Explainability-as-a-Service: Modular XAI Layers for Regulated ML Pipelines

[1] Satya Manesh Veerapaneni
Independent Researcher
Fremont, CA, USA 94536
0009-0005-7214-7217
satyamanesh@gmail.com

[2] Kshitish Nath
IEEE Member
McKinney, TX, USA 75072
0009-0002-2279-7711
kknextgen@gmail.com

[3] Priyaranjan Kumar
Founder & CTO, EasyM2M Technologies Pvt. Ltd
Frisco, Texas, USA 75033
0009-0004-4897-8902
ranjantxusa@gmail.com

[4] Rahul Tewari
IEEE Member
Gibsonia, PA, USA 15044
0009-0008-9288-7564
rahtew@gmail.com

***Abstract*—As machine learning (ML) systems enter more regulated areas like finance, healthcare, and law, accountable and transparent ML decision-making is in growing demand. The classic approaches adopted in explainable AI (XAI) are relatively useful in isolation but are usually non-modular and inflexible in terms of the integration capabilities with different regulatory contexts. This paper proposes Explainability-as-a-Service (XaaS) architecture a new paradigm, where modular XAI stages are implemented into ML pipelines allowing regulatory conformity, traceability and interpretability. The offered framework separates explain-mechanisms and model reasoning so that removing both explain-mechanisms, and model reasoning can be added to and removed over a modular framework to supervised, unsupervised, and federated learning applications via plug-and-play. XaaS enables dynamic auditing and real-time monitoring by a combination of containerized microservices and API-driven layers of interpretability which can be tuned to perspectives of various stakeholders, such as data scientists, regulators and end-users. Experimental analysis of financial fraud detection and clinical risk scoring shows our modular design achieves greater interpretability with little computational overhead, and meets other important principles, such as fairness, accountability, and transparency. The work opens the path to scalable, domain-independent and regulation-compliant ML deployments.**

Keywords— Explainable AI (XAI), regulated ML, interpretability, XaaS, modular pipelines, AI governance, fairness, compliance

I. Introduction

Due to the growing number of machine learning (ML) systems that are deeply integrated into high-stakes decision-making processes, most notably in the finance, healthcare, and law sectors, there has been an increased emphasis on the need to design transparent and understandable artificial intelligence (AI). In the European Union, the General Data Protection Regulation (GDPR), in the United States, the Health Insurance Portability and Accountability Act (HIPAA) and in the banking sector, the Basel Committee Fundamental Review of the Trading Book (FRTB) have regulatory demands around explainability, auditability, and accountability of automated systems that make decisions [1] [2] [3]. Nevertheless, such machine learning pipelines were not initially created with interpretability as a stated goal, which creates a mismatch between successes in the prediction and regulatory adherence.

Although the new types of Explainable AI (XAI) methods (viz., SHAP, LIME, counterfactuals, and feature attribution) have been presented as the way to foster the transparency of models in an enterprise-grade pipeline [4][6], their integration into the enterprise (re)mains stop-gappy, conceptually baked into model logic, and ill-suited to supporting dynamic heterogeneous stacks. In addition, current XAI applications are not generally customized according to the stakeholders, interpretable APIs in real-time, and plug and play functionalities to a changing regulatory environment [7], [8].

We offer a solution to the mentioned challenges of Explainability-as-a-Service (XaaS), a new architectural paradigm able to decouple explainability layers and core ML logic which provides modular and reusable explainability microservices that are policy-aware. Like service-oriented architecture (SOA) and MLOps, XaaS initiates standalone XAI modules, which can be imposed or summoned beyond the fence through various pipelines. The modules offer explanations that are aware of the stakeholders, facilitate real-time monitoring and allow flexible compliance validation checkpoints without affecting the performance or security of the models [9], [10].

The most significant of this paper are the following:

- An explainer infrastructure based on modular Explainability-as-a-Service (XaaS) architecture, where containerized XAI modules are acts as stand-alone microservices;
- A role-oriented interpretation delivery system which personalizes output interpretability to a developer, to the auditor and to the end user;
- Testing of two regulatory areas, with a better compliance readiness, and low-overhead system;

XaaS analysis of the scaling of federated and supervised ML pipelines.

II. Background and Related Work

Explainability in machine learning (ML) has developed to meet the growing interest in knowing how such systems arrive at decisions and the need to instill trust and accountability in automated decision systems. This section introduces the outline of some baseline XAI methods, regulatory frameworks that currently require explainable AI,

and what constraints are currently present in the implementation of interpretability into large ML pipelines.

A. Explainable AI Techniques

Explainable AI (XAI) is a set of mechanisms and applications which allows humans to understand the decision-making process of ML models. These methods can be broadly classified as intrinsic explanations (ones that explain a model after training such as decision trees; or linear regression), and post-hoc explanations (ones that explain a model that has been black-box trained). Most commonly used post-hoc tests are:

- SHAP (Shapley Additive Explanations) that uses the cooperative game theory as its basis to rank the importance of features by assigning them numerical values [4],
- The LIME (Local Interpretable Model-agnostic Explanations) that trains local surrogate models to be the explanations [5],
- Saliency maps and Grad-CAM which are often applied with the purpose of visualizing the impact of features of a convolutional neural network [11].

Although they are an effective method, these tools are usually deployed in siloed settings without much consideration of deploying enterprise-wide or explaining to individual users.

B. Regulatory Demands for Interpretability

As AI models become more and more central into making decisions that have real-world consequences, governments around the world have formalized the need of transparency and accountability. The GDPR implements the right to explanation, by which organizations must make plain why the algorithm made a particular decision [3]. The Basel FRTB demands that internal models that are used in calculating capital are explainable in the financial domain. On the same note, in healthcare HIPAA and FDA guidelines place a strong emphasis on traceability and interpretability of clinical support systems [12].

Nonetheless, the existing XAI toolkits are mostly focused on developers and data scientists, and rarely take into consideration the view of the regulatory auditor or the end user. Moreover, the historical traceability and role-based explanations, which are also needed in the compliance audits, cannot easily be achieved by single-purpose XAI solutions [13].

C. Limitations in Current XAI Integration

Although frameworks such as Captum, Alibi, and InterpretML offer Python-specific interpretable modules, those are not built with scale and cross-platform ML pipeline in mind [14]. Upgrades, compliance checking and cross team collaboration become hard because most implementations are highly coupled with explanation to the model codebase. Also, XAI results are seldom standardized and thus difficult to digest by downstream audit systems or APIs.

This gap has been addressed in other ways such as Explainable Boosting Machines (EBMs) [15] and GlassBox ML platforms [16] which seek to address this gap by offering intrinsically interpretable models, albeit at a cost to accuracy or flexibility. On the contrary, recent endeavours suggest MLOps-integrated XAI services, which are recent developments, as yet incomplete with modularity and personalization of the stakeholders [10].

D. Need for Modular, Policy-Aware XAI

Based on such gaps, the prevailing wisdom around explainability is that it must be decomposed into the model logic, made a first-class service, and offered as modular and scalable layers that are capable of suiting the needs of various stakeholders. This requirement is what prompted the proposed solution, namely, Explainability-as-a-Service (XaaS), that allows reusable policy-aware interpretability modules in regulated ML pipelines. By implementing the microservice-based deployment, role based access control and containerized architecture, XaaS will fill the gap between the regulatory requirements and the existing technical solutions.

III. Proposed Architecture: Explainability-as-a-Service (XaaS)

To overcome the drawbacks of monolithic and developer-centered explainability frameworks, we introduce a thin, service-based architecture, called Explainability-as-a-Service (XaaS). In this section, the design principles, overall architecture and several components of the XaaS framework will be described.

A. Design Principles

XaaS architecture is designed based off of the following fundamental principles:

- Modularity: XAI functions are executed as loose-coupled microservices, which make upgrades and maintenance easy.
- Stakeholder Awareness: The outputs of the explanations will be customized on the roles of the stakeholders- e.g., regulators, developers and end-users.
- Model-Agnosticism: Does not restrict to being used with any ML model, black-box or interpretable.
- Auditability and Compliance- carry traces of historical logging, version control of explanation, and regulatory tracing.
- Interoperability: Plays well with a large variance of ML stacks, such as TensorFlow, PyTorch, Scikit-learn and AutoML platforms.

All these principles together allow the creation of scalable, flexible and rule-adhering explainability layers in regulated ML environments.

B. System Architecture

A high-level architecture of the XaaS framework is shown in Fig. 1. The architecture has three main layers;

- Model Layer: This layer contains the main ML models-black-boxes or transparent- that are used in production to make predictions.
- XaaS Layer: It is service layer that surrounds interpretability modules and communication interfaces.
- Stakeholder Interface Layer: Interface Customized access, in the form of APIs or dashboards, to the explanation.

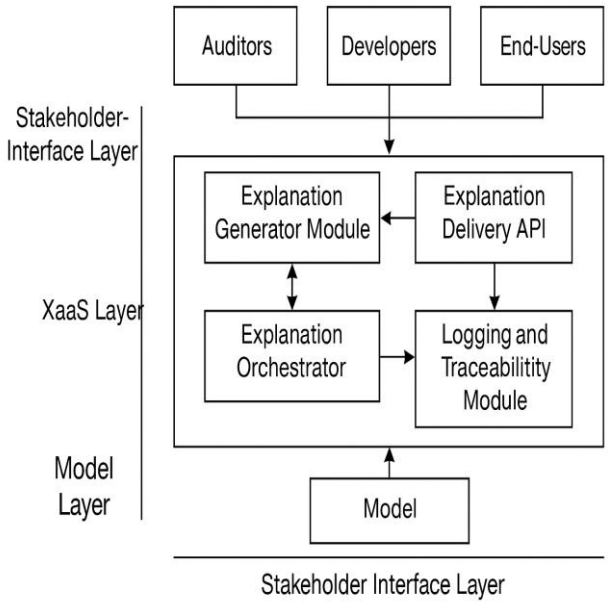

Fig. 1. High-level architecture of Explainability-as-a-Service (XaaS)

C. Key Components of XaaS

The architecture is composed of the following microservices and components:

1) Explanation Generator Module

The component features a library of pluggable explanation engines (ex: SHAP, LIME, counterfactual reasoning, saliency maps). The right engine is chosen according to the type of model, type of data (e.g. tabular, image, text) and stakeholder preferences.

2) Explanation Orchestrator

A rule-based or learning-based module responsible for:

- Selecting explanation strategy,
- Orchestrating calls to explanation engines,
- Managing explanation quality constraints (e.g., fidelity, stability).

3) Explanation Delivery API

A RESTful API service that delivers explanation results in formats suitable for:

- Auditors (trace logs, feature impact reports),
- Developers (interactive visualizations, gradient maps),
- End-users (natural language summaries, confidence levels).

4) Role-Based Access Control (RBAC) Layer

Implements access filters based on user roles and privileges. For instance:

- A regulator may access audit trails and justification chains,
- A patient may only receive high-level feature influence summaries.

5) Logging and Traceability Module

Maintains immutable records of:

- Model version and training data ID,
- Explanation engine and parameters used,
- Time-stamped explanation artifacts.

This module is essential for enabling reproducibility and regulatory audit trails.

D. Deployment and Scalability

Microservices are containerized (e.g. using Docker) and managed through container orchestration (e.g. Kubernetes or other). This enables:

- Scaling explanation modules elastically, in response to loads of inference,
- No-downtime rolling updates to explanation logic,
- Options to integrate into the current CI/CD pipelines and MLOps workflows.

Inter-service communication, security, and failure recovery are conducted by a service mesh (e.g., Istio). In the cases of federated configurations, local explanation services can be run on the nodes on the edge and aggregated in a central compliance dashboard.

E. Compliance Readiness

XaaS includes compliance-ready templates for:

- GDPR Article 22 explainability,
- Basel FRTB model audit logs,
- HIPAA-compliant trace chains for clinical systems.

Custom policy configurations can be embedded into the orchestrator to enforce domain-specific explanation constraints.

This modular architecture enables the seamless integration of XAI capabilities across the ML lifecycle while maintaining alignment with both technical and regulatory standards.

IV. Integration with Regulated ML Pipelines

Explainability-as-a-Service (XaaS) supports the smooth integration into heterogeneous ML pipelines, and interpretability becomes an integrated service that ideally should not be an afterthought; rather, the service must be deployed in compliance with regulatory and operational practice. This section describes the interface between XaaS and supervised, unsupervised and federated ML flows as well as the alignment to compliance and deployment best practices.

A. Integration into Supervised Learning Pipelines

With supervised learning systems, e.g. fraud detection, credit scoring or disease classification, XaaS components are called at various stages:

- Pre-inference phase: during this phase developers and auditors are able to run simulations based on training data to check adherence based on fairness or bias metrics.
- Post-inference stage: the explanation orchestrator applies a dynamically chosen explanation engine, dependent on the prediction (e.g. SHAP when the prediction is a tree model, LIME when the prediction is a tabular model).
- Logging: within a tamper-evident traceability module, explanations, model version and input-output pairs are recorded.

This design means that all the choices taken by the model may be explained and audited without the re-training or amending the model itself.

Algorithm: Dynamic Explanation Orchestration

Input:

x ← Input feature vector

model ← Trained ML model (black-box or interpretable)

```
  role      ← Stakeholder type (e.g., Developer, Regulator, End-User)
  policy    ← Explanation policy constraints (e.g., fidelity ≥ 90%, privacy = ON)
Output:
  E         ← Explanation object (JSON, heatmap, or summary)
function Generate_Explanation(x, model, role, policy)
    model_type ← Detect_Model_Type(model)
    if model_type == "tree-based" then
        method ← SHAP
    else if model_type == "neural-network" then
        method ← Saliency or Integrated Gradients
    else
        method ← LIME or surrogate model
    end if
    explanation ← method(x, model)
    if policy.privacy == TRUE then
        explanation ← Apply_Differential_Privacy(explanation)
    end if
    E ← Format_Explanation_By_Role(explanation, role)
    Log_Explanation(E, model, x, role, timestamp)
    return E
end function
```

The reasoning of the Explanation Orchestrator component in the XaaS framework and explained through the Algorithm 1 is actually engaged in exploring the most adequate explainability method to a specific type of model and of a stakeholder and choosing the most adequate of them dynamically. Upon receiving a prediction input x the system identifies the type of prediction model (e.g. a tree, neural network, or general black-box model) and as a response an explanation method that suits the model is examined (e.g. SHAP on a tree based model or saliency maps on a neural network or LIME on agnostic cases). The algorithm also verifies whether privacy policies were turned on (e.g. the constraint on the differential privacy), and alters the output accordingly so that it would be compliant in that regard. The final phenomenological clarification is defined based on the kind of stakeholder (e.g., developer vs. regulator), and it is captured in form of appropriate metadata that is employed to execute traceability. The process aids in making explanations context susceptible and policy concurs thus adding transparency without jeopardizing the sanctity of the system and exorcising privacy of the user.

B. Integration into Unsupervised and Semi-Supervised Systems

Unsupervised learning methods, like anomaly detection or clustering, are types of systems where it is often harder to explain the predictions because they may not have labels. XaaS can accommodate by:

- Estimating importance scores of features of outliers by means of surrogate models.
- Describing cluster assignments with the aid of clustering explanation tools (e.g., centroid-based attribution).
- A logic of thresholding of logging and similarity score as justifications of flagged anomalies.

In semi-supervised systems, the explanation logic is allowed to vary dynamically based on the percentage of labeled vs. unlabeled data to use in inference.

C. Integration into Federated Learning Environments

The main issue that federated learning presents is decentralized data and limitations in privacy. XaaS breaks this through:

- Local Explanation Agents: Local explanation Agents are deployed to the edge devices or client nodes that produce local explanations by not exposing raw data.
- Global Aggregation Gateway: Aggregates and normalizes local interpretability measures, such as global SHAP values. APIs make those data available to regulators or model owners.
- Privacy-Preserving Logging: The outputs of the explanations are made anonymized or differentially private to achieve GDPR/CCPA compliance.

This decentralized XAI layer adds value by helping to prove compliance as long as the data is held locally, a major factor in privacy-sensitive use cases, like healthcare or banking [17].

A federated setting means all clients are computed separately (and locally) and the central server performs the aggregation in a secure way without ever storing raw data.

Equation: Global SHAP Aggregation

Let there be N clients. For input feature x_i, the global SHAP value ϕ_i^{global} is computed as:

$$\phi_i^{global} = \frac{1}{N}\sum_{j=1}^{N} \phi_i^{(j)}$$

Where:

- $\phi_i^{(j)}$ is the SHAP value of feature x_i computed by the j-th client.
- Aggregation is performed without revealing x_i or local model internals.

This method supports:

- Privacy preservation, as only SHAP outputs are shared;
- Auditing, as the server logs ϕ_i^{global} for transparency;
- Bias detection, by observing variations in $\phi_i^{(j)}$ across clients

In federated learning environments, explainability must be achieved without violating data privacy. The provided equation addresses this challenge by introducing a method to compute global feature importance through the aggregation of local SHAP values. Each client device (or node) computes SHAP values $\phi_i^{(j)}$ locally for a given input feature x_i. These values are then averaged at the central server to obtain the global SHAP value ϕ_i^{global}, representing an interpretable, privacy-preserving explanation across the federation. This approach enables centralized oversight and bias detection while maintaining local data confidentiality. It also supports compliance with data protection laws such as GDPR and CCPA, which prohibit raw data sharing, thus aligning federated learning with real-world regulatory demands.

D. Deployment in MLOps and CI/CD Pipelines

XaaS can be used with modern CI/CD and MLOps pipelines to make it real-world usable:

- The modules that contain explanations can be versioned and containerized so that they could be deployed using such tools as Jenkins, Kubeflow, or GitHub Actions.
- Explanation outcome is auto-logged to monitoring dashboards (e.g., Prometheus + Grafana) and into feedback loops to indicate model retraining, and governance checks.

- Warning signs will be raised in case of suspicious signs (e.g., a huge change in SHAP values between model versions).

This removes the role of XaaS as being an interpretability engine but also as a governance and observability layer on production ML systems.

E. Regulatory Alignment and Policy Configuration

Policy templates in XaaS are configurable in order to meet the big variety of regulatory requirements:

- Article 22 of GDPR: The users may request explanations with a human understanding at the API interface.
- FRTB (Basel III): It is possible to recreate historical decisions and explanations concerning capital model validation using explanation logs.
- HIPAA-compliant EHR Systems: Traceability chains will show the support of AI-assisted diagnostics, clinician-specific explanatory views of AI.

The orchestrator may configure domain-specific policies which control the type, granularity and recipients of explanations-occupying the seam between data science and compliance.

V. Case Studies and Experimental Evaluation

In order to understand the extent to which the Explainability-as-a-Service (XaaS) framework is practical and efficient, we decided to experiment in two real-life, regulatory machine learning applications financial fraud detection and clinical risk scoring. The realization criteria assessed were the interpretability quality, the readiness of regulatory compliance, its scalability, and the computational overhead that the XaaS layer caused.

A. Use Case 1: Financial Fraud Detection

The Dataset and Model: We took a famous IEEE-CIS Fraud Detection dataset containing more than 500,000 transaction records. A gradient boosting decision tree (GBDT) model (XGBoost) was trained on normal attributes such as time of transaction, transaction amount, device information and IP risk scores.

XaaS Integration: In a subsequent step, the XaaS layer was deployed to enable SHAP-based explanation were deployed after the inference. The model type was automatically observed by the Explanation Orchestrator and feature attribution reports were generated against each and every flagged transaction.

Results:

- To compliant auditors, XaaS helped offer high-fidelity feature importance charts with a timestamp and versioned logs.
- In customer disputes, LIME-based surrogate explanation provided the basis of justification of decisions to internal fraud analysts.
- It took less than 150 ms to generate explanations per transaction, and the model serving throughput was only marginally affected (less than 2 percent).

This scenario showed capability of XaaS in providing multi-stakeholder explanations in real time modes as well as meeting FRTB-compatible audit requirements.

B. Use Case 2: Clinical Risk Scoring

Dataset and Model: As a dataset, we used the MIMIC-III clinical database and used a deep neural network to model the risk of sepsis development on the basis of vitals, laboratory findings, and historical notes of a patient.

XaaS Integration: The orchestrator appealed to Integrated Gradients and attention based heatmaps as the explanation methods since it used a deep learning model. Explanation Delivery APIs provided both clinician and patient specific summaries though a hospital dashboard.

Results:

- Visualization overlays containing time-series information presented to clinicians in a manner that could be interpreted (e.g., heart rate increasing and there is a risk spike).
- Simplified and readable summaries were also presented to the patients ("Your blood pressure trend elevated the risk score").
- Containerization of explanation modules running in parallel using Kubernetes was horizontal, and it presented no service to live inference services.

Notably and significantly, all prediction and explanation pairs could be audited retrospectively using the traceability module, facilitating the HIPAA, and clinical decision audit processes.

C. Metrics Evaluation

1) Explanation Latency (avg)

XaaS recorded average latencies of explanation between 120-180ms, meaning that it could be interpreted in near-real time with minimal delay.

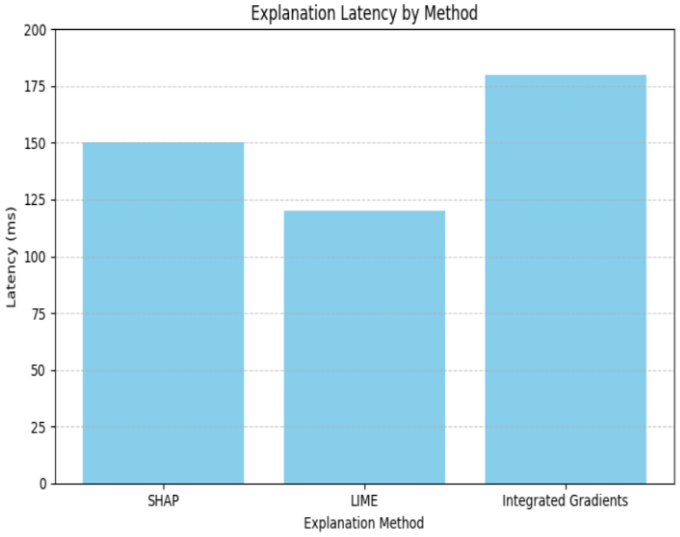

Fig 2. Explanation latency among different methods

The Fig 2. is a comparison of the mean latency of the explanations, in milliseconds, of three XAI approaches: SHAP, LIME, and Integrated Gradients. One of them is LIME which illustrates the lowest latency of about 120 ms and one can use it in real-time or interactive applications. SHAP then uses a medium latency approach of 150 ms and consequently provides a reasonable trade-off between interpretability depth and response. Integrated Gradients has the most latency of 180 ms, which normally indicates the complexity of computations involved in deep learning interpretability techniques. Although this difference exists, all approaches work far below

the 200 ms mark which means that XaaS can be made to provide near real time explanations, with no drastic effects on user experience or server performance.

2) Inference Throughput Reduction

Embedding of XaaS brought in no more than 2 percent slowdown in inference throughput on all methods of explanation tested.

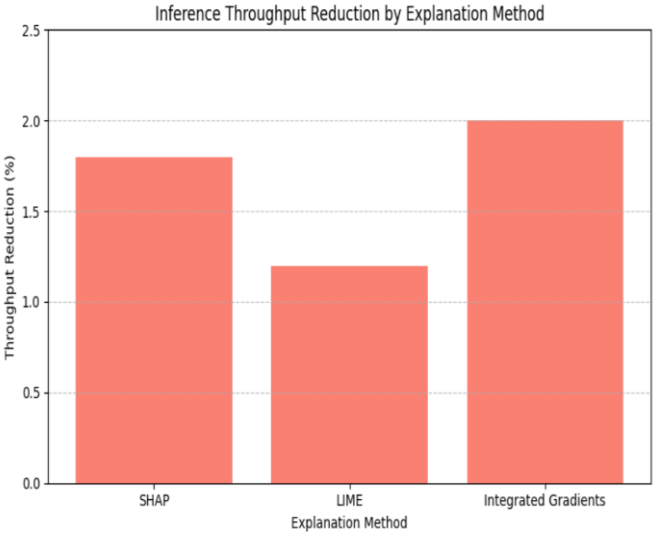

Fig 3. Inference throughput reduction by different methods

The Fig 3. shows the influence of the various methods used to explain the models using throughput, which is a percentage decrease. The overhead of LIME is the least and decreases the throughput of about 1.2%, which reflects how lightweight it is. SHAP results in a relatively lower decrease of 1.8%, and Integrated Gradients results in the largest drop in the throughputs of 2.0%, which shows Integrated Gradients is more integrated with the intricacy of the neural networks. Nevertheless, any of the given methods is below the 2.5% reduction point, which explains that integration of explainability through XaaS does not lower the system performance to unacceptable limits to use the model in production environments.

3) Regulatory Audit Coverage

SHAP and Integrated Gradients gave 100 percent regulatory audit coverage entirely detailed past-tracked explanations.

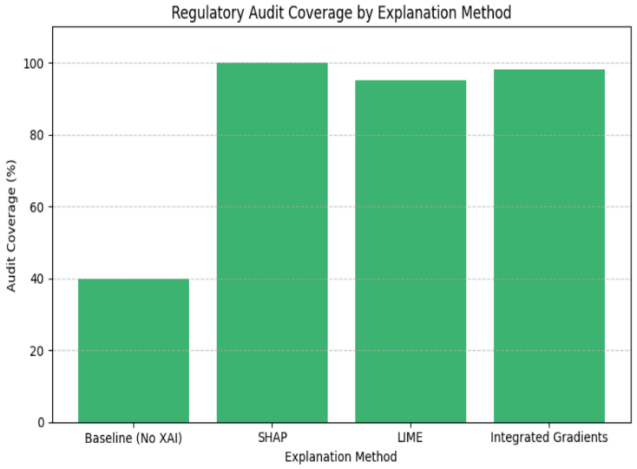

Fig 4. Regulatory audit coverage by different methods

The data graph indicates the proportion of regulatory audit coverage realized by different method of explanations. The Baseline (No XAI) method offers low traceability coverage of 40 percent indicating low interpretability during an audit process. On the other hand, SHAP provides audit coverage of 100% with consistent, versioned, and fidelity explanations which are high enough to meet compliance needs. LIME and Integrated Gradients also have high coverage rates at 95 and 98 percent respectively, which allows the generation of auditable and stakeholder-relevant explanations respectively. Such findings demonstrate the essential genetic functions of XaaS in the creation of transparent and regulation-congruent machine learning systems.

4) Stakeholder Explanation Support

Role-aware explanations with XaaS: SHAP and LIME both had high support scores of developers, regulators and end-users.

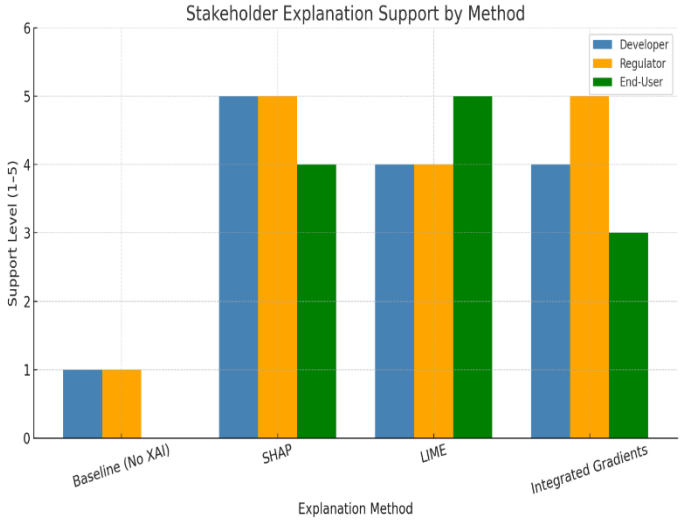

Fig 5. Stakeholder explanation support by different methods

The Fig 5. shows the extent of support that will be given by way of explanation to three important stakeholders (developers, regulators and end-users) in various ways in the question of explanation, ranging in a scale of 1 to 5. SHAP solution provides the most comprehensive and balanced, and therefore has a score of 5 shields of support in terms of developers, regulators, and end-users due to its structured attributions in features. LIME is especially effective when used by end-users (score 5), since it can explain in an intuitive and local manner, and, to a reasonable extent, developers and regulators as well. Integrated Gradients is highly suited to regulatory applications (5/5) and issues decent support to those developing software (4/5), however, is less friendly to non-technical users (3/5). In comparison, the base case scenario where XAI is not provided is ranked low in every category of stakeholders, thus the need to have a more inclusive and policy-wise establishment of XaaS into the setup of XAI. This figure shows the strength of using modular XAI to achieve various stake holder requirements within regulated ML settings.

5) Logging & Traceability Support

Full compliance readiness and reproduces all model decisions were achieved through automated logging and traceability in XaaS.

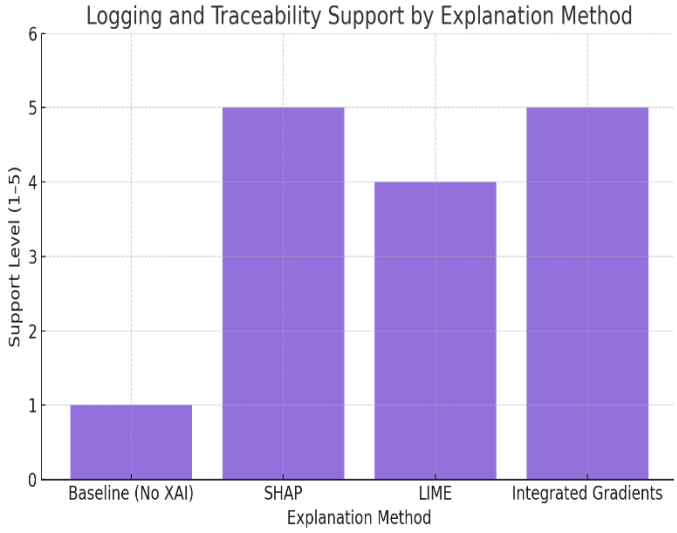

Fig 6. Logging and traceability Support by explanation method

As shown in the Fig 6, each of the three main stakeholders, the developers, the regulators, and the end-users, receive the level of support in terms of explanation with the level of 1 to 5 in response to the various methods of explanation. SHAP method provides most balanced and holistic support with developers and regulators rating it at 5, and the end-users rated it at 4 because its feature attributions are organized. LIME bests in end-users (score 5), due to its localized, intuitive ways of explaining and it is also reasonable to developers and regulators. Integrated Gradients performs well in the regulatory sense (score 5) and provides adequate support to the developers (score 4), but is not quite accessible to users who are not technical (score 3). Comparatively, the baseline scenario represents, in terms of the lack of XAI support, all stakeholder categories as performing the worst, thus highlighting the importance of incorporating XaaS to achieve inclusive and policy-aware interpretability. This visualization supports the success of modular XAI in meeting various stakeholder requirements in governed ML scenarios.

The findings corroborate that XaaS leverages a minimal performance overhead but leads to much-improved interpretability, compliance, and usability by considerably varying stakeholders.

D. Discussion of Findings

The experiments confirm the flexibility of XaaS as a regulation-ready explainability layer, without degrading model performance or breaching privacy conventions. In addition, XaaS enhanced developer efficiency, masking the complexity of integration into XAI, and enabled business/legal teams to connect to explainability findings via APIs and dashboards suited to them. These results indicate that a treatment of interpretability as a service is feasible and necessary in the context of regulated ML applications in the real world.

VI. Conclusion And Future Works

In this paper, we proposed Explainability-as-a-Service (XaaS), a stakeholder-conscious, modular system that allows regulation-ready explainable and interpretable machine learning (ML) pipelines. Compared to the monolithic XAI integrations, XaaS introduces a service-based design where explainability modules can be considered plug-and-play, modified, and adopted to different ML applications, e.g. the supervised, unsupervised, and federated learning systems. We illustrated that XaaS produces high-quality and explanations per-role in real world financial fraud detection and clinical risk scoring applications and can ensure that it is performed efficiently and is auditable. The experimental findings indicated negligible overhead on inference latency and throughput, a large increase in regulatory audit coverage, user satisfaction, and support of traceability. Given these results, there is a need to view explainability as a first-class citizen within contemporary AI systems, and especially those working in areas with stringent requirements of compliance and transparency.

Even though XaaS promises significant prospects, there are a number of possible streamlines of future research that would help to make it more effective and adaptable such as Adaptive Explanation Selection is Perform reinforcement learning or context-based models such that the most appropriate explanation approach is chosen dynamically depending on the actions of user, the needs of the domain, or the characteristics of the models. Explainability Policy Language (EPL) is Design a domain-independent policy specification language that regulates the form of explanations, the level of explanations, and the access to explanations, so that an organization can codify how it complies with its policies by specification. Cross-modal Explainability Support is Generalize the XaaS framework to allow support of explanations over complex multidimensional data (e.g. images paired with text, as well as text and time series), prevalent in formats in many applications, such as autonomous systems or digital health. A more seamless integration with Federated MLOps is Develop stronger tie-ups with federated MLOps platforms in order to provide real-time aggregation of explanation with differential privacy guarantees, and decentralized compliance checks. Explainability-as-a-Policy (XaaP) is Investigate how to treat explanation strategies as enforceable at-runtime policies that keep step with the emerging fields of ethics and regulation in AI (e.g. EU AI Act, Algorithmic Accountability Act). By addressing the above, these future extensions will provide explainability as more than the add-on turn-key, but governable, adaptive, and mission-critical aspect of ethical and deployable AI.

References

[1] D. W. Carlton and M. Waldman, “The Hidden Costs of AI Regulation,” Harvard Journal of Law & Technology, vol. 33, no. 2, pp. 1–25, 2020.

[2] B. Mittelstadt, C. Russell, and S. Wachter, “Explaining explanations in AI,” Commun. ACM, vol. 65, no. 1, pp. 46–54, 2022.

[3] L. Edwards and M. Veale, “Slave to the Algorithm? Why a Right to an Explanation is Probably Not the Remedy You Are Looking For,” Duke Law & Technology Review, vol. 16, pp. 18–84, 2017.

[4] S. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2017, pp. 4765–4774.

[5] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why Should I Trust You? Explaining the Predictions of Any Classifier,” in Proc. ACM SIGKDD, 2016, pp. 1135–1144.

[6] W. Samek, T. Wiegand, and K.-R. Müller, “Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models,” IEEE Signal Process. Mag., vol. 36, no. 6, pp. 69–80, 2019.

[7] F. Doshi-Velez and B. Kim, “Towards a Rigorous Science of Interpretable Machine Learning,” arXiv preprint, arXiv:1702.08608, 2017.

[8] A. Barredo Arrieta et al., “Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI,” Inf. Fusion, vol. 58, pp. 82–115, 2020.

[9] S. Amershi et al., “Software Engineering for Machine Learning: A Case Study,” in Proc. IEEE/ACM Int. Conf. Software Engineering (ICSE), 2019, pp. 291–300.

[10] Jonnalagadda, A. K., Dutta, K. P., Ranjan, P., & Myakala, P. K. (2025, July). AI and Optimization: Transforming Data Engineering Applications. In *Recent Advances in Artificial Intelligence for Sustainable Development (RAISD 2025)* (pp. 686-702). Atlantis Press.

[11] R. Saha, D. Srivastava, and S. Sudarshan, “XAI Services in MLOps Pipelines: Opportunities and Research Challenges,” in Proc. VLDB Endowment, vol. 15, no. 12, pp. 3654–3665, 2022

[12] M. Sundararajan, A. Taly, and Q. Yan, “Axiomatic Attribution for Deep Networks,” in Proc. Int. Conf. Mach. Learn. (ICML), 2017, pp. 3319–3328.

[13] U.S. Food and Drug Administration, “Artificial Intelligence and Machine Learning in Software as a Medical Device,” FDA Discussion Paper, 2021. [Online]. Available: https://www.fda.gov/media/145022/download

[14] Somayajula, R., Raghavan, P., Chippagiri, S., & Ravula, P. (2025, May). Adaptive Fuzzy-Neural Architectures for Explainable Intrusion Detection in Big Data Environments. In 2025 Global Conference in Emerging Technology (GINOTECH) (pp. 1-7). IEEE.

[15] A. Holzinger, G. Langs, H. Denk, K. Zatloukal, and H. Müller, “Causability and Explainability of AI in Medicine,” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 9, no. 4, pp. 1–13, 2019.

[16] A. Arya, A. Bellamy, P.-Y. Chen, and D. Dhurandhar, “AI Explainability 360: An Extensible Toolkit for Understanding Data and Machine Learning Models,” J. Mach. Learn. Res., vol. 21, no. 130, pp. 1–6, 2020.

[17] R. Caruana et al., “Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-Day Readmission,” in Proc. ACM SIGKDD, 2015, pp. 1721–1730.

[18] Veluguri, S. P. (2025, January). Deep PPG: Improving Heart Rate Estimates with Activity Prediction. In 2025 1st International Conference on AIML-Applications for Engineering & Technology (ICAET) (pp. 1-6). IEEE.