DEVELOPING MACHINE LEARNING-BASED MODELS FOR OPTIMAL MAINTENANCE SCHEDULING WITHIN THE FOOD INDUSTRY

R. Oosthuizen[1*], C. Bisset[2] and C. Du Plessis[3]
[1,2,3]School of Industrial Engineering
North-West University, South Africa
[1]Ruhanoo13@gmail.com, [2]Chanelbisset26026856@gmail.com, [3]Chantelle.Coetzee@nwu.ac.za

ABSTRACT

In the context of the food industry, precisely the company used as a case study, unplanned machine downtimes are mainly caused by ineffective maintenance scheduling that impacts organisational profit. To address this challenge, the study explores predictive maintenance within the food industry and its application to reduce machine inefficiencies and improve overall decision-making. First, a theoretical background on predictive maintenance and machine learning is provided, followed by the development of the random forest and decision tree models. Company data is pre-processed, and the models are trained and tested using scientific methods from academic literature, including cross-validation and hyperparameter tuning. One-year future predictions are made for three retail line machines, aiding in proactive maintenance decision-making to reduce unplanned machine downtime. Subsequently, this study contributes towards academia and industry by providing actionable insights for optimising maintenance scheduling and production processes in the food industry.

Keywords: Machine learning, Maintenance scheduling, Food industry, Predictive maintenance

[*] Corresponding Author

1 INTRODUCTION

All machines, equipment, and devices responsible for producing products are bound to wear and tear. Records of maintenance activities date back to ancient Egyptian times. An old Egyptian document dated 600 b.c. mentions a stoppage of cedar wood required to maintain the sacred boat of Amun Ra [1]. The Industrial Revolution incentivised the need for maintenance strategies to be redesigned to adhere to the advancements in technology in the industrial sector. **Table 1** discusses the impact of the Industrial Revolution on maintenance practices.

Table 1: The correlation between the industrial revolutions and maintenance types [1]

Industrial revolution	Type of Maintenance	Overview of the correlation between the Industrial Revolution and maintenance type
Industry 1.0	Reactive	The first industrial revolution began in England, transforming energy sources, transportation, information transfer, and manufacturing. James Watt's invention of the steam engine in 1765 marked the beginning. The increased complexity of machines and increased productivity led to the evolution of maintenance methods. Reactive maintenance was costly and caused unanticipated downtime and productivity losses.
Industry 2.0	Preventive	The Second Industrial Revolution began in 1870, revolutionising society with mass manufacturing and inventions. As machines became more complex, preventative maintenance emerged to reduce downtime and improve equipment performance. Manufacturers now focus on routine inspections, maintenance, and repairs to prevent breakdowns and maintain equipment's overall performance.
Industry 3.0	Productive	Advancements in manufacturing, computer technology, and marketing and management procedures marked the Third Industrial Revolution. Productive Maintenance (PM) emerged after World War II, combining Corrective and Preventive Maintenance with an analytical, data-driven approach. PM improves equipment lifespans, reduces downtime, and reduces costs. Reliability-centered Maintenance (RCM) and Total Productive Maintenance (TPM) were developed during this time. RCM focuses on determining maintenance needs for physical assets, addressing issues like neglected maintenance or wear, and improving machine uptime, cost-effectiveness, and risk awareness. TPM, developed by Seichi Nakajima in 1971, emphasises employee involvement, efficiency, and machinery preservation. Both techniques aim to minimise downtime and mitigate accidents involving various professions, including maintainers and operators.
Industry 4.0	Predictive	The fourth industrial revolution is rapidly expanding due to the global interconnection of the Internet. Predictive maintenance, a method combining big data analytics and artificial intelligence, aims to identify patterns and trends in asset failure. It requires real-time monitoring and alarms based on predictive methods like regression analysis. Fundamental elements include sensors, cyber-physical

		systems, the Internet of Things, big data, cloud computing, networks, artificial intelligence, mobile networks, and WiFi.

This paper will specifically focus on predictive maintenance. An important predictive maintenance component is proactive maintenance. Instead of depending exclusively on set schedules or reactive repairs, it entails carrying out maintenance tasks proactively based on the equipment's predicted demands [2]. Maintenance staff may plan and schedule maintenance work, guaranteeing optimal resource allocation and minimal disturbance to production by predicting future equipment failures. Proactive maintenance combined with predictive analytics will be crucial in developing a company's predictive maintenance strategy plan.

Predictive systems are fundamentally dependent on continuous improvement. A predictive system is iteratively updated and improved based on feedback, performance reviews, and new data. Updating prediction models, modifying maintenance strategies, implementing new sensor technologies, and incorporating customer and equipment operator input are all examples of continuous improvement.

1.1 Types of machine learning techniques

Some standard predictive modelling techniques in machine learning include:

- **Linear regression:** This type of regression model falls within supervised learning. It is a statistical approach that simulates the relationship between one or more independent variables and the dependent variable [3]. The method finds the best-fitting line by fitting a linear equation to the training data and minimising the difference between the predicted and actual values. During training, the method adjusts the slope and intercept to reduce the mean squared error between the predicted and actual values.
- **Logistic regression:** This type of regression model falls within supervised learning. It is a statistical method that describes and explains correlations between binary dependent variables and one or more nominal, interval, or ratio-level independent variables. By using the known values of other variables and a discrete target variable, logistic regression enables you to forecast the unknown values of the target variable. The modelling technique is primarily used for binary classification tasks, where the objective is to predict the probability of an observation belonging to one or more classes [4]. The model uses the output obtained by the linear regression function as input and a sigmoid function (S-form curve) to estimate the probability for the specific class.
- **Decision trees:** A powerful and understandable predictive modelling method used for both classification and regression applications within supervised learning. It is a supervised learning technique with a tree-like structure where each leaf node represents the final prediction or class label, and each interior node represents a choice based on a feature [5]. Making predictions by making a series of binary decisions depending on the values of the input characteristics, decision trees divide the feature space into regions. Recursively building the tree with the most illuminating elements at the top creates a hierarchical collection of simple rules to grasp and comprehend.
- **Gradient boosted model:** This ensemble machine learning method is used for classification and regression tasks within supervised learning. Its foundation is boosting, which joins several decision trees to produce a more reliable and accurate prediction model. Gradient boosting builds decision trees progressively, each aiming to fix the mistakes caused by the one before it [6]. The model emphasises data items that prior trees misclassified or projected incorrectly to increase overall prediction accuracy. It is an effective model for handling complex, non-linear relationships in data due to its flexible nature, handling various data types and feature combinations.

- **Neural networks:** Artificial neural networks (ANNs), often known as neural networks, are a class of strong and adaptable machine learning models that take inspiration from the structure and operation of the human brain. The machine learning model uses both supervised and unsupervised learning. They can comprehend complicated patterns and correlations among the data and are especially effective at processing complex, high-dimensional data. Neurons, the linked units of neural networks, are arranged into three levels: an input layer, one or more hidden layers, and an output layer [7]. Every neuron receives an input, performs a mathematical transformation, and then generates an output that serves as the input for the layer below it. Weights are assigned to the connections between neurons. Backpropagation, which uses optimisation techniques to reduce the error between expected and actual results, adjusts these weights during training.
- **Random forest:** This is a popular and effective ensemble learning method used by classification and regression applications within supervised learning. It is a member of the bagging family of algorithms and combines the predictions of multiple decision trees to boost generalisation performance, decrease overfitting, and increase accuracy [8]. Each decision tree in a random forest is trained using a randomly chosen portion of the training data, and a randomly selected subset of characteristics is considered at each split point during tree building. The trees' variety and unpredictability contribute to the reduction of variance and increase the model's resistance to noisy or complicated input.

1.2 Case studies and applications

A few case studies regarding predictive maintenance will be discussed in this section. The real-life application of these studies will be discussed, and commonalities will be identified. The largest chemical business in the world, "Badische Anilin und Soda Fabrik" (BASF), translates to Baden Aniline and Soda Factory in English. The company has made digitalisation a part of its corporate strategy to streamline maintenance procedures and minimise unplanned downtime. To provide remote monitoring and control of its power distribution substation's operations and asset health, the Beaumont, Texas, facility used EcoStruxure Asset Advisor from Schneider Electric. The IoT-enabled solution uses asset sensors for continuous asset condition monitoring and predictive analytics to find possible asset failure concerns. By collaborating with Schneider Electric's Connected Services Hub, BASF additionally benefits from individualised, proactive guidance on avoiding breakdowns and enhancing maintenance strategies. Over 100 condition variables for 63 substation assets were gathered, measured, and computed in the use case. The digital dashboard offers 24/7 visibility into the substation's global health index and particular asset statuses [9]. The cloud-based system connects assets, enabling the plant to monitor the condition of the motors powering the process, motor control centres, and electrical distribution equipment. The plant can make informed decisions, optimise asset health, and increase the effectiveness of vital electrical distribution assets with round-the-clock data access and professional advice, which will eventually increase plant uptime, performance, productivity, and safety.

Alcoa's predictive maintenance strategy attempts to make the company more stable and profitable. At its Fjarðaál aluminium smelter in Iceland, a proof-of-concept (POC) experiment was conducted to automate predictive maintenance and decrease downtime and maintenance costs. The system is scaled to at least 1,000 assets in Fjarðaál and is enterprise-wide scalable. Senseye was chosen to integrate and synchronise the plant's OSIsoft PI ecosystem's equipment sensor data with maintenance data in its Oracle EAM solution. Without any settings or alert levels, the system automatically forms models and commences learning. Isolated peaks in raw data illustrate patterns and hidden failures. Predictive analytics determines what is happening and why, offering prognostic recommendations about the asset's remaining usable life. Operators are automatically alerted of situations that require attention and can further investigate the fault for more information. Alcoa decreased unscheduled machinery downtime by up to 20% and achieved full ROI in 4-6 months [9]. After completing the POC, the solution expanded to other sections of the business, emphasising identifying the best corporate reasons to continue.

Duke Energy Renewables, an owner/operator of wind and solar farms in the United States, has effectively automated the profiling and detection of faulty contactors on one of its wind turbines. The turbines have six contactors that ensure smooth generator ramp-up and synchronisation with the electrical grid. When any of the six contactors fails, the turbine experiences downtime for 2 to 10 days for diagnostic and repair. Turbine contactors fail more frequently at one location than at others. Automating the prediction of approaching failure would enable predictive maintenance, reduce downtime, lower maintenance costs, and improve spare parts inventory management.

A model was trained to search for the contactor fault error code and credible leading indicator signals using Seeq's automatic profiling tool. The model discovered 12 instances of the error code, most of which occurred within a few months preceding a breakdown [9]. No false positives existed before the outage, and no further fault codes or forecasts were generated once all contactors were replaced. Once approved, the model may run indefinitely and provide automated notification when something goes wrong. The results are encouraging, and Duke Energy Renewables intends to apply this model to the other turbines at the site and investigate additional failure mechanisms. The return on the capability to detect a generator before failure and respond proactively can save the company hundreds of thousands of dollars.

The case study uses the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology, which provides detailed guidance for implementing data mining and consists of the following six phases: Business understanding, data understanding, data preparation, modelling, evaluation, and deployment [10]. Data mining falls within data pre-processing and involves cleaning data to improve models' accuracy. The case mainly considers the prediction of failures and the root causes of two bottleneck machines in the engine component line. High dimensional data is obtained from a newly installed sensor system, control system (programable logic controller), production monitoring system, and maintenance system.

From the case studies, it is evident that predictive maintenance is an essential topic in all industries. Based on the current knowledge regarding the application of machine learning for predictive maintenance, this study aims to propose a predictive maintenance strategy plan as an example that will enable any company to make proactive decisions, effectively reducing machine inefficiencies experienced. Due to confidentiality, the data or the company name may not be displayed. However, the article will provide a guideline on how to use machine learning to enable proactive maintenance decisions.

2 RESEARCH METHOD

Figure 1 provides a holistic view of the inputs that were required, the process that was conducted, and the output that was delivered to achieve the research aim.

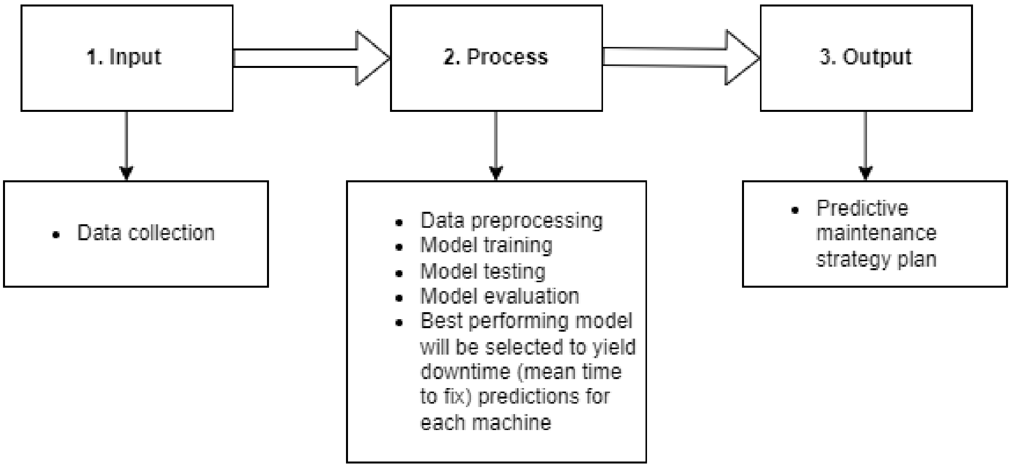

Figure 1: Holistic view of the road towards achieving the project's aim

Two machine-learning models were trained, validated, and tested on pre-processed datasets. The model that yielded the best results when comparing the performance metric results of the selected models was used to predict future mean time to fix (downtime) values for each machine within the production line. The machine learning model predictions were used to develop a predictive maintenance strategy plan with literature to assist the company used as a case study in making proactive maintenance decisions, reducing asset failures, and improving equipment longevity. The following approach/process will be used to develop the machine learning models for the various retail line machines [11]:

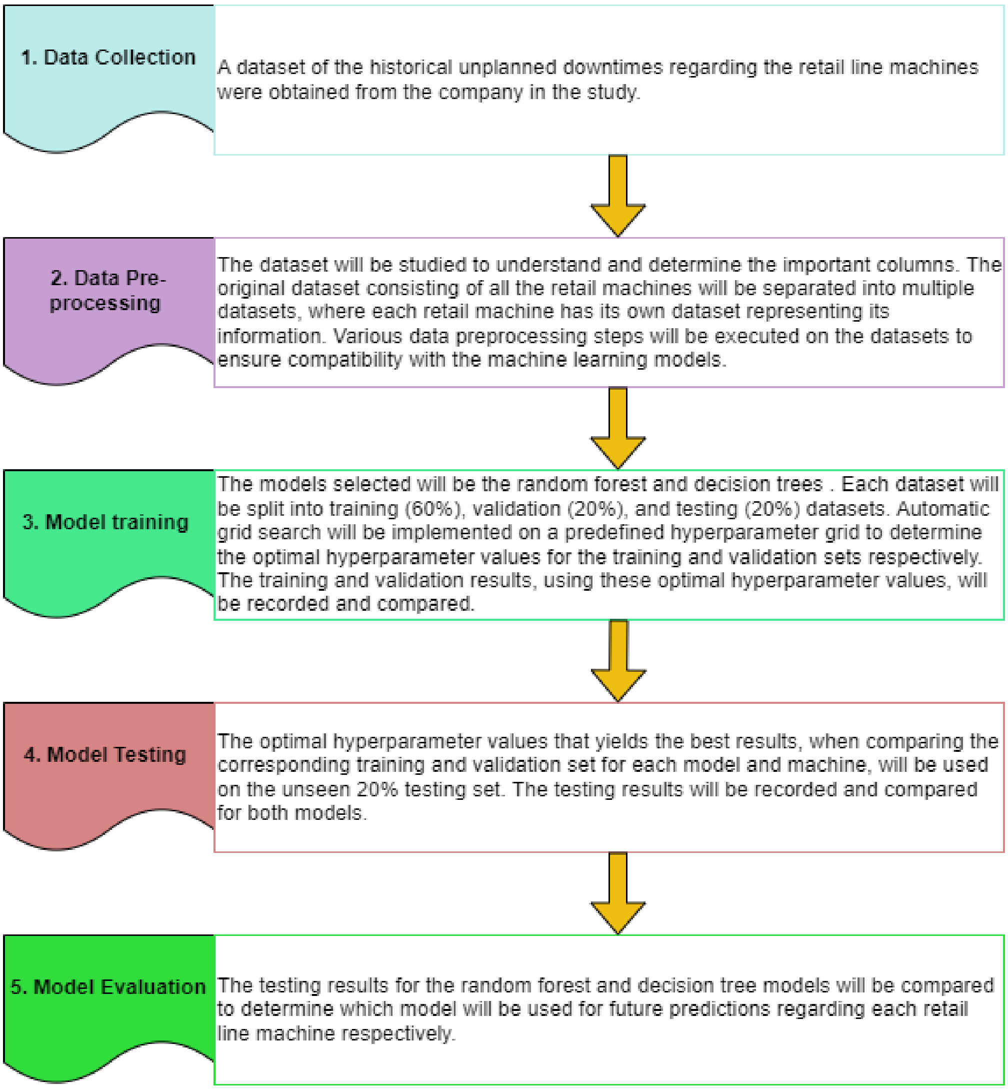

Figure 1: Scientific method for the development of the machine learning models [11]

The scientific method discussed in **Figure 2** was used to provide a conceptual design solution for predicting the mean time to fix for the different production line machines. The random forest and decision tree machine learning models were used to predict the mean time to fix (unplanned machine downtime) for each production line machine dataset. For this study, three production line machines were used as an example.

3 THE APPLICATION OF MACHINE LEARNING

3.1 Verification and validation of machine learning models

3.1.1 Data pre-processing

The following preliminary pre-processed steps were executed on the original raw dataset using R-studio, Excel, and Python (Spyder IDE):

I. Removed unnecessary columns using Excel and import data into R-studio:

The data columns were reduced to only three, namely Date, Machine, and Downtime. All other columns were irrelevant and not required for analysis or model development purposes since they didn't provide any meaningful insights. R-Studio software was used to apply data pre-processing to the Excel sheet containing the three columns (Date, Machine, Downtime). The Excel dataset was imported into R-Studio using the "readxl" package.

II. Downtime column pre-processing using R-Studio:

All the rows containing a 0 or N/A entry in the Downtime column were eliminated from the dataset.

III. Aggregation of duplicate entries using R-Studio:

Duplicate entries of the same machine having multiple downtimes in one day were summed together, ensuring unique data entries for the dataset.

IV. Separation of the dataset to obtain a dataset for each retail line machine using Excel:

The preliminary pre-processed dataset was exported back to Excel, where the dataset was separated using the Machine column. The filter function in Excel was used to create a separate dataset for each retail line machine, resulting in 6 distinct Excel sheets (one for each machine).

V. Removal of Machine column for each newly formed dataset using Excel:

Each of the newly formed datasets, representing one retail line machine each, resulted in the removal of the machine column for each of the newly formed datasets since this column would not provide any meaningful insight for the machine learning models.

VI. Outlier removal for the Downtime columns using Python (Spyder IDE):

The final data pre-processing step was done using Python, where each dataset's outliers (in the Downtime column) were removed using the Interquartile Range method (IQR). The rest of the model development process was executed using Python (Spyder IDE).

VII. Final pre-processed datasets

This process resulted in three pre-processed datasets in total. Each dataset consists of two columns: Date and Downtime.

3.1.2 Training, validation and testing of the regression models

In this section, model training and model testing of the scientific method for developing machine learning models are executed. The results obtained from the machine learning models were evaluated using the performance metrics: MSE (Mean Square Error), RMSE (Root Mean Square Error) and R^2.

I. Machine 1

The model training procedure was followed to obtain Machine 1's training - and validation set results. **Appendix A** illustrates the training and validation results obtained for Machine 1 for both machine learning models. Figure 2 provides an outline regarding the training and testing methods used. **Table 2** demonstrates the training and validation results for machine 1.

Table 2: Comparison of the training - and validation set results for Machine 1

Machine learning Model:	Conclusion When comparing training - and validation results:
Random Forest	The training set yielded an R^2 Value of 0.77, which is better than the validation set value of 0.73. In conclusion, the optimal hyperparameter values for the training set were used on the unseen testing set.
Decision Tree	The training set yielded a better MSE and R^2 value; thus, the optimal hyperparameter values for the training set were used on the unseen testing set.

After training and validating both machine learning models, model testing was done using the conclusions made in **Table 2**. The model testing procedure was followed to obtain the testing set results for the random forest and the decision tree model, illustrated in **Table 3**.

Table 3: Testing set results for Machine 1

Random Forest Model (Testing Set)	
Optimal hyperparameter values (Training set)	**Testing set results (Performance metrics)**
n_estimators = 150 max_depth = None min_samples_split = 5 min_samples_leaf = 2	MSE = 188.43 MAE = 10.34 R^2 = 0.67
Decision Tree Model (Testing Set)	
Optimal hyperparameter values (Training set)	**Testing set results (Performance metrics)**
max_depth = 5 min_samples_split = 5 min_samples_leaf = 2	MSE = 275.04 MAE = 11.59 R^2 = 0.54

Table 3 illustrates that the random forest model provided better MSE, MAE, and R^2 values when compared to the decision tree model. The random forest model was used to make future predictions regarding the unplanned downtime Machine 1 experienced. The same process was repeated for the other machines in the following subsection.

II. Machine 2

The training – and validation set results for the random forest and the decision tree model are illustrated in Appendix A. Figure 2 provides an outline regarding the training and testing methods used. Comparing the training - and validation set results for the random forest and the decision tree model, respectively, the following conclusions were drawn:

Table 4: Comparison of the training - and validation set results for Machine 2

Machine learning Model:	Conclusion When comparing training - and validation results:
Random Forest	The training set yielded better MSE, MAE, and R^2 values than the validation set values. In conclusion, the optimal hyperparameter values for the training set were used on the unseen testing set.
Decision Tree	The validation set yielded better MSE, MAE, and R^2 values than the training set values; thus, the optimal hyperparameter values for the validation set were used on the unseen testing set.

After training and validating both machine learning models, model testing was done using the conclusions made in **Table 4**. The same model testing procedure was repeated for Machine 2 to obtain the testing set results for the random forest and the decision tree model, illustrated in **Table 5**.

Table 5: Testing set results for Machine 2

Random Forest Model (Testing Set)	
Optimal hyperparameter values (Training set)	**Testing set results**
n_estimators = 50 max_depth = None min_samples_split = 5 min_samples_leaf = 2	MSE = 385.51 MAE = 14.51 R^2 = 0.61
Decision Tree Model (Testing Set)	
Optimal hyperparameter values (Validation set)	**Testing set results**
max_depth = None min_samples_split = 5 min_samples_leaf = 2	MSE = 281.04 MAE = 11.56 R^2 = 0.72

Table 5 illustrates that the decision tree model provided a better MSE, MAE, and R^2 value when compared to the random forest model. The decision tree model was used to make future predictions regarding the unplanned downtime Machine 2 experienced. The exact process was repeated for Machine 3 in the following subsection.

III. Machine 3

The training – and validation set results for the random forest and the decision tree model are illustrated in Appendix A. Figure 2 provides an outline regarding the training and testing methods used. Comparing the training - and validation set results for the random forest and the decision tree model, respectively, the following conclusions were drawn:

Table 6: Comparison of the training - and validation set results for Machine 3

Machine learning Model:	**Conclusion When comparing training - and validation results:**
Random Forest	The training set yielded a better R^2 Value of 0.72 when compared to the validation set value of 0.52. In conclusion, the optimal hyperparameter values for the training set were used on the unseen testing set.
Decision Tree	The training set yielded a better R^2 value of 0.83 than the validation set value of 0.78; thus, the optimal hyperparameter values for the training set were used on the unseen testing set.

The conclusions in **Table 6** were followed to obtain model testing results for both regression models, which are displayed in **Table 7**.

Table 7: Testing set results for Machine 3

Random Forest Model (Testing Set)	
Optimal hyperparameter values (Training set)	**Testing set results**
n_estimators = 150 max_depth = 5 min_samples_split = 2 min_samples_leaf = 2	MSE = 427.05 MAE = 16.88 R^2 = 0.71
Decision Tree Model (Testing Set)	
Optimal hyperparameter values (Training set)	**Testing set results**
max_depth = 10	MSE = 223.61

min_samples_split = 2 min_samples_leaf = 2	MAE = 12.11 R^2 = 0.85

Table 7 illustrates that the decision tree model provided a better MSE, MAE, and R^2 value than the random forest model. The decision tree model was used to make future predictions regarding the unplanned downtime Machine 3 experienced. In the following subsection, the same process was repeated for Machine 4.

3.2 Future predictions

3.2.1 Machine 1

This section illustrates and discusses the future predictions for Machine 1.

Future predictions were made from the last recorded data entry (in the pre-processed Machine 1 dataset) up until one year into the future. **Figure 3** illustrates the future predictions obtained for Machine 1 using the best random forest model.

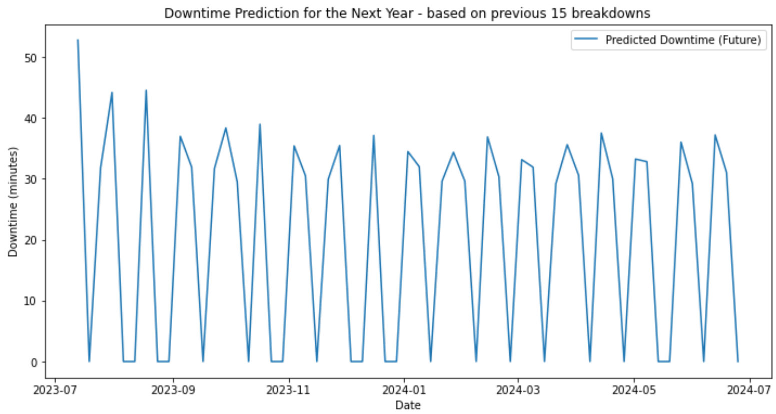

Figure 3: Future Predictions for Machine 1

The actual historical pre-processed dataset of Machine 1 was used to calculate the average amount of days between two consecutive breakdowns, which yielded approximately 12 days. A Future Dates data frame was created, storing the future dates for which predictions were made. It was selected that predictions should be made for every sixth day (in six-day intervals) based on half of the average interval between two consecutive breakdowns (12 days). The best random forest model was used to make downtime predictions for these specific days in the Future Dates data frame. A prediction threshold was created to filter out predictions that were considered insignificant. A prediction threshold of 75% of the mean for the downtime column was selected. Any predicted downtime value below this threshold value was considered insignificant and was recorded as 0 predicted downtime. The prediction results displayed in **Figure 3** are an output of the dates for which the model predicts a significant downtime event will occur and the corresponding predicted downtime duration for this event.

3.2.2 Machine 2

This section illustrates and discusses the future predictions for Machine 2.

Future predictions were made from the last recorded data entry (in the pre-processed Machine 2 dataset) up until one year into the future. **Figure 4** illustrates the future predictions obtained for the Labeller machine using the best decision tree model.

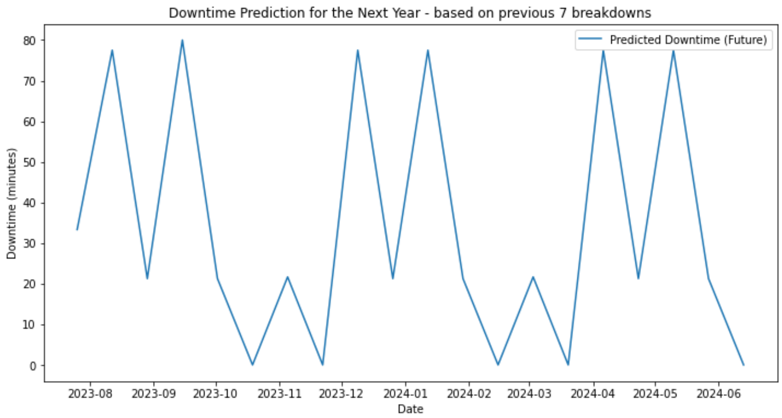

Figure 4: Future Predictions for Machine 2

The actual historical pre-processed dataset of Machine 2 was used to calculate the average amount of days between two consecutive breakdowns, which yielded approximately 34 days. A Future Dates data frame was created, storing the future dates for which predictions were made. It was selected that predictions should be made for every 17th day (in 17-day intervals) based on half of the average interval between two consecutive breakdowns (34 days). The best decision tree model was used to make downtime predictions for these specific days in the Future Dates data frame. A prediction threshold was created to filter out predictions that were considered insignificant. A prediction threshold of 50% of the mean for the downtime column was selected. Any predicted downtime value below this threshold value was considered insignificant and was recorded as 0 predicted downtime. The prediction results displayed in **Figure 4** are the output of the dates for which the model predicts a significant downtime event will occur and the corresponding predicted downtime duration for this event.

3.2.3 Machine 3

This section illustrates and discusses the future predictions for Machine 3.

Future predictions were made from the last recorded data entry (in the pre-processed Machine 3 dataset) up until one year into the future. **Figure 5** illustrates the future predictions obtained for the BEV Capper machine using the best decision tree model.

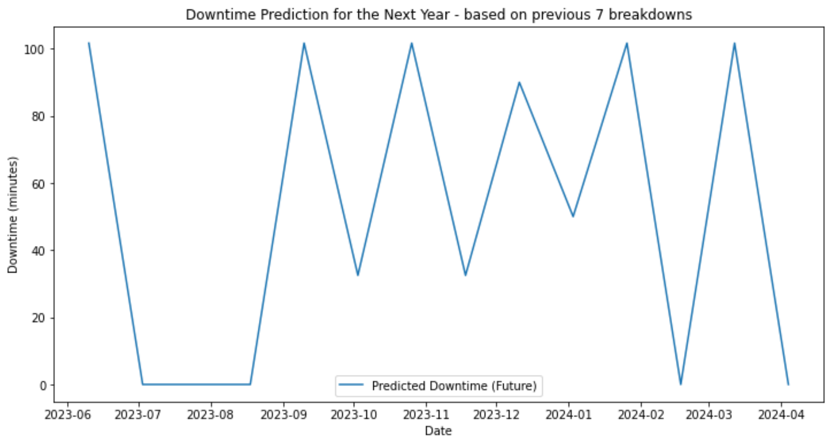

Figure 5: Future Predictions for Machine 3

The actual historical pre-processed dataset of Machine 3 was used to calculate the average amount of days between two consecutive breakdowns, which yielded approximately 46 days. A Future Dates data frame was created, storing the future dates for which predictions were made. It was selected that predictions should be made for every 23rd day (in 23-day intervals) based on half of the average interval between two consecutive breakdowns (46 days). The best decision tree model was used to make downtime predictions for these specific days in the Future Dates data frame. A prediction threshold was created to filter out predictions that were considered insignificant. A prediction threshold of 75% of the mean for the downtime column was selected. Any predicted downtime value below this threshold value was considered insignificant and was recorded as 0 predicted downtime. The prediction results in **Figure 5** are the output of the dates for which the model predicts a significant downtime event will occur and the corresponding predicted downtime duration for this event. The same process will be repeated for the Graham Sleever machine in the next section. Predictive maintenance strategy plan

The predictive maintenance strategy plan is a set of actions and guidelines that leverage the predictive models developed for the retail line machines. The following predictive maintenance strategy plan was developed for the retail line machines by utilising the model predictions:

3.2.4 Machine 1 (6-day interval predictions):

- ✓ High-priority machine for maintenance
- ✓ Four days before a predicted downtime event: Perform a thorough visual inspection and condition-based assessment on Machine 1. Identify any possible warning signs or abnormalities regarding the machine's performance [12].
- ✓ Three days before the predicted downtime event, based on the inspection findings, Ensure the required spare parts are available for repair/maintenance [13].
- ✓ Two days before the predicted downtime event: Perform the required maintenance on the machine based on the findings obtained from the inspection/assessment. If no warning signs or abnormalities were identified during the inspection, essential routine maintenance should be performed [14].

3.2.5 Machine 2 (17-day interval predictions):

- ✓ Five days before a predicted downtime event: Perform a thorough visual inspection and condition-based assessment on the Machine 2. Identify any possible warning signs or abnormalities regarding the machine's performance [12].
- ✓ Four days before the predicted downtime event, based on the inspection findings, Ensure the required spare parts are available for repair/maintenance [13].
- ✓ Two days before the predicted downtime event: Perform the required maintenance on the machine based on the findings obtained from the inspection/assessment. Basic routine maintenance should be performed if no warning signs or abnormalities were identified during the inspection [14].

3.2.6 Machine 3 (23-day interval predictions):

- ✓ Five days before a predicted downtime event: Perform a thorough visual inspection and condition-based assessment on the Machine 3. Identify any possible warning signs or abnormalities regarding the machine's performance [12].
- ✓ Four days before the predicted downtime event, based on the inspection findings, Ensure the required spare parts are available for repair/maintenance [13].
- ✓ Two days before the predicted downtime event: Perform the required maintenance on the machine based on the findings obtained from the inspection/assessment. Basic routine maintenance should be performed if no warning signs or abnormalities were identified during the inspection [14]. Implementing this predictive maintenance strategy plan by utilising the predictions made by the machine learning models can yield the following benefits for company X:

- **Reduced unplanned machine downtime:** Proactively addressing potential machine-related issues before they lead to unplanned machine downtime will increase the production rate obtained by the retail production line [15].
- **Efficient resource allocation:** Utilising the predictions can ensure that resources such as spare parts and labour are allocated efficiently, thus reducing operational expenses [16].
- **Preventive maintenance schedule:** Focusing on the specific dates for which the model indicates a higher risk for unplanned machine downtime can further reduce operational expenses and disruptions [16].
- **Threshold-based alerts** Can reduce unplanned machine downtime and production losses [17].
- **Optimised budget allocation:** Machine predictions can improve financial planning and allocation of funds for maintenance-related tasks [18].
- **Improved overall equipment efficiency (OEE):** Implementing the maintenance strategy plan for the retail line machines can improve the OEE, which indicates better equipment utilisation, reduced waste, and an increased production rate [19].
- **Customer satisfaction:** Reducing the unplanned machine downtimes experienced by the retail line machines will improve the production rate, and consequently, customer demand can be more easily satisfied [20].

4 CONCLUSIONS AND FUTURE RECOMMENDATIONS

Machine learning models were developed using historical unplanned machine downtime. The developed models were verified and validated. Future predictions for each retail line machine were delivered and used to develop a predictive maintenance strategy plan. Implementing the predictive maintenance strategy plan will enable Company X to make proactive decisions regarding maintenance. Possible benefits regarding this implementation include reduced unplanned machine downtime, improved production rate, efficient resource allocation, optimised maintenance budget allocation, and an improved OEE [16]. The project only utilises the random forest and the decision tree machine learning model. Other machine learning techniques can also be used to develop a predictive model. The machine learning models in this study were developed using only two columns. Additional feature columns can be implemented to improve the predictive power of the models possibly.

It is important to note that developing machine learning models is a continuous process that requires constant updates and modifications as new data is obtained. Continuously monitoring the model's performance and accuracy to make informed modifications is crucial to the benefits the model will provide the company.

5 REFERENCES

[1] P. Poor, D. Zenisek, and J. Basl, "Historical overview of Maintenance Management Strategies: Development," [Online]. Available: https://www.researchgate.net/profile/David-Zenisek/publication/335444202_Historical_Overview_of_Maintenance_Management_Strategies_Development_from_Breakdown_Maintenance_to_Predictive_Maintenance_in_Accordance_with_Four_Industrial_Revolutions/links/5f9141ce458515b7cf93872f/Historical-Overview-of-Maintenance-Management-Strategies-Development-from-Breakdown-Maintenance-in-Accordance-with-Four-Industrial-Revolutions.pdf. [Accessed: Jul. 20, 2023].

[2] T. Goncalves, "Proactive vs. Reactive Maintenance," Fiix. [Online]. Available: https://www.fiixsoftware.com/blog/reactive-proactive-maintenance/. [Accessed: Jul. 20, 2023].

[3] "Linear regression in machine learning [with examples]," Linear Regression in Machine Learning [with Examples]. [Online]. Available: https://www.knowledgehut.com/blog/data-science/linear-regression-for-machine-learning. [Accessed: Jul. 20, 2023].

[4] "Logistic regression in machine learning," GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/understanding-logistic-regression/. [Accessed: Jul. 20, 2023].

[5] A. Thevapalan and J. Le, "R decision trees tutorial: Examples & code in R for Regression & Classification," DataCamp. [Online]. Available: https://www.datacamp.com/tutorial/decision-trees-R. [Accessed: Jul. 20, 2023].

[6] "Predictive modelling techniques- a comprehensive guide [2023]," ProjectPro. [Online]. Available: https://www.projectpro.io/article/predictive-modelling-techniques/598#mcetoc_1g5gnldk23h. [Accessed: Jul. 20, 2023].

[7] "What are neural networks?," IBM. [Online]. Available: https://www.ibm.com/topics/neuralnetworks#:~:text=Neural%20networks%2C%20also%20known%20as,neurons%20signal%20to%20one%20another. [Accessed: Jul. 20, 2023].

[8] S. E. R, "Understand random forest algorithms with examples (updated 2023)," Analytics Vidhya. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/. [Accessed: Jul. 20, 2023].

[9] "6 case studies illuminate the value of predictive and prescriptive ...," Plant Services. [Online]. Available: https://www.plantservices.com/predictive-maintenance/predictive-maintenance/article/11290707/6-case-studies-illuminate-the-value-of-predictive-and-prescriptive-maintenance. [Accessed: Jul. 20, 2023].

[10] E. T. Bekar, P. Nyqvist, and A. Skoogh, "An intelligent approach for data pre-processing and analysis in predictive maintenance with an industrial case study," Advances in Mechanical Engineering, vol. 12, no. 5, p. 168781402091920, May 2014. doi: 10.1177/1687814020919207.

[11] J. Kamiri and G. Mariga, "Research methods in machine learning: A content analysis," International Journal of Computer and Information Technology, vol. 10, no. 2, Mar. 2021. doi: 10.24203/ijcit.v10i2.79.

[12] N. Sakib and T. Wuest, "Challenges and opportunities of condition-based Predictive Maintenance: A Review," Procedia CIRP, vol. 78, pp. 267–272, Jan. 2018. doi: 10.1016/j.procir.2018.08.318.

[13] I. de Pater and M. Mitici, "Predictive maintenance for multi-component systems of repairables with remaining-useful-life prognostics and a limited stock of spare components," Reliability Engineering & System Safety, vol. 214, p. 107761, Oct. 2021. doi: 10.1016/j.ress.2021.107761.

[14] A. Bousdekis, K. Lepenioti, D. Apostolou, and G. Mentzas, "Decision making in Predictive maintenance: Literature review and research agenda for industry 4.0," IFAC-PapersOnLine, vol. 52, no. 13, pp. 607–612, 2019. doi: 10.1016/j.ifacol.2019.11.226.

[15] M. Shahin, F. F. Chen, A. Hosseinzadeh, and N. Zand, "Using machine learning and deep learning algorithms for downtime minimisation in manufacturing systems: An early Failure Detection Diagnostic Service," The International Journal of Advanced

Manufacturing Technology, vol. 128, no. 9–10, pp. 3857–3883, Aug. 2023. doi: 10.1007/s00170-023-12020-w.

[16] A. Bousdekis, D. Apostolou, and G. Mentzas, "Predictive maintenance in the 4th Industrial Revolution: Benefits, business opportunities, and managerial implications," IEEE Engineering Management Review, vol. 48, no. 1, pp. 57–62, 2020. doi: 10.1109/emr.2019.2958037.

[17] P. Zhao et al., "Advanced correlation-based anomaly detection method for predictive maintenance," 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), Aug. 2017. doi: 10.1109/icphm.2017.7998309.

[18] T. Cerquitelli et al., "Data-Driven Predictive Maintenance: A methodology primer," Information Fusion and Data Science, pp. 39–73, Aug. 2021. doi: 10.1007/978-981-16-2940-2_3.

[19] Y. Prasetyawan and I. Rachmayanti, "Proposing predictive maintenance strategy to increase OEE through system upgrade scenarios and AHP," IOP Conference Series: Materials Science and Engineering, vol. 1072, no. 1, p. 012031, Feb. 2021. doi: 10.1088/1757-899x/1072/1/012031.

[20] K. Mishra and S. K. Manjhi, "Failure prediction model for predictive maintenance," 2018 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), Feb. 2019. doi: 10.1109/ccem.2018.00019.

6 APPENDIX A

6.1 Machine 1

Table 8: Training - and validation set results for Machine 1

Random Forest Model				
Hyperparameter Grid (Automatic grid search)	**Optimal hyperparameter values (Training set)**	**Optimal hyperparameter values (Validation set)**	**Training Results (Performance metrics)**	**Validation Results (Performance metrics)**
n_estimators: [50, 100, 150] max_depth: [None, 5, 10] min_samples_split: [2, 5] min_samples_leaf: [1, 2]	n_estimators = 150 max_depth = None min_samples_split = 5 min_samples_leaf = 2	n_estimators = 50 max_depth = 10 min_samples_split = 5 min_samples_leaf = 2	MSE = 185.42 MAE = 10.55 R^2 = 0.77	MSE = 142.39 MAE = 8.67 R^2 = 0.73
Decision Tree Model				
Hyperparameter Grid	**Optimal hyperparameter values (Training set)**	**Optimal hyperparameter values (Validation set)**	**Training Results (Performance metrics)**	**Validation Results (Performance metrics)**
max_depth: [None, 5, 10, 20] min_samples_split: [2, 5, 10] min_samples_leaf: [1, 2, 4]	max_depth = 5 min_samples_split = 5 min_samples_leaf = 2	max_depth = 5 min_samples_split = 10 min_samples_leaf = 4	MSE = 362.75 MAE = 14.75 R^2 = 0.56	MSE = 384.20 MAE = 13.88 R^2 = 0.33

6.2 Machine 2

Table 9: Training - and validation set results for Machine 2

Random Forest Model				
Hyperparameter Grid (Automatic grid search)	**Optimal hyperparameter values (Training set)**	**Optimal hyperparameter values (Validation set)**	**Training Results (Performance metrics)**	**Validation Results (Performance metrics)**
n_estimators: [50, 100, 150] max_depth: [None, 5, 10] min_samples_split: [2, 5] min_samples_leaf: [1, 2]	n_estimators = 50 max_depth = None min_samples_split = 5 min_samples_leaf = 2	n_estimators = 50 max_depth = None min_samples_split = 2 min_samples_leaf = 2	MSE = 182.89 MAE = 9.94 R^2 = 0.71	MSE = 208.41 MAE = 10.40 R^2 = 0.69
Decision Tree Model				
Hyperparameter Grid	**Optimal hyperparameter values (Training set)**	**Optimal hyperparameter values (Validation set)**	**Training Results (Performance metrics)**	**Validation Results (Performance metrics)**
max_depth: [None, 5, 10] min_samples_split: [2, 5] min_samples_leaf: [1, 2]	max_depth = 5 min_samples_split = 2 min_samples_leaf = 1	max_depth = None min_samples_split = 5 min_samples_leaf = 2	MSE = 128.42 MAE = 7.73 R^2 = 0.80	MSE = 69.54 MAE = 6.26 R^2 = 0.91

6.3 Machine 3

Table 10: Training - and validation set results for Machine 3

Random Forest Model				
Hyperparameter Grid (Automatic grid search)	**Optimal hyperparameter values (Training set)**	**Optimal hyperparameter values (Validation set)**	**Training Results (Performance metrics)**	**Validation Results (Performance metrics)**
n_estimators: [50, 100, 150] max_depth: [None, 5, 10] min_samples_split: [2, 5] min_samples_leaf: [1, 2]	n_estimators = 150 max_depth = 5 min_samples_split = 2 min_samples_leaf = 2	n_estimators = 100 max_depth = 10 min_samples_split = 5 min_samples_leaf = 2	MSE = 265.08 MAE = 12.20 R^2 = 0.72	MSE = 216.28 MAE = 9.45 R^2 = 0.52
Decision Tree Model				
Hyperparameter Grid	**Optimal hyperparameter values (Training set)**	**Optimal hyperparameter values (Validation set)**	**Training Results (Performance metrics)**	**Validation Results (Performance metrics)**
max_depth: [None, 5, 10] min_samples_split: [2, 5] min_samples_leaf: [1, 2]	max_depth = 10 min_samples_split = 2 min_samples_leaf = 2	max_depth = 5 min_samples_split = 2 min_samples_leaf = 2	MSE = 151.52 MAE = 7.36 R^2 = 0.83	MSE = 99.77 MAE = 6 R^2 = 0.78