Selecting the Optimal Machine Learning Framework for LiDAR-Based Railroad Inspection and Maintenance at Strukton Rail: A Comprehensive Evaluation

TIM UILKEMA, University of Twente, The Netherlands

The surge in machine learning (ML) applications has made robust ML operational frameworks imperative. Strukton Rail, a railway solutions company, faces significant challenges, including managing extensive LiDAR data, deploying models efficiently on the cloud, maintaining version control, and ensuring user-friendly solutions. In response to these challenges, this research not only evaluates but also provides a structured approach for selecting ML frameworks based on criteria relevant to the needs of Strukton Rail. This contributes to the domain by providing a blueprint for future ML framework selection in similar contexts. The study bridges the gap between ML model experimentation and real-world application, aiming to boost the efficiency of ML applications in practice. Overall, the framework deemed most suitable for Strukton Rail through this process was ClearML.

Key Words: Machine learning (ML), MLOps, ML framework, ML lifecycle management, Framework evaluation.

1 INTRODUCTION

Machine learning (ML) has significantly influenced various sectors, including predictive maintenance [5], presenting both advancements and challenges. While extensive research has been conducted on ML model development, less attention has been given to deployment, particularly to transitioning models from the experimental stage to real-world implementations [7].

Strukton Rail, a railway infrastructure solutions company, intends to harness ML to improve their railway inspection and maintenance efficiency. The Ambient Intelligence research group at Saxion Hogeschool is tasked with the development of the ML model. Strukton Rail faces challenges in implementing this model, such as managing extensive LiDAR data, efficient cloud deployment of ML models and maintaining version control. An ML framework helps developers create, deploy, and maintain a ML application. The primary aim of this research is to select the most suitable ML framework to address these challenges.

Numerous ML frameworks exist today, each with a different set of features, capabilities, and scopes within the ML lifecycle. Determining the most suitable framework given specific requirements can be a complex task, highlighting a knowledge gap in the field.

TScIT 39, July 9, 2022, Enschede, The Netherlands
© 2023 University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

This research strives to bridge this gap by answering the following research question:

RQ: Based on defined criteria, which machine learning framework provides the highest overall compatibility with Strukton Rail's use case?

To thoroughly address this question, the following sub-questions are explored:

RQ1: What are the specific requirements and constraints for a machine learning framework in Strukton Rail's context?
RQ2: How does each considered machine learning framework score based on the defined criteria?

By addressing these questions, this study aims to provide a practical guide for ML framework evaluation tailored to the specific needs of Strukton Rail. Furthermore, it also establishes a methodological blueprint for future ML framework evaluation. To answer the research questions a three-step approach is taken, a literature review into ML and the practical implementation of ML (see Section 2), requirements interview (see Section 3) and thirdly based on the findings in these sections a methodology for the evaluation of ML frameworks was developed (see Section 4). The taken approach is visualized in Figure 1, providing a more detailed version to the outlined three step approach. The results from this evaluation are presented in Section 5.1. This research hopes to contribute to narrowing the gap between ML model development and real-world application.

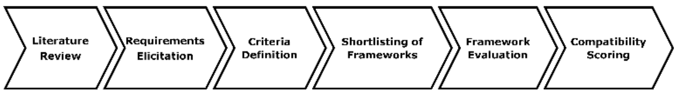

Figure 1. Methodology to select most compatible ML framework.

2 LITERATURE REVIEW

2.1 The Rise of MLOps

The ML lifecycle, spanning from data collection to model deployment, encompasses numerous challenges. These include, but are not limited to, managing cloud deployment, extensive version control, ensuring reproducibility, and handling a constant stream of new data [1,7,9]. Having to accommodate for users with diverse technical skill levels further complicates this process [12].

Machine Learning Operations (MLOps) has risen to meet these challenges, building upon the principles of DevOps, a set of principles for streamlining software development, to address the unique complexities of ML applications [11]. Bridging the gap between ML applications and DevOps, MLOps standardizes and streamlines the ML lifecycle, enabling rapid experimentation, efficient deployment, and enhanced team collaboration [1,4,7,16]. In addition, MLOps automates intricate aspects of ML model deployment and maintenance, helping teams to keep up with ML advancements and quickly deploy new models [1].

2.2 Stages of the Machine Learning Workflow

The ML workflow consists of nine critical stages, as shown in Figure. 2 [2]. This figure illustrates the iterative nature of ML. In this context, "iterative nature" refers to the ongoing process of developing, experimenting with, and continuously improving ML models, even post-delivery of the initial solution. This iterative approach facilitates constant learning and adaptation, leading to more effective and efficient ML models over time [16].

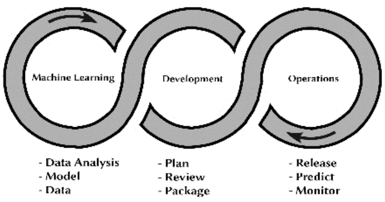

Figure 2. Nine critical stages in ML workflow [25].

2.3 Implementing Machine Learning in Real-World Applications

As the field of ML continues to mature, its practical applications are becoming more widespread and impactful across various sectors [3]. There is a need for further exploration into how ML can be leveraged for railway maintenance [10]. Existing research [13] has laid a foundation for comparing ML frameworks, particularly focusing on managing ML artifacts to enhance reproducibility and repeatability in the ML pipeline. The importance of such work is underlined by the central role reproducibility plays in MLOps principles [7]. After all, the reliability and performance of ML models depend on the consistency of input data and training processes [15]. However, it is crucial to note that the performance of ML models may vary with new data for different applications. The principles of MLOps provide a solid framework for the effective implementation of ML models across diverse applications.

A comprehensive examination of ML frameworks that considers the entire MLOps lifecycle (from data collection to model deployment) remains under-explored. This research aims to bridge that gap, offering an assessment of ML frameworks to better guide their implementation in practice, particularly in railway maintenance.

Success in ML deployment depends largely on three integral elements [14] (three Vs of MLOps): Velocity, Validation, and Versioning.

Velocity is key in ML's experimental process. Essentially, ML engineers need to quickly test and refine ideas. So, they prefer environments that prioritize rapid experimentation and easy debugging [14].

Validation is the second key to success in ML deployment. Given the increased cost of errors the further along they are in the ML workflow, it's highly advised to test changes and actively monitor pipelines for bugs as early as possible in the process[14]. This early validation enables a faster iteration cycle and therefore increases efficiency [14]. Hence, model validation tools are vital for enhancing successful ML deployment.

Versioning is the third important key in ML deployment. It's unrealistic to predict all potential issues in advance, therefore, it is helpful to store and manage multiple versions of production models and datasets. Asset management is one of the commonly reported challenges in ML deployment[6]. When faced with problematic models in production, ML engineers often resort to switching the model to a simpler, historical, or retrained version [14].

ML frameworks play a vital role in managing these three aspects of the ML lifecycle. Ideally, an effective evaluation process of these tools should consider the following general criteria derived from the three Vs of MLOps. For Velocity, an ML framework should provide an environment that supports rapid experimentation. The framework needs to facilitate robust tools for model validation and testing. Lastly, the ML framework should offer version control, enabling the storage and management of multiple versions of models and datasets. Such features contribute to a successful ML deployment[14].

3 REQUIREMENTS ELICITATION

As the research and future development of criteria is based on insights gathered from stakeholder interviews the interview process is outlined in this section. This approach allows the thesis to more closely follow the research progress.

3.1 Methodology

The initial step in the requirements elicitation process was to identify potential ML frameworks for Strukton Rail. A comprehensive spreadsheet of frameworks, provided by the Ambient Intelligence research group at Saxion Hogeschool, served as the starting point. Prior research into ML frameworks was performed by this research team and it consists of 22 potential frameworks. Researching and evaluating all 22 frameworks was outside the scope of this research and therefore, stakeholder interviews were to be conducted to shortlist this selection and evaluate the most promising frameworks.

These stakeholder interviews serve two main purposes. First, they allow for the calibration of the category's weights based on the insights and priorities of the stakeholders. Second, they offer valuable context into the project that helped the subsequent evaluation of the selected ML frameworks. Given the project's technical nature and a small number of stakeholders, interviews were chosen as they allow in-depth, personalized exploration of each stakeholder's viewpoint.

Two key stakeholders were identified and interviewed. The primary goal was to gather insights regarding the project's specific needs, constraints, and goals.

The identified stakeholders were a researcher at Saxion Hogeschool involved in the development of the deep learning model, and a BIM (Building Information Modelling) manager from Strukton Rail, representing the end-user perspective. There was a difference in the focus of the interview between both stakeholders. The researcher provided insight into the requirements and constraints concerning the training and experimentation of the model. Whereas the BIM manager was focused on the deployment and user-friendly maintenance of the ML lifecycle.

The interviews were structured with a list of questions to fall back on (see Appendix A). These were specific to the stakeholder's scope within the project. The questions were open-ended. The interview was encouraged to be open and thus discuss different important requirements for the frameworks. The open-ended conversations enabled a free-flowing exchange of ideas and a thorough exploration of the project's context.

3.2 Results

During the interviews, several key requirements and constraints emerged that initiated the creation of criteria to fairly evaluate multiple frameworks on the categories which encompass such requirements. Interview findings from both the researcher at Saxion Hogeschool and the BIM manager at Strukton are summarized as follows:

Ease of Use and Community Support: The stakeholders emphasized the importance of a framework that is not code-heavy and preferably has an intuitive interface. Additionally, having a vibrant and helpful community was considered essential. This guided the weighting of the Ease of Use and Community Support criteria.

Version Control: The stakeholders indicated the importance of both model and data versioning, with a specific interest in seeing the presence of data versioning. This is reflected in the weighting of the Version Control criteria.

Data volume: It was highlighted that the volume of the data being handled would reach hundreds of GBs. This insight emphasized the need for robust handling and processing capabilities in the chosen ML framework.

Cloud Deployment: The stakeholders conveyed their intent for eventual cloud deployment. This requirement reinforced the importance of Cloud Support in the evaluation criteria.

While these findings are significant, they do not represent an exhaustive list of important evaluation aspects. Instead, they highlight the key insights from the interviews that influenced the weightage of corresponding categories in Section 4.3.1. As both stakeholders represented a different scope within the project there was not any contrasts in preferences. Without these stakeholder insights, the literature research into ML and MLOps might have led to different category prioritization.

4 METHODOLOGY

Building on the insights from the literature review and stakeholder interviews, an evaluation system was created to assess ML frameworks. This section proceeds in three distinct stages. Initially, it explains the selection criteria established for the evaluation. Following that, it continues the process employed to shortlist the candidate ML frameworks. Finally, it outlines the methodology used to rank these selected frameworks against the predefined criteria.

4.1 Criteria Definition

To allow for a fair and objective ranking of different ML frameworks, a detailed criteria list based on insights from stakeholder interviews was developed. General important principles were examined through the literature review. Criterion such as model validation was established from this literature review. The aim of this criterion is to evaluate the "Validation" element of MLOps. Criteria related to the literature review were more focused on understanding how ML models are developed and managed. One category with criteria related to this is Model Management Features. All categories evaluated are Ease-of-use, Scalability, Documentation/Support, Cloud support, Version control, Model management features, and Licensing. To avoid subjectivity, particularly in categories such as 'Ease-of-use', concrete criteria for each category were created, along with a scoring rubric. For instance, one parameter for 'Ease-of-use' was the presence of model architecture visualization, which enhances user understanding of the underlying model architecture. Although not all criteria are objective and binary, extensive effort was made to define fair evaluation criteria for each category. The scoring rubric, further detailed in Table 1, defines specific scores for each criterion based on these parameters, facilitating a more objective comparison of the frameworks.

4.2 Shortlisting of Frameworks

Based on the interview's insight and literature review, a preliminary selection of the most promising frameworks was made. The list of frameworks to be evaluated consisted of:

1. **ClearML** [18]: An open-source ML Operations (MLOps) tool designed to automate the process of tracking, organizing, and optimizing ML experiments and models.
2. **Comet.ml** [19]: A cloud-based ML platform enabling data scientists to track, compare, explain, and reproduce their ML experiments.
3. **MLFlow** [20]: An open-source platform developed by Databricks, specializing in ML experiment tracking, reproducibility, and model deployment.
4. **OmegaML** [21]: A flexible, open-source platform that focuses on data operations and model management, allowing quick model deployment.
5. **Polyaxon** [22]: An open-source platform for ML automation and management. It supports the entire lifecycle of ML and deep learning applications.
6. **PyTorch** [17] **with TorchStudio**[8]: PyTorch is an open-source ML library known for its flexibility and ease of use, and TorchStudio is a visual tool for ML development built on top of PyTorch, providing a graphical interface and MLOps features.
7. **TensorFlow Extended (TFX)** [23]: TFX, built on top of TensorFlow is Google's end-to-end platform for deploying production-ready ML pipelines. TensorFlow is a widely used, open-source ML library with comprehensive features.
8. **Weights & Biases (W&B)** [24]: A toolset for ML that allows for tracking/visualizing metrics, model prediction outputs, and hyperparameter tuning.

All of these tools work with ML libraries and thus presumably also with TensorFlow and PyTorch. However, TensorFlow and PyTorch were chosen as they had dedicated MLOps platforms built on top.

Table 1. Rubric for evaluating ML framework.

Ease-of-use	0	1	2
GUI	No GUI feature	Presence of GUI	
Experiment/Metric visualization	Not Possible	Possible	
Model architecture visualization	Not possible	Setup or extension required	Native feature
Dashboard	Not possible	Through extension	Native feature
Hyperparameter tuning tools	Not possible or extensive setup	Not native but possible through extension/setup	Native feature
Pipeline Visualization	Not available or difficult setup	(Data)/Input analyzer	End-to-end visualization
Real-time visual progress/experiment tracking tools	Not available	Available	

Documentation	0	1	2
Comprehensive documentation	No documentation or outdated	Somewhat comprehensive documentation/unclear	Comprehensive/clear documentation
Community forum	Not available	Available but inactive/outdated	Active community forum
Stack Overflow	Latest questions/tagged topic > 14 days	2-14 days	Latest question < 48 hours
Official support	No official support channel	Official support channel	
Training materials	No official training material/No recent contributed material	Some training material available	Extensive official and community-contributed training material
Updates	No update since 2023	Update with minimal release notes/at least 2 months old	Active/extensive update, most recent < 2 months
FAQ	Not available	Available	
Official youtube videos	Not available	Available	

Model Management Features	0	1	2
Experiment Tracking	Not available	Limited tracking	Full-tracking availability
Data Tracking	Not available	Limited tracking	Full-tracking availability
Model Tracking	Not available	Available	
Automatic Pipeline Resumption	Not available	Available	
Pipeline Scheduler	Not available	Available	
Model deployement	Not available	Available	
Model testing and validaton tools	Not available	Minimal tools available	Robust features for model validation and testing

Version Control	0	1	2
Model Versioning	Not available	Available	
Data Versioning	Not available	Available	

Licensing	0	1	2
Cost per year (in euros)	> 1000	200-1000	< 200
Unlimited Users	Per user basis	No restrictions on users (or not relevant)	
Restrictions (posed by license terms)	Restrictive license type	Open source/limited restrictions	

Cloud Support	0	1	2
Cloud-services	No support	Limited services/extensive setup	Supports major cloud services/simple setup

Table 2. Results for ML framework evaluation

Results	Ease-of-use	Versioning	Documentation Support	Model Management Features	Cloud Support	Scalability	Licensing	Compatibility Score
ClearML	5,0	5,0	4,6	4,0	5,0	5,0	3,8	4,7
W&B	4,1	5,0	5,0	5,0	5,0	5,0	1,3	4,6
Comet	3,2	5,0	3,1	3,5	5,0	5,0	2,5	3,9
MLFlow	2,3	5,0	4,2	3,5	2,5	3,8	5,0	3,6
TensorflowXTD	3,2	2,5	4,2	4,5	5,0	3,8	5,0	3,7
Polyaxon	2,3	2,5	3,8	1,5	5,0	5,0	0,0	2,9
PyTorch	2,7	0,0	4,6	3,5	5,0	3,8	5,0	3,0
OmegaML	0,9	0,0	1,5	5,0	5,0	5,0	0,0	2,2

4.3 Framework Evaluation

To thoroughly examine the capabilities of the selected ML frameworks, an evaluation was conducted based on the set of predetermined criteria previously explained. All frameworks were evaluated on these criteria and ranked according to the scoring rubric. Each framework was extensively researched through its documentation and relevant websites concerning the ML framework. These provided the sources for scoring a framework on a criterion. The sources are provided in Criteria References. The final score of a category is calculated by summing all individual criteria scores and dividing this by the total possible score for all criteria. Criteria with a maximum score of 2 naturally have more influence on the average category score than criteria with a maximum score of 1. This is intentional as some criteria are seen as more influential and there was nuance between scoring a 0 or a 1. The average category score is then normalized on a 1-5 scale where a 1 represents a poor score and 5 a high score. This results in each category having an average score based on concretely defined criteria.

4.3.1 Compatibility Scoring

To find the most suitable framework relevant to this project a Compatibility Score for each ML Framework is computed. This score quantifies the overall compatibility of each ML Framework with the needs and constraints identified for the project.

The Compatibility Score was computed by assigning weights to each category, reflecting their relative importance as identified through the stakeholder interviews. These weights reflect the varying impact of each category on the overall compatibility of the ML framework with the project. The aim of this structured and methodical evaluation was to obtain an objective understanding of each framework's ability to meet the project's needs and constraints. This scoring system provides a direct, quantifiable comparison among the ML frameworks. Compatibility Score is calculated as a weighted average using these weights:

1. *Ease-of-use (EoU): 25%*
2. *Version Control (VC): 20%*
3. *Documentation/Support (DS): 15%*
4. *Model Management Features (MMF): 15%*
5. *Cloud Support (ClS): 10%*
6. *Scalability (SC): 10%*
7. *Licensing (L): 5%*

This robust evaluation process, intertwined with literature study and stakeholder engagement, was designed to allow for a comprehensive, balanced, and objective assessment of potential ML frameworks, ultimately guiding the selection of the best-suited framework for this project.

5 RESULTS

5.1 Machine Learning Framework Evaluation Results

Using the outlined methodology all frameworks are evaluated and the average category scores are displayed in Table 2. Average category scores are calculated using the scoring rubric outlined in Section 4.1. To improve readability, only the average category scores are displayed, not all individual criteria scores.

6 DISCUSSION

6.1 Challenges in Evaluating Different Frameworks

Evaluating multiple frameworks introduced various challenges. The impracticality/feasibility of installing and testing each one resulted in a heavy reliance on framework documentation, creating potential gaps in experiential understanding.

6.2 Establishing a Rating System

Establishing a fair and concrete rating system posed significant challenges due to the difficulty in quantifying subjective aspects such as ease of use and compatibility with varying levels of ML expertise. Criteria for ranking were not all objective and there was still some subjectivity in how a ML framework scored. As this research is centred around Strukton Rail the ranking system is subject to specific requirements and biases which restrict the universality of the ranking system.

6.3 Recommendation for Strukton Rail

The research question in this thesis was designed to identify the most compatible ML framework for Strukton Rail.
The application of this research methodology has led to the selection of ClearML as the most compatible solution. This selection is backed by a comprehensive criteria-based evaluation outlined in Section 4. This effectively demonstrates the compatibility of ClearML with Strukton Rail's needs. It is important to note that the weightings given to each category, derived from interviews, may not fully capture objectivity. Under these weights, ClearML emerged as the top choice. However, if the priority were to shift towards Model Management Features over Ease-of-Use, W&B would emerge as the preferable selection.

It is important to mention the challenging nature of practical ML implementation. This is eemphasized by the fact that only one in two organizations manages to progress beyond pilots and proofs of concept [11]. Consequently, while this thesis provides a strong recommendation for ClearML based on the rigorous evaluation performed, the ultimate success of ML deployment at Strukton Rail will depend on careful and strategic implementation.

6.4 Future Work

While this study provides a groundwork for evaluating ML frameworks, it's largely specific to Strukton Rail.
Future work should however incorporate unique insights from their organization through activities such as stakeholder interviews.

Hands-on testing of chosen frameworks could be considered for deeper understanding, and further refinement of the rating system. This could help in better quantifying subjective aspects.

7 CONCLUSION

This research explored the process of identifying a suitable ML framework for Strukton Rail. To do this, a structured methodology was used, deriving evaluation criteria from stakeholder interviews, and establishing a comprehensive rating system.

While the process faced some limitations due to practical challenges, it offered valuable insights into the project's context. ClearML was selected as the most suitable framework for Strukton Rail, demonstrating strong usability, cloud support, community strength, model management features and data handling capabilities.

This study serves as a blueprint for future work in this area. However, the methodology needs to be adjusted to cater to each organization's unique requirements and circumstances. Despite acknowledging the limitations, this research emphasizes the crucial role of a well-chosen ML framework in successful ML implementation.

In conclusion, this research follows a systematic and context-driven approach in selecting an ML framework, contributing to a successful deployment of ML within Strukton Rail.

REFERENCES

[1] Sridhar Alla and Suman Kalyan Adari. 2021. *Beginning MLOps with MLFlow: Deploy Models in AWS SageMaker, Google Cloud, and Microsoft Azure*. Apress, Berkeley, CA. DOI:https://doi.org/10.1007/978-1-4842-6549-9
[2] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. 2019. Software Engineering for Machine Learning: A Case Study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, IEEE, Montreal, QC, Canada, 291–300. DOI:https://doi.org/10.1109/ICSE-SEIP.2019.00042
[3] Jacques Bughin, Eric Hazan, Sree Ramaswamy, Tera Allas, Peter Dahlström, Nicolaus Henke, and Monica Trench. 2017. ARTIFICIAL INTELLIGENCE THE NEXT DIGITAL FRONTIER? Retrieved from https://www.mckinsey.com/~/media/mckinsey/industries/advanced%20electronics/our%20insights/how%20artificial%20intelligence%20can%20deliver%20real%20value%20to%20companies/mgi-artificial-intelligence-discussion-paper.ashx
[4] Nipuni Hewage and Dulani Meedeniya. 2022. Machine Learning Operations: A Survey on MLOps Tool Support. (2022). DOI:https://doi.org/10.48550/arXiv.2202.10169
[5] Julio Hurtado, Dario Salvati, Rudy Semola, Mattia Bosio, and Vincenzo Lomonaco. 2023. Continual learning for predictive maintenance: Overview and challenges. *Intell. Syst. Appl.* 19, (September 2023), 200251. DOI:https://doi.org/10.1016/j.iswa.2023.200251
[6] Samuel Idowu, Daniel Struber, and Thorsten Berger. 2021. Asset Management in Machine Learning: A Survey. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, IEEE, Madrid, ES, 51–60. DOI:https://doi.org/10.1109/ICSE-SEIP52600.2021.00014
[7] Dominik Kreuzberger, Niklas Kühl, and Sebastian Hirschl. 2023. Machine Learning Operations (MLOps): Overview, Definition, and Architecture. *IEEE Access* 11, (2023), 31866–31879. DOI:https://doi.org/10.1109/ACCESS.2023.3262138
[8] Robin Lobel. TorchStudio. *TorchStudio*. Retrieved June 26, 2023 from https://www.torchstudio.ai/
[9] Andrei Paleyes, Raoul-Gabriel Urma, and Neil D. Lawrence. 2023. Challenges in Deploying Machine Learning: A Survey of Case Studies. *ACM Comput. Surv.* 55, 6 (July 2023), 1–29. DOI:https://doi.org/10.1145/3533378
[10] Juan Pineda-Jaramillo and Francesco Viti. 2023. MLOps in freight rail operations. *Eng. Appl. Artif. Intell.* 123, (August 2023), 106222. DOI:https://doi.org/10.1016/j.engappai.2023.106222
[11] Khalid Salama, Jarek Kazmierczak, and Donna Schut. 2021. Practitioners guide to MLOps: A framework for continuous delivery and automation of machine learning. (May 2021).
[12] Sebastian Schelter, Felix Biessmann, Tim Januschowski, David Salinas, Stephan Seufert, and Gyuri Szarvas. 2015. On Challenges in Machine Learning Model Management. (2015).
[13] Marius Schlegel and Kai-Uwe Sattler. 2022. Management of Machine Learning Lifecycle Artifacts: A Survey. *SIGMOD Rec.* 51, 4 (2022).
[14] Shreya Shankar, Rolando Garcia, Joseph M. Hellerstein, and Aditya G. Parameswaran. 2022. Operationalizing Machine Learning: An Interview Study. Retrieved July 1, 2023 from http://arxiv.org/abs/2209.09125
[15] Matei Zaharia, Andrew Chen, Aaron Davidson, Ali Ghodsi, Sue Ann Hong, Andy Konwinski, Siddharth Murching, Tomas Nykodym, Paul Ogilvie, Mani Parkhe, Fen Xie, and Corey Zumar. Accelerating the Machine Learning Lifecycle with MLflow.
[16] Yizhen Zhao. Machine Learning in Production: A Literature Review.
[17] 2023. PyTorch. *Wikipedia*. Retrieved June 26, 2023 from https://en.wikipedia.org/w/index.php?title=PyTorch&oldid=1161155029
[18] What is ClearML? | ClearML. Retrieved June 26, 2023 from https://clear.ml/docs/latest/docs/
[19] Comet ML - Build better models faster. Retrieved June 26, 2023 from https://www.comet.com/site/
[20] MLflow - A platform for the machine learning lifecycle | MLflow. Retrieved June 26, 2023 from https://mlflow.org/
[21] MLOps simplified. Deliver Data Products, not just models. *omegaml*. Retrieved June 26, 2023 from https://www.omegaml.io

[22] Open source Machine Learning at scale with Kubernetes - MLOps Lifecycle for data-scientists & machine-learning engineers - Model & Data Centric AI. *Polyaxon*. Retrieved June 27, 2023 from https://polyaxon.com/
[23] TFX | ML Production Pipelines. *TensorFlow*. Retrieved June 26, 2023 from https://www.tensorflow.org/tfx
[24] Weights & Biases – Developer tools for ML. Retrieved June 27, 2023 from https://wandb.ai/site/, http://wandb.ai/site
[25] Machine Learning Operations (MLOps). *Neal Analytics*. Retrieved July, 2023 from https://nealanalytics.com/expertise/mlops/

CRITERIA REFERENCES

References used in evaluating ML frameworks on established criteria.

[1] Akinwande Komolafe. 2023. Top Model Versioning Tools for Your ML Workflow. neptune.ai. Retrieved June 2023 from https://neptune.ai/blog/top-model-versioning-tools
[2] Gideon Mendels. 2018. Building reliable machine learning models with cross-validation. Comet. Retrieved June 2023 from https://www.comet.com/site/blog/building-reliable-machine-learning-models-with-cross-validation/
[3] Slack. Join MLflow on Slack. Slack. Retrieved June 2023 from https://slack.com/join/shared_invite/zt-1xj105pvl-bOc7ri2ZgiqasJ1p1cFUmA
[4] 2020. How to Display Model Evaluation Metrics Using MLflow Search API. Databricks. Retrieved June 2023 from https://www.databricks.com/blog/2020/02/18/how-to-display-model-metrics-in-dashboards-using-the-mlflow-search-api.html
[5] 2020. omega|ml community. omega|ml community. Retrieved June 2023 from https://community.omegaml.io/forum/discussions
[6] 2022. Model Evaluation in MLflow. Databricks. Retrieved June 2023 from https://www.databricks.com/blog/2022/04/19/model-evaluation-in-mlflow.html
[7] 2023. BSD licenses. Wikipedia. Retrieved June 2023 from https://en.wikipedia.org/w/index.php?title=BSD_licenses&oldid=1158850428
[8] 2023. allegroai/clearml. Retrieved June 2023 from https://github.com/allegroai/clearml/blob/5824e27adae42302e3b8b13d10584e4a0fe4209b/LICENSE
[9] Why do you need to use W&B to track your ML project? | by LittleBigCode - AI Solution Creator | BEYOND DATA by LittleBigCode | Medium. Retrieved June 2023 from https://medium.com/hub-by-littlebigcode/why-do-you-need-to-use-weights-biases-to-track-your-ml-project-f093562287d8
[10] What is ClearML? | ClearML. Retrieved June 2023 from https://clear.ml/docs/latest/docs/
[11] Welcome to TensorFlow's global community. Retrieved June 2023 from https://www.tensorflow.org/community
[12] Welcome to PyTorch Tutorials — PyTorch Tutorials 2.0.1+cu117 documentation. Retrieved June 2023 from https://pytorch.org/tutorials/
[13] Weights & Biases – Developer tools for ML. Retrieved June 2023 from https://wandb.ai/site
[14] wandb/CHANGELOG.md at main · wandb/wandb · GitHub. Retrieved June 2023 from https://github.com/wandb/wandb/blob/main/CHANGELOG.md
[15] W&B Docs | Weights & Biases Documentation. Retrieved June 2023 from https://docs.wandb.ai/
[16] W&B Community. Retrieved June 2023 from https://community.wandb.ai/
[17] Tutorials Archives - Comet. Retrieved June 2023 from https://www.comet.com/site/blog/category/tutorials/
[18] Tutorial — MLflow 2.4.1 documentation. Retrieved June 2023 from https://mlflow.org/docs/latest/tutorials-and-examples/tutorial.html
[19] The TFX User Guide | TensorFlow. Retrieved June 2023 from https://www.tensorflow.org/tfx/guide
[20] The complete guide to ML model visualization with Tensorboard | cnvrg.io. Retrieved June 2023 from https://cnvrg.io/tensorboard-guide/
[21] The Best Tools for Machine Learning Model Visualization. Retrieved June 2023 from https://neptune.ai/blog/the-best-tools-for-machine-learning-model-visualization
[22] Terms of Service. Comet. Retrieved June 2023 from https://www.comet.com/site/terms-of-service/
[23] TensorBoard | TensorFlow. Retrieved June 2023 from https://www.tensorflow.org/tensorboard
[24] Releases · pytorch/pytorch. Retrieved June 2023 from https://github.com/pytorch/pytorch/releases
[25] Releases · omegaml/omegaml. GitHub. Retrieved June 2023 from https://github.com/omegaml/omegaml/releases
[26] Releases · mlflow/mlflow. Retrieved June 2023 from https://github.com/mlflow/mlflow/releases
[27] Releases · allegroai/clearml. GitHub. Retrieved June 2023 from https://github.com/allegroai/clearml/releases
[28] PyTorch Forums. Retrieved June 2023 from https://discuss.pytorch.org/
[29] PyTorch documentation — PyTorch 2.0 documentation. Retrieved June 2023 from https://pytorch.org/docs/stable/index.html
[30] Polyaxon Docs. Retrieved June 2023 from https://polyaxon.com/docs/
[31] polyaxon · Discussions · GitHub. Retrieved June 2023 from https://github.com/orgs/polyaxon/discussions
[32] MLOps simplified. Deliver Data Products, not just models. omegaml. Retrieved June 2023 from https://www.omegaml.io
[33] MLOps for humans — omega|ml 1.1 documentation. Retrieved June 2023 from https://omegaml.github.io/omegaml/
[34] mlflow-users - Google Groepen. Retrieved June 2023 from https://groups.google.com/g/mlflow-users
[35] mlflow-resources/MLflow_FAQ.md at master · amesar/mlflow-resources · GitHub. Retrieved June 2023 from https://github.com/amesar/mlflow-resources/blob/master/MLflow_FAQ.md
[36] Issues · tensorflow/tensorflow. GitHub. Retrieved June 2023 from https://github.com/tensorflow/tensorflow
[37] Hyperparameter Tuning With MLflow Tracking | by aditi kothiya | The Startup | Medium. Retrieved June 2023 from https://medium.com/swlh/hyperparameter-tuning-with-mlflow-tracking-b67ec4de18c9
[38] Getting Started with omega|ml — omega|ml 1.1 documentation. Retrieved June 2023 from https://omegaml.github.io/omegaml/admin/kickstart.html#client-configuration
[39] Features. Retrieved June 2023 from https://www.torchstudio.ai/features/
[40] FAQ - W&B Community. Retrieved June 2023 from https://community.wandb.ai/faq
[41] FAQ - Polyaxon. Retrieved June 2023 from https://polyaxon.com/faq/
[42] FAQ. PyTorch Forums. Retrieved June 2023 from https://discuss.pytorch.org/faq
[43] End to End ML pipelines with MLflow | by Haythem tellili | Medium. Retrieved June 2023 from https://medium.com/@haythemtellili/end-to-end-ml-pipelines-with-mlflow-projects-63a11baa2dd1
[44] Docs Home - Comet Docs. Retrieved June 2023 from https://www.comet.com/docs/v2/
[45] Contact Us | Comet ML. Retrieved June 2023 from https://www.comet.com/site/about-us/contact-us/
[46] Contact ClearML: We Are Here to Help You. Retrieved June 2023 from https://clear.ml/contact-us/
[47] ClearML Community: Built by the Team and Folks Like You. Retrieved June 2023 from https://clear.ml/community/
[48] Automated Hyperparameter Optimization Made Easy - ClearML. Retrieved June 2023 from https://clear.ml/blog/hyperparameter-optimization-hpo-is-simpler-than-you-think/
[49] Artifacts. Comet. Retrieved June 2023 from https://www.comet.com/site/products/artifacts-dataset-management/
[50] A Beginner's Guide to TorchStudio, The PyTorch IDE. Retrieved June 2023 from https://www.assemblyai.com/blog/beginners-guide-to-torchstudio-pytorch-only-ide/
[51] 07. PyTorch Experiment Tracking - Zero to Mastery Learn PyTorch for Deep Learning. Retrieved June 2023 from https://www.learnpytorch.io/07_pytorch_experiment_tracking/

APPENDIX A

Questions created prior to the interview.

- What volume of data do you anticipate your system will need to handle on a daily basis?
- How quickly will your system need to process data? What's the expected latency for data processing and decision making?
- How important is it to have a robust community and support network behind the machine learning framework you choose?
- What level of documentation would you expect for the ML framework?
- Are there any specific systems or databases that the ML framework must be compatible with?
- Would you need to export or import models from other frameworks?
- How scalable does the machine learning framework need to be to accommodate potential growth in data volume or complexity of models?

- What types of machine learning models do you anticipate needing?
- What is the technical proficiency of the team that will be using the ML framework? Would they prefer a code-heavy framework or a drag-and-drop interface?
- What are your requirements regarding data security and privacy?
- Are there any compliance regulations that the ML framework needs to meet?
- What are your needs in terms of model deployment? Do you need the framework to support mobile deployment, real-time processing, batch processing, etc.?
- How important is model monitoring, versioning, and lifecycle management for you?
- How important is it for your team to be able to experiment with different models and configurations?
- How essential is it for your machine learning experiments to be reproducible?
- What is your preference when it comes to deploying the machine learning framework: on-site hardware or a cloud-based service? Can you elaborate on the reasons for your preference?
- How would you evaluate the trade-offs between a cloud-based machine learning framework and an on-premise solution in terms of cost, performance, and data security?
- In your own view, what would be the 'dream' machine learning solution or application for Strukton Rail? What would it look like and what kind of impact would it have on the company and the broader industry?