TRANSMISI: JURNAL ILMIAH TEKNIK ELEKTRO, 26, (2), APRIL 2024
p-ISSN 1411-0814 e-ISSN 2407-6422

DEVELOPMENT OF TIME-SERIES-BASED MLOPS ARCHITECTURE FOR PREDICTING SALES QUANTITY IN MICRO, SMALL, AND MEDIUM ENTERPRISES (MSMES)

Salsabila Putri Lesmarna[1], Farrikh Alzami[1, *)], Ifan Rizqa[1], Abu Salam[1], Diana Aqmala[2], Rama Aria Megantara[1] and Ricardus Anggi Pramunendar[1]

[1] Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang, Jawa Tengah, Indonesia
[2] Faculty of Economics and Business, Universitas Dian Nuswantoro, Semarang, Jawa Tengah, Indonesia

[*)] *E-mail: alzami@dsn.dinus.ac.id*

Abstract

Micro, Small, and Medium Enterprises (MSMEs) constitute a significant portion of the economy in many developing countries, playing a vital role in employment generation and economic growth. Sales performance is a critical factor for MSMEs, influenced by various internal and external factors. Time-series analysis offers a valuable tool to predict sales quantities by analyzing historical data and identifying patterns and trends. In this context, the SARIMAX (Seasonal Autoregressive Integrated Moving Average with Exogenous Variables) model emerges as a suitable method to forecast future sales, leveraging both historical data and external variables. This research explores the synergy between time-series analysis, specifically SARIMAX modeling, and MLOps (Machine Learning Operations). Finally, this research aims to provide a framework for the practical application of MLOps to enhance sales forecasting and decision-making processes within MSMEs, fostering their growth and sustainability in a competitive market landscape.

Keywords: MLOps, Hopsworks, Prediction, Sales, MSMEs

Abstrak

Usaha Mikro, Kecil, dan Menengah (UMKM) merupakan bagian penting dari perekonomian di banyak negara berkembang, memainkan peran vital dalam penciptaan lapangan kerja dan pertumbuhan ekonomi. Kinerja penjualan adalah faktor kritis bagi UMKM, dipengaruhi oleh berbagai faktor internal dan eksternal. Analisis deret waktu (*time series analysis*) menawarkan alat yang berharga untuk memprediksi jumlah penjualan dengan menganalisis data historis serta mengidentifikasi pola dan tren. Dalam konteks ini, model SARIMAX (*Seasonal Autoregressive Integrated Moving Average with Exogenous Variables*) muncul sebagai metode yang sesuai untuk meramalkan penjualan di masa depan, dengan memanfaatkan data historis dan variabel eksternal. Penelitian ini mengeksplorasi sinergi antara analisis deret waktu, khususnya pemodelan SARIMAX, dan MLOps (Machine Learning Operations). Akhirnya, penelitian ini bertujuan untuk menyediakan kerangka kerja (*framework*) untuk penerapan praktis MLOps guna meningkatkan peramalan penjualan dan proses pengambilan keputusan dalam UMKM, mendorong pertumbuhan dan keberlanjutan mereka dalam lanskap pasar yang kompetitif

kata kunci: MLOps, Hopsworks, Prediksi, Penjualan, UMKM

1. Introduction

Micro, Small, and Medium Enterprises (MSMEs) are a type of business characterized by their small to medium scale, limited workforce, and assets [1]. MSMEs play a vital role in the national economy and constitute the majority of employment in developing countries [2]. Sales are a crucial aspect for MSMEs, as they directly impact their performance and growth. Several factors influence sales volume, including internal factors such as product or service quality, the consistency of the business owner, and innovation in marketing strategies [3]. Additionally, external factors also play a role, such as market trends and other elements that can influence the sales performance of MSMEs [4].

In addressing the dynamics and fluctuations in sales volume, time-series analysis plays a crucial role. Time-series analysis serves to predict sales quantities by analyzing historical data to identify patterns and trends that can be utilized to forecast future sales [5], [6]. By scrutinizing past sales data, MSMEs can pinpoint trends, changes in consumer behavior, and other factors that may impact future sales. Moreover, time-series analysis aids MSMEs in tackling inventory shortages and optimizing sales [7]. To optimize the use of time-series analysis,

particularly in the context of MSMEs, appropriate methods or models are required.
One suitable model that can be employed is the SARIMAX (Seasonal Autoregressive Integrated Moving Average with Exogenous Variables) model, which is an extension of the ARIMA (Autoregressive Integrated Moving Average) model [8]. SARIMAX is a time-series forecasting model used to predict future values of the dependent variable based on past values and the values of other independent variables. In several studies, SARIMAX has been used to forecast electricity consumption [9], predict chickenpox cases [10], and forecast the spread of COVID-19 [11]. In these studies, SARIMAX has been demonstrated to provide accurate forecasts and assist in decision-making based on the expected outcomes.

In this scenario, the relevance of MLOps, or machine learning operations, becomes apparent. MLOps combines best practices from software development and artificial intelligence to operate machine learning models in a production environment [12]. In the case of MSMEs, MLOps is needed to streamline their machine learning workflows, reducing the time and effort required for model development, training, and deployment [13]. However, the implementation of MLOps is still relatively rare compared to DevOps, which has seen rapid development and widespread adoption [14]. MLOps in large organizations, such as IBM, Amazon, and Microsoft, have garnered significant attention, but studies focusing on how MLOps can be concretely and effectively applied in the MSME environment are still limited. However, MLOps offers numerous advantages, including efficiency, automation of machine learning workflows, which can save time and reduce errors [15]. In terms of reliability, MLOps can help ensure that machine learning models are robust and accurate by providing tools for versioning, testing, and monitoring. Furthermore, in the aspect of collaboration, MLOps encourages collaboration between data scientists and operations teams, fostering better communication and more effective problem-solving [16].

Thus, in this research, we are developing an MLOps architecture based on time-series data to predict sales quantities in the context of micro, small, and medium enterprises (MSMEs) with the aim of simplifying sales forecasting. By integrating the principles of MLOps and time-series data analysis, this study aims to assist MSMEs in addressing the challenges posed by fluctuations in sales volume, enabling them to make smarter and more informed business decisions. By identifying the advantages, constraints, and challenges of merging these two paradigms, this research will help fill the knowledge gap in the literature regarding the concept and application of MLOps in the context of time-series analysis.

2. Methods

Our research methodology consists of several stages: first, we process the dataset. Second, we process the dataset to save features for forecasting in Hopsworks. Third, we train the dataset using the SARIMAX algorithm and save it to the model registry Hopsworks. Fourth, we use the model to predict the sales quantity with actionable insight. Fifth, we deploy the machine learning model using Streamlit and Hugging Face. Figure 1 shows a diagram of the overall process stages.

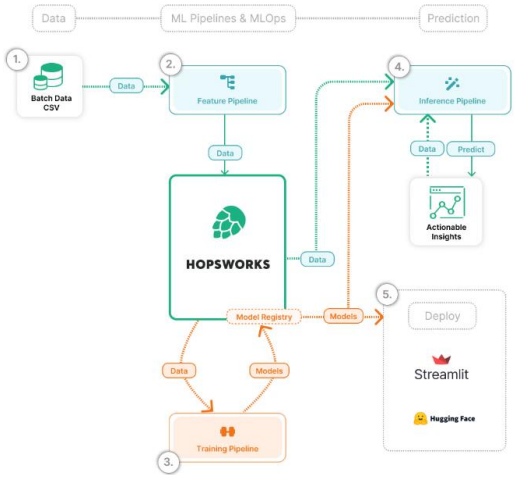


Figure 1. Flow MLOps Architecture

For a detailed explanation of Figure 1. Can be described as follows:

2.1. MLOps

MLOps is a method that combines software engineering and machine learning to manage the entire lifecycle of a machine learning model, from development to deployment and maintenance [12]. MLOps aims to streamline the process of building, testing, and implementing machine learning models, making it more efficient [12]. It involves the use of tools and techniques such as version control, integration, and deployment, as well as monitoring and logging to ensure that machine learning models are developed and deployed consistently and reproducibly [15].

2.2. Hopsworks

Hopsworks is an open-source artificial intelligence (AI) platform that provides a comprehensive set of services for managing the entire data lifecycle in machine learning (ML) and deep learning (DL) pipelines [17]. Hopsworks organizes and stores ML experiments for easy reproducibility and archiving. Within Hopsworks, models are trained with pre-selected hyperparameter settings using a number of GPUs and the RingAll-Reduce method

for distributed data training. The output of the training process is a model stored within HopsFS [18].

Hopsworks also includes a multi-tenancy environment based on projects, allowing students to collaborate in groups. It supports designing, debugging, and running deep learning workflows at scale. All stages of the workflow can be horizontally scaled because the platform manages the entire stack, from resource management (YARN with GPU support) to API support for running distributed training and hyperparameter optimization experiments. Hopsworks provides tools and infrastructure to streamline the development, deployment, and monitoring of ML models, including the implementation of feature pipelines, training pipelines, and batch inference pipelines for MLOps [18].

2.3. SARIMAX

SARIMAX is a time-series forecasting algorithm that combines the Seasonal Autoregressive Integrated Moving Average (SARIMA) model with external variables [19]. This algorithm is used to predict the future values of a time series based on past values and additional factors that may influence it. SARIMAX is particularly useful for modeling time-series data that exhibit seasonality and for incorporating external variables not present in the time series itself [20].

2.4. Streamlit

Streamlit is a Python framework designed for building web applications. Within this framework, it is possible to create interactive interfaces containing visualizations [21]. Streamlit is utilized to construct web applications for data science and machine learning, where users can interact with machine learning models to make predictions, for instance [22]. This allows users to engage with machine learning models by providing input and receiving predictions or recommendations.

2.5. Hugging Face

Hugging Face is a platform specifically designed to host and develop machine learning (ML) projects. Its primary focus is to facilitate the sharing of datasets, pre-trained ML models, and applications built using these models. The platform also offers collaborative features such as issues and pull requests to support the growth and development of ML artifacts. With over 100,000 repositories, Hugging Face presents a promising source for empirical studies of ML projects and interactions within the community [23].

3. Results and Discussion

In this research, we outline the phases involved in the development of a Time-Series-Based MLOps Architecture for Predicting Sales Quantity in Micro, Small, and Medium Enterprises (MSMEs). The steps are explained as follows:

3.1. Batch Data

The dataset used in this study is from Perrin Freres Monthly Champagne Sales at https://www.kaggle.com/datasets/galibce003/perrin-freres-monthly-champagne-sales. The dataset has several variables. The variables are shown in Table 1.

Table 1. Variable Data

Variable	Description
month	Time of sales
sales	Sales quantity

3.2. Feature Pipeline

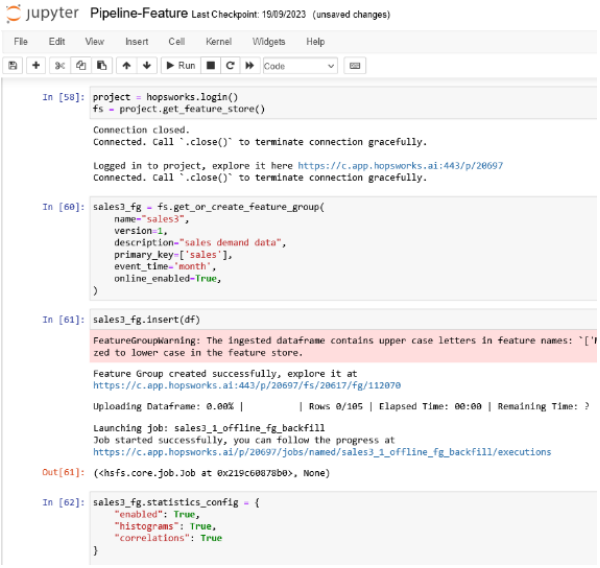


Figure 2. Feature Pipeline on Jupiter Notebook

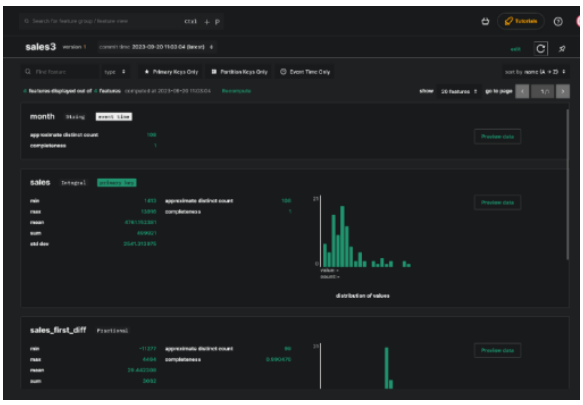


Figure 3. Feature Statistics in Hopsworks

A feature pipeline is a sequence of data preprocessing and feature engineering steps that transform raw data into a

format suitable for training ML models. In Hopsworks, you can implement a feature pipeline using the feature engineering capabilities provided by the platform. This involves data preprocessing, feature extraction, and transformation tasks. Figure 2 shows the implementation of the feature pipeline in Jupiter Notebook.

In this process, the required features will be stored in the Hopsworks Feature Store. Within Hopsworks, we can view data previews, data statistics, and correlations among the features within the data. The data display stored in Hopsworks can be seen in Figure 3.

3.3. Training Pipeline

```
In [7]: # set up dates
start_time = "2004-01-01"
end_time = "2012-09-01"

# create a training dataset
version, job = feature_view.create_training_data(
    start_time=start_time,
    end_time=end_time,
    description='Description of a dataset',
)

Training dataset job started successfully, you can follow the progress at
https://c.app.hopsworks.ai/p/20697/jobs/named/sales3_1_create_fv_td_200920230

VersionWarning: Incremented version to `1`.
```

Figure 4. Implementation of Training Pipeline on Jupiter Notebook

The Training Pipeline retrieves data from Hopsworks, processes it through a split train-test operation, and conducts modeling using an appropriate machine learning algorithm based on the objective. Models are trained with Hopsworks, a CI/CD workflow can be set up where experiments are tracked by Hopsworks, and every model created is published to a model registry. Each project has its own private model registry, so when working on a development project, the model is typically published to the project's private development registry. Figure 4 shows the implementation of a training pipeline for forecasting sales in Jupiter Notebook. We set the start and end dates for training data. After that, we use the training data for modeling with the algorithm SARIMAX and save the model into the Hopsworks.

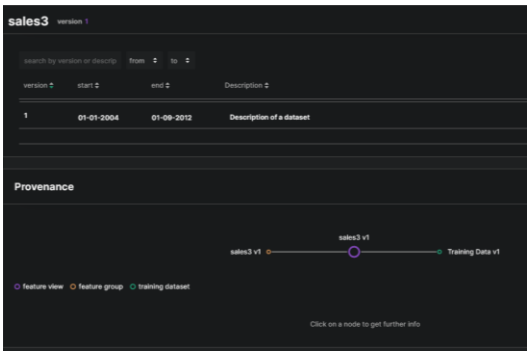


Figure 5. Training Version in Hopsworks

The training pipeline's results within Hopsworks include monitoring, versioning of training, and model storage with versioning, including metrics. The training version is shown in Figure 5. This makes it easy to monitor and track the progress of different training iterations and model versions. By maintaining a record of model versions as shown in Figure 6 and associated metrics, it becomes convenient to compare and analyze the performance improvements or changes in various iterations. This ensures a streamlined and transparent approach to managing the development and improvement of machine learning models within the Hopsworks platform.

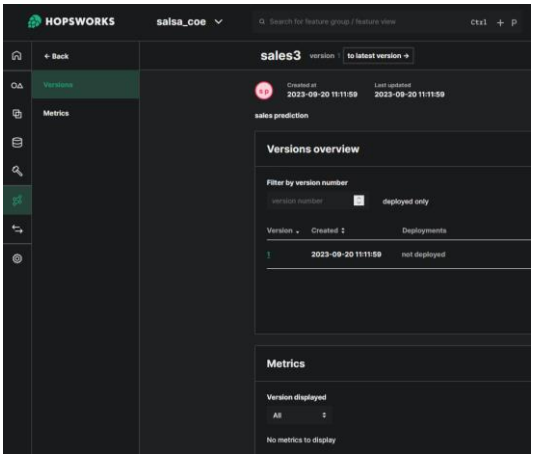
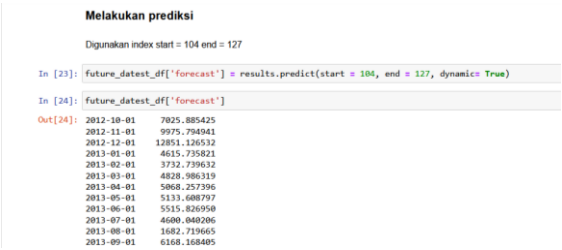


Figure 6. Model Registry in Hopsworks

3.4. Batch Inference Pipeline



Melakukan prediksi

Digunakan index start = 104 end = 127

```
In [23]: future_datest_df['forecast'] = results.predict(start = 104, end = 127, dynamic= True)

In [24]: future_datest_df['forecast']

Out[24]: 2012-10-01     7025.885425
         2012-11-01     9975.794941
         2012-12-01    12851.126532
         2013-01-01     4615.735821
         2013-02-01     3732.739632
         2013-03-01     4828.986319
         2013-04-01     5068.257396
         2013-05-01     5133.608797
         2013-06-01     5515.826950
         2013-07-01     4600.040206
         2013-08-01     1682.719665
         2013-09-01     6168.168405
```

Figure 7. Implementation Batch Inference Pipeline in Jupiter Notebook

A batch inference pipeline is used to make predictions on a batch of data using a trained ML model. In this stage, we can set up batch inference pipelines that take input data, apply the trained model to make predictions, and store the results. These prediction results can be actionable insights because they can be used to take concrete actions or make decisions. In the case of MSMEs (micro, small, and medium-sized enterprises), forecasting sales figures for the next 24 months can be highly valuable for business planning and decision-making. The implementation of this stage is shown in Figure 7.

3.5. Deployment

Deployment involves two stages: creating a Streamlit interface saved in a py file and deploying it on the Hugging Face platform. Inside the py script file, don't forget to integrate the model that was previously saved in

Hopsworks. The next step is to deploy the Streamlit interface on the Hugging Face platform. To integrate it with Hopsworks, you do this by configuring secret variables in the Hugging Face settings. The result of deployment is shown in Figure 8.

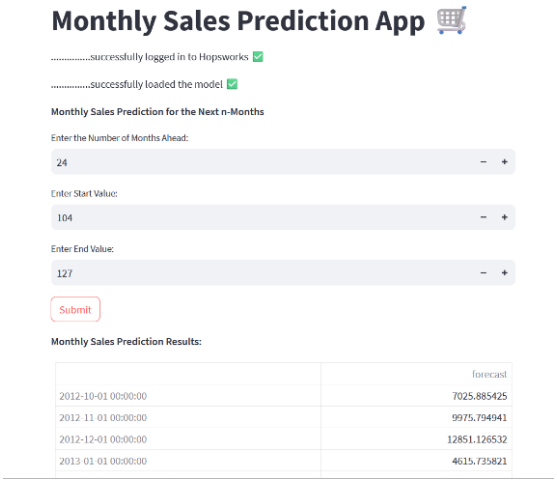


Figure 8. The Result of Deployment

3.6. Monitoring Performance

Regularly monitor the performance and reliability of the MLOps architecture. Check the logs and metrics generated by Hopsworks and the Hugging Face application. For maintenance in Hopsworks, you can follow the steps below:

- Model Performance Monitoring: Continuously monitor model performance metrics in the production environment to ensure that the model functions correctly. Metrics such as accuracy.
- Logging Critical Information: Ensure that system logs and model logs in Hopsworks are up-to-date and contain important information, such as training time, performance metrics, or relevant error messages. This will aid in analysis and troubleshooting if issues arise during training or modeling.

4. Conclusions

The study's focus on developing an MLOps architecture for sales quantity prediction in MSMEs demonstrates the potential for simplifying sales forecasting and enabling smarter decision-making. Identifying the advantages, constraints, and challenges of merging these two paradigms contributes to bridging the gap in the literature regarding MLOps in the context of time-series analysis. The methodology outlined a comprehensive process that involved data preprocessing, model training, and deployment using platforms like Hopsworks, Streamlit, and Hugging Face. The SARIMAX model, well-suited for time-series forecasting with external variables, played a central role. Finally, this research aims to empower MSMEs by enhancing their sales forecasting capabilities, enabling them to navigate the challenges of fluctuating sales volumes, and fostering their growth and sustainability in a competitive market landscape. The fusion of time-series analysis and MLOps offers a promising avenue for improving business outcomes and decision-making processes for MSMEs.

Acknowledgements

We sincerely express our gratitude to the Directorate General of Higher Education, Research, and Technology, Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia, for funding a portion of this project through the Kedaireka Program. This work has also received support from Dian Nuswantoro University (UDINUS) through the Center of Excellence in Science and Technology, UDINUS, and the Association of Food and Beverage Souvenir Entrepreneurs in Central Java. This is related to the grant contract document entitled "Implementation of Supply Chain Management System for MSMEs in the Food and Beverage Souvenir Sector, A Case Study in Central Java" with contract number 4501/F.9.02/UDN-01/IV/2023.

References

[1]. O. S. Taiwo, A. Hakan, and Ç. Savaş, "Modeling the Impacts of MSMEs' Contributions to GDP and their Constraints on Unemployment: The Case of African's Most Populous Country," *Stud. Bus. Econ.*, vol. 17, no. 1, pp. 154–170, Apr. 2022, doi: 10.2478/sbe-2022-0011.

[2]. Jhon Montalvo-Garcia, Juan Bernardo Quintero, and Bell Manrique-Losada, *Crisp-dm/smes: A data analytics methodology for non-profit smes*, vol. 1041. London: Springer, 2020.

[3]. I. R. Riana and L. Nafiati, "Pengaruh Persepsi Etika Bisnis Islam, Persepsi Kualitas Produk, dan Persepsi Kualitas Pelayanan terhadap Tingkat Penjualan UMKM di Kota Yogyakarta," *J. REKSA Rekayasa Keuang. Syariah Dan Audit*, vol. 8, no. 1, p. 59, Feb. 2021, doi: 10.12928/j.reksa.v8i1.3871.

[4]. F. Cuandra, "Analisis Tingkat Penjualan Melalui Faktor Internal Maupun Faktor Eksternal terhadap UMKM Kuliner Kota Batam," *J. Progres Ekon. Pembang. JPEP*, vol. 6, no. 2, p. 123, Aug. 2021, doi: 10.33772/jpep.v6i2.19794.

[5]. X. Qi, K. Hou, T. Liu, Z. Yu, S. Hu, and W. Ou, "From Known to Unknown: Knowledge-guided Transformer for Time-Series Sales Forecasting in Alibaba." arXiv, Sep. 22, 2021. Accessed: Sep. 26, 2023. [Online]. Available: http://arxiv.org/abs/2109.08381

[6]. H. Ge and L. Fang, "Prediction Model of Physical Goods Sales based on Time Series Analysis," *Front. Bus. Econ. Manag.*, vol. 5, no. 2, pp. 90–97, Sep. 2022, doi: 10.54097/fbem.v5i2.1670.

[7]. S. Wang and Y. Yang, "M-GAN-XGBOOST model for sales prediction and precision marketing strategy making of each product in online stores," *Data Technol. Appl.*, vol. 55, no. 5, pp. 749–770, Oct. 2021, doi: 10.1108/DTA-11-2020-0286.

[8]. N. Kumar, V. Jain, K. Joshi, and I. Dawar, "Prediction of epidemic disease cases using ARIMA and SARIMAX models," in *2023 Sixth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU)*, Riyadh, Saudi Arabia: IEEE, Mar. 2023, pp. 201–205. doi: 10.1109/WiDS-PSU57071.2023.00049.

[9]. J. Au, J. S. Jr, B. Spanswick, and J. Santerre, "Forecasting Power Consumption in Pennsylvania During the COVID-19 Pandemic: A SARIMAX Model with External COVID-19 and Unemployment Variables," vol. 3, no. 2, 2020.

[10]. W. Skaf, A. Tosayeva, and D. T. Várkonyi, "Towards Automatic Forecasting: Evaluation of Time-Series Forecasting Models for Chickenpox Cases Estimation in Hungary." arXiv, Oct. 04, 2022. Accessed: Oct. 02, 2023. [Online]. Available: http://arxiv.org/abs/2209.14129

[11]. S. Sahoo, "A Comprehensive Analysis and prognostication of COVID-19 (SARS-Cov-2) Outbreak situation in India," Open Science Framework, preprint, Jun. 2021. doi: 10.31219/osf.io/v9n7s.

[12]. D. Kreuzberger, N. Kühl, and S. Hirschl, "Machine Learning Operations (MLOps): Overview, Definition, and Architecture," *IEEE Access*, vol. 11, pp. 31866–31879, 2023, doi: 10.1109/ACCESS.2023.3262138.

[13]. T. Masood and P. Sonntag, "Industry 4.0: Adoption challenges and benefits for SMEs," *Comput. Ind.*, vol. 121, p. 103261, Oct. 2020, doi: 10.1016/j.compind.2020.103261.

[14]. B. M. A. Matsui and D. H. Goya, "MLOps: a guide to its adoption in the context of responsible AI," in *Proceedings of the 1st Workshop on Software Engineering for Responsible AI*, Pittsburgh Pennsylvania: ACM, May 2022, pp. 45–49. doi: 10.1145/3526073.3527591.

[15]. L. Faubel *et al.*, "Towards an MLOps Architecture for XAI in Industrial Applications," 2023, doi: 10.48550/ARXIV.2309.12756.

[16]. A. Singla, "Machine Learning Operations (MLOps): Challenges and Strategies," *J. Knowl. Learn. Sci. Technol. ISSN 2959-6386 Online*, vol. 2, no. 3, pp. 333–340, Aug. 2023, doi: 10.60087/jklst.vol2.n3.p340.

[17]. D. H. Hagos *et al.*, "Scalable Artificial Intelligence for Earth Observation Data Using Hopsworks," *Remote Sens.*, vol. 14, no. 8, p. 1889, Apr. 2022, doi: 10.3390/rs14081889.

[18]. A. A. Ormenis, "Horizontally Scalable ML Pipelines with a Feature Store," 2019.

[19]. A. M. Elshewey *et al.*, "A Novel WD-SARIMAX Model for Temperature Forecasting Using Daily Delhi Climate Dataset," *Sustainability*, vol. 15, no. 1, p. 757, Dec. 2022, doi: 10.3390/su15010757.

[20]. T. Andrianajaina, D. T. Razafimahefa, R. Rakotoarijaina, and C. G. Haba, "Grid Search for SARIMAX Parameters for Photovoltaic Time Series Modeling," *Glob. J. Energy Technol. Res. Updat.*, vol. 9, pp. 87–96, Dec. 2022, doi: 10.15377/2409-5818.2022.09.7.

[21]. M. P. Keerthi, G. S. Reddy, V. S. Raghava, and K. B. Reddy, "Streamlit Interface for Multiple Disease Diagnosis," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 2, pp. 1159–1164, Feb. 2023, doi: 10.22214/ijraset.2023.49166.

[22]. Dr. J. N. Padmaja, A. V. Kanth, P. V. Reddy, and B. A. Rao, "Web Application for Emotion-Based Music Player using Streamlit," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 2, pp. 342–347, Feb. 2023, doi: 10.22214/ijraset.2023.49019.

[23]. A. Ait, J. L. C. Izquierdo, and J. Cabot, "HFCommunity: A Tool to Analyze the Hugging Face Hub Community," in *2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Taipa, Macao: IEEE, Mar. 2023, pp. 728–732. doi: 10.1109/SANER56733.2023.00080.