www.anaconda.com/ (accessed Jan. 29, 2022).

[169] “Visual Studio Code - Code Editing. Redefined.” https://code.visualstudio.com/ (accessed Aug. 25, 2022).

[170] “Google Colab.” https://research.google.com/colaboratory/faq.html (accessed Jan. 29, 2022).

[171] Mathworks, “MATLAB - Mathworks - MATLAB & Simulink,” *Www.Mathworks.Com*, 2016. https://www.mathworks.com/products/matlab.html (accessed Jan. 25, 2022).

[172] Deap Project, “DEAP documentation — DEAP 1.3.1 documentation,” 2020. https://deap.readthedocs.io/en/master/ (accessed Jan. 25, 2022).

[173] “PyGAD - Python Genetic Algorithm! — PyGAD 2.16.1 documentation.” https://pygad.readthedocs.io/en/latest/ (accessed Jan. 25, 2022).

[174] “pyeasyga — pyeasyga 0.3.1 documentation.” https://pyeasyga.readthedocs.io/en/latest/ (accessed Jan. 25, 2022).

[175] Pallets organization, “Welcome to Flask -- Flask Documentation (2.0.x),” *Pallets Projects*, 2010. https://flask.palletsprojects.com/en/2.0.x/ (accessed Jan. 29, 2022).

[176] “TensorFlow.” https://www.tensorflow.org/ (accessed Jan. 25, 2022).

[177] “PyTorch.” https://pytorch.org/ (accessed Jan. 25, 2022).

[178] “pandas - Python Data Analysis Library.” https://pandas.pydata.org/ (accessed Jan. 25, 2022).

[179] “NumPy.” https://numpy.org/ (accessed Jan. 25, 2022).

[180] “Mibel – Mercado Ibérico de Electricidade.” https://www.mibel.com/ (accessed Jan. 23, 2022).

[181] A. Frank and A. Asuncion, “{UCI} Machine Learning Repository.” 2010, [Online]. Available: https://archive.ics.uci.edu/ml/index.php.

[182] Y. Zhuo and Z. Ge, “Data Guardian: A Data Protection Scheme for Industrial Monitoring Systems,” *IEEE Trans. Ind. Informatics*, vol. 18, no. 4, pp. 2550–2559, Aug. 2021, doi: 10.1109/tii.2021.3103765.

[183] E. A. Mityukov, A. V. Zatonsky, P. V. Plekhov, and N. V. Bilfeld, “Phishing detection model using the hybrid approach to data protection in industrial control system,” in *IOP Conference Series: Materials Science and Engineering*, May 2019, vol. 537, no. 5, p. 052014, doi: 10.1088/1757-899X/537/5/052014.

[184] “sklearn.model_selection.GridSearchCV — scikit-learn 1.0.2 documentation.” https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html (accessed Apr. 20, 2022).

[185] “tf.keras.wrappers.scikit_learn.KerasClassifier | TensorFlow.” http://man.hubwiz.com/docset/TensorFlow.docset/Contents/Resources/Documents/api_docs/python/tf/keras/wrappers/scikit_learn/KerasClassifier.html (accessed Apr. 20, 2022).

[186] L. J. Wells, J. A. Camelio, C. B. Williams, and J. White, “Cyber-physical security challenges in manufacturing systems,” *Manuf. Lett.*, vol. 2, no. 2, pp. 74–77, 2014, doi: 10.1016/j.mfglet.2014.01.005.

[187] G. A. Jaafar, S. M. Abdullah, and S. Ismail, “Review of Recent Detection Methods for HTTP DDoS Attack,” *J. Comput. Networks Commun.*, vol. 2019, 2019, doi: 10.1155/2019/1283472.

[188] U. P. D. Ani, H. (Mary) He, and A. Tiwari, “Review of cybersecurity issues in industrial critical infrastructure: manufacturing in perspective,” *J. Cyber Secur. Technol.*, vol. 1, no. 1, pp. 32–74, Jan. 2017, doi: 10.1080/23742917.2016.1252211.

[189] V. Mullet, P. Sondi, and E. Ramat, “A Review of Cybersecurity Guidelines for Manufacturing Factories in Industry 4.0,” *IEEE Access*, vol. 9. Institute of Electrical and Electronics Engineers Inc., pp. 23235–23263, 2021, doi: 10.1109/ACCESS.2021.3056650.

[190] S. R. Chhetri, A. Canedo, and M. A. Al Faruque, “KCAD: Kinetic cyber-attack detection method for cyber-physi*gn, Digest of Technical Papers, ICCAD*, Nov. 2016, vol. 07-10-Nove, doi: 10.1145/2966986.2967050.

[191] Z. DeSmit, A. E. Elhabashy, L. J. Wells, and J. A. Camelio, “An approach to cyber-physical vulnerability assessment for intelligent manufacturing systems,” *J. Manuf. Syst.*, vol. 43, pp. 339–351, Apr. 2017, doi: 10.1016/j.jmsy.2017.03.004.

[192] Á. L. Perales Gómez, L. Fernández Maimó, A. Huertas Celdrán, F. J. García Clemente, M. Gil Pérez, and G. Martínez Pérez, “SafeMan: A unified framework to manage cybersecurity and safety in manufacturing industry,” in *Software - Practice and Experience*, Mar. 2021, vol. 51, no. 3, pp. 607–627, doi: 10.1002/spe.2879.

[193] Microsoft Corporation, “Microsoft Excel,” 2018. https://office.microsoft.com/excel (accessed Jan. 29, 2022).

[194] B. Mota, L. Gomes, P. Faria, C. Ramos, and Z. Vale, “Production line dataset for task scheduling and energy optimization - Schedule Optimization,” Nov. 2020, doi: 10.5281/ZENODO.4106746.

[195] B. Mota, “Joint Optimization of Production and Maintenance for Cost-effective Manufacturing and Demand Response Participation Dataset - Energy Cost Optimization with Energy Selling,” Sep. 2022, doi: 10.5281/ZENODO.7055698.

[196] B. Mota, “Joint Optimization of Production and Maintenance for Cost-effective Manufacturing and Demand Response Participation Dataset - Total Cost and Machine Occupancy Deviation Optimization,” Sep. 2022, doi: 10.5281/ZENODO.7055580.

[197] B. Mota, L. Gomes, P. Faria, C. Ramos, and Z. Vale, “Production line dataset for task scheduling and energy optimization - Demand Response Participation,” Nov. 2020, doi: 10.5281/ZENODO.4247232.

[198] “sklearn.ensemble.GradientBoostingClassifier — scikit-learn 1.0.2 documentation.” https://scikit-

learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html (accessed Apr. 29, 2022).

[199] "sklearn.svm.SVC — scikit-learn 1.0.2 documentation." https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html (accessed Apr. 29, 2022).

[200] "sklearn.ensemble.RandomForestClassifier — scikit-learn 1.0.2 documentation." https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (accessed Apr. 20, 2022).

[201] "sklearn.model_selection.RandomizedSearchCV — scikit-learn 1.0.2 documentation." https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html (accessed Apr. 20, 2022).

[202] S. Matzka, "Explainable Artificial Intelligence for Predictive Maintenance Applications," in *Proceedings - 2020 3rd International Conference on Artificial Intelligence for Industries, AI4I 2020*, Sep. 2020, pp. 69–74, doi: 10.1109/AI4I49448.2020.00023.

[203] B. Mota, "Joint Optimization of Production and Maintenance for Cost-effective Manufacturing and Demand Response Participation Dataset - Maintenance Optimization," Sep. 2022, doi: 10.5281/ZENODO.7055573.

[204] B. Mota, "Joint Optimization of Production and Maintenance for Cost-effective Manufacturing and Demand Response Participation Dataset - Machine Breakdown Event," Sep. 2022, doi: 10.5281/ZENODO.7055586.

[205] B. Mota, "Joint Optimization of Production and Maintenance for Cost-effective Manufacturing and Demand Response Participation Dataset - Joint Optimization of Production and Maintenance," Sep. 2022, doi: 10.5281/ZENODO.7060462.

[206] M. Khorram, P. Faria, Z. Vale, and C. Ramos, "Sequential tasks shifting for participation in demand response programs," *Energies*, vol. 13, no. 18, p. 4879, Sep. 2020, doi: 10.3390/en13184879.

[207] M. Khorram, P. Faria, O. Abrishambaf, and Z. Vale, "Consumption optimization in an office building considering flexible loads and user comfort," *Sensors (Switzerland)*, vol. 20, no. 3, p. 593, Jan. 2020, doi: 10.3390/s20030593.

[208] V. Vakkuri, K. K. Kemell, J. Kultanen, and P. Abrahamsson, "The Current State of Industrial Practice in Artificial Intelligence Ethics," *IEEE Softw.*, vol. 37, no. 4, pp. 50–57, Jul. 2020, doi: 10.1109/MS.2020.2985621.

[209] T. Hagendorff, "The Ethics of AI Ethics: An Evaluation of Guidelines," *Minds Mach.*, vol. 30, no. 1, pp. 99–120, Mar. 2020, doi: 10.1007/s11023-020-09517-8.

[210] A. F. Winfield, K. Michael, J. Pitt, and V. Evers, "Machine ethics: The design and governance of ethical ai and autonomous systems," *Proc. IEEE*, vol. 107, no. 3, pp. 509–517, Mar. 2019, doi: 10.1109/JPROC.2019.2900622.