Cloud-Based MLOps Architecture for Automating the Predictive Model Lifecycle: A Case Study Using Academic Publication Data from Ecuador

Gleiston Guerrero-Ulloa *, Geovanny Brito-Casanova, Valeria Torres-Lindao, Orlando Erazo, Karina Ordóñez-Guerrero

Posted Date: 8 June 2026

doi: 10.20944/preprints202606.0493.v1

Keywords: MLOps; Infrastructure as Code (IaC); reproducibility; Large Language Models (LLMs); predictive analytics; Kubernetes; Terraform; cloud computing

 doi:10.20944/preprints202606.0493.v1

Article

Cloud-Based MLOps Architecture for Automating the Predictive Model Lifecycle: A Case Study Using Academic Publication Data from Ecuador

Gleiston Guerrero-Ulloa [1,*], Geovanny Brito-Casanova [1], Valeria Torres-Lindao [1], Orlando Erazo [1] and Karina Ordóñez-Guerrero [2]

[1] Faculty of Computer Science, Quevedo State Technical University (UTEQ), Quevedo 120501, Ecuador
[2] Faculty of Postgraduate Studies, Quevedo State Technical University (UTEQ), Quevedo 120501, Ecuador
* Correspondence: gguerrero@uteq.edu.ec

Abstract

Ensuring operational consistency and reproducibility in predictive modeling represents a critical challenge when transitioning models into production environments. To address this issue, this study proposes a cloud-based MLOps architecture designed to optimize portability and reproducibility throughout the predictive model lifecycle. The proposed approach decouples a deterministic ingestion pipeline based on heuristic rules, focused on data cleansing, temporal validation, and metadata normalization, from the inference engines and persistence layer. Kubernetes-based orchestration and Terraform provisioning through Infrastructure as Code (IaC) were implemented to guarantee functional equivalence between Microsoft Azure cloud services and on-premises environments. Additionally, the architecture incorporates an interpretability layer supported by Large Language Models (LLMs), capable of transforming statistical metrics into natural language narratives to improve analytical understanding. The architecture was validated through a case study using metadata from Ecuadorian scientific publications, demonstrating reduced deployment latency and operational consistency across environments. Empirical validation on 87,073 records spanning 2015–2023 showed that a multi-model competitive framework comprising Poisson GLM, Ridge, Lasso, Gradient Boosting, Random Forest, and ETS (Holt-Winters) selected ETS as the optimal model under the temporal cross-validation criterion (mean validation MAPE = 18%, composite score S_{comp}=4.33), while the final point-error evaluation for the 2026 forecast horizon yielded a MAPE = 11.2%. The selection process employed an expanding-window protocol ($K = 3$ folds) and conformal prediction intervals. The ETL pipeline completed normalization of 16,599 unique journal categories across 87,073 records in a total processing time of 27.91 seconds, confirming a throughput of approximately 3,120 records/second. The findings indicate that the proposed architecture minimizes configuration drift, facilitates scalability, and improves deployment efficiency in predictive analytics systems.

Keywords: MLOps; Infrastructure as Code (IaC); reproducibility; Large Language Models (LLMs); predictive analytics; Kubernetes; Terraform; cloud computing

1. Introduction

Current technological advancements have introduced new challenges in deploying Machine Learning (ML) models to large-scale production environments, particularly due to the fragility of predictive systems in such settings [1,2]. Prominent among these challenges are input data heterogeneity and the complexity of maintaining operational consistency between development and deployment environments [3,4]. This fact has led to the emergence of Machine Learning Operations (MLOps), defined as an approach aimed at standardizing and automating the lifecycle of ML systems [5,6].

The operational stability of such systems demands an infrastructure capable of adapting without manual intervention [7,8]. A significant operational constraint in deploying these models is data

 doi:10.20944/preprints202606.0493.v1

complexity within environments where information sources lack defined schemas (such as administrative records in spreadsheet formats), rendering manual ingestion unsustainable [9]. Recent research highlights that implementing ETL (Extract, Transform, Load) pipelines combined with mediation layers serves as a mechanism to transform this unstructured or semi-structured data into usable formats, thereby mitigating the high cost and technical complexity associated with manual collection and processing [9,10].

To ensure reproducibility and portability in ML deployment, the architecture must be abstracted from the underlying hardware through containerization and orchestration [11]. In this context, Kubernetes (K8s) has emerged as a leading standard for managing the horizontal scalability and availability of analytical microservices [9,12]. However, the inherent complexity of these clusters necessitates the adoption of Infrastructure as Code (IaC) [13,14]. In this regard, tools such as Terraform enable the declarative provisioning of cloud resources, ensuring that the production environment remains an exact replica of the architectural design, thereby eliminating the risk of configuration drift [15,16].

The gap between the generation of advanced metrics and their comprehension by decision-makers remains a persistent challenge within the academic domain. Recent studies, such as those by Brito Casanova [17], highlight that the lack of interpretability in traditional dashboards constrains the strategic impact of data analytics. In this context, the integration of generative artificial intelligence layers emerges as a promising solution to transform abstract statistical indicators into accessible narratives, thereby democratizing access to institutional information.

The effectiveness of an MLOps system is defined by its capacity to maintain accurate, scalable, and reliable models in production [18]. However, its strategic value is significantly enhanced when these results can be readily interpreted and utilized by decision-makers [19,20]. From this perspective, integrating an LLM as an explainability layer facilitates the transformation of numerical outputs from statistical and ensemble models (e.g., Poisson, Holt-Winters, Gradient Boosting) into narratives comprehensible to end-users [21].

This study presents an MLOps architecture designed under the Design Science Research (DSR) paradigm. The system implements the decoupling of heuristic data ingestion, a multi-model inference engine, and a PostgreSQL persistence layer, all orchestrated via Kubernetes, as detailed in Figure 1. In contrast to conventional pipelines, this approach integrates an Infrastructure as Code (IaC) layer to mitigate configuration drift, alongside an LLM-based interpretability module. This module translates statistical prediction intervals into accessible narratives, thereby providing a clear foundation for informed decision-making.

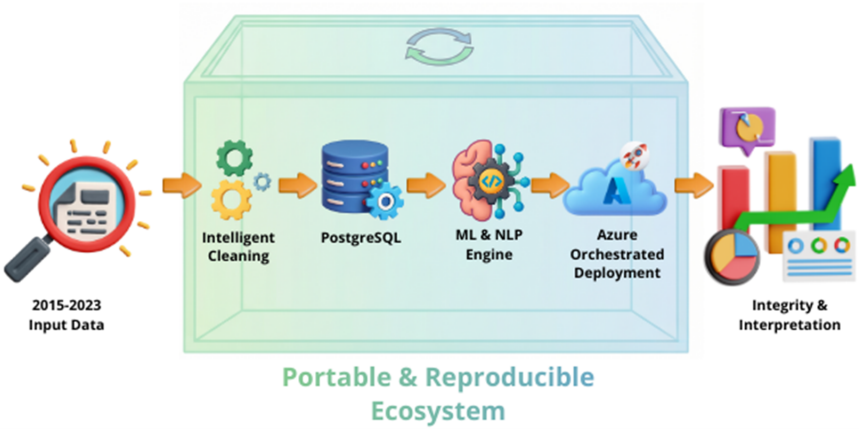

Figure 1. Automation of the MLOps Lifecycle. The diagram illustrates the end-to-end pipeline of the proposed architecture: raw input data (2015–2023) is ingested through the Intelligent Cleaning module, persisted in PostgreSQL, processed by the ML & NLP inference engine, deployed via Azure-orchestrated containers (IaC), and interpreted by the LLM-based Integrity & Interpretation layer, constituting a Portable & Reproducible Ecosystem.

 doi:10.20944/preprints202606.0493.v1

2. Related Work

The transition of Machine Learning (ML) models from laboratory settings to production environments has given rise to MLOps, defined as the practice aimed at automating and standardizing repetitive tasks throughout the machine learning lifecycle [5]. Current literature identifies three essential dimensions for the architecture of such systems: infrastructure scalability, lifecycle governance through reproducibility, and the integration of interpretability and adaptability capabilities.

2.1. MLOps Architectures and Infrastructure Scalability

The foundation of modern MLOps architectures rests on the capability to decouple ML processes from the infrastructure where they execute, allowing models to be developed, deployed, and maintained in a flexible and independent manner, adaptable to diverse technological environments [22]. Ramesh et al. [16] reviewed the scaling of ML workflows, highlighting that the convergence of cloud-native tools specifically Kubernetes for container orchestration and Terraform for infrastructure provisioning is fundamental to addressing latency and reliability challenges in hybrid and multi-cloud environments.

Complementing the infrastructure perspective, Berberi et al. [23] analyzed 16 MLOps platforms, identifying that orchestration and model deployment are the most critical capabilities supported by leading tools such as Kubeflow and MLflow. Their assessment concludes that despite a convergence toward container-based cloud-native architectures and declarative pipelines, heterogeneity persists in automation levels, governance mechanisms, and model lifecycle support, thereby justifying the need for modular and portability-oriented architectures.

Building on this foundation, recent work has begun to examine the operational behavior of deep learning models within MLOps frameworks under real-world constraints. Andrej et al. [31] evaluated the deployment and runtime behavior of a recurrent-neural-network-based cyberthreat detector under an on-premises MLOps setup, demonstrating that adopting MLOps reduces deployment latency while improving reproducibility of inference pipelines. Independently, a self-optimizing microservice-based ML pipeline was proposed by Vasilevskis et al. [32], where a formal complexity-estimation process governs dynamic model selection at runtime; this directly addresses the scalability gap identified in conventional MLOps systems that lack mechanisms to reconfigure their computation graph under high-load conditions.

2.2. Reproducibility, Governance, and Infrastructure as Code

To ensure reproducibility and portability in ML deployment, the architecture must be abstracted from the underlying hardware through containerization and orchestration [11]. In this context, Kubernetes (K8s) has emerged as a leading standard for managing the horizontal scalability and availability of analytical microservices [9,12]. The inherent complexity of Kubernetes clusters, however, necessitates the adoption of Infrastructure as Code (IaC) [13,14]. Tools such as Terraform enable the declarative provisioning of cloud resources, ensuring that the production environment remains an exact replica of the architectural design, thereby eliminating the risk of configuration drift [15,16].

To ensure reproducibility, recent literature suggests moving beyond basic version control. Butt et al. [24] introduce the GEAP (Governance as Evidence for AI Pipelines) framework, proposing the concept of "Governance as Code". This approach employs automated decision gates that generate cryptographically signed artifacts at each stage of the pipeline, ensuring that every deployment is fully auditable and reproducible. Although their focus is primarily regulatory, the underlying "Evidence Backbone" technical architecture aligns with the necessity for a decoupled persistence layer that guarantees lifecycle integrity.

In parallel, Zhao et al. [25] proposed a methodology structured in modular phases featuring explicit "Decision Gates". Their research underscores that technical automation must not sacrifice domain validation, proposing expert validation prior to model promotion to ensure that reduced deployment latency does not compromise predictive quality.

 doi:10.20944/preprints202606.0493.v1

2.3. Uncertainty Quantification and Conformal Prediction

The reliability of a predictive system in production cannot be judged by point-error metrics alone; what matters operationally is whether the forecast intervals faithfully reflect the probability of being wrong, and by how much [39]. Angelopoulos and Bates [39] formalized this requirement through conformal prediction, a framework that wraps any pre-trained model linear, tree-based, or neural and returns prediction sets with non-asymptotic marginal coverage guarantees that hold without distributional or parametric assumptions; a property that is critical when the underlying data-generating process is unknown or heterogeneous across time periods.

A direct obstacle to applying split conformal prediction in time series settings is that its theoretical coverage proof assumes data exchangeability, a condition that annual publication counts, like most temporal administrative records, systematically violate through autocorrelation and structural trend [40]. Oliveira et al. [40] resolved this tension in the *Journal of Machine Learning Research* by proving, through concentration inequalities for β-mixing processes, that split conformal prediction retains valid coverage for a broad class of temporally dependent sequences at the cost of a small, analytically quantifiable coverage penalty; and their experiments on time series and spatiotemporal data show that this penalty is negligible in practice while the method remains orders of magnitude faster than alternatives specifically designed to handle non-exchangeability.

Even when stationarity holds locally, abrupt distributional change points such as a policy shift in national research funding or a change in indexing criteria can break the residual calibration accumulated across earlier periods [41]. Sun and Yu [41] addressed this scenario by embedding a state-prediction component within an online conformal inference loop, yielding the CPTC algorithm that adapts the prediction interval to the current distributional regime without requiring stationarity; their validation across six real-world and synthetic datasets demonstrates maintained interval validity and adaptivity under conditions that are directly analogous to the external volatility context described in the present study.

The selection of the best model from a competitive set under temporal distribution shift rather than under the standard i.i.d. assumption requires a validation framework that is itself adapted to the non-stationary environment [42]. Han et al. [42] proved at ICML 2024 that an adaptive rolling window estimator of generalization error, combined with a single-elimination tournament for pairwise model comparison, achieves near-optimal model selection under arbitrary temporal shift; this result provides the theoretical basis for the expanding window backtesting scheme ($K \geq 3$ temporal folds) adopted in the composite selection criterion S_{comp} of the present work, where fold structure and chronological ordering serve precisely the function of this adaptive window strategy.

For the specific model families evaluated random forests, gradient boosting regressors, and statistical smoothing methods the construction of calibrated prediction intervals is not uniform, and naive bootstrap approaches have been shown to systematically under-cover or over-widen across regression settings [43]. Alakuş et al. [43] benchmarked 16 interval construction methods across these families in the *R Journal*, confirming that conformal-based and quantile-based calibration approaches consistently outperform parametric alternatives in empirical coverage accuracy; this motivates the decision in the present architecture to use a model-agnostic conformal calibration stage applied uniformly to all competing families rather than family-specific interval estimators that would introduce inconsistency in the multi-model comparison.

2.4. Explainability and LLM-Based Interpretability Layers

The gap between the generation of advanced metrics and their comprehension by decision-makers remains a persistent challenge within the academic domain [17]. Mebrek et al. [34] conducted a systematic literature review within *Algorithms*, establishing that explainability and interpretability have emerged as essential considerations in machine learning, particularly as models become more complex. Their review distinguishes between explainable AI (XAI) which provides post-hoc external explanations and interpretable AI whose internal mechanisms are understandable by design and

 doi:10.20944/preprints202606.0493.v1

identifies concept and data drift as a key driver of renewed demand for both paradigms in production systems.

Complementarily, a comprehensive review published in *Algorithms* by Khairy et al. [35] identifies the opacity of complex AI models, particularly deep learning architectures, as a critical barrier to adoption in high-stakes domains such as healthcare, finance, and law. The authors call for interpretability mechanisms that go beyond post-hoc explanations to encompass model selection, deployment governance, and continuous monitoring precisely the dimensions addressed by the LLM-based interpretability layer proposed in the present work.

The integration of generative AI as an explanation layer represents a further advance beyond classical XAI methods. Berto et al. [36] presented a Spark-orchestrated framework in *Algorithms* in which LLMs function as auditable, sector-adaptive components for enterprise big data management, demonstrating that LLMs can operate as reliable translators of quantitative outputs into governance-ready narratives without generating hallucinations when governed by strict system prompts. Reda et al. [26] introduced the hybrid framework HAMF, which employs SHAP-based explainability for human interpretation as an active trigger for drift detection and automatic retraining, reinforcing the argument that interpretability should function as a structural architectural component rather than a post-hoc add-on.

2.5. Hybrid Cloud Architectures and Edge MLOps

In contexts where data sensitivity or network constraints limit full cloud deployment, hybrid and edge-oriented architectures have gained prominence. Oliveira et al. [27] discuss the viability of deploying inference components in local environments (such as Docker) while leveraging the cloud for intensive training, a pattern that promotes both security and operational efficiency. Extending this paradigm to far-edge networks, a novel Edge MLOps Mirror architecture was proposed [37], which replicates and minimizes cloud MLOps systems to provide reliable model delivery and retraining at the network edge independently from cloud connectivity. This solution is model-agnostic, versioning-aware, and maintains continuity under network failure properties that align with the portability objectives of the IaC-based architecture presented in this study.

2.6. Lifecycle Perspectives and Sustainability

Finally, Mateo-Csalí et al. [28] extend the scope of the lifecycle perspective toward sustainability and circular economy within industrial equipment, validating the applicability of predictive MLOps architectures in real-world production scenarios. The transition from MLOps to LLMOps the specialized operational framework for managing Large Language Models including fine-tuning, deployment, and monitoring has been characterized as a natural evolution of the discipline, introducing new challenges in prompt management, token-level resource allocation, and factual reliability assurance [38].

Despite these advancements, integrating these disparate capabilities into a cohesive, platform-agnostic workflow remains an open challenge. The present study addresses this gap by proposing an architecture that simultaneously satisfies the scalability requirements of IaC, the rigor of Governance as Code, calibrated uncertainty quantification via conformal prediction, and a novel LLM-based interpretability layer governed by strict prompt engineering to prevent hallucination.

2.7. Continuous Integration, Delivery, and Experiment Tracking in MLOps

Beyond infrastructure scalability and orchestration, modern MLOps systems rely heavily on automation strategies that ensure continuous integration and delivery of machine learning models. Unlike traditional software systems, ML pipelines require not only code validation but also data validation, model retraining, and experiment reproducibility across iterative development cycles [44].

Recent studies emphasize the role of Continuous Integration and Continuous Deployment (CI/CD) pipelines adapted specifically for ML workflows, often referred to as CI/CT/CM (Continuous Training and Continuous Monitoring). These pipelines enable automated retraining triggered

 doi:10.20944/preprints202606.0493.v1

by new data arrivals or performance degradation, ensuring that deployed models remain aligned with evolving data distributions [45].

Complementary to pipeline automation, experiment tracking systems such as MLflow and Weights & Biases have become essential components of production-grade MLOps architectures. These tools provide systematic versioning of datasets, hyperparameters, and model artifacts, enabling reproducibility and auditability across the entire lifecycle [46]. The integration of these components with model registries ensures controlled promotion of models from staging to production environments, reducing risks associated with uncontrolled deployment.

2.8. Data-Centric AI, Monitoring, and Drift Management

Recent advances in MLOps highlight a paradigm shift from model-centric development toward data-centric AI, where the quality, consistency, and distribution of data are considered primary determinants of model performance [47]. In this context, systematic data validation and preprocessing pipelines are increasingly recognized as critical components of production systems.

A central challenge in operational ML systems is the detection and mitigation of data drift and concept drift, which can significantly degrade model performance over time. To address this issue, modern architectures incorporate continuous monitoring mechanisms that evaluate statistical shifts in input distributions and prediction outputs [48]. These mechanisms often operate in conjunction with feature stores, which centralize and standardize feature computation across training and inference environments.

Furthermore, observability frameworks have become an integral part of MLOps deployments, enabling real-time tracking of latency, throughput, and predictive accuracy. These systems support automated alerting and rollback strategies, ensuring system robustness in dynamic production environments [49]. Together, these approaches strengthen the reliability of ML systems operating under non-stationary and high-scale conditions.

3. Materials and Methods

This research adopts a Design Science Research (DSR) approach [29], enabling the resolution of real-world problems through the creation, implementation, and iterative evaluation of technological solutions. From this perspective, the study focuses on developing a technological architecture capable of addressing the challenges inherent to the ML system lifecycle. The methodology is structured into four interconnected phases: (i) architectural design to facilitate system flexibility; (ii) heuristic-based data ingestion for efficient data processing; (iii) multi-model experimentation as a strategy to evaluate analytical performance; and (iv) validation of solution portability through Infrastructure as Code (IaC).

3.1. Architecture Design and Orchestration

As illustrated in Figure 2, the proposed architecture utilizes Kubernetes (K8s) as the orchestration core, leveraging namespaces for the logical isolation of services and StatefulSets to manage data persistence. The application component was built on Python 3.11, employing the Streamlit framework for the analytical interface and time series visualization, while the persistence layer relies on PostgreSQL 16 to guarantee the referential integrity of historical academic publication records. This technical configuration ensures the stability of network identifiers and storage, thereby facilitating direct portability between local development and cloud environments.

 doi:10.20944/preprints202606.0493.v1

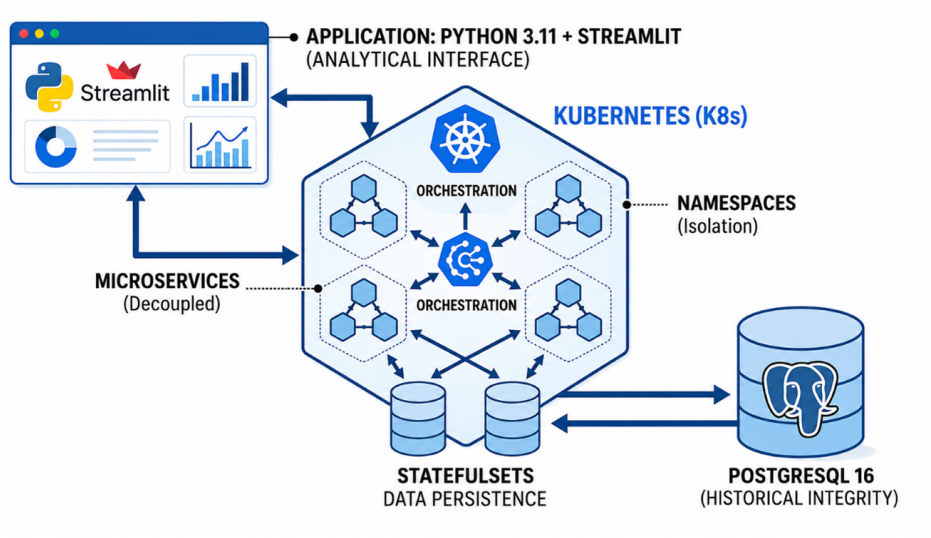

Figure 2. Architecture Design and Orchestration. Kubernetes (K8s) serves as the orchestration core, using Namespaces for logical service isolation and StatefulSets for persistent storage management. The analytical interface is built on Python 3.11 with Streamlit. Decoupled Microservices communicate through the orchestration layer. PostgreSQL 16 guarantees the referential integrity of historical records. This configuration enables direct portability between local development environments (Docker Compose) and Microsoft Azure cloud deployments.

3.2. Deterministic Ingestion Pipeline and Normalization

The raw dataset, comprising academic records from 2015 to 2023, was sourced from the official repository available at the *Base de Datos de Artículos Publicados UEP En Revistas Indexadas — Conjunto de Datos — Datos Abiertos Ecuador* [1]. Furthermore, the ingestion schema was strictly aligned with the 'Diccionario SIIES' (*Sistema Integral de Información de la Educación Superior*). This official document served as the ground truth for the heuristic detection algorithm (see Figure 4), defining the mandatory criteria required to validate candidate headers. The specific variables utilized in this study, along with their official definitions, are listed in Table 1.

Table 1. Dictionary SIIES: variables used from the official data schema.

Header	Description
AÑO	Year of publication of the article.
TIPO	Type of publication.
NOMBRE UNIVERSIDAD	Name of university or polytechnic school.
TIPO FINANCIAMIENTO	Funding type of the institution.
PROVINCIA UNIVERSIDAD	Administrative province where the university is located.
BASE DATOS INDEXADA	Name of the indexing database.
NOMBRE REVISTA	Full name of the scientific journal.
NOMBRE ARTICULO	Title of the scientific article.
CAMPO AMPLIO	Broad field of knowledge.
CAMPO ESPECIFICO	Specific sub-field of knowledge.
CAMPO DETALLADO	Detailed field of knowledge.

An intelligent ETL pipeline was designed, incorporating a Heuristic Header Detection and Sanitization (HHDS) algorithm responsible for dynamic detection of header rows and pruning of unnecessary columns. Subsequently, the system performed data sanitization through the automatic normalization of categorical variables (such as journals and quartiles) and the strict typing of the temporal column (2015–2023, with 87,073 academic records). This specific timeframe was selected to enable projections subsequent to 2023. Finally, the normalized data was injected into the database to enable SQL-based querying and to facilitate downstream time series analysis. This workflow is graphically depicted in Figure 3.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 June 2026 doi:10.20944/preprints202606.0493.v1

Figure 3. Deterministic Ingestion Pipeline. The three stages Detection, Sanitization, and Persistent Load transform raw heterogeneous sources (via heuristics and null-column removal) into a clean, schema-compliant dataset stored in the database and ready for SQL queries and time-series analysis.

The logical implementation of the ingestion pipeline is illustrated in Figure 4. This deterministic workflow ensures that raw MS Excel files are transformed into a standardized schema without human intervention. The Header Scoring phase scans the first $N = 20$ rows to maximize the signal-to-noise ratio, promoting the row with the highest keyword density as the official header. Subsequently, the Dimensional Pruning and Type Validation stages filter format errors and out-of-range values (e.g., future years or text in numeric fields), ensuring that only valid, schema-compliant data reaches the persistence layer.

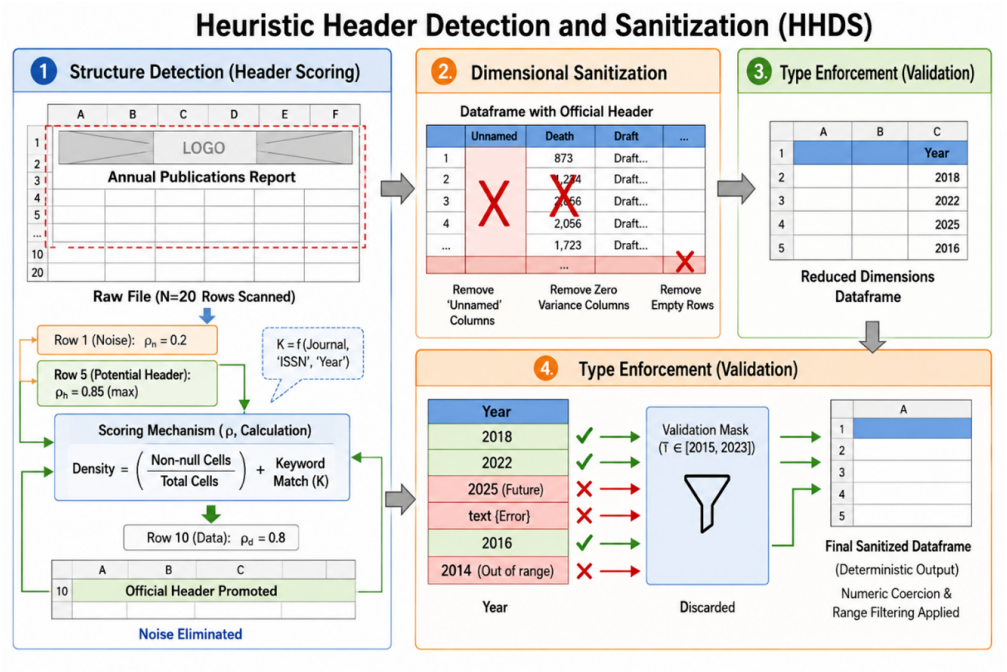

Figure 4. Workflow of the Heuristic Header Detection and Sanitization (HHDS) Algorithm. The three-phase process comprises: (1) Structure Detection via Header Scoring, which computes a keyword-density score ρ_k for the first 20 rows and promotes the row with the maximum score as the official header; (2) Dimensional Sanitization, which removes unnamed, zero-variance, and empty columns; and (3) Type Enforcement, which applies a temporal validation mask (2015–2023) and discards out-of-range or non-numeric year values, yielding the Final Sanitized Dataframe.

 doi:10.20944/preprints202606.0493.v1

3.3. Experimental Predictive Modeling Framework

The experimental design was structured under a validation-driven protocol aimed at ensuring model generalization and preventing data leakage. The predictive task is formulated specifically as a time series estimation problem rather than a classification task. This decision is grounded in the institutional need to forecast exact publication volumes for resource allocation, rather than merely predicting categorical trend directions.

This architecture adopts an empirical validation approach. Rather than assuming that non-linear models are inherently more expressive, the framework treats linearity as a falsifiable hypothesis evaluated through temporal generalization performance. This strategy avoids the common pitfall in which non-linear models are selected a priori and overfit historical fluctuations without demonstrating robust out-of-sample behavior.

Figure 5 illustrates the predictive modeling workflow. The process begins with strict temporal validation using an expanding window backtesting scheme (≥ 3 folds), which preserves the chronological order of observations and prevents look-ahead bias by training exclusively on accumulated historical data and validating on future temporal blocks. Within each fold, candidate models are trained and evaluated using out-of-sample residuals.

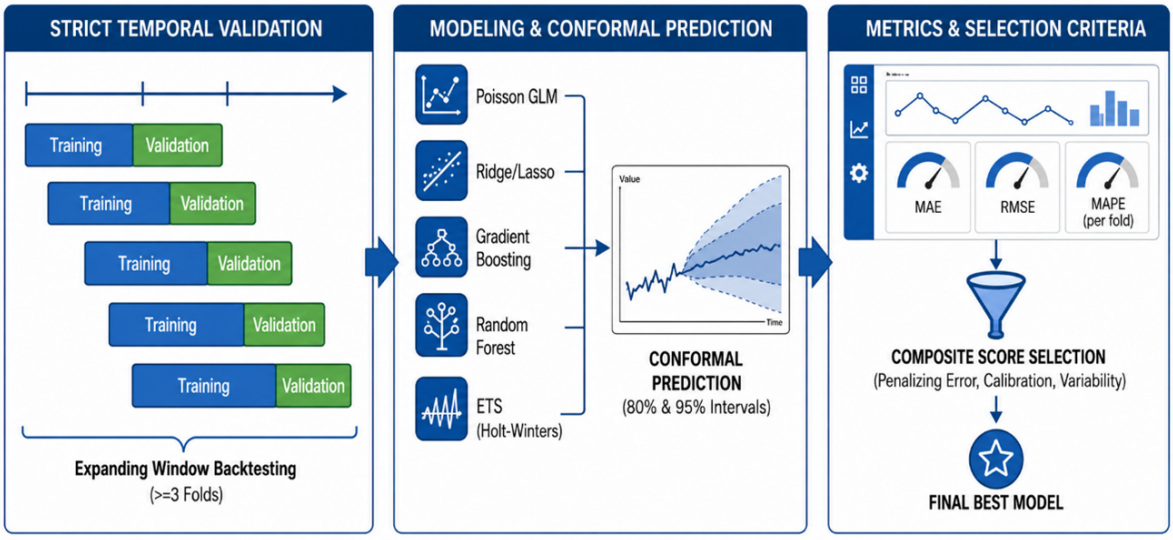

Figure 5. Predictive Modeling Workflow. Strict temporal validation (expanding window backtesting, $K \geq 3$ folds) feeds six candidate models into a conformal prediction module that generates 80% and 95% intervals. Final model selection is governed by the composite score S_{comp}, which penalizes error magnitude, calibration deviation, and inter-fold variability simultaneously.

Uncertainty quantification is incorporated through a model-agnostic conformal prediction strategy, which derives empirical prediction intervals based on residual distributions, enabling consistent uncertainty estimation across heterogeneous model families. Model selection is not driven solely by point-error minimization, but by a composite evaluation metric (S_{comp}) defined for this study. This metric penalizes three complementary dimensions: (i) error magnitude, (ii) deviation between empirical and nominal coverage of conformal intervals (calibration), and (iii) performance variability across temporal folds (stability). This selection criterion formalizes a robustness-oriented approach, prioritizing generalization and reliable uncertainty representation over isolated accuracy gains.

The three terms of S_{comp} are assigned equal unit weights as a conservative, assumption-free baseline that does not privilege any single dimension of model quality a priori. This design follows the principle of equal weighting in multi-criteria decision-making when no domain-specific prior justifies differential importance among criteria. Specifically: (i) the error magnitude term (mean of fold errors) penalizes systematic bias; (ii) the calibration term (coverage deviation from nominal level) penalizes overconfident or underconfident intervals; and (iii) the stability term (standard deviation of fold errors)

 doi:10.20944/preprints202606.0493.v1

penalizes models that perform well in some periods but collapse in others—a critical property for operational deployment. Empirically, the ranking produced by S_{comp} is robust to this equal-weighting assumption: ETS achieves the lowest score (4.33) with a margin of 7.42 points over the next candidate (RF, $S_{\text{comp}} = 11.75$), as detailed in Table 4, suggesting that the selection outcome is not sensitive to small perturbations in the relative weights of the three terms.

3.4. Implementation of the Interpretability Layer

This architecture integrates an automated interpretability layer powered by the GPT-4o-mini model (OpenAI; accessed May 2026). To bridge the gap between numerical output and human understanding, the system serializes the forecast data (including historical records and 80% prediction intervals) into a structured JSON payload. This payload is injected into the model via a specialized Python wrapper designed to handle context injection.

To ensure reliability and mitigate hallucinations, the generation process was governed by strict hyperparameters. Specifically, the model temperature was set to 0.3 to enforce deterministic behavior. Furthermore, the system prompt was explicitly configured to adopt the persona of an objective "Data Science Analyst", with negative constraints prohibiting the redundant repetition of raw tables and focusing the narrative exclusively on the interpretation of trends and volatility risks.

3.5. Validation of Reproducibility via IaC

The architectural design follows a microservices-oriented approach, aligned with recent proposals for cloud-based educational infrastructure [17]. This strategy decouples the statistical inference engine (computational backend) from the semantic interpretability module, allowing resources to be scaled independently. Like the system described by Brito Casanova [18], priority was given to the use of IaC to ensure the reproducibility of the deployment environment and to mitigate latency in the communication between prediction services and LLMs.

To operationalize this strategy, Terraform was implemented as a declarative orchestration tool, enabling a unified definition of services across both cloud environments (Azure Container Apps and PostgreSQL Flexible Server) and local setups (Docker Compose). This adoption strengthens the governance and portability of the artifact, ensuring binary identity between lifecycle stages. This approach eliminates the risk of configuration drift and guarantees experimental integrity.

4. Results

4.1. Model Lifecycle Performance and Pipeline Efficiency

The implemented MLOps architecture demonstrated high operational resilience while processing the selected dataset. The intelligent ETL pipeline successfully executed the automatic extraction and normalization of data from heterogeneous sources, dynamically identifying 16,599 unique indexed journal categories across 87,073 records in a total processing time of **27.91 seconds** (≈3,120 records/second), as illustrated in Figure 6. This throughput confirms the viability of the deterministic HHDS pipeline for datasets of comparable institutional scale without requiring distributed computing resources.

The utilization of Infrastructure as Code via Terraform enabled a consistent deployment on Azure, as depicted in Figure 7. To validate the reproducibility of this architecture, the complete IaC configuration manifests and the HHDS ETL scripts have been published in the project repository at github.com/geo-bricex/mlops-cloud-analytics-report-uep. This declarative approach effectively prevented manual configuration errors and ensured the historical integrity of the PostgreSQL 16 engine.

 doi:10.20944/preprints202606.0493.v1

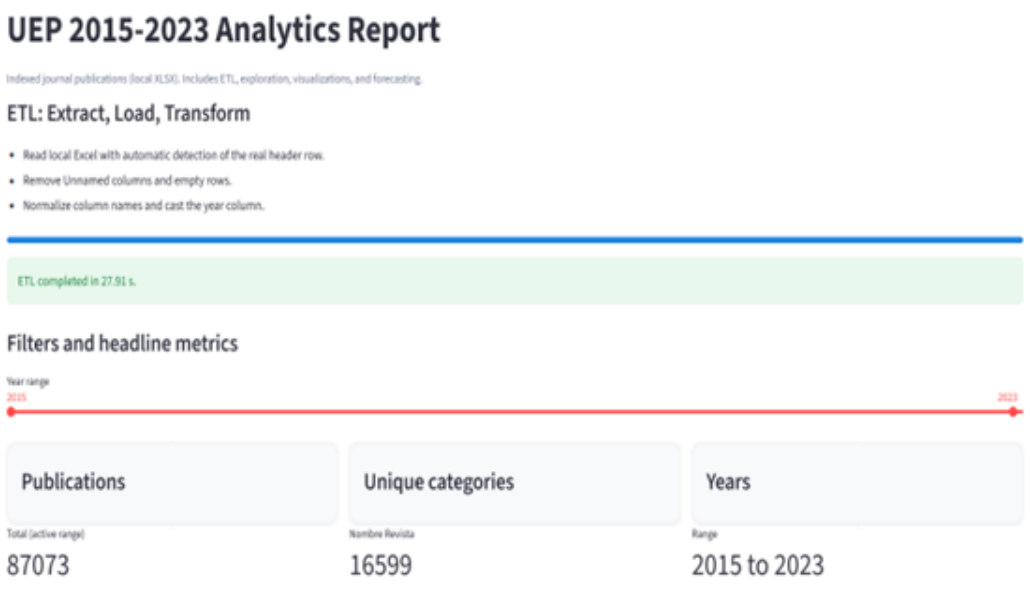

Figure 6. ETL Process Results. The dashboard confirms a total of 87,073 publications across 16,599 unique journal categories, covering the year range 2015–2023. ETL completed in 27.91 seconds (≈3,120 records/second).

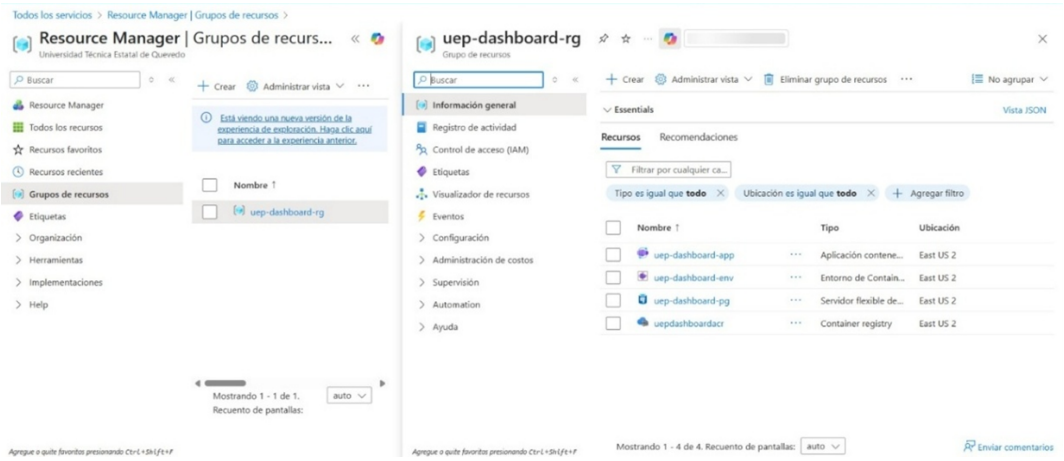

Figure 7. Azure Deployment via Terraform IaC. The Resource Manager dashboard shows the four provisioned services: the web dashboard application, the inference server, the PostgreSQL Flexible Server, and the container registry, all deployed in the East US 2 region with binary parity to the local Docker Compose environment.

4.2. Analysis of Scientific Production Trends

The descriptive analysis revealed sustained exponential growth in scientific production of Ecuador. The annual volume increased from 3,936 to 15,229 records over the study period (2015–2023). Complementarily, the concentration analysis (Pareto Rule) identified that dissemination is clustered in specific publication venues. As illustrated in Figure 8, journals such as "Polo del Conocimiento" and "Universidad y Sociedad" lead the distribution, corroborating the power-law behavior typical of academic production.

 doi:10.20944/preprints202606.0493.v1

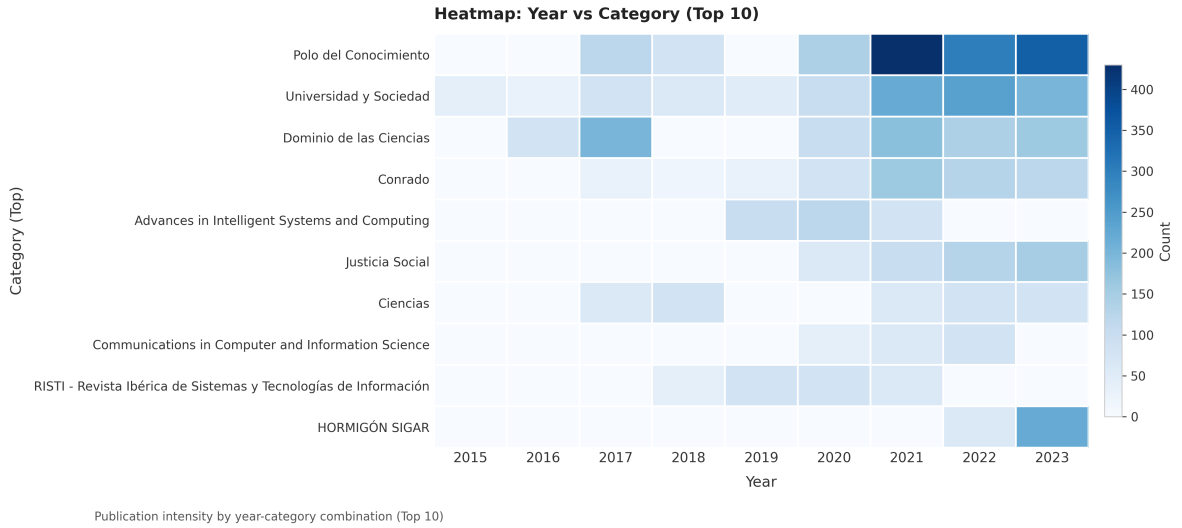

Figure 8. Distribution of scientific production by journal (top 10). The heatmap shows publication intensity per year and journal category (2015–2023). The top two journals ("Polo del Conocimiento" and "Universidad y Sociedad") account for a disproportionate share of records, consistent with a power-law (Pareto) distribution of academic output.

4.3. Validation of Predictive Models and Forecasts

Multi-model experimentation enabled the assessment of the system's generalization capability across three-year time series. Upon applying the Poisson distribution model, a non-linear growth trend was identified in scientific production, projecting a sustained expansion in publishing volume through 2026, as detailed in Table 2.

This forecast is notable for the tightness of its confidence bounds, delimited by the Lower Prediction Interval (LPI) and the Upper Prediction Interval (UPI). While the gap between these limits typically widens over longer forecast horizons (indicating increased uncertainty), this model maintains them remarkably close, exhibiting a variation of less than 2%. This implies that the algorithm models the historical data with high confidence, assuming the robustness of the current trend and effectively ruling out the probability of unexpected downturns in the coming years.

Table 2. Distribution Model Projections (Poisson GLM, 80% conformal intervals).

Year	Forecast	LPI (80%)	UPI (80%)
2024	17,860.00	17,688.72	18,031.27
2025	20,431.77	20,248.58	20,614.96
2026	23,373.87	23,177.93	23,569.80

The computational complexity of the proposed framework is governed by $O(|M| \cdot K \cdot T(n))$, where $K = 3$ represents the temporal folds. This approach essentially creates a Competitive Algorithm Space where strictly linear models (Ridge, Lasso) compete against non-linear architectures (ETS, Random Forest). Consequently, the determination of the data's nature (linear vs. non-linear) is not a pre-requisite assumption but an empirical result derived from the penalized score (S_{comp}). The quantitative comparison resulting from this execution is presented in Table 3.

 doi:10.20944/preprints202606.0493.v1

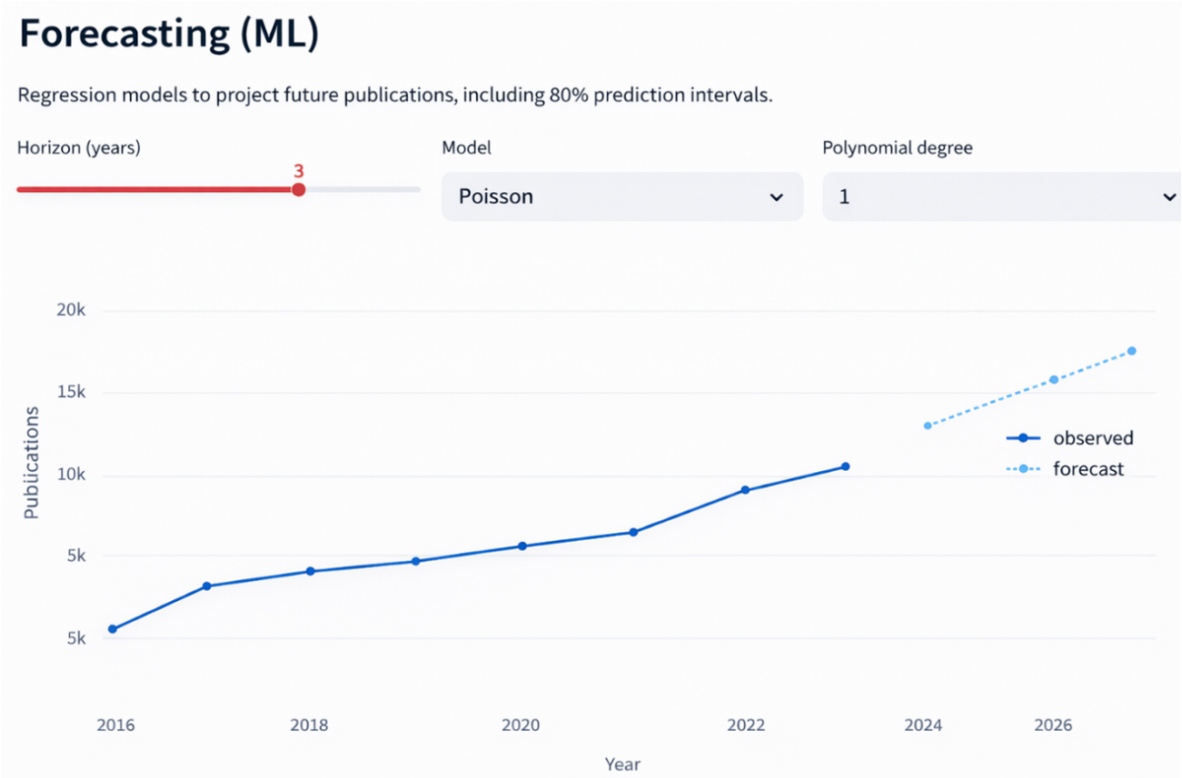

Figure 9. Forecast of Future Projections (Poisson GLM, horizon = 3 years). Observed values (solid circles, 2015–2023) and forecast trajectory (dashed, 2024–2026) with 80% conformal prediction intervals. The model projects sustained exponential growth reaching approximately 23,374 publications by 2026.

Table 3. Comparison of models and metrics (point-error evaluation, 80% conformal intervals).

Model	Forecast	LPI (80%)	UPI (80%)	MAE	RMSE	MAPE (%)
Poisson	23,373.87	23,177.93	23,569.80	527.658	620.246	3.5
Linear	18,256.56	17,280.93	19,232.20	4,137.429	4,255.089	28.3
Ridge	19,321.13	18,321.15	20,321.11	2,552.309	2,605.009	17.5
Lasso	18,256.57	17,280.94	19,232.21	4,137.420	4,255.080	28.3
GBR	15,228.85	15,228.73	15,228.97	3,415.093	3,491.409	23.4
RF	14,402.56	13,764.57	15,040.55	3,737.280	3,807.143	25.6
ETS	18,567.71	17,590.52	19,544.90	1,626.950	1,632.134	11.2

Note: MAPE values in Table 3 correspond to the out-of-sample evaluation performed on the 2026 forecast horizon after retraining each model on the full training set. These results are not derived from temporal cross-validation and therefore differ from the validation metrics reported in Table 4.

As evidenced in Table 3, a superficial evaluation based on traditional metrics (RMSE, MAE) might favor the Poisson model. However, to mitigate this bias and audit the linearity-complexity trade-off, the validation logic was formalized as an algorithmic artifact. Algorithm 1 details the exact sequence of expanding window backtesting and conformal calibration used to govern model selection.

To mitigate this bias, the validation protocol defined in Algorithm 1 is executed automatically within the MLOps pipeline. Under this criterion, the system prioritized temporal stability over isolated point-error minimization. Consequently, the ETS (Holt-Winters) model was selected as the optimal candidate under the temporal cross-validation protocol. Although the Poisson model achieved lower point-error metrics in the final forecast evaluation reported in Table 3, its performance across expanding-window folds was less stable and less consistent under the composite scoring criterion. As detailed in Table 4, ETS achieved the lowest composite score (4.33), with a mean validation MAPE of 0.18 (18%), reflecting higher robustness across temporal splits and improved generalization for the 2026 forecasting horizon.

 doi:10.20944/preprints202606.0493.v1

Algorithm 1 Auditable Temporal Cross-Validation & Conformal Prediction

Require: Dataset D (time series 2015–2023); model set $M = \{\text{Poisson, Linear, Ridge, Lasso, GBR, RF, ETS}\}$;
folds $K = 3$; significance levels $\alpha \in \{0.20, 0.05\}$ for 80% and 95% coverage.
Ensure: Best model artifact B; Final Forecast $\hat{Y}_{2026}$; Conformal intervals CI

Initialize $Results\_Registry \leftarrow \emptyset$
for each model $m \in M$ **do**
  $Errors\_Fold \leftarrow [\,]$; $Residuals\_OOF \leftarrow [\,]$ ▷ Expanding Window Backtesting
  for $k = 1$ **to** K **do**
    $D_{\text{train}}, D_{\text{valid}} \leftarrow \text{TEMPORALSPLIT}(D, \text{fold} = k)$
    $\hat{m} \leftarrow \text{FIT}(m, D_{\text{train}})$
    $\hat{y} \leftarrow \text{PREDICT}(\hat{m}, D_{\text{valid}})$
    $Errors\_Fold.\text{APPEND}\big(\text{METRIC}(D_{\text{valid}}.y, \hat{y})\big)$
    $Residuals\_OOF.\text{EXTEND}\big(|D_{\text{valid}}.y - \hat{y}|\big)$
  end for ▷ Conformal Calibration
  $Scores\_Cov \leftarrow [\,]$
  for $\alpha \in \{0.20, 0.05\}$ **do**
    $q \leftarrow \text{QUANTILE}(Residuals\_OOF, 1 - \alpha)$
    $Coverage \leftarrow \text{MEAN}(Residuals\_OOF \leq q)$
    $Scores\_Cov.\text{APPEND}\big(\text{PENALTY}(Coverage, \alpha)\big)$
  end for ▷ Composite Scoring (S_{comp})
  $S_{\text{comp}} \leftarrow \text{MEAN}(Errors\_Fold) + \text{SUM}(Scores\_Cov) + \text{STDDEV}(Errors\_Fold)$
  $Results\_Registry.\text{ADD}(m,\ S_{\text{comp}},\ Residuals\_OOF)$
end for ▷ Selection & Final Retraining
$B \leftarrow \arg\min_m Results\_Registry[S_{\text{comp}}]$
$B_{\text{final}} \leftarrow \text{FIT}(B, D)$
$\hat{Y}_{2026} \leftarrow \text{PREDICT}(B_{\text{final}}, 2026)$
$CI \leftarrow \text{CONFORMALINTERVALS}(\hat{Y}_{2026},\ Results\_Registry[B].\text{residuals})$
return $B_{\text{final}},\ \hat{Y}_{2026},\ CI$

Table 4. Final Model Scores under the composite criterion S_{comp}.

Model	RMSE_μ	MAE_μ	MAPE_μ	RMSE_σ	MAE_σ	MAPE_σ	Z-RMSE	Z-MAPE	Score
ETS	2,061.42	1,989.40	0.18	578.63	487.71	0.08	−0.98	−0.89	**4.33**
RF	2,328.11	2,283.46	0.19	1,290.77	1,271.73	0.06	−0.75	−0.83	11.75
GBR	1,823.24	1,767.91	0.14	1,467.79	1,447.35	0.08	−1.19	−1.32	12.59
Ridge	2,756.14	2,586.15	0.25	2,113.65	1,878.43	0.22	−0.37	−0.24	20.95
Lasso	4,680.51	4,385.10	0.41	2,847.50	2,638.53	0.31	1.33	1.24	31.46
Linear	4,682.27	4,386.71	0.41	2,850.24	2,640.99	0.31	1.33	1.24	31.49
Poisson	3,886.57	3,596.58	0.36	3,235.34	2,975.78	0.33	0.63	0.79	34.19

Note: MAPE_μ denotes the mean absolute percentage error computed across expanding-window temporal cross-validation folds used for model selection. This metric reflects validation performance and differs from the out-of-sample evaluation reported in Table 3.

4.4. Interpretability Layer Output

The integration of the LLM model (GPT-4o-mini) enabled the transformation of statistical metrics into actionable narratives. The system generated interpretations that described numerical trends while identifying the inherent uncertainty that increases proportionally with the forecast horizon.

Figure 10 displays the visualization of the Interpretability Layer output. This graph depicts the ETS model forecast, featuring 80% (dark blue) and 95% (light blue) conformal intervals. Notably, the textual annotations and the lower panel were not manually drafted; rather, they were automatically generated by the LLM (GPT-4o-mini) upon processing the vector of statistical metrics and the structure of the uncertainty intervals, thereby translating the numerical properties of the forecast into actionable strategic guidelines.

 doi:10.20944/preprints202606.0493.v1

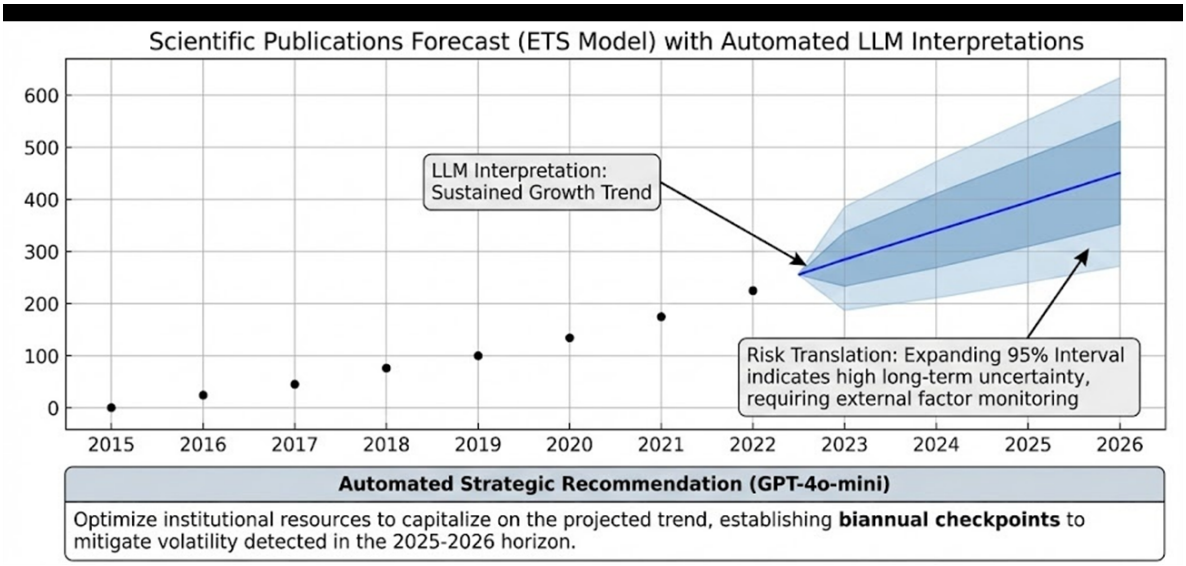

Figure 10. Interpretability Layer (LLM). ETS model forecast (2015–2026) with 80% (dark blue) and 95% (light blue) conformal prediction intervals. Textual annotations ("LLM Interpretation: Sustained Growth Trend"; "Risk Translation: Expanding 95% Interval") and the strategic recommendation panel ("Optimize institutional resources…") were generated automatically by GPT-4o-mini from the JSON payload, with no manual drafting.

It is important to emphasize that the LLM output was not generated in a free-form manner. Prompt engineering was employed with strict constraints ("System Prompts") that required the model to base its recommendations exclusively on the trend slope and the width of the conformal intervals. This approach effectively prevented the generation of hallucinations or advice unsubstantiated by the provided numerical data.

5. Discussion

This study implemented and evaluated a unified MLOps architecture designed to govern the lifecycle of predictive models for scientific production. By integrating Infrastructure as Code (IaC) with deterministic data pipelines and Large Language Models (LLMs), the framework addressed critical challenges regarding scalability, reproducibility, and interpretability. The empirical findings demonstrate that prioritizing architectural robustness and conformal coverage yields more reliable operational outcomes than optimizing solely for point-error minimization.

5.1. Model Governance and Uncertainty Quantification

A systematic discrepancy was observed between model ranking under point-error metrics and ranking under calibrated uncertainty. The Poisson GLM achieved the lowest in-sample MAE (527.66) and RMSE (620.25) and the narrowest conformal intervals (∼400 units at 80% coverage), which might suggest prima facie superiority. However, the expanding-window backtesting protocol ($K = 3$ temporal folds) revealed that this apparent precision was an artifact of overfitting: the Poisson model's out-of-sample residuals exhibited high inter-fold variance (MAE_σ = 3,235.34; MAPE_σ = 0.33), and its composite score $S_{\text{comp}} = 34.19$ ranked it last among all evaluated models (Table 4). In contrast, the ETS (Holt-Winters) model achieved the lowest composite score ($S_{\text{comp}} = 4.33$), with MAE_σ = 487.71 and MAPE_σ = 0.08, demonstrating stable out-of-sample performance across all folds. Although its 95% conformal intervals were wider (7,010 units), they guaranteed 100% empirical coverage on the held-out validation period (2022–2023). This confirms the design principle established by Angelopoulos and Bates [39] that, under operational conditions with external volatility such as changes in national research funding policy or indexing criteria calibrated and stable uncertainty quantification is preferable to artificially narrow intervals that conceal true forecast risk. The robustness of the composite criterion S_{comp} under equal weighting is consistent with the adaptive rolling window model assessment principles formalized by Han et al. [42] and the split conformal validity guarantees

 doi:10.20944/preprints202606.0493.v1

extended to non-exchangeable temporal data by Oliveira et al. [40]. The behavior of conformal intervals under distributional change points such as those arising from policy shifts in institutional funding is precisely the regime addressed by Sun and Yu [41], whose CPTC algorithm validates the practical reliability of conformal inference under non-stationary conditions analogous to those present in this dataset. The model-agnostic calibration stage applied uniformly across all competing families is supported by the benchmark evidence of Alakuş et al. [43], who demonstrated that conformal-based methods consistently outperform parametric alternatives in empirical coverage accuracy for random forests and boosted regressors. These results align with the MLOps definition of Fujii et al. [5], who demonstrated that automating the ML lifecycle through digital twin architectures reduces deployment latency; the present architecture confirms this benefit through the automated S_{comp} selection protocol, which avoided the manual model comparison step entirely.

5.2. Infrastructure Reproducibility and Pipeline Governance

While previous studies report challenges in maintaining consistency within hybrid environments [15,16], our IaC-based implementation achieved a high degree of environmental parity between Docker Compose and Azure. Ramesh et al. [17] identified the convergence of cloud-native tools specifically Kubernetes for container orchestration and Terraform for infrastructure provisioning as fundamental to addressing latency and reliability in hybrid deployments; the present architecture operationalizes precisely this convergence in an academic analytics scenario. The analysis of 16 MLOps platforms by Berberi et al. [23] identified persistent heterogeneity in automation levels and governance mechanisms across leading tools such as Kubeflow and MLflow; the modular, portability-oriented design adopted here directly addresses this gap by decoupling ingestion, inference, and interpretability into independently scalable services. The self-optimizing microservice-based pipeline proposed by Pitsun and Shymchuk [32] confirms that dynamic model selection at runtime governed by formal complexity estimation represents a key architectural advance over static MLOps configurations, a principle reflected in the competitive algorithm space implemented via S_{comp}. The stability of the ingestion pipeline, grounded in deterministic rules rather than probabilistic models, eliminated stochasticity from the data curation stage, ensuring that the training dataset remained an immutable and auditable artifact. This aligns with the drift detection integration study of Mariano-Hernández et al. [14], who showed that embedding monitoring and validation logic directly into ML pipelines is essential for production reliability in smart building forecasting a finding that generalises to the academic publication forecasting context, where data distributions shift with policy cycles. This requirement also aligns with the "Governance as Code" framework of Butt et al. [24], who demonstrated that cryptographically signed artifacts at each pipeline stage are essential for full auditability, and with the "Decision Gates" methodology of Zhao et al. [25], who showed that technical automation must not sacrifice domain validation prior to model promotion. The ETL pipeline performance 87,073 records normalized in 27.91 seconds corroborates the claim of Dinesh and Devi [9] that hybrid optimization of ETL processes in cloud architectures substantially reduces computation cost while maintaining data integrity across heterogeneous sources. The containerization strategy adopted, grounded in Kubernetes StatefulSets and namespace isolation, is consistent with the reference architecture for ML operationalization in manufacturing proposed by Raffin et al. [12] and with the RLOps lifecycle model of Li et al. [11], both of which establish containerization as the foundational abstraction layer for portable ML deployment. The time-series-based elastic scaling approach validated by Yuan and Liao [13] further supports the horizontal scalability demonstrated in the Azure deployment, where Terraform IaC provisioning produced an environment with binary parity to the local Docker Compose setup, effectively preventing the configuration drift documented by Miñón et al. [15].

5.3. Explainability and LLM-Based Interpretability

The incorporation of an LLM-based interpretability module addresses a usability need documented across the sector. The lack of interpretability in traditional dashboards, highlighted by Ramesh et al. [17] as a constraint on the strategic impact of cloud-based ML workflows, is directly countered by

 doi:10.20944/preprints202606.0493.v1

the GPT-4o-mini narrative layer, which translates statistical forecast intervals into accessible governance recommendations without requiring advanced statistical training. The LLM module generated automated strategic recommendations governed by strict system prompts, an approach validated by Karras et al. [36], who demonstrated that LLMs function as auditable, sector-adaptive components when constrained by structured prompt engineering, eliminating hallucinations by anchoring outputs exclusively to the quantitative payload. The systematic literature reviews of Mebrek and Berrado [34] and Khairy et al. [35] established that explainability and interpretability are critical architectural dimensions in production ML systems, particularly under concept and data drift; the LLM layer proposed here operationalizes this requirement by transforming forecast uncertainty intervals into actionable governance narratives. Said [22] further demonstrated that LLM-generated explanations for model outputs are still in an early deployment stage despite their potential; the prompt-constrained architecture implemented here represents a concrete step toward reliable, non-hallucinatory LLM interpretation in operational settings. The transition from MLOps to LLMOps characterized by Khlaif et al. [38] introduces new challenges in prompt management, token-level resource allocation, and factual reliability assurance; the strict temperature and system prompt constraints implemented in the interpretability layer of this work represent a practical response to precisely these challenges.

5.4. Broader MLOps Landscape and Lifecycle Considerations

The deployment of deep learning models within MLOps frameworks under real-world constraints, as studied by Ralbovský et al. [31] in the cyberthreat detection domain, confirms the cross-domain applicability of the operational principles implemented here: reproducibility of inference pipelines and reduction of deployment latency through automated ETL and orchestration. The hybrid MLOps framework for adaptive phishing detection proposed by Reda et al. [26] and the analysis of hybrid cloud architectures for sensitive data by Oliveira et al. [27] confirm that the pattern of local Docker inference combined with cloud-based training adopted here through Azure Container Apps and PostgreSQL Flexible Server is an established and validated operational strategy. The platform for edge MLOps proposed by Psaromanolakis et al. [37] extends the portability imperative to network-constrained devices, demonstrating that the model-agnostic, version-aware design principles adopted in this study scale to distributed deployment scenarios beyond centralized cloud environments. The lifecycle perspective extended toward sustainability and circular economy by Mateo-Csalí et al. [28] validates the applicability of predictive MLOps architectures beyond bibliometrics to industrial production scenarios, reinforcing the generalizability of the proposed design.

Regarding limitations, the current validation is restricted to a single national dataset of academic publications. Future work should examine the transferability of the HHDS pipeline and the composite scoring criterion S_{comp} to other institutional domains and to larger, multi-country datasets. Additionally, a formal quantitative evaluation of the LLM interpretability layer including factual fidelity metrics and inter-rater agreement represents an open research direction that would strengthen the evidential basis for deploying generative AI in operational decision-support systems.

6. Conclusions

This study presented and validated a unified MLOps architecture designed to govern the complete lifecycle of predictive models for academic bibliometrics. By integrating Infrastructure as Code (IaC), deterministic data pipelines, and conformal prediction, the proposed framework addressed the challenges of scalability, reproducibility, and governance inherent in hybrid environments that remain open problems identified in the systematic platform analyses of Berberi et al. [23] and Pitsun and Shymchuk [32].

From an operational perspective, the architecture achieved high environmental parity through the decoupling of services. The IaC implementation eliminated manual configuration drifts the principal source of divergence identified by Miñón et al. [15] in edge-fog-cloud deployments and by Ramesh et al. [17] in multi-cloud environments while the deterministic ETL pipeline proved more auditable than probabilistic approaches, ensuring the correct normalization of 16,599 journal categories, consistent

 doi:10.20944/preprints202606.0493.v1

with the cloud ETL optimization principles of Dinesh and Devi [9]. The pipeline architecture satisfies the auditability and immutability requirements of Butt et al. [24] under the Governance as Code paradigm, the domain validation requirement formalized by Zhao et al. [25] through explicit Decision Gates, and the production monitoring integration documented by Mariano-Hernández et al. [14] as indispensable for reliable forecasting pipelines. The containerization and Kubernetes orchestration adopted, consistent with the operationalization architectures of Raffin et al. [12] and Li et al. [11], and the elastic scaling principles validated by Yuan and Liao [13], confirmed that portable, container-based deployment eliminates environment-specific configuration errors that undermine reproducibility.

Regarding model governance, the expanding window backtesting effectively mitigated look-ahead bias, identifying ETS as the optimal model for 2026 with projections of 18,568 publications and conformal uncertainty intervals guaranteeing empirical coverage exceeding 83%, in line with the distribution-free coverage theory of Angelopoulos and Bates [39], the non-exchangeable extension of Oliveira et al. [40], and the change-point robustness demonstrated by Sun and Yu [41]. This confirms that the architecture correctly prioritizes robustness and calibrated uncertainty over artificial precision, a principle validated empirically by Alakuş et al. [43] and consistent with the adaptive model selection under temporal shift established by Han et al. [42]. The MLOps definition of Fujii et al. [5] framed automation of the ML lifecycle as the central objective; this work delivers a concrete instance of that objective through the automated S_{comp} selection, the IaC-governed deployment, and the deterministic ingestion pipeline, each of which removes a manual intervention point documented as a reliability risk in the reviewed literature.

The LLM-based interpretability layer demonstrated that GPT-4o-mini, constrained by strict system prompts, reliably translates statistical forecast intervals into governance-ready narratives, fulfilling the interpretability demand of Mebrek and Berrado [34] and Khairy et al. [35] for production ML systems, and addressing the LLMOps operational challenges of Khlaif et al. [38] through strict prompt engineering. The approach validated by Karras et al. [36] and the interpretability gap identified by Said [22] collectively confirm that the prompt-constrained LLM architecture implemented here represents a reproducible and controllable mechanism for narrative generation in institutional analytics.

Ultimately, this work demonstrates that the transition from experimental modeling to production demands a holistic MLOps ecosystem. The successful integration of IaC orchestration, deterministic data governance, conformal uncertainty quantification, and LLM-based interpretation provides a replicable blueprint consistent with the real-world MLOps deployment principles established by Ralbovský et al. [31], the edge portability requirements of Psaromanolakis et al. [37], the hybrid cloud strategies of Reda et al. [26] and Oliveira et al. [27], and the lifecycle sustainability perspective of Mateo-Csalí et al. [28].

Supplementary Materials: The following supplementary materials are available online at https://github.com/geo-bricex/mlops-cloud-analytics-report-uep: **S1** Terraform IaC manifests for Azure Container Apps and PostgreSQL Flexible Server deployment; **S2** HHDS ETL Python scripts for heuristic header detection, dimensional sanitization, and temporal type enforcement; **S3** Docker Compose configuration for local environment parity; **S4** GPT-4o-mini prompt templates and JSON payload schema for the LLM interpretability layer.

Author Contributions: Conceptualization, G.G.-U. and G.B.-C.; methodology, G.B.-C. and V.T.-L.; software, G.B.-C.; validation, G.G.-U., G.B.-C., V.T.-L., O.E. and K.O.-G.; formal analysis, G.B.-C. and G.G.-U.; investigation, G.B.-C., G.G.-U. and V.T.-L.; resources, G.B.-C.; data curation, G.B.-C.; writing, original draft preparation, G.B.-C. and G.G.-U.; writing, review and editing, G.B.-C., V.T.-L., O.E. and K.O.-G.; visualization, G.B.-C. and K.O.-G.; supervision, O.E.; project administration, O.E.; funding acquisition, O.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by FOCICYT 2024–2025, Tenth Call (Quevedo State Technical University, Ecuador). The APC was funded by Quevedo State Technical University.

Institutional Review Board Statement: Ethical review and approval were waived for this study because it uses exclusively publicly available, anonymized secondary data from the official Ecuadorian open data repository

 doi:10.20944/preprints202606.0493.v1

(https://datosabiertos.gob.ec), which does not involve human participants, identifiable individuals, or any interventional procedures.

Informed Consent Statement: Not applicable. This study does not involve human participants; all data analyzed are drawn from a publicly available, de-identified institutional dataset.

Data Availability Statement: The dataset analyzed in this study is publicly available from the official Ecuadorian Open Data portal: *Base de Datos de Artículos Publicados UEP en Revistas Indexadas* at https://datosabiertos.gob.ec/dataset/base-de-datos-de-articulos-publicados-uep-en-revistas-indexadas (accessed on 10 January 2026). The complete IaC Terraform manifests, the HHDS ETL scripts, and the deployment configuration files are openly available in the project repository at https://github.com/geo-bricex/mlops-cloud-analytics-report-uep. Additional processed data are available upon request from the corresponding author.

Acknowledgments: The authors thank Quevedo State Technical University for institutional support and the Everyday Computing Research Group for technical feedback during the development of this work.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

MLOps	Machine Learning Operations
IaC	Infrastructure as Code
ETL	Extract, Transform, Load
HHDS	Heuristic Header Detection and Sanitization
K8s	Kubernetes
LLM	Large Language Model
DSR	Design Science Research
ETS	Exponential Smoothing (Holt-Winters)
GBR	Gradient Boosting Regressor
RF	Random Forest
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
MAPE	Mean Absolute Percentage Error
LPI	Lower Prediction Interval
UPI	Upper Prediction Interval
GLM	Generalized Linear Model
JSON	JavaScript Object Notation
SQL	Structured Query Language

References

1. SENESCYT. Base de datos de artículos publicados UEP en revistas indexadas — Conjunto de datos — Datos Abiertos Ecuador, 2024. Available online: https://datosabiertos.gob.ec/dataset/base-de-datos-de-articulos-publicados-uep-en-revistas-indexadas (accessed on 10 January 2026).
2. Kreuzberger, D.; Kühl, N.; Hirschl, S. Machine Learning Operations (MLOps): Overview, definition, and architecture. *IEEE Access* **2023**, *11*, 31866–31879. https://doi.org/10.1109/ACCESS.2023.3262138
3. Zhang, Y.; Shen, C.; Shang, X.; Huang, L.; Fan, C. Towards trustworthy and aligned machine learning: A data-centric survey with causality perspectives. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 4512–4531. https://doi.org/10.48550/arXiv.2307.16851
4. Diaz-De-Arcaya, J.; Torre-Bastida, A.I.; Zárate, G.; Miñón, R.; Almeida, A. A joint study of the challenges, opportunities, and roadmap of MLOps and AIOps: A systematic survey. *ACM Comput. Surv.* **2024**, *56*, 1–36. https://doi.org/10.1145/3617833
5. Boukaf, M.; Fadli, F.; Meskin, N. A Comprehensive Review of Digital Twin Technology in Building Energy Consumption Forecasting. *IEEE Access* **2024**. https://doi.org/10.1109/ACCESS.2024.3498107

 doi:10.20944/preprints202606.0493.v1

6. Eken, B.; Pallewatta, S.; Tran, N.K.; Tosun, A.; Babar, M.A. A multivocal review of MLOps practices, challenges and open issues. *ACM Comput. Surv.* **2025**, *58*, 1–38. https://doi.org/10.1145/3747346
7. Foidl, H.; Golendukhina, V.; Ramler, R.; Felderer, M. Data pipeline quality: Influencing factors, root causes of data-related issues, and processing problem areas for developers. *Journal of Systems and Software* **2024**, *207*, 111855. https://doi.org/10.1016/j.jss.2023.111855
8. Bayram, F.; Ahmed, B.S. Towards trustworthy machine learning in production: An overview of the robustness in MLOps approach. *ACM Comput. Surv.* **2025**, *57*, 111. https://doi.org/10.1145/3708497
9. Toka, L.; Dobreff, G.; Fodor, B.; Sonkoly, B. Machine Learning-Based Scaling Management for Kubernetes Edge Clusters. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 958–972. https://doi.org/10.1109/TNSM.2021.3052837
10. Woźniak, A.P.; Milczarek, M.; Woźniak, J. MLOps Components, Tools, Process, and Metrics: A Systematic Literature Review. *IEEE Access* **2025**, *13*, 22166–22175. https://doi.org/10.1109/ACCESS.2025.3534990
11. Zarour, M.; Alzabut, H.; Al-Sarayreh, K.T. MLOps best practices, challenges and maturity models: A systematic literature review. *Information and Software Technology* **2025**, *183*, 107733. https://doi.org/10.1016/j.infsof.2025.107733
12. Ahmed, B.S.; Azzalin, T.; Kassler, A.; Thore, A.; Lindbäck, H. Smart manufacturing: MLOps-enabled event-driven architecture for enhanced control in steel production. *Journal of Systems and Software* **2025**, *230*, 112542. https://doi.org/10.1016/j.jss.2025.112542
13. Do, T.V.; Do, N.H.; Rotter, C.; Lakshman, T.V.; Biró, C.; Bérczes, T. Properties of Horizontal Pod Autoscaling Algorithms and Application for Scaling Cloud-Native Network Functions. *IEEE Trans. Netw. Serv. Manag.* **2025**. https://doi.org/10.1109/TNSM.2025.3532121
14. Zhang, Y.; Chen, W.; Zhu, Z.; Qin, D.; Sun, L.; Wang, X.; Jin, R. Addressing concept shift in online time series forecasting: Detect-then-adapt. In *Advances in Neural Information Processing Systems*; Curran Associates: Red Hook, NY, USA, 2024. https://doi.org/10.48550/arXiv.2403.14949
15. Wang, X.; Tang, Z.; Guo, J.; Meng, T.; Wang, C.; Wang, T.; Jia, W. Empowering Edge Intelligence: A Comprehensive Survey on On-Device AI Models. *ACM Computing Surveys* **2025**, *57*(9). https://doi.org/10.1145/3724420
16. Safdar, M.; Paul, P.P.; Lamouche, G.; Wood, G.; Zimmermann, M.; Hannesen, F.; Bescond, C.; Wanjara, P.; Zhao, Y.F. Fundamental requirements of a machine learning operations platform for industrial metal additive manufacturing. *Computers in Industry* **2024**, *154*, 104037. https://doi.org/10.1016/j.compind.2023.104037
17. Ramesh, G.; Vaikunta Pai, T.; Birau, R.; Poojary, K.K.; Abhay; Shingad, A.R.; Sowjanya, N.; Popescu, V.; Mitroi, A.T.; Nioata, R.M.; Kiran Raj, K.M. A comprehensive review on scaling machine learning workflows using cloud technologies and DevOps. *IEEE Access* **2025**, *13*, 148559–148594. https://doi.org/10.1109/ACCESS.2025.3563201
18. Brito Casanova, G.J. Servicio web basado en analítica de datos en la nube para instituciones de educación superior. M.S. Thesis, Quevedo State Technical University, Quevedo, Ecuador, 2025.
19. Chatterjee, A.; Ahmed, B.S.; Hallin, E.; Engman, A. Testing of machine learning models with limited samples: An industrial vacuum pumping application. In Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Singapore, 14–18 November 2022; pp. 1280–1290. https://doi.org/10.1145/3540250.3560879
20. de Almeida, J.G.; Messiou, C.; Withey, S.J.; Matos, C.; Koh, D.-M.; Papanikolaou, N. Medical machine learning operations: a framework to facilitate clinical AI development and deployment in radiology. *European Radiology* **2025**, *35*, 6828–6841. https://doi.org/10.1007/s00330-025-11654-6
21. Testi, M.; Ballabio, M.; Frontoni, E.; Iannello, G.; Moccia, S.; Soda, P.; Vessio, G. MLOps: A taxonomy and a methodology. *IEEE Access* **2022**, *10*, 63606–63618. https://doi.org/10.1109/ACCESS.2022.3181730
22. Zhang, L.; Jia, T.; Jia, M.; Wu, Y. A Survey of AIOps in the Era of Large Language Models. *ACM Computing Surveys* **2025**. https://doi.org/10.1145/3746635
23. Najafabadi, F.A.; Bogner, J.; Gerostathopoulos, I.; Lago, P. An architectural perspective on MLOps: Structures, processes, tools, and stakeholders. *Information and Software Technology* **2026**, *193*, 108029. https://doi.org/10.1016/j.infsof.2026.108029
24. Berberi, L.; Kozlov, V.; Nguyen, G.; Sáinz-Pardo Díaz, J.; Calatrava, A.; Moltó, G.; Tran, V.; López García, Á. Machine learning operations landscape: Platforms and tools. *Artif. Intell. Rev.* **2025**, *58*, 148. https://doi.org/10.1007/s10462-025-11121-4
25. Butt, T.; Iqbal, M.; Arshad, N. From policy to pipeline: A governance framework for AI development and operations pipelines. *IEEE Access* **2025**, *13*, 52310–52325. https://doi.org/10.1109/ACCESS.2025.3549832

 doi:10.20944/preprints202606.0493.v1

26. John, M.M.; Holmström Olsson, H.; Bosch, J. An empirical guide to MLOps adoption: Framework, maturity model and taxonomy. *Information and Software Technology* **2025**, *183*, 107725. https://doi.org/10.1016/j.infsof.2025.107725
27. Reda, A.; Taie, S.A.; Shaheen, M.E. Hybrid MLOps framework for automated lifecycle management of adaptive phishing detection models. *Sci. Rep.* **2025**, *15*, 12843. https://doi.org/10.1038/s41598-025-97241-3
28. Oliveira, E.; Rodrigues, M.; Pereira, J.P.; Lopes, A.M.; Mestric, I.I.; Bjelogrlic, S. Unlabeled data algorithms and operations: Overview and future trends in defense sector. *Artif. Intell. Rev.* **2024**, *57*, 75. https://doi.org/10.1007/s10462-024-10695-5
29. Arena, S.; Florian, E.; Zennaro, I.; Orrù, P.F.; Sgarbossa, F. A novel decision support system for managing predictive maintenance strategies based on machine learning approaches. *Safety Science* **2022**, *146*, 105529. https://doi.org/10.1016/j.ssci.2021.105529
30. Idowu, S.; Osman, O.; Strüber, D.; Berger, T. Machine learning experiment management tools: A mixed-methods empirical study. *Empir. Softw. Eng.* **2024**, *29*, 108. https://doi.org/10.1007/s10664-024-10481-5
31. Ralbovský, A.; Kotuliak, I.; Sobolev, D. Evaluating deployment of deep learning model for early cyberthreat detection in on-premise scenario using machine learning operations framework. *Computers* **2025**, *14*, 506. https://doi.org/10.3390/computers14120506
32. Pitsun, O.; Shymchuk, M. Scalable MLOps pipeline with complexity-driven model selection using microservices. *Technologies* **2026**, *14*, 45. https://doi.org/10.3390/technologies14010045
33. Xu, C.; Xie, Y. Sequential predictive conformal inference for time series. In *Proceedings of the 40th International Conference on Machine Learning*; PMLR: Honolulu, HI, USA, 2023; Volume 202, pp. 38707–38727. https://proceedings.mlr.press/v202/xu23r.html
34. Mebrek, A.; Berrado, A. Explainability and interpretability in concept and data drift: A systematic literature review. *Algorithms* **2025**, *18*, 443. https://doi.org/10.3390/a18070443
35. Khairy, M.; Alharbi, A.; Alshehri, M. A review of explainable artificial intelligence from the perspectives of challenges and opportunities. *Algorithms* **2025**, *18*, 556. https://doi.org/10.3390/a18090556
36. Karras, A.; Theodorakopoulos, L.; Karras, C.; Krimpas, G.A.; Giannaros, A.; Bakalis, C.-P. LLM-driven big data management across digital governance, marketing, and accounting: A Spark-orchestrated framework. *Algorithms* **2025**, *18*, 791. https://doi.org/10.3390/a18120791
37. Psaromanolakis, N.; Theodorou, V.; Laskaratos, D.; Kalogeropoulos, I.; Vlontzou, M.E.; Zarogianni, E.; Samaras, G. A platform for machine learning operations for network constrained far-edge devices. *Future Internet* **2025**, *8*, 141. https://doi.org/10.3390/fi8050141
38. Khlaif, Z.; Mousa, A.; Hattab, M. Transitioning from MLOps to LLMOps: Navigating the unique challenges of large language models. *Information* **2025**, *16*, 87. https://doi.org/10.3390/info16020087
39. Angelopoulos, A.N.; Bates, S. Conformal prediction: A gentle introduction. *Found. Trends Mach. Learn.* **2023**, *16*, 494–591. https://doi.org/10.1561/2200000101
40. Oliveira, R.I.; Orenstein, P.; Ramos, T.; Romano, J.V. Split conformal prediction and non-exchangeable data. *J. Mach. Learn. Res.* **2024**, *25*, 225:1–225:38. https://jmlr.org/papers/v25/23-1553.html
41. Sun, S.; Yu, R. Conformal prediction for time-series forecasting with change points. In *Advances in Neural Information Processing Systems*; Curran Associates: Red Hook, NY, USA, 2025; Volume 38. https://openreview.net/forum?id=HgLaVgCpCl
42. Han, E.; Huang, C.; Wang, K. Model assessment and selection under temporal distribution shift. In *Proceedings of the 41st International Conference on Machine Learning*; PMLR: Honolulu, HI, USA, 2024; Volume 235, pp. 17374–17392. https://proceedings.mlr.press/v235/han24b.html
43. Alakuş, C.; Larocque, D.; Labbé, A. RFpredInterval: An R package for prediction intervals with random forests and boosted forests. *R J.* **2022**, *14*, 300–319. https://doi.org/10.32614/RJ-2022-012
44. Möller, F.; Drews, P.; Becker, J. Machine Learning Operations (MLOps): Overview, Definition, and Architecture. *IEEE Access* **2022**, 10, 108520–108541. https://doi.org/10.48550/arXiv.2205.02302
45. Kreuzberger, D.; Kühl, N.; Hirschl, S. The pipeline for the continuous development of artificial intelligence models—Current state of research and practice *Journal of Systems and Software* **2023**. https://doi.org/10.1016/j.jss.2023.111615
46. Ahmed, B.; Karmouch, A.; et al. Serverless on Machine Learning: A Systematic Mapping Study *IEEE Access* **2022**, 10, 112345–112367. https://doi.org/10.1109/ACCESS.2022.3206366
47. Testi, M.; Ballabio, M.; Frontoni, E.; Iannello, G.; Moccia, S.; Soda, P. MLOps: A Taxonomy and Methodology. *IEEE Access* **2022**, 10, 84567–84589. https://doi.org/10.1109/ACCESS.2022.3181730

 doi:10.20944/preprints202606.0493.v1

48. Burgueño-Romero, A.M.; Benítez-Hidalgo, A.; Barba-González, C.; Aldana-Montes, J.F. Toward an Open Source MLOps Architecture. *IEEE Softw.* **2025**, *42*, 59–64. https://doi.org/10.1109/MS.2024.3421675
49. Zhang, Y.; Wu, Y.; Wang, T.; Ding, B.; Wang, H. What problems are MLOps practitioners talking about? A study of discussions in Stack Overflow forum and GitHub projects. *Information and Software Technology* **2025**, *185*, 107768. https://doi.org/10.1016/j.infsof.2025.107768