

Holistic Production Overview: Using XAI for Production Optimization



Sergi Perez-Castanos, Ausias Prieto-Roig, David Monzo,
and Javier Colomer-Barbera

1 Use Case Context

The **eXplainable MANufacturing Artificial Intelligence (XMANAI)** Ford use case focuses on managing the complexity of manufacturing in large production lines. These lines are composed of multiple work stations that mostly work sequentially, creating a direct dependence among them. Moreover, each of the stations is also made up of diverse machines and assets, with interconnected processes that range from fully automated to manual labor. All this considered, in a manufacturing line there are numerous complexities and challenges that can arise, making it difficult to anticipate and address potential production problems effectively. For example, a minor problem undetected, and thus not solved, at an asset at the beginning of the line may propagate affecting with great impact to the subsequent stations at the end of the line, and creating bottlenecks that hamper to reach the production goals. These kind of difficulties emphasize the crucial need for developing intelligent systems to ensure that the objective quality and quantity of produced items reach each industry goal, while keeping the production times within a profitable margin; such systems should not only anticipate unwanted situations during production, but in order to help the line operators to make proper decisions to manage them, it is also of great relevance for them to know the root causes that may cause production deviations.

The Ford use case has been designed in order to tackle these inherent complexities applying AI-based optimization systems, complemented with an explainability layer, applied on a real engine production line to monitor the overall standards that

S. Perez-Castanos (✉) · A. Prieto-Roig · D. Monzo
Tyrus AI, Valencia, Spain
e-mail: sergi.perez@tyris.ai; ausias.prieto@tyris.ai; david.monzo@tyris.ai

J. Colomer-Barbera
Ford, Valencia, Spain
e-mail: jcolome5@ford.com

are needed, and therefore minimizing deviations on the expected production. One of the main challenges addressed in this case is the high variability in engine types and their corresponding components. In an engine manufacturing facility, different engine derivatives are produced to meet the requirements of various vehicle models. Each engine type may require specific components, such as the engine crankcase, fuel pump, oil pump, clutch, and more. Managing the diverse range of components and ensuring their availability and correct installation on the assembly line can be a daunting task. The sheer number of engine types and components increases the likelihood of errors, delays, and production bottlenecks.

Another complexity lies in the planning and scheduling of production batches. Currently, manual processes driven by the expertise of the MP&L (Material Planning & Logistics) team and production staff are employed to manage weekly production batches. However, accurately determining the optimal batch size, sequencing, and allocation of resources is a complex task. The planning engineer relies on customer demand and their own experience to make decisions, which can result in suboptimal production plans and resource allocation. This can lead to inefficiencies, increased downtime, and compromised production capacity.

Moreover, unforeseen issues and disruptions during the manufacturing process can significantly impact production efficiency. Shift foremen must make decisions on the fly to minimize planned stoppages and address unexpected failures on the assembly line. Without a comprehensive understanding of the root causes and potential solutions for such issues, decision-making becomes challenging, and it becomes difficult to maintain consistent line availability and performance.

By developing AI models specifically designed for optimizing production on the engines line, these challenges can be effectively addressed. AI systems can analyze real-time and batch data simultaneously acquired from various systems to identify patterns, detect anomalies, and provide recommendations for line optimization. With advanced Machine Learning techniques, AI models can simulate different scenarios, predict the impact of changes, and suggest the best course of action to maximize line performance. These AI models can assist operators and engineers in making data-driven decisions, reducing errors, improving resource allocation, and minimizing downtime.

Overall, the complexities and uncertainties inherent in engine manufacturing highlight the critical need for AI models to optimize production. By leveraging AI technologies, manufacturers can enhance their ability to anticipate and tackle potential problems, leading to increased efficiency, reduced costs, and improved overall performance on the assembly line.

The current situation at Ford Engine Plant does not allow the power of quasi-real-time data to be harnessed for decision-making. There are records on the status of the different operations on the production line, the quantity of engines produced and their parts, quality reports, and production plans. Despite having this information, there is not a centralized database and all the information is disaggregated in different corporate databases. This lack of centralized information is the first problem that needs to be solved in order to optimize the different processes that occur on the production line. This problem implies another one, which is the lack of

artificial intelligence applied to the different decision-making processes due to the impossibility of taking advantage of all the available data. The proposed application aims to mitigate these problems by means of a set of functionalities that will be explained in the following sections.

This use case consists of a set of actions related to the current status of the line within a shift. By means of the information provided by the different disaggregated data sources, it is possible to analyze this information jointly to establish trends and to make predictions about anomalous situations in the line or the total amount of produced engines at the end of the shift. Thus, this use case is focused on the estimation of the production at the end of the shift, detection of unwanted scenarios, and simulations of new hypothetical situations, while giving insights on the assets that may cause potential deviations regarding the expected production goals.

Ford internal databases have different information about the status of operations (whether an operation is cycling a new component, waiting for a new part, blocked or in another possible state), operation failures, cycle times (both actual and design time), number of parts produced in a shift and data related to the quality of the parts produced. In this use case, different data sources related to production data will be joined to represent the historical status of the production line and to make predictions about the number of engines produced at the end of the shift following the current trend of the line. Both information will help the business experts to understand the significant deviations that may occur between the predicted (planned) production and the actual engines produced at the end of the shift.

2 XAI Approach

The application of explainability techniques is not a merely technical process, since explainability is precisely the bridge that connects intelligent systems with the users who are using them. Therefore, it is important that the end-users of XAI solutions are involved in the design process of explainable systems, looking for a human-in-the-loop approach to be followed. Therefore, in order to provide a solid explainability layer to the AI system developed in relation to the Ford use case, two key activities were carried out prior to the development of the XAI models.

The initial activity involved identifying the specific XAI needs of the end-users, i.e. the operators of the engine production line. This step is crucial in understanding the requirements and preferences of the stakeholders who will be interacting with the AI system. By closely collaborating with the end-users, their expectations and concerns regarding the interpretability of the AI models are effectively captured. This process ensures that the subsequent selection of Machine Learning models and explainability tools is aligned with the identified needs of the end-users.

Consequently, the second activity focused on selecting the appropriate methods that fulfill the identified XAI needs. Drawing from a range of available techniques, the selection process took into consideration the specific requirements and constraints of the manufacturing problem. The chosen methods were evaluated based

on their capability to provide interpretable insights into the decision-making process of the AI models. Through this meticulous selection process, the chosen methods effectively address the XAI needs identified during the initial activity.

By conducting these two activities, the system can ensure that the AI and explainability aspects of the demonstrator are tailored to the requirements of the end-users at the plant. This approach fosters a collaborative and user-centric approach, guaranteeing that the selected methods provide meaningful and actionable explanations. Ultimately, the identification of explainability needs and the subsequent selection of appropriate methods enable the development of explainable AI models that empowers plant personnel to understand and trust the decisions made by the AI models, facilitating effective decision-making and driving the optimization of manufacturing processes.

2.1 Identification of XAI Needs

Prior to the development of XAI models, several tasks need to be addressed to effectively solve the intended problem. The general workflow follows the next steps.

The initial step is identifying the relevant data sources and determining the technical requirements necessary for data collection, storage, and processing. This involves understanding the data ecosystem within the manufacturing environment, including sources such as corporate systems, maintenance records, tooling systems, and real-time data acquisition. Concretely, for this problem, the data employed are related to the status of the production of the line and quality data.

Next, it is essential to assess the AI needs specific to the manufacturing problem at hand. This includes identifying the key challenges and objectives, such as optimizing production, minimizing downtime, and improving resource allocation. Understanding the desired outcomes helps in defining the scope and purpose of the AI models to be developed. The objective for this problem consists on finding deviations from the expected productions and preventing line bottlenecks.

Simultaneously, it is important to recognize the explainability needs of the stakeholders involved. This entails considering the requirements for transparency, interpretability, and trust in the decision-making process. Different stakeholders may have varying levels of expertise and understanding of AI systems, so it is crucial to determine the appropriate level of explanation needed to ensure effective collaboration and decision-making. For this problem, it is essential to understand which elements of the line and to what degree they have influenced predicted deviations and analyze the best action to fit it.

By applying XAI models, several advantages can be achieved in the manufacturing context. The foremost advantage is that the end-user involved in the production lines will be able to make better decisions based on results of the analysis of the available data and understand why these results are originated, in terms of knowing the specific assets that have the most influence on them.

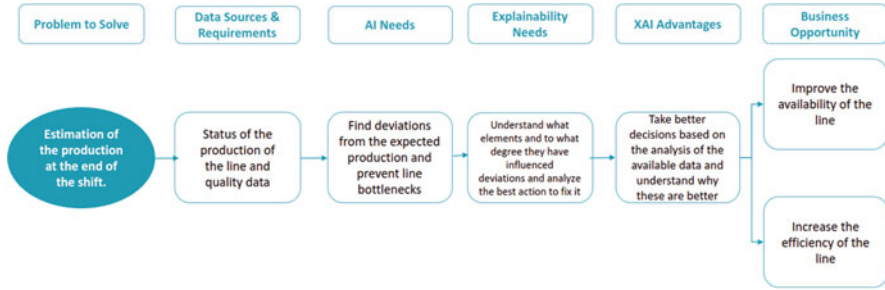


Fig. 1 Identification of AI and explainability needs for the use case

The combination of XAI models and their advantages opens up significant business opportunities in this use case. Specifically, the application of XAI models will affect positively the line in two aspects: firstly, the line will increase its availability, preventing line bottlenecks by quick fixing issues that can affect the whole production, and secondly, the efficiency of the production will also be increased, by reducing unnecessary maintenance stops.

Overall, the tasks preceding the development of XAI models involve identifying data sources, technical requirements, AI needs, explainability needs, recognizing the advantages of using XAI models, and uncovering the resulting business opportunities. In Fig. 1, a diagram of the whole identification process is presented.

By addressing these aspects, it is possible to identify which Machine Learning Models and explainability tools meet the XAI needs of the demonstrator as it is explained in the next section.

2.2 Hybrid Models

In the context of the **XMANAI** project, a hybrid model [2] refers to the combination of two components: a Machine Learning model and an explainability tool used to interpret the results produced by the Machine Learning model.

The component with the Machine Learning model is responsible for learning patterns and making predictions based on input data. It leverages algorithms and techniques to extract information, generalize from training examples, and generate predictions for new data instances. The Machine Learning model may include various approaches such as decision trees [3] or random forests [4].

On the other hand, the component with the explainability tool is employed to provide insights into the decision-making process of the Machine Learning model. It helps to uncover the underlying factors, features, or patterns that influence the model’s predictions. The explainability tool enhances transparency and interpretability by providing explanations, visualizations, or metrics that shed light on how the Machine Learning model arrives at its results.

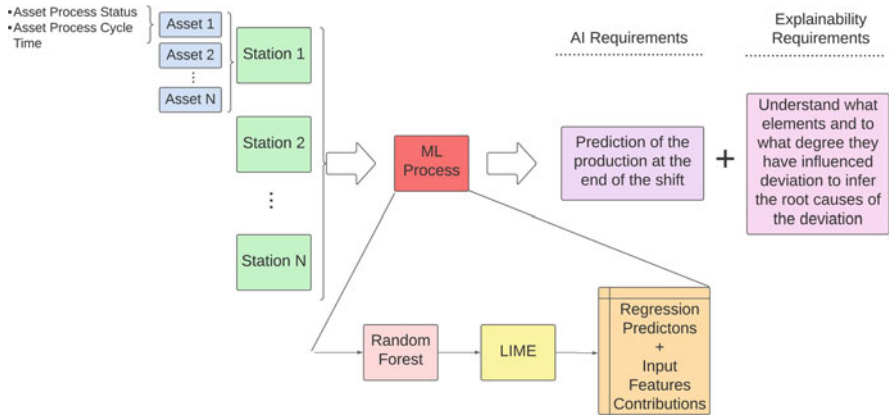


Fig. 2 Ford experiment diagram

By combining these two components, the hybrid model aims to address the “black box” nature of some Machine Learning algorithms, providing added transparency. The explainability tool provides insights into the most relevant features on a specific prediction, the relative contributions of those features, and how they have impacted on the final model decision.

Since the goal of this approach is the development and training of a model for the estimation of the production at the end of a shift, a regression model has been determined to be a good fit to solve the problem, and thus it has been selected for that purpose. As this estimation is a regression task, this model can be employed to that end and the output prediction of the system will be the number of engines produced at the end of one production shift. Based on the comparison of this prediction with regard to the expected production, the plant manager and the operators will be able to make decisions to correct potential deviations.

In order to solve this use case, an experiment based on the architecture shown in Fig. 2 has been built in order to configure the use case assets and train the AI model. From left to right, the use case data asset will be generated retrieving operation data directly from the different production line stations of the engine assembly line. Those assets will feed the Machine Learning pipeline performing a regression prediction on the number of engines that will be produced at the end of the shift. Different ML predictive models have been evaluated to analyze which of them provide more accurate results. Finally, Random Forest was selected as the most suitable AI base-model for this goal.

Therefore, the explainability requirement of this problem is to understand what elements and how much have influenced the deviation so as to infer the root causes of the deviation as mentioned above. Considering the explainability requirement, LIME [8] has been selected as the optimal XAI Tool due to its feature contribution explanation, which gives much value to the model explanation in this use case scenario. Local Interpretable Model-Agnostic Explanations (LIME) is an

explainability tool that provides interpretable explanations for individual predictions made by Machine Learning models. It creates local surrogate models to approximate the behavior of the original model, generating feature importance weights that indicate the relative influence of each input feature. LIME is model-agnostic, making it applicable to various models without requiring access to their internal workings. Its explanations enhance transparency and understanding, fostering trust in complex Machine Learning systems.

Taking into account the selected Machine Learning model and the explainability tool, the expected output of the ML pipeline is a prediction which indicates the expected performance of the production at the end of the shift, which ultimately will be compared with the actual production goals to prevent potential deviations. Thus, the explanations given should help to understand what the key elements are and how much they have influenced on production deviations, and supporting the end-users to infer root causes that could be tackled to minimize the impact.

2.2.1 Interpretation of XAI Outputs

Using a Random Forest as a model and LIME as an explainability tool, an example of explainability is shown in Fig. 3 where an entire shift of production is analyzed to estimate the production of the current shift through the regression of the input data.

On the left side, it can be seen that the model predicts a value of 19.16. This value has to be interpreted as the average value of engines produced by the line in time intervals of 10 minutes. For this use case, values in the range between 15 and 20 are

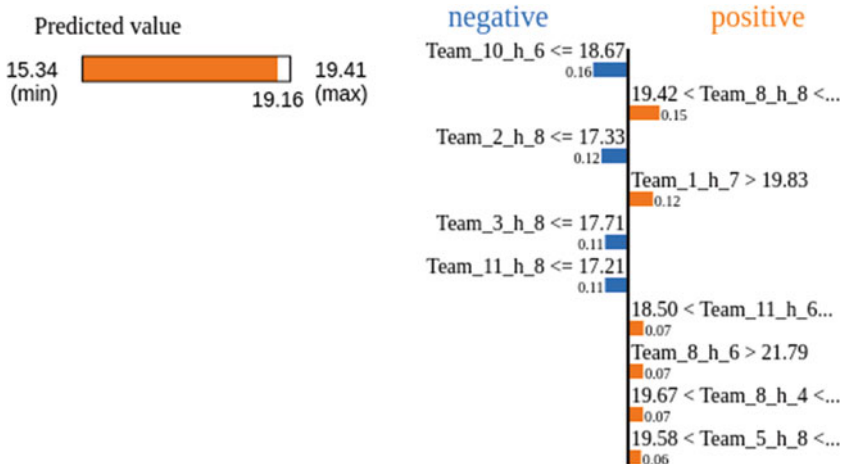


Fig. 3 Lime explainability result example. On the left side, it is shown the ML prediction information. On the right side, it can be see, the contributions (both positive and negative) of the different input features

considered as proper values, representing a nominal behavior of the line. When the value is lower than 15, a deviation from the expected value has to be considered.

On the right side, the information provided by the [LIME](#) explainability tool is presented. As this technique consists on a feature relevance representation, the different rows in the diagram represents the input features, coming from the input data used for this concrete prediction, that are contributing the most to this concrete output and in what degree they are contributing. The negative or positive side represents how much they are affecting to an upper or lower value from the average output value of the training data employed to train the model.

2.3 Graph Machine Learning Models

Graph Machine Learning (Graph ML) [7] models are a class of Machine Learning algorithms designed to operate on structured data represented as graphs. In contrast to traditional Machine Learning approaches that work with tabular data or sequential data, graph ML models leverage the inherent relational information present in graph structures to make predictions or gain insights.

Graph ML models are particularly suited for tasks that involve complex relationships and dependencies among data points. They find applications in various domains, including social network analysis, recommendation systems, bioinformatics, fraud detection, knowledge graph completion, and many more [10].

This topology of models can be correlated with the complex structure of interconnected processes present at production plants of manufacturing industries. Thus, a research line within XMANAI has been dedicated to determine the fit of this type of approach to provide IA-based predictive models adapted to the specifics of manufacturing considering the addition of an explainability layer. Specifically, among our experiments with the data, Heterogeneous Graph ML models were selected as they are focused on Explainability. After a research among different beyond-state-of-the-art options, we were inclined to use in this case [Tensorflow GNN \(TFGNN\)](#) [5]. This library is capable of handling Heterogeneous Graphs to fit a dynamic number of nodes and edges, with different node classes and edge classes, and their own set of data. This also leverages Tensorflow's [1] graph execution, which speeds up the training and inference process, allowing for faster iteration and less waiting time.

2.3.1 Graph Models

Graph models offer distinct advantages over regular models in the context of predicting the output of a manufacturing line. By leveraging the inherent relational structure present in the data, graph models can incorporate contextual information and capture the interdependencies between different components. Manufacturing

lines often exhibit complex relationships, such as cascading effects or feedback loops, which graph models excel at representing and exploiting. Additionally, graph models can effectively utilize both the graph topology and the node-level features associated with each component, enabling them to learn embeddings that encode intrinsic properties and interactions.

The transferability capabilities of graph models allow them to generalize knowledge across different manufacturing line states with shared characteristics. Furthermore, graph models provide interpretability and explainability by analyzing learned embeddings and the influence of components on the output.

GNNs have gained significant attention in recent years due to their ability to capture both the local and global structural information of the graph. They leverage a combination of node-level features, graph topology, and message passing mechanisms to update the representations of nodes throughout the network layers. GNNs can learn expressive node embeddings that encode both the inherent features of the nodes and their relationships with neighboring nodes.

During the training stage of the graph ML models, they are often optimized using gradient-based methods, where the gradients are computed through backpropagation. Graph ML models can be trained in a supervised manner, where labeled data is available, or in an unsupervised or semi-supervised manner, where only a subset of the data is labeled or no labels are available.

Graph ML models can provide various types of outputs that capture different aspects of the graph data. For example, in node classification tasks, the model may assign a label or class to each node in the graph, indicating the predicted category or behavior of the corresponding entity. Graph ML models can also generate outputs related to link prediction, where they estimate the likelihood or presence of connections between nodes in the graph. Additionally, graph ML models may produce node embeddings or representations that capture the learned features and relationships of each node, enabling downstream tasks or further analysis.

In the scenario analyzed, we work on temporal slices of the manufacturing line and we will be predicting the output at the final slice of the manufacturing line, encoded in the graph training as an attribute of the nodes. This has been proven to be possible, per example by Google in their paper [6], where they improved the state of the art by reducing the loss by 6% (using RMSLE).

2.3.2 Explainability Techniques

For the explainability, we used [Graph Attention \(GAT\)](#) [9] layers to collect the weight that each input is given by the model. Graph Machine Learning not only allows us to adapt both the inference and explainability to the layout of the manufacturing process when operators are added or removed, but it also allows the model to have more contextual information about what operators are connected or what kinds of data there is, which in the end, this improves the accuracy of the model and explanations

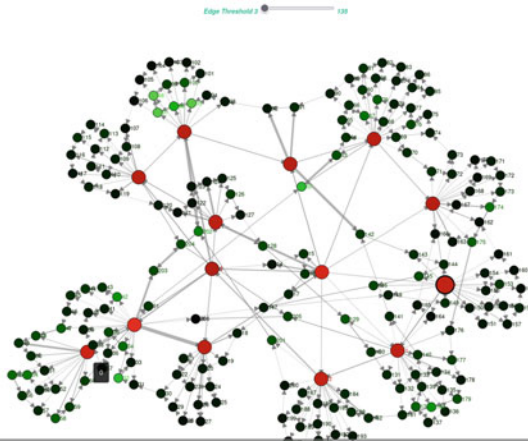


Fig. 4 Graph ML GAT contribution weights full graph. Brighter colors and wider edges means greater contribution

Upon completion of training and explanation, we obtain a graph that depicts the manufacturing line, including all the relationships between its components and their respective importance.

In Fig. 4, we present a visualization that represents the connections between operators (green dots) and teams (red dots) in our project. The relevance of each node is indicated by the darkness of its color, with darker colors indicating less relevance and brighter colors indicating more relevance. The size of the edges connecting the nodes also represents the strength of the connection, with larger edges denoting more relevant connections.

Specifically, the green dots represent operators, and a connection between two green dots implies that those operators are connected in line, with the next operator receiving the output of the previous one. Similarly, the red dots represent teams, and connections between red dots indicate that one team receives the output of the previous team's work. Additionally, there are connections between the red and green dots, indicating which operators belong to which team.

Figure 5 is a modified version of the first graph, where we have removed some of the less relevant connections. This filtering process results in retaining only groups of highly influential operators, represented by the remaining connections.

The utility of presenting this information to a user is that it provides a clear understanding of the relationships and dependencies between operators and teams in the project. By visualizing the graph, the user can identify the most relevant operators and teams based on their brightness and the size of their connections. This information helps in understanding the flow of work, identifying key contributors, and potentially optimizing the project by focusing on the most influential aspects.

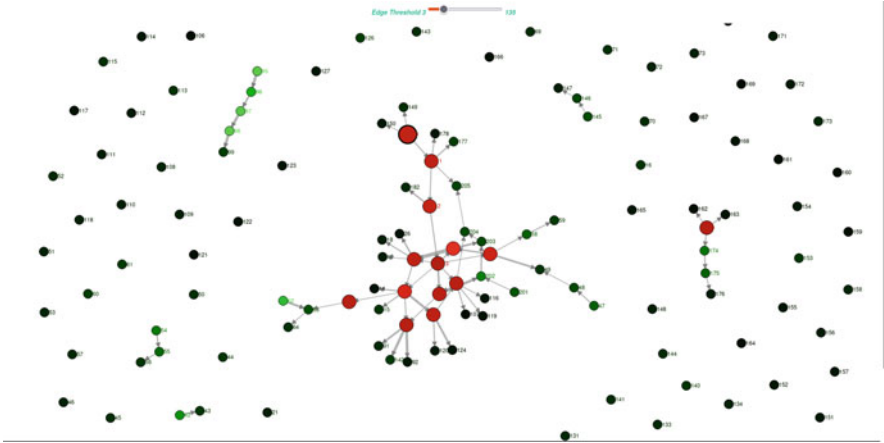


Fig. 5 Graph ML GAT contribution weights trimmed graph to only highest contributors. Brighter colors and wider edges means greater contribution

By removing less relevant connections in the second graph, the user can quickly grasp the core components and key dependencies, simplifying the visualization and highlighting the most critical elements. This focused view is simpler and straighter.

Finally, if this were to be too hard to understand for users, this has the potential to be added in a map of the line to better represent the positions of each team and operator.

3 XMANAI Platform Usage

All training of Hybrid models and the exploration of the available data is performed under the XMANAI platform. For this, Ford uploads their datasets to the platform, ensuring the data’s security and privacy by sharing it under a contractual agreement while retaining ownership.

The platform provides tools for visualizing the data, allowing us to gain insights and understand its characteristics. After visualizing the data, we can further explore the problem and develop solutions using notebooks available on the platform, leveraging its computational capabilities and pre-built libraries.

To create a hybrid model and explainable artificial intelligence (XAI) tool, we can upload their model and associated artifacts to the platform. We can configure the model with its hyperparameters with a viable explainability tool, and fine-tuning them to achieve optimal performance. The output is a session which will be used for training, and that will ensure that the model can always be used with the selected explainability tool.

After creating the session, we can use the platform's pipeline functionality to establish a streamlined process for training both the model and the explainer component.

Once the pipeline is set up, Ford can run it whenever they wish to make inferences or predictions based on the trained model. The pipeline ensures consistent and automated execution, saving time and effort.

Finally, we can visualize the explanations provided by the XAI tool through the platform's visualization capabilities. This allows them to gain interpretability and insights into how the model arrived at its predictions, aiding in decision-making and model validation.

4 Achievements, Conclusions, and Open work lines

In the following paragraphs, we will delve into the accomplishments we have made in the areas of data veracity, data ingestion automation, data processing and analytics, hybrid and graph model development, explainability methods, manufacturing app creation, and infrastructure enhancement.

In terms of data veracity, we have allocated a team of programmers to thoroughly review and ensure that all machines adhere to the standardized manufacturing data reporting. This serves as the starting point for all other aspects, as the availability of accurate data is paramount. Without reliable data, we cannot construct models that accurately represent the reality of our manufacturing processes.

We have also made significant progress in automating the data ingestion process. Previously done manually, the data ingestion is now automated, creating dumps of data periodically and producing new batches of data that can be used to retrain existing models.

Our focus on data processing and data analytics has involved extracting features from different datasets and determining the best approach for ingesting the input data to achieve relevant results.

Furthermore, we have been actively working on developing hybrid models that combine machine learning algorithms with explainability tools. Specifically, we have utilized [Local Interpretable Model-Agnostic Explanations](#) to create explanations that help identify the parts of the production line that are affecting specific predictions. Alongside this, we have also been working on graph models that provide explanations in a similar manner to our hybrid models.

Additionally, we have utilized the [eXplainable MANufacturing Artificial Intelligence](#) platform to develop a manufacturing app, as described in Sect. 3. This app has been designed for use by demonstrators in their respective use cases.

To accommodate the on-premise segment of the platform and effectively host the entire backend infrastructure for our final applications, we have recently acquired a dedicated physical server. This acquisition ensures efficient and reliable hosting for our platform.

In conclusion, our achievements in data veracity have ensured a solid foundation for our manufacturing processes by guaranteeing accurate and reliable data. This has enabled us to construct models that accurately represent the reality of our operations. The automation of data ingestion has not only saved valuable time and resources but also provided us with fresh batches of data for continuous model improvement. Our focus on data processing and analytics has allowed us to extract meaningful insights and achieve relevant results. The development of hybrid and graph models, coupled with explainability tools, has enhanced our understanding of predictions and identified the specific factors influencing them. The implementation of explainability methods has increased transparency and trust in our models' outputs, empowering us to make informed decisions. Furthermore, the creation of the manufacturing app has facilitated streamlined processes and improved collaboration among demonstrators in their respective use cases. Lastly, our investment in infrastructure, including the acquisition of a dedicated physical server, ensures a robust and efficient hosting environment for our final applications.

As we look toward the future, one of our key objectives is to bridge the gap between scientific explanations and end-user accessibility. While our current focus on developing explanations has been rooted in scientific rigor, we recognize the need to make these explanations more comprehensible and user-friendly for a non-technical audience. Our next step involves refining and adapting the explanations derived from our models, transforming them into a format that is easily understandable and meaningful to end-users. By translating complex technical concepts into accessible language and visualizations, we aim to empower the end-users to make informed decisions and derive maximum value from our manufacturing processes. This user-centric approach will enable us to deliver explanations that not only provide scientific insights but also serve as practical tools for enhancing productivity, efficiency, and overall business profitability.

References

1. Abadi, M., et al.: TensorFlow, Large-scale Machine Learning on Heterogeneous Systems (2015). <https://doi.org/10.5281/zenodo.4724125>
2. Amini, M., et al.: Discovering injury severity risk factors in automobile crashes: a hybrid explainable AI framework for decision support. *Reliab. Eng. Syst. Saf.* **226**, 108720 (2022). <https://doi.org/10.1016/j.res.2022.108720>. <https://www.sciencedirect.com/science/article/pii/S0951832022003441>
3. Breiman, L.: Classification and regression trees. In: The Wadsworth & Brooks/Cole (1984)
4. Brieman, L.: Random Forests Machine Learning (2001)
5. Ferludin, O., et al.: TF-GNN: graph neural networks in TensorFlow. *CoRR* abs/2207.03522 (2022). <http://arxiv.org/abs/2207.03522>
6. Kapoor, A., et al.: Examining covid-19 forecasting using spatio-temporal graph neural networks. In: *MLG workshop @ KDD'2020, epiDAMIK workshop @ KDD'2020* (2020). <https://arxiv.org/abs/2007.03113>
7. Liu, Z., et al.: Heterogeneous graph neural networks for malicious account detection. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge*

- Management (CIKM '18). Association for Computing Machinery, New York (2018), pp. 2077–2085. <https://doi.org/10.1145/3269206.3272010>
8. Ribeiro, M.T., et al.: Model-agnostic interpretability of machine learning (2016). arXiv preprint arXiv:160605386
 9. Wang, X., et al.: Heterogeneous graph attention network. In: The World Wide Web Conference (WWW '19). Association for Computing Machinery, New York (2019), pp. 2022–2032. <https://doi.org/10.1145/3308558.3313562>
 10. Xia, F., et al.: Graph learning: a survey. *IEEE Trans. Artif. Intell.* **2**(2), 109–127 (2021). <https://doi.org/10.1109/TAI.2021.3076021>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

