FEDERAL RURAL UNIVERSITY OF PERNAMBUCO

POSTGRADUATE PROGRAM IN APPLIED INFORMATICS

ANDRÉ CARLOS SANTOS DE ASSIS

INTERPRETABILITY AND TRUST IN MACHINE LEARNING MODELS

RECIFE – PE

2025

ANDRÉ CARLOS SANTOS DE ASSIS

INTERPRETABILITY AND TRUST IN MACHINE LEARNING MODELS

Master's dissertation submitted to the Postgraduate Program in Applied Informatics at the Federal Rural University of Pernambuco as partial fulfillment of the requirements for the degree of Master in Applied Informatics.

ADVISOR: Prof. Dr. Ermeson Andrade

RECIFE – PE

2025

Dados Internacionais de Catalogação na Publicação
Sistema Integrado de Bibliotecas da UFRPE
Bibliotecário(a): Suely Manzi – CRB-4 809

A848i Assis, André Carlos Santos de.
Interpretability and trust in machine learning models / André Carlos Santos de Assis. – Recife, 2025.
86 f.

Orientador(a): Ermeson Carneiro de Andrade.

Dissertação (Mestrado) – Universidade Federal Rural de Pernambuco, Programa de Pós-Graduação em Informática Aplicada, Recife, BR-PE, 2026.

Inclui referências e apêndice(s).

1. Inteligência artificial . 2. Usabilidade de software. 3. Aprendizado de computador. 4. Ambiente virtual 5. Ciclo de vida de desenvolvimento de software. I. Andrade, Ermeson Carneiro de, orient. II. Título

CDD 004

To my family, my foundation and refuge, for
reminding me every day why it is worth
carrying on.

Acknowledgements

To God, my first and greatest gratitude. His presence sustained my uncertain days, renewed my strength, and illuminated my choices.

To my wife, Ana Beatriz, companion at all times. Your affection, patience, and constant encouragement made every stage of this journey possible. You celebrated my small victories and welcomed my stumbles, reminding me why we began.

To my parents, Gercilda and Aluizio, for sowing in me the value of study and perseverance. You taught me that knowledge is the path and character is destiny. The example of honest work, faith, and love I received at home was the foundation that supported me in the most difficult choices of this academic and personal journey.

To Professors Dr. Jamilson Dantas and Fumio Machida, my sincere gratitude for their collaboration on the work developed through constructive criticism and research partnerships.

To Professor Dr. Ermeson Andrade, my profound gratitude. Since my undergraduate studies, your guidance has gone beyond theory. You believed in my potential and reminded me, at every stage, of the value of what we built. In moments of doubt, you offered direction, serenity, and encouragement. Your presence was essential.

To God, all the honor and all the glory.

Now unto him that is able to do exceeding
abundantly above all that we ask or think,
according to the power that worketh in us.
(Bible — Ephesians 3:20)

Abstract

Machine learning systems increasingly support decisions in domains where transparency, reliability, and user trust are essential. As models become more complex, the tension between predictive performance and explainability becomes a critical concern for real-world adoption. Although several techniques provide localized or post-hoc explanations, the literature still lacks integrated approaches that evaluate how interpretability affects performance, support transparent model understanding, and strengthen confidence in intelligent environments. This dissertation addresses these gaps through three complementary investigations. First, it analyzes transparent and opaque models under different workloads, revealing practical trade-offs among accuracy, latency, and users' ability to understand model behavior. Second, it introduces the Interpretable Models Analysis Tool (IMAT), a visual analytics solution that represents the internal computation flow of Multilayer Perceptrons through flowcharts and structured textual artifacts, enhancing traceability and transparency in tasks such as sentiment analysis. Third, it applies explainability to software aging detection by integrating visual diagrams with natural-language explanations generated by Large Language Models, enabling engineers to examine decision paths, identify influential features, and increase confidence in maintenance decisions. The results show that interpretability can be incorporated into machine-learning workflows with controlled impact on performance. The main contributions include empirical criteria for selecting between transparent and opaque models, a visual tool for interpretable MLP analysis, and an explainability-based approach that strengthens transparency and trust in predictive maintenance and intelligent environments.

Keywords: Explainable AI, Interpretability, Machine Learning, Intelligent Environments, Software Aging

List of symbols

AI	*Artificial Intelligence*
CNN	*Convolutional Neural Network*
CPU	*Central Processing Unit*
DBMS	*Database Management System*
EU	*European Union*
I/O	*Input/Output*
IMAT	*Interpretable Models Analysis Tool*
ISAAT	*Interpretable Software Aging Analysis Tool*
LIME	*Local Interpretable Model-agnostic Explanations*
LLM	*Large Language Model*
ML	*Machine Learning*
MLOps	*Machine Learning Operations*
MLP	*Multilayer Perceptron*
MNIST	*Modified National Institute of Standards and Technology*
RAM	*Random Access Memory*
SHAP	*SHapley Additive exPlanations*
SVM	*Support Vector Machine*
XAI	*Explainable Artificial Intelligence*

Contents

1 Introduction

Interpretability in Machine Learning (ML) models has become a central topic in the field of Artificial Intelligence (AI), as increasingly complex algorithms begin to influence decisions in different domains such as healthcare, finance, engineering, and transportation. The growth of computational power and advances in modeling techniques have enabled the development of highly accurate solutions that, due to their complexity, operate as "black boxes". This lack of transparency raises concerns about accountability, bias, and reliability, demanding methods that make it possible to understand the internal reasoning of the models (MOLNAR, 2020). The field of Explainable Artificial Intelligence (XAI) thus emerges with the purpose of balancing performance and transparency, enabling humans to understand, validate, and trust automated systems (MILLER, 2019).

Interpretability concerns the extent to which a human can understand a model's internal mechanisms or anticipate how changes in inputs affect outputs, whereas explainability focuses on how these mechanisms and decisions are communicated to a specific audience (MARCINKEVIČS; VOGT, 2023). In this sense, explainability acts as a bridge between complex machine learning models and human understanding, providing contextualized, user-oriented justifications that support validation, accountability, and trust (MILLER, 2019). Although the literature distinguishes these concepts by emphasizing structural transparency in interpretability and user-facing communication in explainability (LIAO; VARSHNEY, 2021), this work adopts a pragmatic and applied perspective in which both terms are treated in a closely aligned manner, consistent with common practice in applied machine learning research and with the articles that support this dissertation.

The need for explainability becomes concrete in high impact decisions where errors, bias, or misuse create measurable harm. Examples include clinical decision support, credit underwriting and loan pricing, fraud detection, hiring and performance assessment, and predictive maintenance in critical infrastructure, where false alarms increase costs and missed detections increase risk. In these settings, explanations support auditing, incident analysis, accountability to affected stakeholders, and safer human oversight (SARTOR et al., 2020). This practical demand is reinforced by regulation and governance frameworks. For example the European Union (EU) AI Act establishes transparency and information duties for certain AI systems and imposes specific obligations for high risk systems,

including documentation and information to enable appropriate use and oversight. EU data protection practice links automated decision making to information duties on profiling and automated processing, increasing pressure for traceable and communicable model behavior in sensitive contexts (LOGNOUL, 2025).

ML models can generally be classified as transparent or opaque, according to their capacity for explanation. Transparent models, such as linear regressions and decision trees, are inherently interpretable, making it easy to trace how input variables influence outputs. Opaque models, such as deep neural networks, support vector machines (SVM), and random forests, achieve more accurate results but sacrifice clarity regarding their internal functioning (ANGELOV et al., 2021). This distinction gives rise to the so-called trade-off between performance and interpretability, in which the pursuit of higher accuracy tends to reduce decision transparency. In critical applications, model explanation is just as important as accuracy, since the lack of understanding about how a decision was made can make its adoption unfeasible, even if the results are statistically correct (CHADDAD et al., 2023).

Beyond promoting understanding, interpretability is also an essential component of trust in intelligent systems. The literature shows that clear explanations increase user acceptance and facilitate validation by domain experts (ROSENBACKE et al., 2024). However, superficial or inconsistent explanations can have the opposite effect, reducing the system's credibility (CHEUNG; HO, 2025). For this reason, different approaches have been explored to integrate explainability and performance. Among them, post-hoc methods stand out, such as Local Interpretable Model-Agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP), which offer localized justifications for specific predictions (RIBEIRO et al., 2016; LUNDBERG; LEE, 2017). Although useful, these methods do not fully address the trust and transparency challenge, as they explain individual decisions without exposing the model's global behavior.

Recent advances in the area have resulted in a diversified ecosystem of explainability techniques and tools, with different levels of maturity and application. Studies such as those by LIAO; VARSHNEY and ARYA et al. indicate that user-centered approaches, which combine interactive visualization and natural language explanations, increase the understanding and acceptance of AI systems. At the same time, industrial research indicates that although most organizations recognize the importance of explainability,

few have concrete strategies for incorporating it into their processes (GIOVINE et al., 2024). These findings show that the current challenge is not limited to the development of interpretable methods, but to their effective integration into the design, operation, and auditing practices of intelligent systems.

The investigations in this dissertation presented address interpretability and trust from three complementary perspectives. The first discusses the balance between performance and transparency, comparing models with different levels of complexity in intelligent environments (ASSIS et al., 2025b). The second proposes a tool focused on the analysis and visualization of interpretable models, capable of representing the internal flow of computation and facilitating the understanding of their decisions (ASSIS et al., 2025a). The third applies XAI principles to the context of software engineering, using visual explanations and natural language to support the detection and prevention of software aging in computational systems (ASSIS et al., 2025c).

This dissertation contributes to this area by bringing together experimental analyses, tools, and applications that demonstrate the feasibility of aligning interpretability and performance in machine learning models. The results obtained reinforce that transparency should not be treated as a technical obstacle, but as a fundamental attribute for the reliability and adoption of AI-based solutions. Thus, this dissertation proposes a reflection on how explainability principles can be incorporated in a practical and continuous way into the development cycle of intelligent systems, promoting the responsible and ethical use of machine learning technologies in different contexts.

1.1 Objectives

This work aims to investigate how interpretability and trust can be fostered in machine learning models, especially in contexts where performance and transparency need to coexist. The specific objectives are:

1. Evaluate how different levels of interpretability (transparent models and opaque models) influence performance metrics such as accuracy, response time, and understanding of results in machine learning environments.
2. Propose a visualization and explanation tool for interpretable models that allows users to understand decisions, compare alternative models, and support the auditing

as well as the supervision of machine learning models.

3. Apply the concepts of explainability in a real software engineering scenario, investigating how visual explanations and natural language can assist in detecting software aging and in promoting engineers trust in the adoption of predictive maintenance models.

1.2 Structure of the Document

Section 2 introduces the articles that support this dissertation, highlighting their main contributions and results. Section 3 concludes the work by presenting the contributions, limitations, and potential directions for future research.

2 Articles that Support this Thesis

This chapter presents the works developed and published to support the main objective of this dissertation. All articles form a joint investigation into interpretability, explainability, and trust in machine learning models.

2.1 Articles Compiled in this Thesis

This section summarizes the main objectives and contributions of each work compiled in this thesis. Each subsection refers to the journal or conference where the corresponding work was published.

2.1.1 Journal of Reliable Intelligent Environments

Assis et al. «The performance-interpretability trade-off: A comparative study of machine learning models». In *Journal of Reliable Intelligent Environments*, v. 11, n. 1, p. 1, 2025. DOI: 10.1007/s40860-024-00240-0.

This article comparatively investigates models with different degrees of transparency, contrasting inherently interpretable approaches (for example, logistic regression and decision trees) with opaque models (for example, SVM, neural networks, and random forests). The motivation comes from the debate on the trade-off between performance and interpretability and its effects on trust and adoption in real-world problems. The study approaches this problem using metrics of accuracy, response time, and explainability, discussing its practical relevance in intelligent environments. The experiments were conducted with varied workloads and datasets, in order to observe the behavior of the algorithms under different demands. The results indicated that opaque models achieve higher accuracy, while transparent models exhibited lower latency and better explanatory capability. The analysis emphasizes that performance gains do not eliminate the need for human understanding, especially when automated decisions affect sensitive processes. Therefore, the choice of model must balance predictive metrics with transparency requirements.

The analysis emphasizes that performance gains do not eliminate the need for

human understanding, especially when automated decisions affect sensitive processes. As an implication, the study reinforces that model evaluation should consider more than average accuracy, incorporating response time, interpretability, and alignment with the context of use. In operational terms, it is advisable to prioritize interpretable models when decision traceability, rule inspection, and interaction with domain experts are required. When the use of more complex models becomes necessary, their adoption should be accompanied by mechanisms that support understanding and validation in a manner appropriate to the application context, even though the present study does not explore such mechanisms directly.

2.1.2 Brazilian Workshop on Social Network Analysis and Mining (BraSNAM)

Assis et al. «IMAT: A Tool for Analyzing Interpretable Machine Learning Models». In: *Brazilian Workshop on Social Network Analysis and Mining (BraSNAM)*, SBC, 2025, p. 134–147. DOI: 10.5753/brasnam.2025.8924.

This work presents the IMAT, a tool aimed at the analysis of interpretable models that uses visualizations of the decision flow. The proposal is aligned with the advancement of user-centered approaches, in which interfaces and visual artifacts mediate the understanding of the model's internal process and qualify the interaction with domain experts, with the main objective of making more accessible the visualization of the chain of calculations and the tracing of attribute contributions in models employed in real applications. IMAT's architecture integrates the generation of flowcharts to elucidate operations in internal layers, as well as structured textual summaries, offering users elements for auditing and decision support. In an applied study of sentiment analysis on tweets with the MultiLayer Perceptron (MLP) model, the tool made it possible to observe data transformations, activations per layer, and justifications for predictions, reducing uncertainties about the model's behavior. This perspective complements widely used post-hoc methods (for example, LIME and SHAP), adding a holistic view of the processing flow and of the artifacts that support the decision.

The results suggest that interactive visualization can help improve the comprehensibility of and trust in model operation on the part of non-expert users, in addition to promoting continuous auditing practices. The main contribution lies in bringing

XAI theory closer to everyday, traceable use, in which the operator can relate inputs, internal transformations, and outputs in a transparent way.

2.1.3 Workshop on Software Aging and Rejuvenation (WoSAR)

Assis et al. «Exploring Explainability in Machine Learning Models for Software Aging Detection». In: *Workshop on Software Aging and Rejuvenation (WoSAR)*, 2025.

This study applies principles of interpretability and explanation to the domain of software aging, a phenomenon in which long-running systems tend toward performance degradation and failures. The problem is particularly sensitive in critical environments, where predictions must be accompanied by tangible justifications to guide maintenance interventions. The Interpretable Software Aging Analysis Tool (ISAAT) proposed in the paper aims to support software aging detection by generating explanatory artifacts that reveal how the MLP-based model produces its predictions. The approach combines structured records collected during training and inference, including weights, biases, and layer activations, with visualization mechanisms that convert these records into flowcharts. These diagrams allow users to navigate from a high-level view of the processing pipeline to detailed operations within each layer.

In parallel, ISAAT produces natural-language summaries derived from the same technical evidence, ensuring that the explanations remain consistent with the underlying computations. By making the decision paths, activations, and internal transformations explicit, the tool enhances operational reliability and provides actionable information for detecting aging trends and planning rejuvenation strategies. As a contribution, the study demonstrates that structured visualizations combined with domain-oriented textual explanations can support engineers in assessing risks, interpreting model behavior, and monitoring aging indicators in a continuous and transparent manner.

2.2 Authorship Statement

The author of this dissertation is the principal author of the works presented in this chapter. A detailed description of the author's contribution to each work is provided in the corresponding appendices. None of the works included in this compendium have

been submitted as part of any other dissertation compendium, nor will they be included in such compendiums in the future.

3 Final Considerations

This dissertation examined how interpretability and trust can be incorporated into machine learning models through comparative analysis, tool development, and application in operational contexts. By bringing together three independent but complementary studies, the work reinforces that transparency is not only a desirable attribute in intelligent systems but also an operational requirement in scenarios where decisions must be validated, justified, and monitored.

The comparative experiments demonstrated that opaque models tend to achieve higher accuracy, while interpretable models provide more predictable latency, clearer decision processes, and greater acceptance by domain specialists. These findings indicate that the choice between transparent and opaque approaches should consider not only predictive performance but also response time, auditability, and the specific demands of the application environment.

The IMAT tool contributed by offering visual and textual artifacts capable of clarifying how an MLP processes inputs and reaches decisions. This form of structured interpretability supports failure diagnosis, communication among technical teams, and more informed model selection. Likewise, the ISAAT study showed that transparency can strengthen reliability in software aging detection by exposing how computation flows through the model and producing traceable explanations aligned with the monitored indicators.

The results demonstrate that integrating interpretability into machine learning systems is feasible without compromising their practical usefulness. They also highlight that explanations, when grounded in the internal evidence of the model, can support decision-making in technical environments, reduce uncertainty, and improve governance over automated processes. The contributions presented in this dissertation offer concrete steps toward building machine learning solutions that are not only accurate but also justifiable and aligned with the needs of real-world operation.

3.1 Contributions

This work contributes to the field of interpretability and trust in machine learning models, with a focus on responsible adoption in different scenarios. The main contributions include:

1. **Comparative analysis between transparent and opaque models:** We provide evidence, through controlled experiments, of how ML models behave in terms of accuracy, response time, and explainability, presenting the practical trade-offs involved in model choice in intelligent environments.
2. **Tool for interpretable analysis:** We propose a tool that integrates visualizations of the internal decision flow, summaries in natural language, and interpretability metrics, supporting auditing, traceability, and communication between technical teams and domain experts.
3. **Integration of visualizations and LLM-mediated explanations:** We demonstrate the combination of MLP flowcharts with natural language explanations generated by language models to increase decision transparency, support audits, and facilitate the adoption of explainable solutions in operational contexts.
4. **Application in different domains:** How visual and textual explanations can support software aging detection and sentiment analysis, generating verifiable artifacts for decision support and prioritization of interventions.

3.2 Limitations

This section discusses limitations directly observed in the articles and experiments that compose this work.

1. **Data diversity**: The comparative evaluation between transparent and opaque models considered a limited number of datasets and domains (for example, MNIST for images and a news dataset for text), which restricts generalization to other types of data, multi-label tasks, imbalanced scenarios, and environments with high noise. In the software aging study, the focus was on a single Database Management System (DBMS) service and on one primary software aging indicator, Random Access Memory (RAM), which reduces the scope of operational variables considered.
2. **Model coverage and complexity of explanations**: The proposed tools (IMAT

and ISAAT) mainly support MLP, which limits applicability to other algorithms, for example, CNNs (Convolutional Neural Networks) and Random Forest. Moreover, the generated flowcharts tend to grow in complexity as the architecture increases, which may hinder readability for non-technical users and demand additional summarization mechanisms.

3. **Validation with users and LLM-mediated explanations**: The evaluation of the usefulness of explanations was centered on controlled scenarios and audiences with technical familiarity. Longitudinal studies with diverse user profiles are lacking, measuring impact on trust, diagnosis, auditing, and decision-making. In the case of using large language models to generate explanatory summaries, the risk of inaccuracies or omissions remains, requiring human review and traceability to the underlying technical artifacts.

3.3 Future Work

The results presented provide opportunities to deepen the investigation of interpretability and trust in machine learning models in operational contexts. One direction is to broaden the diversity of data, tasks, and domains, incorporating multi-label scenarios and imbalanced datasets, as well as evaluating other popular models such as Random Forest and CNN. It is also relevant to integrate complementary explanation approaches (counterfactuals, partial dependence, permutation importance, local and global decompositions) and to assess their stability under controlled perturbations. A further opportunity for future investigation is to extend the experiments to include operational metrics (such as cost of explanation generation, variability under peaks, energy consumption, and robustness to noise) in order to bring the evaluation closer to real production conditions.

A further research direction is to establish a precise working definition of explainability for the proposed artifacts and to formalize how it should be measured at different levels. Following the evaluation taxonomy that distinguishes functionally grounded, human grounded, and application grounded assessments, future studies can combine proxy metrics such as faithfulness and robustness with controlled user tasks that quantify understanding, and with end to end validation in realistic operational settings.

Building on these evaluation principles, a next step involves conducting usability studies with participants from diverse backgrounds in order to assess how different forms of explanation affect trust, diagnosis time, and decision quality, thereby informing practical guidelines for adoption.

Another line of advancement involves the evolution of the proposed tools. This includes incorporating mechanisms for hierarchical summarization of extensive flowcharts, investigating explanation strategies integrated with MLOps (Machine Learning Operations) pipelines for continuous auditing, and defining quality indicators for the explanations generated. Longitudinal usability studies with audiences of different backgrounds can quantify impacts on trust, diagnosis time, and decision quality, providing evidence for practical adoption guidelines.

Finally, in the context of sentiment analysis on tweets, it is important to improve the handling of context, multilingual content, irony, and emojis, as these factors influence both prediction quality and the reliability of explanations. For software aging, future work may extend the study to distributed or multi-service environments and incorporate additional operational indicators, such as CPU usage, I/O activity, or network latency. Broader datasets and real-world logs may also strengthen the evaluation of both detectors and explanation strategies. These directions can enhance the integration of interpretability with continuous monitoring and operational decision-making.

Bibliography

ANGELOV, P. P.; SOARES, E. A.; JIANG, R.; ARNOLD, N. I.; ATKINSON, P. M. Explainable artificial intelligence: an analytical review. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, Wiley Online Library, v. 11, n. 5, p. e1424, 2021.

ARYA, V.; BELLAMY, R. K.; CHEN, P.-Y.; DHURANDHAR, A.; HIND, M.; HOFFMAN, S. C.; HOUDE, S.; LIAO, Q. V.; LUSS, R.; MOJSILOVIĆ, A. et al. Ai explainability 360: Impact and design. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2022. v. 36, n. 11, p. 12651–12657.

ASSIS, A.; DANTAS, J.; ANDRADE, E. Imat: Uma ferramenta para análise de modelos de aprendizado de máquina interpretáveis. In: SBC. **Brazilian Workshop on Social Network Analysis and Mining (BraSNAM)**. [S.l.], 2025. p. 134–147.

_____. The performance-interpretability trade-off: A comparative study of machine learning models. **Journal of Reliable Intelligent Environments**, Springer, v. 11, n. 1, p. 1, 2025.

ASSIS, A.; MACHIDA, F.; ANDRADE, E. Exploring explainability in machine learning models for software aging detection. In: **Workshop on Software Aging and Rejuvenation (WoSAR)**. [S.l.: s.n.], 2025.

CHADDAD, A.; PENG, J.; XU, J.; BOURIDANE, A. Survey of explainable ai techniques in healthcare. **Sensors**, MDPI, v. 23, n. 2, p. 634, 2023.

CHEUNG, J. C.; HO, S. S. The effectiveness of explainable ai on human factors in trust models. **Scientific Reports**, Nature Publishing Group UK London, v. 15, n. 1, p. 23337, 2025.

GIOVINE, C.; ROBERTS, R.; POMETTI, M.; BANKHWAL, M. Building ai trust: The key role of explainability. **URL https://www. mckinsey. com/capabilities/quantumblack/our-insights/building-ai-trust-the-key-role-of-explainability**, 2024.

LIAO, Q. V.; VARSHNEY, K. R. Human-centered explainable ai (xai): From algorithms to user experiences. **arXiv preprint arXiv:2110.10790**, 2021.

LOGNOUL, M. Regulation (eu) 2024/1689 laying down harmonised rules on artificial intelligence (artificial intelligence act–ai act). **Revue du Droit des Technologies de l'information**, Larcier, n. 3-4, p. 145–189, 2025.

LUNDBERG, S. M.; LEE, S.-I. A unified approach to interpreting model predictions. **Advances in neural information processing systems**, v. 30, 2017.

MARCINKEVIČS, R.; VOGT, J. E. Interpretable and explainable machine learning: A methods-centric overview with concrete examples. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, Wiley Online Library, v. 13, n. 3, p. e1493, 2023.

MILLER, T. Explanation in artificial intelligence: Insights from the social sciences. **Artificial intelligence**, Elsevier, v. 267, p. 1–38, 2019.

MOLNAR, C. **Interpretable machine learning**. s.l.: Lulu. com, 2020.

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. " why should i trust you?" explaining the predictions of any classifier. In: **Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining**. [S.l.: s.n.], 2016. p. 1135–1144.

ROSENBACKE, R.; MELHUS, Å.; MCKEE, M.; STUCKLER, D. How explainable artificial intelligence can increase or decrease clinicians' trust in ai applications in health care: systematic review. **Jmir Ai**, JMIR Publications Toronto, Canada, v. 3, p. e53207, 2024.

SARTOR, G.; LAGIOIA, F. et al. The impact of the general data protection regulation (gdpr) on artificial intelligence. European Parliament, 2020.

APÊNDICES

APPENDIX A – The performance-interpretability trade-off: A comparative study of machine learning models

Assis et al. «The performance-interpretability trade-off: A comparative study of machine learning models». In *Journal of Reliable Intelligent Environments*, v. 11, n. 1, p. 1, 2025. DOI: 10.1007/s40860-024-00240-0.

Author's contribution: The author was responsible for developing methodology, experiment setup and execution, data processing, resource generation (including tables, charts, and images), and writing the original draft.

The Performance-Interpretability Trade-off: A Comparative Study of Machine Learning Models

André Assis[1†], Jamilson Dantas[2†], Ermeson Andrade[1†]

[1]Department of Computing, Federal Rural University of Pernambuco, Recife, Pernambuco, Brazil.
[2]Computer Science Center, Federal University of Pernambuco, Recife, Pernambuco, Brazil.

Contributing authors: andre.csassis@ufrpe.br; jrd@cin.ufpe.br; ermeson.andrade@ufrpe.br;
[†]These authors contributed equally to this work.

Abstract

Machine learning models are increasingly being integrated into various aspects of society, impacting decision-making processes across domains such as healthcare, finance, and autonomous systems. However, as these models become more complex, concerns about their transparency and interpretability have emerged. Transparent models, which provide detailed and understandable explanations, stand in contrast to opaque models, which often achieve higher accuracy but lack interpretability. This study presents a comparative analysis, examining the performance and explainability of transparent models (K-Nearest Neighbors (KNN), Decision Trees, and Logistic Regression) and opaque models (Convolutional Neural Networks (CNN), Random Forests, and Support Vector Machines (SVM)) in an intelligent environment. Our experimental evaluation explores the balance between performance (accuracy and response time) and explainability, a crucial aspect for the effective deployment of Artificial Intelligence (AI) in smart systems. Our results indicate that opaque models such as CNN, SVM, and Random Forest achieved higher accuracy (up to 98% on MNIST and 95% on Fake and Real News) compared to transparent models (up to 94% on MNIST and 92% on Fake and Real News). However, transparent models exhibited faster response times and greater interpretability, especially under high workload conditions, highlighting the trade-off between performance and interpretability.

Keywords: Machine Learning, Transparency, Interpretability, Model Comparison.

1 Introduction

Explainable Artificial Intelligence (XAI) emerges as a crucial response to the challenges of transparency and trust faced by systems utilizing Artificial Intelligence (AI). As AI systems advance in capability and complexity, it becomes increasingly imperative to develop methods that not only enable users to understand and validate the decisions made, but also ensure accountability and ethical use of AI technologies [1]. XAI aims to balance performance (such as assertiveness and response time) with explainability, ensuring that processes not only operate effectively but also do so in a way that is comprehensible to those who use them. This field is gaining increasing attention from researchers, policymakers, and industry leaders as the need for transparent algorithms becomes more evident, driving researchers and developers to seek solutions that satisfy both performance and explainability in AI applications [2].

The importance of XAI is particularly critical in domains where automated decisions can have significant implications, such as healthcare, finance, and legal sectors. In these areas, the ability to explain and validate AI system decisions is essential to ensure user adoption and technological acceptance, as users often hesitate to trust systems they do not fully understand [3]. Transparency, therefore, is not just about regulatory compliance but is a fundamental element in building lasting trust between AI technology and its users, ensuring that automated decisions are fair, accountable, and acceptable to all involved [4]. Moreover, achieving a delicate balance between transparency and performance is crucial. While transparency may offer understandable explanations, AI system must also perform optimally to meet real-world demands. However, achieving this balance can be challenging, as the need for clear explanations may conflict with the need for high performance. Thus, finding the right equilibrium is crucial to ensure that AI systems not only make informed decisions but also do so efficiently and effectively.

AI algorithms, particularly in machine learning, can be broadly classified into two categories based on their explainability: transparent and opaque[1]. Transparent algorithms, such as KNN, Decision Trees, and Logistic Regression, are recognized for their simpler structures and ease of interpretation, making them accessible to end users. On the other hand, opaque algorithms, including CNN, Random Forests, and SVM, are known for their ability to achieve high accuracy in complex tasks, yet they often come with internal complexities that obscure decision-making processes. However, as is the case with all aspects of life, each of these algorithms possesses its own unique set of advantages and disadvantages in terms of transparency, accuracy, and performance, making them suitable for certain applications while unsuitable for others [5]. For example, transparent algorithms like KNN and Decision Trees are well-suited for tasks where interpretability is crucial, such as credit risk assessment in finance, while opaque algorithms like CNN and SVM demonstrate exceptional performance in tasks where accuracy is crucial, such as facial recognition in security systems.

Currently, there is a wide variety of studies in the literature that focus on XAI. For instance, [6] investigated the essential role of XAI in promoting trust via transparency, highlighting various techniques aimed at enabling stakeholders to comprehend and

[1] In this context, the terms *AI algorithms* and *AI models* are used interchangeably to refer to computational systems trained on data to perform specific tasks in the field of artificial intelligence.

trust AI applications. In [7], the authors focused on the growing demand for accountability and transparency in the medical sector, particularly addressing challenges presented by the opaque nature of deep learning systems. They categorize various approaches to interpretability, highlighting its significance in the context of medical AI. In [8], the diverse needs and expectations of stakeholders in XAI are discussed. The study provides a model for evaluating and developing explainability approaches that cater to these needs across different domains. However, these studies often narrow their focus to specific techniques or algorithms, such as interpretability methods or model-agnostic approaches, and often do not conduct a comprehensive analysis of the trade-offs among response time, accuracy, and explainability.

This work explores the trade-offs between response time, accuracy, and explainability in intelligent environments, which require models that are both high-performing and interpretable for users. We aim to understand how machine learning models can be responsibly developed to meet the unique demands of these environments. To this end, we compared transparent algorithms such as Decision Trees, Logistic Regression, and KNN with opaque algorithms like Random Forests, SVM, and CNN. These comparisons were conducted using two datasets: MNIST for image classification and a fake news detection dataset for text classification, followed by statistical analysis to identify significant differences among the groups. The results showed that opaque algorithms achieved higher accuracy due to their ability to handle complex data patterns, but at the cost of longer response times under heavy workloads, reflecting a trade-off with computational efficiency. In contrast, transparent models, though less accurate, provided faster responses and greater interpretability, making them more suitable for applications requiring quick decision-making and transparency. This study offers insights to guide the selection of algorithms that align with both operational and ethical needs, contributing to the development of more reliable and ethically sound AI systems across various contexts.

The remainder of this work is organized as follows: Section 2 addresses the theoretical foundation, exploring XAI and the machine learning algorithms used. Section 3 presents the related work. Section 4 details the configuration of the experimental environment used in the tests. Section 5 discusses the results. Finally, Section 6 presents the conclusions and briefly introduces directions for future research.

2 Background

This section provides an overview of XAI, along with the adopted algorithms. We also explain each of these algorithms in the context of the MNIST dataset.

2.1 XAI

XAI aims to create AI systems that humans can understand [9]. While there is no universally agreed-upon mathematical definition for AI explainability or interpretability, Miller suggests that interpretability is about "the degree to which an individual can comprehend the reasoning behind a decision" [10]. Kim *et al.*, in the context of machine learning, define it as "the ability of the outcomes of a model to be predictable by a human" [11]. Essentially, the easier it is for someone to understand how a model

operates and the steps it takes to arrive at its conclusions, the more explainable the model is considered. A model is considered more explainable than another if it offers a clearer understanding of its decision-making process in comparison to the other model.

From another perspective, explainability is defined as the "ability to explain something in terms that are understandable to humans" [12]. In the context of machine learning, Molnar describes interpretable machine learning as using methods and models that enable humans to understand the operations and predictions of these systems [13]. Therefore, explainability is closely related to humans' capacity to comprehend and express the logic behind an algorithm's operations. This leads to the categorization of machine learning models based on their interpretability, which is defined as the ease with which a person can understand the rationale behind a decision or replicate the actions of the model [10].

Models are typically classified into two categories: transparent and opaque. Transparent models have straightforward, easily understandable structures, while opaque models have intricate, "black-box" structures, making them harder to interpret [14]. In this study, we specifically chose transparent models like Decision Trees, Logistic Regression, and KNN because of their simplicity and interpretability. On the other hand, we used opaque models such as Random Forests, SVM, and CNN because they can capture complex patterns, even though they are less interpretable.

2.2 Decision Tree

Decision Tree is a supervised machine learning algorithm applicable to both classification and regression tasks. This means it is capable of predicting both discrete categorical outcomes, like "yes" or "no", and continuous numerical values [15]. Structurally, a Decision Tree consists of a hierarchy of nodes interconnected in a tree-like format. The topmost node, known as the "root node," initiates the decision-making process. The subsequent nodes, termed "branches," represent potential decisions, leading to the "leaf nodes" at the end of each path, where the final outcomes are determined [16]. This hierarchy operates through conditional control statements, with branches delineating decision paths and leaf nodes holding either class labels (in classification tasks) or continuous values (in regression). This model is known for its transparency and interpretability, offering a straightforward understanding of predictive decisions based on input features. However, Decision Trees can suffer from overfitting to the training data, potentially impacting performance on new, unseen data.

The construction of a decision tree involves the sequential selection of attributes to partition the data, seeking to maximize the homogeneity of the classes in each leaf node. This selection is based on impurity metrics, such as entropy and the Gini index, which quantify the heterogeneity of the classes in a dataset. Entropy, defined as $H(S) = -\sum_{i=1}^{C} p_i \log_2 p_i$, where S is the set of samples, C is the number of classes and p_i is the proportion of samples of class i in S, measures the uncertainty associated with the distribution of the classes. The Gini index, defined as $Gini(S) = 1 - \sum_{i=1}^{C} p_i^2$, measures the probability of misclassifying a random sample from the dataset. Algorithms such as ID3 [17] and C4.5 [18] use information gain, which is the reduction in

entropy after splitting the data by an attribute, to select the attribute that best separates the classes. The CART algorithm [19], on the other hand, uses the Gini index as a splitting criterion.

In the MNIST dataset, a decision tree begins by examining all the pixels of all the images and selects the pixel that, at that moment, provides the best separation between the classes of digits based on a purity criterion (such as entropy or the Gini index). This process divides the dataset into two subsets, which are then repeatedly split by new questions about the pixel values, following the logic of maximizing the purity of the classification in each subset. This recursive procedure continues until the leaves of the tree represent a digit class with sufficient confidence or until predefined stopping criteria are met, such as a minimum number of images in a node. The sequential and binary nature of the decisions made at each node of the tree allows the classification process to be easily interpreted and visualized. Each decision is based on clear and objective criteria, allowing complete traceability of the decision path for each classification. The ability to describe exactly why an image was classified in a certain way, following the specific rules learned, makes decision trees highly transparent and interpretable.

2.3 Logistic Regression

Logistic Regression is a statistical model and a type of supervised machine learning algorithm primarily used to estimate the likelihood of a specific event's occurrence. This model not only generates precise predictions but also clarifies the relationships between different variables. Similar to Decision Trees, Logistic Regression is employed to predict discrete categorical outcomes. It aims to uncover the connections between independent (predictor) and dependent (outcome) variables, building a model based on these relationships. This enables Logistic Regression to function as a classifier by predicting the state of the dependent variable using the values of the independent variables [20]. While the dependent variables can be either continuous or categorical, categorical variables must be binarized or transformed into a dichotomous format for the model's application [21]. To ensure transparency, Logistic Regression relies on a linear relationship between input variables and outcomes. This approach enables users to interpret how each predictor affects the model's predictions.

Mathematically, logistic regression models the probability of a sample belonging to the positive class ($y = 1$) using the sigmoid function:

$$P(y = 1|x) = \frac{1}{1 + e^{-(\beta_0+\beta_1 x_1+...+\beta_p x_p)}}, \tag{1}$$

where $x = (x_1, ..., x_p)$ represents the vector of predictor variables, β_0 is the intercept, and $\beta_1, ..., \beta_p$ are the regression coefficients. The sigmoid function maps any real value to the interval [0, 1], ensuring that the output of the model can be interpreted as a probability. The regression coefficients are estimated from the training data using the maximum likelihood method, which seeks to find the parameter values that maximize the probability of observing the data. The interpretation of the coefficients is facilitated by the use of the odds ratio concept, which represents the change in the odds (ratio

of the probability of success to the probability of failure) associated with a one-unit increase in the corresponding predictor variable [22].

To classify the digits from the MNIST dataset, logistic regression calculates a weighted sum of the values of all pixels in an image, where each weight reflects the importance of a specific pixel in distinguishing between the digits. This weighted sum is then passed through a sigmoid function, which maps the result to a range between 0 and 1, interpreted as the probability of the image belonging to a specific class. The model is trained by adjusting these weights to maximize the accuracy of classifications on the training set, using methods like gradient descent. In essence, logistic regression assigns weights to each pixel of the images and uses these weights to calculate the probability of the image belonging to each digit class. The digit with the highest calculated probability is selected as the final classification. Despite its mathematically simplicity, logistic regression facilitates a reasonable interpretation of how the weights assigned to each pixel impact the probability of each class. The linearity of decisions and the direct relationship between input features and output facilitate an understanding and explanation of why an image was classified into a specific category, although the interpretation may become more complex in scenarios with higher-dimensional data or intricate decision boundaries.

2.4 K-Nearest Neighbors

KNN is a supervised machine learning method widely used for solving classification and regression problems. The main characteristic of KNN is its simplicity, where the classification of a data point is determined by the majority votes of its nearest neighbors, with the number "k" representing the number of neighbors considered. This model is based on the premise that similar data points exist in proximity. The performance of KNN can be significantly influenced by the choice of the number of neighbors "k", as well as by the distance metric used to measure the proximity between data points. Despite its simplicity, KNN has been effective across a variety of datasets, especially in applications where the relationship between data attributes is not easily captured by more complex mathematical models [23]. It is often considered an explainable and transparent algorithm due to its simplicity and the intuitive method of classifying observations based on the majority votes of the nearest neighbors, facilitating the understanding and explanation of its decisions [24].

Formally, the KNN algorithm can be defined as follows: given a training set $T = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, where $x_i \in \mathbb{R}^d$ represents the feature vector of the i-th sample and y_i its respective class, and a new sample $x \in \mathbb{R}^d$ to be classified, the KNN algorithm assigns to x the most frequent class among its k nearest neighbors in T [25]. The proximity between samples is determined by a distance metric, such as the Euclidean distance:

$$d(x, x_i) = \sqrt{\sum_{j=1}^{d} (x_j - x_{ij})^2}, \tag{2}$$

where x_j and x_{ij} represent the j-th feature of samples x and x_i, respectively. The value of k directly influences the performance of the classifier. A small value of k can

lead to overfitting, capturing noise and particularities of the training set, while a large value of k can lead to underfitting, oversimplifying the decision boundary.

In classifying images from the MNIST dataset, the KNN model operates as follows: for each test image, the algorithm calculates the distance (often Euclidean) between that image and all the images in the training set. Then, it identifies the "K" closest digits (neighbors) and conducts a majority vote among these "K" neighbors to determine the class of the test image. The effectiveness of KNN depends on the choice of the "K" value and the distance metric used. Generally, KNN identifies the "K" closest (or most similar) digits to a test image based on a measure of distance and classifies the image based on the most common class among these nearby neighbors. The decision-making process of KNN is intuitively understandable because it is based on the premise that similar images tend to belong to the same class. However, the direct reliance on neighboring data points for classification can make the interpretation of decisions more challenging, particularly in scenarios with higher-dimensional data or when the optimal choice of "K" is unclear.

2.5 Random Forest

Random Forest is a supervised machine learning algorithm similar to Decision Trees, commonly used for prediction tasks. This method involves creating an ensemble of Decision Trees, each generated randomly, and synthesizing their outcomes to improve prediction accuracy. Unlike a single Decision Tree, which aims to create a comprehensive model from the entire dataset, Random Forest constructs multiple trees, each from a randomly selected subset of attributes from the full dataset [26]. This technique, is applicable to both classification and regression scenarios. Random Forests are known for their high accuracy in predicting outcomes based on input data. However, their complex and stochastic nature often complicates the interpretability of their results, which affects their overall explainability.

A random forest can be formally defined as an ensemble of B decision trees $\{T_b(x)\}_{b=1}^{B}$, where $x \in \mathbb{R}^p$ represents the feature vector. Each tree $T_b(x)$ is built from a bootstrap sample S_b of the training data and a random subset of m features, selected from the p original features [27]. The prediction of the random forest for a new sample x is given by the aggregation of the predictions of the individual trees:

$$\hat{y}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x) \tag{3}$$

for regression problems, and

$$\hat{y}(x) = \arg\max_{c} \sum_{b=1}^{B} \mathbb{I}(T_b(x) = c) \tag{4}$$

for classification problems, where $\mathbb{I}(A)$ is the indicator function that returns 1 if the condition A is true and 0 otherwise. The randomness in the construction of the trees,

both in the selection of the samples and in the selection of the features, contributes to the diversity of the set of trees and, consequently, to the robustness of the model.

In the MNIST dataset, a Random Forest combines many decision trees, each trained on a random subset of the training data, with the goal of reducing overfitting and improving generalization. Each tree makes an independent prediction, and the final classification is determined by the majority vote of the trees. Due to the randomness in the selection of samples and features, different trees may focus on different aspects of the digits, thus strengthening the robustness and accuracy of the final classification. In essence, a Random Forest creates multiple decision trees, each trained with a random sample of the training data, and classifies based on the vote of the majority of these trees. This helps to reduce overfitting and improve generalization compared to a single decision tree. Although each individual decision tree is transparent, the aggregation of multiple trees and the randomness introduced in the process of feature and sample selection make the model as a whole more difficult to interpret. The complexity increases with the number of trees, making it difficult to understand how specific features influence the final decision.

2.6 Support Vector Machine

SVM is a versatile supervised machine learning algorithm applicable to both regression and classification tasks. In essence, SVM functions by identifying a hyperplane in a dataset with two distinct classes, aiming to optimally separate the data points of these classes. This separation is achieved by positioning the maximum number of data points from each class on their respective sides of the hyperplane while maximizing the distance, known as the margin, between the closest points of the classes and the hyperplane itself. The primary objective of SVM is to enhance this separation margin, as a wider margin indicates a greater generalization capability of the model. This margin plays a crucial role in mitigating the risk of overfitting to the training data, which could otherwise compromise the model's performance on new, unseen data. The defining hyperplane in SVM is determined by a subset of critical data points referred to as support vectors [28]. Due to its reliance on high dimensionality and geometric principles for operation, SVMs are often perceived as complex and less transparent, categorizing them as opaque models.

Formally, the problem of finding the optimal hyperplane in an SVM can be formulated as a convex optimization problem. Given a training set $T = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, where $x_i \in \mathbb{R}^d$ represents the feature vector of the i-th sample and $y_i \in \{-1, 1\}$ its respective class, the optimal hyperplane is defined by:

$$w^T x + b = 0, \tag{5}$$

where $w \in \mathbb{R}^d$ is the weight vector and b is the bias. The margin of separation is given by $2/\|w\|$, and the SVM optimization problem consists of finding w and b that maximize this margin, subject to the constraint that all samples are correctly classified:

$$y_i(w^T x_i + b) \geq 1, \forall i = 1, ..., n. \tag{6}$$

To handle cases where the data is not linearly separable, the SVM uses the kernel trick, which maps the data to a higher dimensional space where linear separation becomes possible. Common kernel functions include the linear, polynomial, and radial basis function (RBF) [29].

In the MNIST dataset, when employing SVM, the first step involves vectorizing the images, transforming each digit from its 28x28 matrix representation into a 784-dimensional vector. The SVM then operates in this high-dimensional space to identify the hyperplanes that best distinguish each class of digits (0 to 9). Using a kernel function, such as the RBF (Radial Basis Function), the SVM is capable of projecting the data into an even larger space where the classes become linearly separable. The decision criterion is based on maximizing the margin between the classes, choosing the hyperplane that leaves the largest possible distance from the closest points of each class, known as the support vectors. Despite its efficacy in classifying complex data, the logic behind the model's decisions is not easily interpretable by humans. The construction of the separation hyperplane in the high-dimensional feature space, determined by the support vectors, does not provide a direct or intuitive explanation of how the features of the data influence the classification.

2.7 Convolutional Neural Network

CNN is a class of deep neural networks, widely used in computer vision tasks such as image and video recognition, recommendation systems, and natural language processing. CNNs are inspired by the organization of the human visual cortex and are distinguished by their ability to automatically capture the importance of local features in images through the convolution process, without the need for manual preprocessing. The structure of a typical CNN includes convolutional layers, pooling layers, and fully connected layers, allowing the network to learn hierarchies of features efficiently [30]. It is considered an opaque algorithm due to its complex internal architecture and the vast number of parameters, which make it difficult to interpret how inputs are transformed into outputs. This "black-box" characteristic presents difficulties in human comprehension of the network's decision-making process, particularly in sensitive applications like medical diagnosis and surveillance. Although the convolutional layers of a CNN can learn hierarchical representations of visual data, the relationship between these representations and the reasoning of the network remains largely non-intuitive to humans [31].

A CNN is composed of multiple layers, including convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply filters (kernels) to the input, extracting relevant features such as edges, corners, and textures. In the context of images, the convolution operation slides the filter over the input image, calculating the dot product between the filter and the region of the image under it. Mathematically, the convolution operation between a filter w of dimensions $k \times k$ and a region x of the input image can be expressed as:

$$(w * x)_{ij} = \sum_{m=1}^{k} \sum_{n=1}^{k} w_{mn} x_{i+m-1, j+n-1}, \tag{7}$$

where $(w * x)_{ij}$ represents the value of the pixel at position (i, j) of the convolution output. Pooling layers reduce the dimensionality of the convolution output, making the representation more robust to small variations in the position of the features. Fully connected layers combine the features extracted by the previous layers to perform the final classification. Learning in a CNN is performed through the backpropagation algorithm, which adjusts the weights of the filters and the connections between the layers in order to minimize the classification error [30].

In the MNIST dataset, a CNN learns to classify digits by automatically identifying complex patterns in image data. Initially, convolutional layers apply filters to extract low-level features, such as edges and corners, from the raw pixels. Subsequent layers combine these initial features into more complex structures (high-level features), such as parts of digits. Finally, fully connected layers use these abstract features to make predictions about the digit class. The network adjusts the parameters of all filters and layers during training to minimize the classification error, using techniques such as backpropagation and gradient descent optimization. The depth and complexity of CNNs, along with the substantial number of parameters and the automatic feature learning, make the decision-making process extremely difficult to interpret. The hidden layers and the nonlinear nature of the transformations applied to the data contribute to the opacity, making it challenging to understand how extracted features affect the final classification.

2.8 Comparison of Algorithm Explainability

Table 1 presents a comparison of the explainability of the algorithms used, accompanied by justifications that highlight their interpretative characteristics. Decision Trees are noted for their high explainability, as they offer a clear visualization of decision paths, facilitating the tracing of how input features lead to classification decisions. Logistic Regression, while providing coefficients for each feature, falls into the moderate explainability category due to its mathematical nature, yet its direct relationship between features and outcome aids in interpretability. KNN, classified as moderately explainable, may vary in explainability based on factors such as the number of neighbors considered, the distance metric used, and the dimensionality of the feature space. Conversely, algorithms like Random Forest, SVM, and CNN exhibit lower explainability due to their complex structures and processes, making it challenging to interpret how features contribute to final outputs.

3 Related Works

Research in XAI is an emerging area that has been gaining increasing interest in the scientific community. However, despite exploring various implementation and methodology aspects, there remains a significant gap in addressing the crucial trade-offs between performance and explainability. According to [32], there is a noticeable trend towards interdisciplinarity in XAI research, with collaborations emerging among fields such as computer science, psychology, and medicine. This collaboration has been found to enrich the development of AI systems, leading to advancements that address not only technical aspects but also the social and ethical implications of AI technologies.

Table 1 Explainability of Transparent and Opaque Algorithms.

Algorithm	Explainability	Justification
Decision Trees	High	Decision Trees offer a clear visualization of decision paths, where each node represents a decision based on feature values. This tree structure allows for easy tracing of how input features lead to a classification decision.
Logistic Regression	Moderate	Logistic Regression provides coefficients for each feature, showing their impact on the probability of classification. Although the model is mathematical, the direct relationship between features and outcome makes it relatively interpretable.
KNN	Moderate	Although KNN is considered moderately explainable, as it classifies new data points based on the most frequent labels among the nearest neighbors, the explainability may vary depending on factors such as the number of neighbors considered (parameter "K"), the distance metric used, and the dimensionality of the feature space.
Random Forest	Low	Random Forest uses ensemble learning from multiple decision trees to improve accuracy and robustness. While individual trees are interpretable, the aggregation of many trees and the randomness introduced in the selection process decrease overall transparency.
SVM	Low	While SVM is effective in handling complex and high-dimensional data, the reliance on support vectors and hyperplanes for classification makes it less intuitive. The transformation of data through kernel functions further complicates its interpretability.
CNN	Low	CNNs are complex models with multiple layers that automatically learn feature hierarchies from data. The deep structure and convolution processes make it difficult to trace how features are combined and transformed into final outputs, leading to lower explainability.

In [33], the authors explored the integration of human-centered approaches in developing and evaluating XAI technologies. They argue that traditional XAI methods often fail to meet the needs and expectations of end-users interacting with AI systems. Their research emphasizes the role of Human-Computer Interaction (HCI) and User Experience (UX) design in improving the explainability of AI systems. They highlight three key roles: guiding technical choices based on users' explainability needs, identifying shortcomings in current XAI methods, and proposing conceptual frameworks for human-compatible XAI. This aims to illustrate how human-centered approaches can guide the development of more effective and trustworthy XAI applications.

In [34], the application of XAI techniques to enhance the transparency and interpretability of AI-driven audit processes is examined. They emphasize popular techniques such as Local Interpretable Model-agnostic Explanations (LIME) and Shapley Additive exPlanations (SHAP), demonstrating their utility in tasks such as assessing the risk of material misstatement. By integrating XAI methods with audit documentation and evidence standards, this research facilitates a connection between

complex AI models and the stringent requirements of auditing practices, equipping auditors with tools to better comprehend and trust AI outputs.

In [35], the authors focused on improving the applicability of XAI in the context of mental health, introducing the TIFU (Transparency and Interpretability For Understandability) framework to enhance the transparency and interpretability of AI decisions. This framework was designed to assist healthcare professionals in understanding and trusting AI systems, facilitating their integration into clinical practices. The study demonstrated that the TIFU framework could significantly improve doctors' interaction with technology, increasing their confidence and willingness to adopt AI solutions. Furthermore, it showed that more comprehensible AI systems could lead to quicker and broader adoption of these technologies in clinical settings, where clarity in decision-making is crucial for effective and safe treatments.

In [2], a systematic review of XAI techniques was conducted, highlighting their implementation across various domains. The review identified a significant gap between the sophistication of the developed techniques and their practical applicability. The study pointed out that despite technical advances, many XAI techniques are still perceived as inaccessible by professionals in non-technical fields, such as medicine and law, who require clear and contextualized explanations. This review shows the urgency of developing XAI approaches that are not only technically robust but also practical, accessible, and customized to the needs of end users.

In [4], the authors provided a detailed analysis of XAI techniques specifically applied in the context of healthcare. This study assessed how different XAI approaches have been developed and implemented to make AI decisions more transparent and understandable for healthcare professionals and patients. The research highlighted the importance of developing AI systems that not only support diagnoses and treatments but also allow users to understand the basis of these recommendations. This is crucial for the acceptance and effectiveness of AI systems in healthcare, where data-based decisions need to be explained clearly to ensure patient adherence and ethical compliance. Similarly in [36], a comprehensive review of XAI techniques applied to medical imaging applications is presented. With deep learning's increasing prominence in medical diagnostics, there is a pressing need for transparent AI methods capable of offering understandable and trustworthy explanations. Their framework categorizes various XAI techniques according to anatomical locations and specific criteria relevant to medical image analysis, while also identifying future research directions aimed at improving AI interpretability in critical healthcare settings.

In [1], the technical and methodological challenges in implementing transparent and auditable XAI systems were discussed. They argued that the complexity of AI models and data sets often impedes the transparency required for effective explainability. This study proposes a more integrated and systematic approach in the development of AI systems, where clarity and comprehensibility are incorporated from the beginning of the design process. Such an approach is essential to ensure that AI systems are reliable and can be effectively audited by regulators and end users. In another work, [37] provided a unique perspective on integrating XAI into the software development process. This study examined how XAI practices could be systematically incorporated

throughout the software development phases to improve not only the quality of AI systems but also their acceptability among end users. The meta-review presented in the article suggested that aligning XAI techniques with traditional software development methodologies could facilitate a broader and more effective adoption of explainability practices, ensuring that AI solutions were developed with a solid foundation in transparency and comprehensibility from the start.

This study shares certain similarities with previous research, as presented in Table 2. However, it is crucial to note that these studies did not explore the relationship between performance and explainability in ML models, nor did they consider an experimental approach. The distinction of this research lies in the comparison between transparent algorithms, such as KNN, Decision Tree, and Logistic Regression, and opaque ones, like CNN, Random Forest, and Support Vector Machines. The goal is to analyze the trade-offs between performance and explainability in a client-server architecture. This work aims to provide researchers, policymakers, and industry leaders with a more accurate assessment of the advantages and disadvantages of each model, facilitating the choice of the most suitable one for different contexts.

Table 2 Comparison with Related Works.

Work	Focus Area	XAI Implementation	Practical Applicability
[32]	Interdisciplinarity in XAI	General XAI principles	Broad auditability and stakeholder understanding
[33]	Integrating human-centered approaches in XAI	Emphasizes HCI and UX for AI explainability	Demonstrated benefits for effective XAI development
[34]	XAI in audit processes	Application of LIME and SHAP in audit integration	Enhances auditor comprehension and trust in AI-driven audits
[35]	XAI in Mental Health	TIFU framework	Focused on healthcare professional adoption
[2]	Systematic review of XAI techniques	Review of techniques	Highlighted need for accessibility in non-technical fields
[4]	XAI in Healthcare	Assessment of XAI techniques	Aimed at healthcare professionals and patient understandability
[36]	XAI in medical imaging	Categorizes XAI techniques by anatomical locations and specific criteria	Enhances transparency and trust in AI for medical diagnostics
[1]	Methodological challenges in XAI	Proposes systematic approach	Aimed at regulatory and end-user auditability
[37]	XAI in Software Development	Integration of XAI practices	Enhances quality and user acceptability of AI systems
Our Study	Trade-offs in XAI	Comparison of transparent and opaque algorithms	Guides model selection for performance and explainability needs

4 Experimental Plan

In this section, we detail the experimental plan adopted in this work, which evaluates the trade-offs between performance and explainability in the context of intelligent environments. These experiments are designed to replicate conditions found in smart systems, where AI models must operate efficiently while providing understandable outcomes to users.

4.1 Datasets and models

To analyze the relationship between performance and explainability, we adopted two datasets: MNIST and Fake and Real News. We use the MNIST dataset from TensorFlow, an open-source machine learning library useful for a variety of applications [38]. The MNIST dataset comprises a large set of handwritten digits, including a training set of 60,000 samples and a testing set of 10,000 samples. This dataset originates from a subset of NIST Special Database 3, with digits written by United States Census Bureau employees, and Special Database 1, which includes digits written by high school students, all in grayscale format [39]. We also use the Fake and Real News dataset, which contains 21,417 articles with real news and 23,502 articles with fake news. This dataset includes the title of the news article, the full text of the article, the subject, and the publication date of the article [40].

In the initial treatment of the MNIST dataset, which consists of images of handwritten digits, preprocessing techniques were adopted to optimize the images for machine learning model training. The images underwent a normalization process, where pixel values were adjusted from a range of 0 to 255 to a range of 0 to 1. This step is crucial to facilitate convergence during model training, as it deals with smaller and more uniform input values. Consistency and uniformity in the input data help to improve the efficiency and effectiveness of the subsequently applied learning algorithms. For the Fake and Real News dataset, which includes texts categorized as false and true news, textual preprocessing was also employed. Initially, texts were cleaned by removing special characters and converting all letters to lowercase to standardize data input. Subsequently, common words that usually do not contribute to distinguishing between categories, known as stopwords, were removed. Additionally, tokenization was applied, segmenting the texts into individual words, followed by vectorization using the Frequency-Inverse Document Frequency (TF-IDF) method, which quantifies the importance of each word in the documents relative to the dataset as a whole. These vectorization techniques are essential for transforming text into a form that machine learning models can effectively process and learn.

In our study, we selected models based on their transparency and complexity. For transparent models, which offer interpretability through their straightforward structures, we chose Decision Tree, Logistic Regression, and KNN. On the other hand, for opaque models, which are more complex and less interpretable, we selected Random Forest, SVM, and CNN.

4.2 Adopted Intelligent Environment

To conduct the performance evaluations in the adopted intelligent environment, we configured a networked setup involving two computers (see Figure 1). One computer functioned as the client, acting as a workload generator to simulate user interactions within a smart environment, while the other served as the server, hosting either transparent models (e.g., Logistic Regression, KNN, and Decision Trees) or opaque models (e.g., SVM, Random Forest, and CNN). This setup is designed to emulate the typical interactions of a real-world intelligent system, where a client device (such as a smartphone or sensor) sends data to a server that processes it using a machine learning model. This configuration enables us to examine the trade-offs between response time, accuracy, and explainability across different models, considering each algorithm's specific demands and constraints. The client-server architecture also reflects real-world smart systems, where distributed processing and networked interactions are common.

The client and server machines were connected to the same local network via Wi-Fi. In each test, the client transmitted data (e.g., images or articles) to the server using socket connections. The server processed these requests and returned the classification results to the client. To ensure accurate measurement, only one model was hosted on the server at a time, allowing for precise assessment of the server's average response time under varying workloads and across different models. After each test, we collected and analyzed the performance metrics to explore the trade-offs. Details regarding the hardware configurations of the server and client are provided in Table 3.

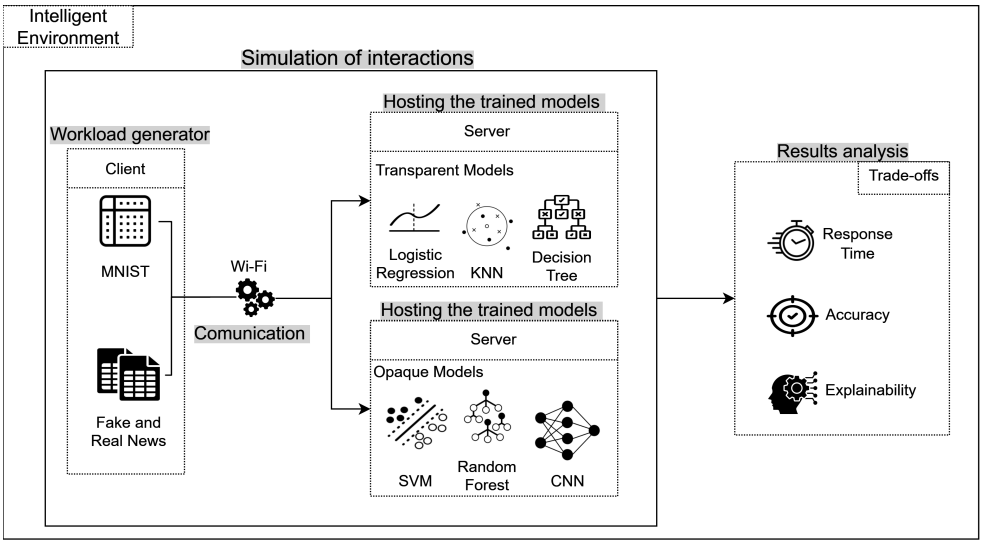

Fig. 1 Adopted intelligent environment.

Table 3 Configuration of Adopted Platforms.

Configuration	Server	Client
CPU	Intel(R) Core(TM) i5-8500T	Intel(R) Core(TM) i7-8565U
Memory RAM	24 Gb DDR4 2666 Mhz	12 Gb DDR4 2400 Mhz
Disk Storage	500 Gb SSD	500 Gb SSD
GPU	Intel(R) UHD Graphics 630	Intel(R) UHD Graphics 620

4.3 Experiment Execution

Before initiating the training of the models, we utilized the RandomizedSearchCV library [41] to optimize the selection of hyperparameters. This tool operates by randomly sampling a combination of possible parameters during training and selects the set that achieves the highest performance based on predefined evaluation criteria. For each model, we conducted experiments aimed at identifying the optimal set of hyperparameters that delivered the maximum accuracy. Table 4 displays the hyperparameters selected for the algorithms applied to the MNIST dataset, while Table 5 presents the hyperparameters used for the Fake and Real News dataset.

Table 4 Hyperparameters selected via RandomizedSearch for the MNIST dataset.

Algorithm	Hyperparameters
KNN	`n_neighbors`: 5 `p`: 2 `weights`: uniform
Decision Tree	`criterion`: gini `max_depth`: None
Logistic Regression	`penalty`: l2 `dual`: False
CNN	`activation`: relu `dropout_rate`: 0.25 `kernel_size`: 4 `num_filters`: 106
Random Forest	`criterion`: gini `max_depth`: None
SVM	`kernel`: rbf `degree`: 3 `gamma`: scale

After training the models, we proceeded with the experimental phase, which involved transmitting images (or articles) from the client computer to the server under varying workloads categorized as 1.0, 0.5, and 0.1 seconds for low, medium, and high workloads, respectively. These specific workloads were chosen to simulate different

Table 5 Hyperparameters selected via RandomizedSearch for the Fake and Real News dataset.

Algorithm	Hyperparameters
KNN	`n_neighbors`: 1 `p`: 2 `weights`: distance
Decision Tree	`criterion`: gini `max_depth`: None
Logistic Regression	`penalty`: l2 `dual`: True
CNN	`activation`: relu `dropout_rate`: 0.25 `kernel_size`: 4 `num_filters`: 96
Random Forest	`criterion`: gini `max_depth`: 150
SVM	`kernel`: linear `degree`: 2 `gamma`: 1.0

levels of server load that could occur in real-world applications. A 1.0-second workload represents a scenario with low demand on the server, similar to situations where the server is handling fewer requests, thus allowing more time for each transaction. A 0.5-second workload signifies a medium load, reflecting a typical operational scenario where the server is managing a moderate number of requests simultaneously. The 0.1-second workload illustrates a high-demand scenario, where the server is under significant load, handling numerous requests in a short time frame. By testing the models under these varying workloads, our goal was to understand how these variations impacted the performance of the deployed models, providing insights into their robustness and reliability in different operational conditions. It is important to note that each model was tested with MNIST and Fake and Real News datasets, processing a total of one thousand requests under each specified workload.

4.4 Analyses

To assess the performance of the models, we employed the accuracy metric, which measures the proportion of correct predictions made by the model in comparison to the total predictions. Given that the datasets used is balanced with evenly distributed classes, accuracy serves as an effective evaluation metric for this experiment. To calculate the accuracy, we used a function from the scikit-learn library in Python, which specializes in computing evaluation metrics for machine learning models. Additionally, to collect the performance of the models, we computed the average response times for each model under varying workloads. Response time is defined as the duration

required for a specific operation or process to complete, starting from the initiation of the request to the point when the response is fully received, including any delays introduced by the network. The response time can be influenced by several factors including the complexity of the model, the nature of the input, and the processing capabilities of the server. Monitoring response time is critical for evaluating a model's performance, ensuring the application's efficiency, and confirming that the models are capable of delivering efficient, scalable, and real-time results.

5 Results

In this section, we present the results of the experiments conducted within intelligent environments, simulating scenarios where AI models must process information efficiently and transparently. After transmitting the images (or articles) from the client to the server under varying workloads, reflecting the dynamic nature of smart systems, we collected and analyzed the accuracies and response times of each model on the two datasets.

5.1 Accuracy analysis

Table 6 shows accuracy results for the MNIST dataset, where opaque models (Random Forest, SVM, and CNN) performed better, achieving accuracies of 0.97, 0.98, and 0.98, respectively, compared to transparent models (Decision Tree, Logistic Regression, and KNN) with accuracies of 0.83, 0.93, and 0.94. The robustness of Random Forest over Decision Tree is due to its structure, combining multiple trees to reduce overfitting. High accuracy in SVM and CNN reflects their effectiveness in managing MNIST's data characteristics. Although Logistic Regression benefits from the dataset's large number of predictor variables, the dataset's complexity limited its accuracy slightly, whereas KNN benefitted from the data distribution.

Table 6 Results for the MNIST Dataset

Model	Accuracy	Explainability
Decision Tree	0.83	Transparent
Logistic Regression	0.93	Transparent
K-Nearest Neighbors	0.94	Transparent
Random Forest	0.97	Opaque
Convolutional Neural Network	0.98	Opaque
Support Vector Machine	0.98	Opaque

Table 7 displays accuracy for the Fake and Real News dataset, showing that opaque models (Random Forest, SVM, CNN) once again outperformed transparent models. Transparent models achieved accuracies of 85, 91, and 92 for Decision Tree, Logistic Regression, and KNN, while opaque models reached 94, 94, and 95, respectively. Random Forest's integration of multiple trees helps handle linguistic complexity, while SVM and CNN efficiently manage textual structures. Logistic Regression performed slightly lower due to limited non-linear capture, and KNN performed well due to similarity-based classification useful in text data.

Table 7 General Model Results for the Fake and Real News dataset

Model	Accuracy	Explainability
Decision Tree	0.85	Transparent
Logistic Regression	0.91	Transparent
K-Nearest Neighbors	0.92	Transparent
Random Forest	0.94	Opaque
Convolutional Neural Network	0.95	Opaque
Support Vector Machine	0.94	Opaque

Comparing the results obtained between the MNIST and the Fake and Real News datasets reveals consistent trends in algorithm performance. For both datasets, opaque algorithms outperformed their transparent in terms of accuracy, suggesting that these models are better suited for handling complex patterns and large feature sets across varied data types. While transparent models provide the benefit of easier interpretability, they generally achieve lower accuracy, indicating a potential trade-off between transparency and performance. The inherent challenges in each dataset (image recognition for MNIST and text analysis for Fake and Real News) further highlight the superior capabilities of opaque models in managing complex patterns and delivering higher accuracy.

5.2 Response time analysis

Figures 2 and 3 show the average response times for each model under varying workloads across the datasets. The x-axis indicates workload levels, and the y-axis represents response times in seconds. For the MNIST dataset (Figure 2), transparent models (Decision Tree, Logistic Regression, and KNN) maintained stable response times at lower workloads, with moderate increases at medium and high workloads. Under high workload, Decision Tree recorded 0.261 seconds, KNN 0.291 seconds, and Logistic Regression 0.239 seconds. In contrast, opaque models had higher times, with Random Forest at 0.415 seconds, CNN at 0.423 seconds, and SVM at 0.384 seconds. Similarly, for the Fake and Real News dataset (Figure 3), transparent models showed consistent response times, with slight increases under medium and high workloads. At high workload, Decision Tree reached 0.334 seconds, KNN 0.315 seconds, and Logistic Regression 0.304 seconds, while opaque models recorded higher times: Random Forest at 0.404 seconds, CNN at 0.444 seconds, and SVM at 0.429 seconds.

In evaluating response times on the MNIST and Fake and Real News datasets, transparent models consistently showed shorter response times across varying workloads, demonstrating efficient data processing. Opaque models, however, had notably longer response times, especially as workloads increased. This trend persisted across both datasets, highlighting the efficiency trade-offs associated with opaque models.

To determine if the differences in response times among the models are statistically significant, we employed the Kruskal-Wallis test, followed by the post-hoc Dunn test. The Kruskal-Wallis test is a non-parametric method used to evaluate whether there are statistically significant differences across multiple groups. This test is particularly

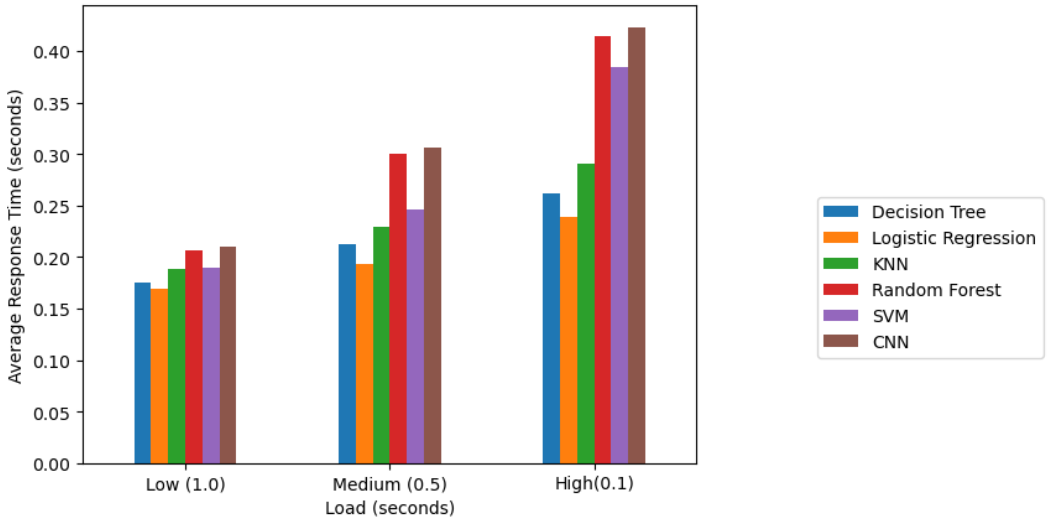

Fig. 2 Average response time per load on the MNIST dataset

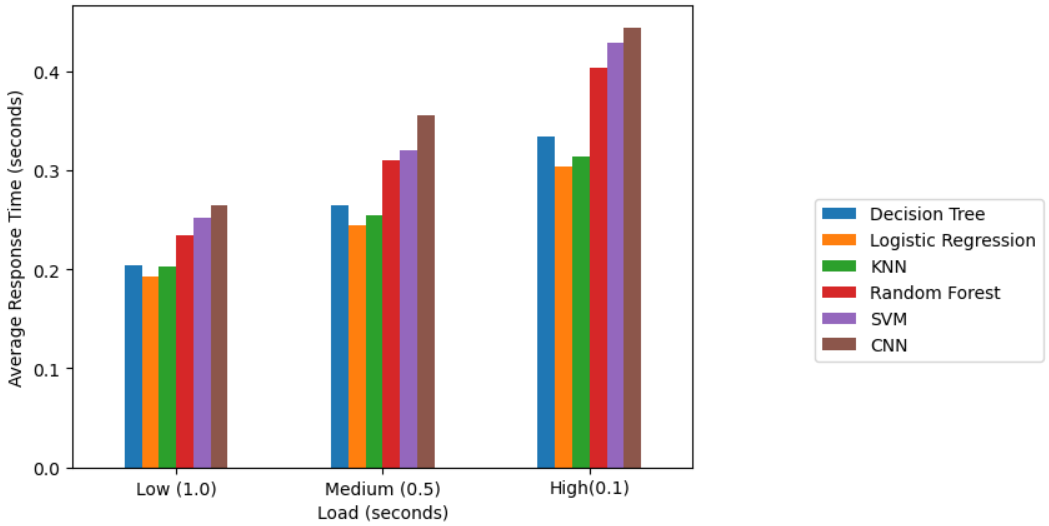

Fig. 3 Average response time per load on the Fake and Real News dataset

useful when the assumption of normality in the data cannot be assured, as is common in real-world datasets [42]. Upon establishing that differences exist among the groups with the Kruskal-Wallis test, we applied the post-hoc Dunn test for pairwise comparisons to identify which specific groups differ significantly from each other [43].

We conducted these statistical tests across all three workload conditions (low, medium, and high). The results from the Kruskal-Wallis test confirmed significant differences among the groups, indicating that at least one model performed differently in terms of response times. Subsequently, the Dunn test provided a multiple comparison matrix, revealing significant differences between all pairs of groups. This comprehensive analysis confirms that the distributions of response times are indeed significantly different across the models under different loads, providing statistical evidence of variation in model performance.

5.3 Trade-off analysis

Figure4 and 5 illustrate the trade-offs among accuracy, explainability, and response time across different machine learning models. Transparent models like Decision Tree and Logistic Regression offer high interpretability and faster response times, but with moderate accuracy. In contrast, opaque models such as SVM and CNN achieve high accuracy but at the cost of lower explainability and longer response times, reflecting the "black box" issue inherent in complex models. KNN and Logistic Regression represent a balanced compromise among these aspects, while Random Forest, though highly accurate, is less efficient in response time and explainability. These figures highlight the essential trade-offs in model selection, where improvements in one dimension often necessitate sacrifices in another, emphasizing the need for careful alignment of model characteristics with application-specific requirements.

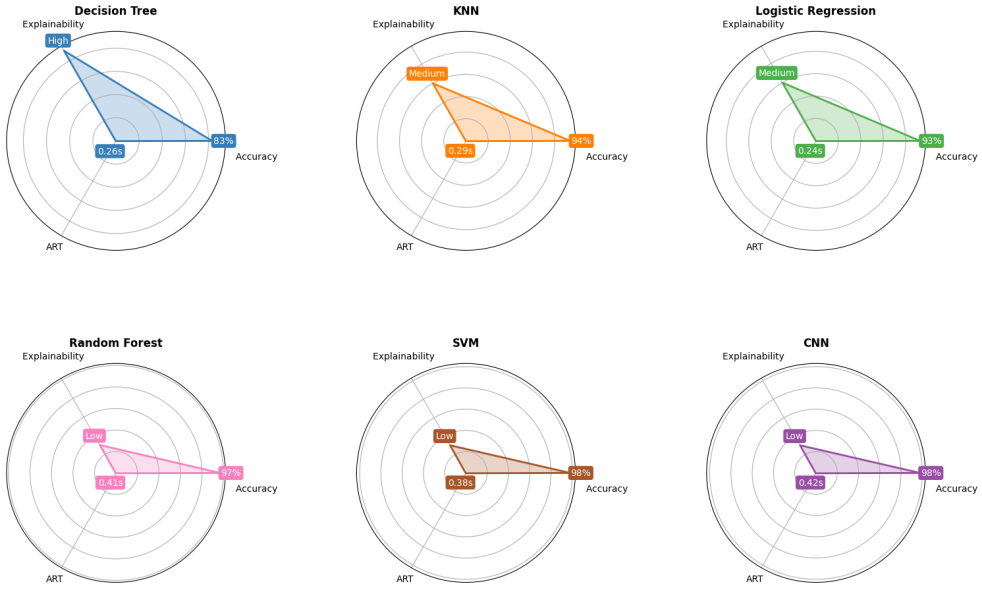

Fig. 4 Comparison among Performance, Explainability and Average Response Time on MNIST dataset.

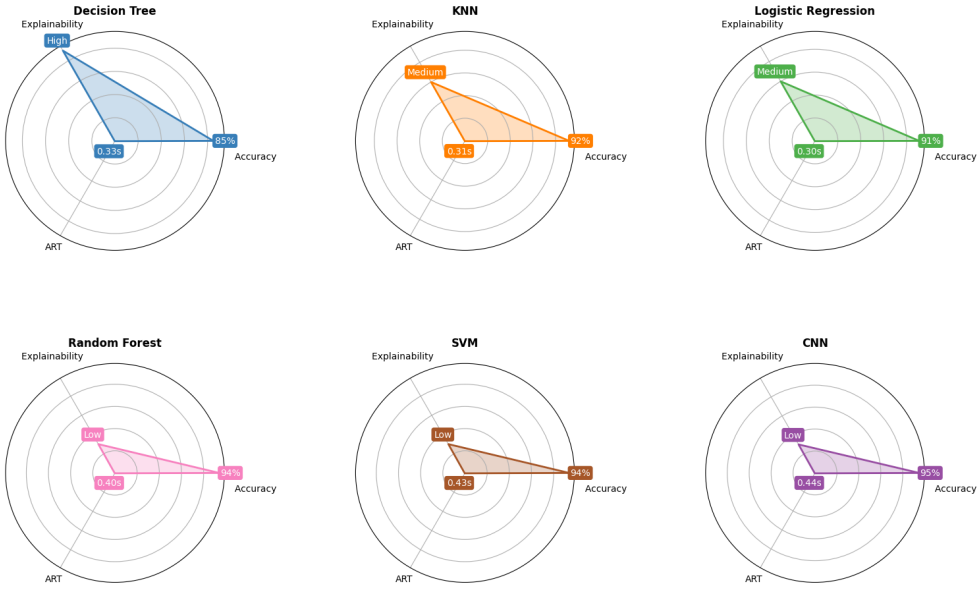

Fig. 5 Comparison among Performance, Explainability and Average Response Time on Fake and Real News dataset.

Choosing between opaque and transparent algorithms depends on the specific needs of an application. For high accuracy, especially in tasks involving complex data patterns like image recognition in the MNIST dataset, Support Vector Machines or Convolutional Neural Networks are recommended. These opaque models excel in handling intricate data effectively but offer limited explainability due to their complex internal structures. On the other hand, for applications where quick response times and transparency are critical, such as interactive tools, K-Nearest Neighbors or Decision Trees are preferable due to their simplicity and direct interpretability. These models allow users to easily trace decision paths and understand how inputs influence outputs. In environments where clarity in decision-making is important, such as in healthcare diagnostics or financial auditing, transparent models like Logistic Regression are advised for their clear linkage between input features and predicted outcomes, even though they may not achieve the highest accuracy. For scenarios that demand robust performance but where transparency is less critical, Random Forests are suitable, beacause they provide improved accuracy over single Decision Trees by aggregating results from multiple trees, reducing variance and avoiding overfitting, although their layered complexity can obscure the exact decision-making process.

6 Conclusions

XAI represents a promising field of study focused on developing techniques that enhance the transparency and comprehensibility of decisions made by AI algorithms. The goal is to increase the transparency of computational systems by allowing users

to clearly understand how decisions are formed and what factors influence them. This is crucial because, despite AI's ability to learn and make decisions effectively, its often opaque nature and the complexity of the models can make it perceived as a "black box".

In this study, we conducted a comprehensive comparison of transparent algorithms, including KNN, Decision Tree, and Logistic Regression, with opaque algorithms, such as CNN, Random Forest, and Support Vector Machines, within intelligent environments. Using the MNIST and Fake and Real News datasets, our evaluation focused on the trade-offs between performance (accuracy and response time) and explainability. The findings emphasize the importance of selecting the right balance of accuracy and transparency for AI systems operating in smart environments, where user trust and system efficiency are important. The results revealed that while transparent models offer greater ease of interpretation and faster responses, they tend to have lower accuracy. Conversely, opaque models, despite being more accurate, exhibited longer response times and poorer explainability due to their intrinsically opaque nature. Therefore, the choice between transparent and opaque models should take into account the user's priorities, with transparent models recommended when explainability and speed are crucial, while opaque models are preferable for tasks requiring higher accuracy.

Future research directions include exploring techniques to enhance the accuracy of transparent models, investigating hybrid approaches that integrate the interpretability of simpler models with the accuracy of complex ones, developing novel explainability methods for opaque models, and assessing the impact of XAI on user trust. Future work could also focus on developing a metric to quantify explainability based on user feedback regarding the effectiveness and quality of explanations provided by XAI models, enabling a more detailed evaluation of these methods.

References

[1] Saeed, W., Omlin, C.: Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities. Knowledge-Based Systems **263**, 110273 (2023)

[2] Islam, M.R., Ahmed, M.U., Barua, S., Begum, S.: A systematic review of explainable artificial intelligence in terms of different application domains and tasks. Applied Sciences **12**(3), 1353 (2022)

[3] Gohel, P., Singh, P., Mohanty, M.: Explainable ai: current status and future directions. arXiv preprint arXiv:2107.07045 (2021)

[4] Chaddad, A., Peng, J., Xu, J., Bouridane, A.: Survey of explainable ai techniques in healthcare. Sensors **23**(2), 634 (2023)

[5] Assis, A., Véras, D., Andrade, E.: Explainable artificial intelligence-an analysis of the trade-offs between performance and explainability. In: 2023 IEEE Latin American Conference on Computational Intelligence (LA-CCI), pp. 1–6 (2023).

IEEE

[6] Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., Qian, B., Wen, Z., Shah, T., Morgan, G., *et al.*: Explainable ai (xai): Core ideas, techniques, and solutions. ACM Computing Surveys **55**(9), 1–33 (2023)

[7] Tjoa, E., Guan, C.: A survey on explainable artificial intelligence (xai): Toward medical xai. IEEE transactions on neural networks and learning systems **32**(11), 4793–4813 (2020)

[8] Langer, M., Oster, D., Speith, T., Hermanns, H., Kästner, L., Schmidt, E., Sesing, A., Baum, K.: What do we want from explainable artificial intelligence (xai)?–a stakeholder perspective on xai and a conceptual model guiding interdisciplinary xai research. Artificial Intelligence **296**, 103473 (2021)

[9] Van Lent, M., Fisher, W., Mancuso, M.: An explainable artificial intelligence system for small-unit tactical behavior. In: Proceedings of the National Conference on Artificial Intelligence, pp. 900–907 (2004). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999

[10] Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artificial intelligence **267**, 1–38 (2019)

[11] Kim, B., Khanna, R., Koyejo, O.O.: Examples are not enough, learn to criticize! criticism for interpretability. Advances in neural information processing systems **29** (2016)

[12] Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608 (2017)

[13] Molnar, C.: Interpretable Machine Learning. Lulu. com, s.l. (2020)

[14] Angelov, P.P., Soares, E.A., Jiang, R., Arnold, N.I., Atkinson, P.M.: Explainable artificial intelligence: an analytical review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **11**(5), 1424 (2021)

[15] Quinlan, J.R.: Learning decision tree classifiers. ACM Computing Surveys (CSUR) **28**(1), 71–72 (1996)

[16] Quinlan, J.R.: Simplifying decision trees. International journal of man-machine studies **27**(3), 221–234 (1987)

[17] Quinlan, J.R.: Induction of decision trees. Machine learning **1**, 81–106 (1986)

[18] Quinlan, J.R.: C4. 5: Programs for Machine Learning. Elsevier, s.l. (2014)

[19] Breiman, L.: Classification and Regression Trees. Routledge, s.l. (2017)

[20] Shevade, S.K., Keerthi, S.S.: A simple and efficient algorithm for gene selection using sparse logistic regression. Bioinformatics **19**(17), 2246–2253 (2003)

[21] Böhning, D.: Multinomial logistic regression algorithm. Annals of the institute of Statistical Mathematics **44**(1), 197–200 (1992)

[22] Hosmer Jr, D.W., Lemeshow, S., Sturdivant, R.X.: Applied Logistic Regression. John Wiley & Sons, s.l. (2013)

[23] Zhang, S., Cheng, D., Deng, Z., Zong, M., Deng, X.: A novel knn algorithm with data-driven k parameter computation. Pattern Recognition Letters **109**, 44–54 (2018)

[24] Uddin, S., Haque, I., Lu, H., Moni, M.A., Gide, E.: Comparative performance analysis of k-nearest neighbour (knn) algorithm and its different variants for disease prediction. Scientific Reports **12**(6256) (2022)

[25] Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE transactions on information theory **13**(1), 21–27 (1967)

[26] Schonlau, M., Zou, R.Y.: The random forest algorithm for statistical learning. The Stata Journal **20**(1), 3–29 (2020)

[27] Breiman, L.: Random forests. Machine learning **45**, 5–32 (2001)

[28] Suthaharan, S., Suthaharan, S.: Support vector machine. Machine learning models and algorithms for big data classification: thinking with examples for effective learning, 207–235 (2016)

[29] Cortes, C.: Support-vector networks. Machine Learning (1995)

[30] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)

[31] Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, pp. 818–833 (2014). Springer

[32] Jacovi, A.: Trends in explainable ai (xai) literature. arXiv preprint arXiv:2301.05433 (2023)

[33] Liao, Q.V., Varshney, K.R.: Human-centered explainable ai (xai): From algorithms to user experiences. arXiv preprint arXiv:2110.10790 (2021)

[34] Zhang, C.A., Cho, S., Vasarhelyi, M.: Explainable artificial intelligence (xai) in auditing. International Journal of Accounting Information Systems **46**, 100572 (2022)

[35] Joyce, D.W., Kormilitzin, A., Smith, K.A., Cipriani, A.: Explainable artificial intelligence for mental health through transparency and interpretability for understandability. npj Digital Medicine **6**(1), 6 (2023)

[36] Velden, B.H., Kuijf, H.J., Gilhuijs, K.G., Viergever, M.A.: Explainable artificial intelligence (xai) in deep learning-based medical image analysis. Medical Image Analysis **79**, 102470 (2022)

[37] Clement, T., Kemmerzell, N., Abdelaal, M., Amberg, M.: Xair: A systematic metareview of explainable ai (xai) aligned to the software development process. Machine Learning and Knowledge Extraction **5**(1), 78–108 (2023) https://doi.org/10.3390/make5010006

[38] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org (2015). https://www.tensorflow.org/

[39] LeCun, Y., Cortes, C., Burges, C.: Mnist handwritten digit database. ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist **2** (2010)

[40] Fake and Real News Dataset. Kaggle (2020). https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset

[41] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)

[42] Elliott, A.C., Hynan, L.S.: A sas® macro implementation of a multiple comparison post hoc test for a kruskal–wallis analysis. Computer methods and programs in biomedicine **102**(1), 75–80 (2011)

[43] McKight, P.E., Najab, J.: Kruskal-wallis test. The corsini encyclopedia of psychology, 1–1 (2010)

APPENDIX B – IMAT: A Tool for Analyzing Interpretable Machine Learning Models

Assis et al. «IMAT: Uma Ferramenta para Análise de Modelos de Aprendizado de Máquina Interpretáveis». In: *Brazilian Workshop on Social Network Analysis and Mining (BraSNAM)*, SBC, 2025, p. 134–147. DOI: 10.5753/brasnam.2025.8924.

Author's contribution: The author was responsible for implementing the IMAT tool, designing the experimental setup, running the comparative experiments, processing the collected data, generating the visual and textual artifacts (including tables, charts, and explanation outputs), and drafting the original manuscript.

IMAT: A Tool for the Analysis of Interpretable Machine Learning Models

André Assis[1], **Jamilson Dantas**[2], **Ermeson Andrade**[1]

[1]Department of Computing – Federal Rural University of Pernambuco (UFRPE)
Recife, PE – Brazil

{andre.assis, ermeson.andrade}@ufrpe.br

[2]Center for Informatics – Federal University of Pernambuco
Recife, PE – Brazil

jrd@cin.ufpe.br

Abstract. *Transparency and interpretability of Artificial Intelligence (AI) and Machine Learning (ML) models are increasingly relevant in applications involving social network analysis and data mining. While advanced models, such as deep learning, provide robust solutions to complex problems, their growing complexity makes it difficult to understand the decisions they make. The lack of transparency can undermine user trust and hinder the widespread adoption of these technologies. To address this challenge, this paper presents IMAT (Interpretable Models Analysis Tool), a tool designed to generate flowcharts that map each stage of data processing in deep learning models. IMAT aims to provide a clear and accessible visualization of the data flow and internal operations of models, from data input to output generation, facilitating their interpretation. Additionally, this work discusses the architecture and functionalities of IMAT and demonstrates its application in sentiment analysis of tweets, using the MLP (MultiLayer Perceptron) algorithm, evaluating the implications and limitations of the obtained results.*

1. Introduction

Artificial Intelligence (AI) and Machine Learning (ML) have become fundamental across various industries and information technology in general. Aiming to solve complex real-world problems, robust deep learning models stand out for their ability to make accurate predictions from large volumes of data. However, the increasing complexity of these models raises critical questions about the interpretability and transparency of the decisions made—aspects that are essential for the acceptance and reliability of the technology, especially in sensitive areas such as medical diagnostics and financial decisions. The lack of transparency in the models can result in distrust from end users, compromising the widespread adoption of these technologies. Moreover, interpretability is important for identifying and correcting biases and errors in the models, ensuring fairness and accuracy in predictions [Assis et al. 2023]. This need for explanations that facilitate understanding of the model becomes even more evident in applications that handle unstructured and highly subjective data, such as those found on social networks.

Social networks, especially Twitter (currently named X), have become rich environments for textual data analysis, enabling the exploration of public opinions on a wide

variety of topics [Silva et al. 2021, Paes et al. 2022]. Sentiment analysis with AI models, which identifies and classifies emotions expressed in texts, has proven particularly valuable for understanding social perceptions in real time. Applicable to large volumes of tweets, this approach provides relevant information for decision-making in areas such as marketing, politics, and public health. However, the correct interpretation of the results of these models requires attention to transparency and reliability. Understanding the criteria that led to a given classification is fundamental to validate the results, detect biases, and ensure the quality of the inferences produced.

One of the main challenges faced is the difficulty of understanding and explaining the internal processes of deep learning models, which often operate as "black boxes," making the paths that lead to certain predictions difficult to understand, even for experts [Alves and ANDRADE 2022]. This lack of transparency hinders the validation and trust in predictions, in addition to making it difficult to identify potential biases and errors. Therefore, it is essential to develop tools that provide a clear and accessible understanding of these models' decision-making processes. Such interpretability tools play a crucial role in the validation and continuous improvement of ML models, providing essential feedback for adjustments and refinements [Cortiz 2021]. By offering a detailed view of the models' internal mechanisms, they enable developers to identify unsuitable patterns, correct biases, and make precise adjustments [Agarwal and Das 2020]. Furthermore, integrating interpretability tools into regular review and update cycles contributes to the creation of fairer, more reliable systems aligned with users' expectations and needs.

Several approaches have been proposed to address these challenges, focusing on techniques that make models more interpretable. Methods such as LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations) are widely used to provide insights into model predictions, offering local explanations that help understand the model's behavior on specific instances [Salih et al. 2023]. These techniques have shown effectiveness in different contexts, standing out for their ability to make the internal mechanisms of ML models visible. Nevertheless, there remains a significant gap in the integration of tools that, in addition to providing explanations, visualize the entire decision-making process in a way that is understandable to end users, facilitating interpretation in industrial and academic environments [Bhattacharya 2022].

To bridge this gap, this work proposes IMAT (Interpretable Models Analysis Tool), a tool developed to build detailed and summarized flowcharts that map each step of data processing in deep learning models. IMAT seeks to offer a clear and accessible visualization of the data flow and the models' internal operations, from data input to the generation of the response, facilitating understanding and increasing model transparency. By providing a graphical and detailed representation of the models' inner workings, IMAT aims not only to increase user trust, but also to provide a powerful tool for the analysis and continuous improvement of ML models, meeting the needs of different users. In addition, this work aims to evaluate the applicability of IMAT through a case study in which we analyze the operation of an MLP-type neural network model applied to sentiment analysis of tweets, using the Sentiment140 dataset.

The structure of this article is organized as follows: Section 2 introduces the concept of Explainable Artificial Intelligence, highlighting its importance and main techniques. Section 3 reviews the principal works related to interpretability in ML, justifying

the need for IMAT. Section 4 describes IMAT's main functionalities. Section 5 presents a case study applied to sentiment analysis of tweets. Finally, Section 6 presents final considerations and suggestions for future research.

2. Explainable Artificial Intelligence

XAI, from the English *eXplainable* Artificial Intelligence, is a research field that seeks to develop methods and techniques that enable human understanding and interpretation of artificial intelligence models. Its main objective is to reduce the opacity of AI models—often referred to as "black boxes"—and make them more transparent and trustworthy for end users [Adadi and Berrada 2018]. The opacity of AI models prevents humans from understanding the underlying decision-making processes, which can lead to distrust and reluctance to adopt these technologies [Meske and Bunde 2020]. Transparency, on the other hand, refers to a system's ability to be understood by its users, while explainability involves the system's ability to provide comprehensible justifications for its decisions and predictions [Gregor and Benbasat 1999].

Interpretability and explainability in Machine Learning, according to authors such as Miller, Kim, and Molnar, refer to the ability of a human to understand and consistently predict a model's decisions, making it transparent and comprehensible [Miller 2019, Kim et al. 2016, Molnar 2020]. This characteristic is fundamental to increasing users' trust in AI systems, especially in critical domains such as healthcare, where algorithmic decisions can have significant consequences. XAI enables the identification and correction of biases, improves model performance, and ensures that models operate fairly and ethically, contributing to the construction of more trustworthy and responsible AI systems [Sundar 2020].

Machine learning models vary in their complexity and, consequently, in their capacity for explanation. Simpler models are generally more transparent, allowing a more intuitive understanding of how they work. In contrast, more complex models tend to be opaque, making it difficult to directly interpret their internal processes [Angelov et al. 2021]. Research in XAI seeks to develop methods that increase the transparency and interpretability of both types of models, providing detailed information about their internal operations and decisions [Arrieta et al. 2020]. Thus, it promotes user trust and facilitates the adoption of AI systems, while ensuring accountability and transparency in critical applications [Adadi and Berrada 2018].

In the context of social networks and data mining, XAI plays a crucial role by ensuring transparency and reliability in analyses. Platforms such as X generate large volumes of textual data, used to identify trends and understand public opinion. Sentiment analysis, for example, is widely applied in marketing, politics, and public health, but its effectiveness depends on the interpretability of the models used. The opacity of deep learning models can undermine the acceptance of their inferences, especially when they impact public policy or business strategies [Søgaard 2023]. XAI methods make it possible to understand the criteria that lead to certain classifications, identify biases, and ensure that models operate fairly and in alignment with users' expectations.

In addition to improving the quality of analyses, the integration of explainable techniques in social network data mining expands the adoption of these technologies. Tools that offer clear visualizations of the models' decision-making process enable both

experts and non-experts to better interpret results and make more well-founded decisions [Arrieta et al. 2020]. Thus, by promoting greater transparency and comprehensibility, XAI strengthens the ethical and responsible use of artificial intelligence in highly dynamic environments.

3. Related Work

The growing complexity of machine learning models has generated significant demand for tools that facilitate understanding of their internal operations. Considering that XAI tools are essential to ensure transparency, trust, and the ethical adoption of AI across sectors, this section reviews the main works and tools related to interpretability in machine learning models, highlighting contributions and the rationale for developing the IMAT tool.

In [Angelov et al. 2021], a comprehensive review of the explainability of artificial intelligence in the context of deep learning is presented. The study addresses a detailed taxonomy and the main challenges related to explainability, based on the National Institute of Standards' principles on explanation, meaning, accuracy, and limits of knowledge. A significant contribution of this article is the classification of XAI techniques into five categories: post-hoc explanations, intrinsic explanations, transparent explanations, interactive explanations, and hybrid explanations. In a practical application context, [Samek and Müller 2019] discuss the importance of interpretability and explainability in AI, especially in critical applications such as medicine. They argue that the lack of interpretability is a significant problem for deep learning techniques and also explore the idea that interpretability is a continuous and dynamic process, emphasizing the importance of collaboration between AI specialists and end users to create accurate and interpretable models.

Over the years, several techniques have been proposed to address the issue of explainability in machine learning models. For example, LIME, developed by [Ribeiro et al. 2016], is a technique that aims to make the predictions of machine learning models more understandable. It works by creating a simplified version of the complex model to explain each individual prediction. This simplified version, usually a linear model, allows the most important features that influenced the model's decision to be identified. LIME stands out for its versatility, as it can be applied to a wide variety of models. Similarly, [Lundberg and Lee 2017] developed SHAP, which uses concepts from game theory to assign each feature a value representing its contribution to the final prediction. This technique offers both local explanations (for a single prediction) and global explanations (for the model as a whole), enabling deeper and more consistent analysis.

In recent years, several tools have been developed to address explainability in machine learning models, including IBM AI Explainability 360 [Arya et al. 2022]. This tool provides a collection of algorithms, interpreters, and visualizations to help users understand and trust AI models. By integrating explainability techniques such as LIME and SHAP to meet different needs and contexts, it provides a platform for analyzing machine learning models. InterpretML [Nori et al. 2019] follows a similar proposal, designed to provide transparent explanations for machine learning models. Supporting techniques such as GAM (Generalized Additive Models) and SHAP, InterpretML enables detailed analysis of model predictions, helping developers better understand the models

they are building and using. The DALEX (Descriptive mAchine Learning EXplanations) tool [Biecek 2018] enables the analysis of machine learning models, offering approaches such as Partial Dependence Plots (PDPs), Accumulated Local Effects (ALE) Plots, Break Down Plots, Ceteris Paribus Plots, and Permutational Variable Importance. It stands out for its ability to provide detailed and accessible explanations, facilitating understanding of the decisions made by machine learning models.

Unlike approaches such as LIME and SHAP, which provide local and global explanations of model predictions, focusing on explaining specific predictions or the contribution of individual variables, IMAT stands out by providing an intuitive and comprehensive visualization of the entire decision-making process. It maps each step of data processing within models in detail, allowing developers to better understand the internal workings of models and identify possible improvements and necessary adjustments. The importance of IMAT lies in its ability to provide a complete and comprehensible view of the decision-making process of machine learning models, promoting trust and broader adoption of these technologies across sectors. Tools such as IBM AI Explainability 360 and DALEX offer interpretability features, but none combine the ability to visually detail the data flow with the ease of use that IMAT provides. IMAT was developed not only to fill gaps identified in the XAI literature, but also to offer a practical and accessible solution that can be applied in various sectors. By making machine learning models more interpretable, IMAT facilitates the ethical and transparent adoption of AI.

4. IMAT

IMAT is a tool designed to provide a comprehensible and detailed analysis of machine learning models. Specifically, the tool exclusively supports the MLP algorithm, focusing on providing a clear and detailed view of this type of model's decision-making process through an architecture composed of a backend implemented in Flask and a frontend developed in HTML, CSS, and JavaScript.

Flask is a Python micro web framework that provides the necessary tools to create web applications in a modular and extensible way [Grinberg 2018]. In this work, the framework is responsible for data processing, model training, and flowchart generation. The frontend offers a user-friendly interface for interacting with the tool, allowing users to upload datasets in CSV format, configure model parameters such as number of layers, activation functions, optimizers, and loss functions, as well as visualize the generated flowcharts. Using TensorFlow, an open-source platform for machine learning, and Keras, a high-level API that facilitates the construction and training of deep learning models [et al. 2015], the model training module enables the training of MLP models with customizable configurations. Thus, the model can be configured with different numbers of layers and units in each layer; activation functions such as ReLU (Rectified Linear Unit), Sigmoid, and Tanh (Hyperbolic Tangent); optimizers such as Adam and RMSprop (Root Mean Square Propagation); and loss functions such as MSE (Mean Squared Error) and MAE (Mean Absolute Error). During training, the model is adjusted to minimize the chosen loss function using the dataset provided by the user.

For each MLP layer, IMAT extracts intermediate outputs, enabling a detailed visualization of how the data are transformed throughout the network. This process involves creating an intermediate model that provides the outputs of each layer for the input data,

helping to understand the flow of information and the transformations applied at each stage. Using the open-source Graphviz library to generate structural graphs and diagram representations based on textual descriptions [Ellson et al. 2001], the tool generates two types of flowcharts: detailed and summarized. The detailed flowchart shows all mathematical operations, weights, biases, and activations in each layer, while the summarized flowchart provides an overview of the data flow through the model without including all detailed calculations. These flowcharts are fundamental to understanding the model's internal behavior and the contributions of each input variable.

The tool's development process followed several stages, from the initial design to implementation and testing. Figure 1 presents IMAT's intuitive and accessible design, developed to be used by users with different levels of knowledge in machine learning through an interface designed to facilitate data loading, model configuration, and result visualization, ensuring that users can easily configure and interpret the models generated by the tool. The tool underwent a series of tests to confirm the accuracy of the generated models and the clarity of the flowcharts. The tests included checks to confirm result consistency, load testing to observe system performance with large volumes of data, and usability evaluations to ensure the interface is intuitive and easy to navigate. In addition, specific tests were carried out to ensure that the tool could process large datasets efficiently, ensuring system responsiveness under different usage conditions.

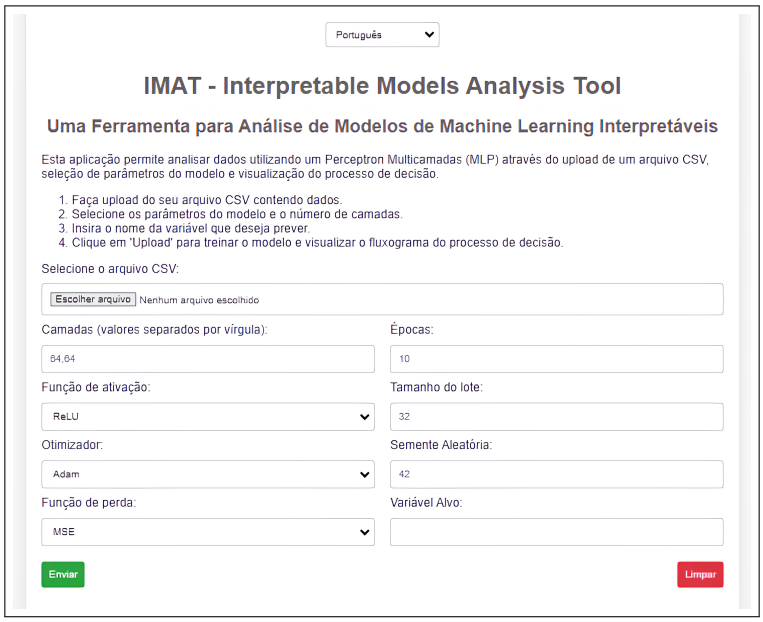

Figure 1. IMAT design.

5. Case Study

In this section, we present a case study that demonstrates the practical use of IMAT through a detailed analysis of the operation of an MLP-type neural network model applied to sentiment analysis of tweets, using the Sentiment140 dataset. We analyze examples of IMAT's application, compare the tool with other existing solutions, and interpret the results obtained, discussing the implications of the findings and the limitations of the study.

The MLP algorithm has a feedforward neural network architecture composed of multiple layers of neurons. In general terms, the MLP consists of an input layer, one or more hidden layers, and an output layer. Each neuron performs a linear operation, multiplying the inputs by weights and adding a bias, then applies a nonlinear activation function—such as ReLU, Sigmoid, or Tanh—to introduce complexity and enable the modeling of nonlinear relationships in the data. During training, the algorithm uses the backpropagation method to adjust weights and biases by minimizing a loss function, allowing the network to learn useful representations of the input data [Taud and Mas 2017].

The Sentiment140 dataset used in this case study consists of labeled tweets, where sentiments are converted into a binary classification task (positive or negative) [Go et al. 2009]. The texts underwent a preprocessing pipeline that included cleaning (removal of URLs, mentions, and stopwords) and transformation into numerical vectors using TF-IDF (Term Frequency–Inverse Document Frequency) with 1000 features. Due to processing limitations, a reduced sample of 1000 records was used for the training stage. The model was configured with one hidden layer of 50 neurons with ReLU activation and an output layer with 1 neuron and Sigmoid activation, which yields a probability representing confidence in the sentiment classification. Figure 2 depicts the summarized flowchart of the model's decision process, where it is possible to visualize the main data-processing steps up to the final prediction. The full model contains 50 neurons in the first layer; due to page limits, only the summarized flowchart with 5 neurons was used.[1]

The flowchart illustrates the model's decision process for identifying the sentiment contained in the sentence "Happy birthdayy! You are my hero, keep up the good work!". In addition to the original sentence, the input node displays the vectorized phrase through the TF-IDF features. In the hidden layer (Hidden Layer 1), which contains 50 neurons with ReLU activation, the flowchart details the calculations for some neurons. For example, for neuron 0, the linear value z_0 is calculated as 1.8391 from the combination of weights (such as -0.0528, -0.0123, 0.0310, -0.0326, and 0.0508) with the input and the addition of the bias (-0.0055). After applying the ReLU function, the neuron yields an activation of 1.8391, indicating a strong contribution to processing. Similarly, other neurons, such as neuron 1 (with $z_1 = 0.0287$) and neuron 2 (with $z_2 = 0.6505$), show how small variations in the calculations can result in low or moderate activations. Thus, the flowchart helps visualize the heterogeneity in the hidden layer's response to the same input, enabling identification of which neurons are effectively activated and which remain inactive.

In the output layer, composed of a single neuron with Sigmoid activation, the extracted information is consolidated to generate the final sentiment prediction. In the example presented, the flowchart indicates that the model classified the tweet as Positive with a confidence of 0.5322. This result is obtained after the data are processed by the hidden layer, where the multiplication, bias addition, and activation calculations are propagated to the final layer. The visualization provided by IMAT significantly contributes to understanding the model, as it allows the data flow and intermediate calculations to be viewed in detail. By showing, for example, how the same input produces varied responses among the neurons in the hidden layer (with some neurons exhibiting robust activations and others resulting in zero), the flowchart makes the internal heterogeneity evident and

[1] All versions are available in our repository.

enables the identification of improvement points, as well as facilitating the diagnosis of possible failures, the assessment of weights and biases, and the implementation of adjustments to improve model performance—making it easier to interpret the nuances present in the final decisions.

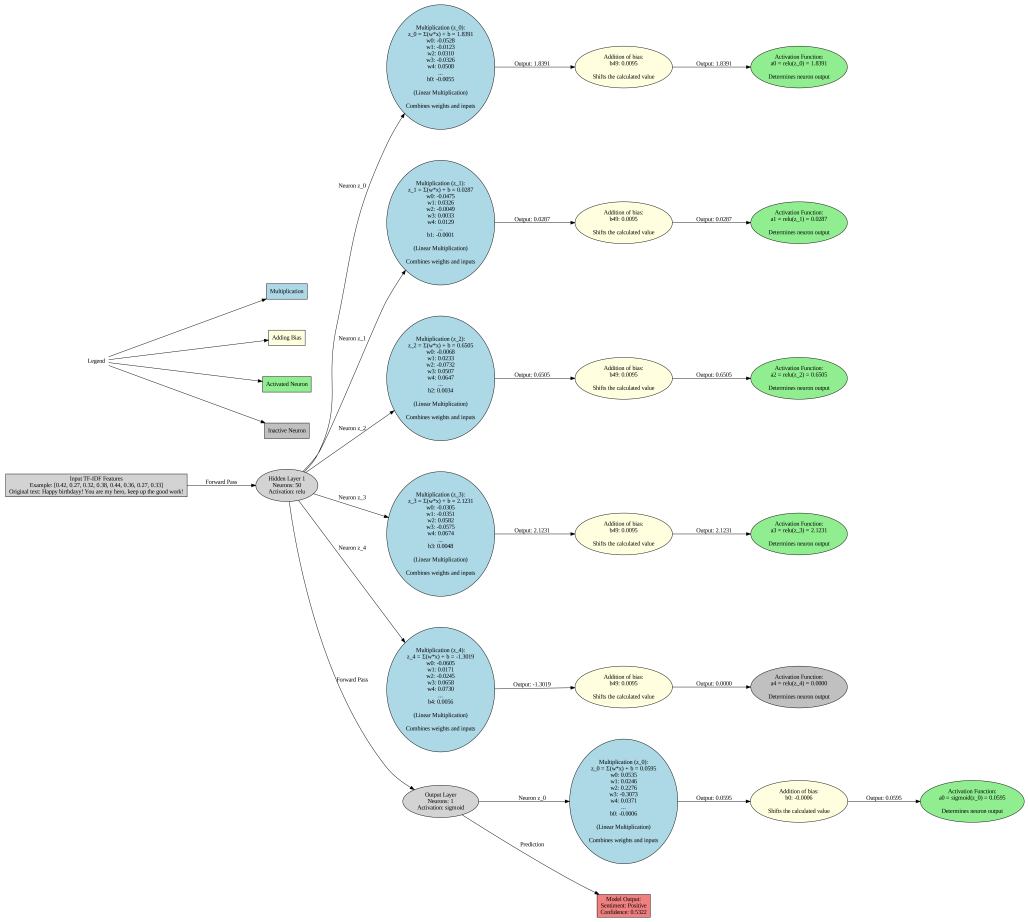

Figure 2. Summarized flowchart for the *Sentiment140* dataset.

To compare IMAT with other explainability tools, we analyzed its functionalities relative to LIME and SHAP. Although LIME and SHAP provide useful explanations for the predictions of machine learning models, IMAT stands out for its ability to generate detailed flowcharts that represent the model's complete decision-making process. While LIME and SHAP focus on local and global explanations of predictions, IMAT provides an intuitive and comprehensive visualization of the model's internal operations, facilitating understanding of the data flow and the transformations applied at each layer of the MLP.

When interpreting the results obtained with IMAT, we observed that the tool facilitated the identification of relevant features and the understanding of their contributions to the model's predictions. On platforms such as X, where public opinions and trends form rapidly, model transparency and interpretability are essential to ensure confidence in data-driven decisions, such as marketing strategies, public policies, or electoral campaigns. However, the study also revealed some limitations, as IMAT, in its current version, supports only the MLP algorithm, limiting its applicability to other types of machine learning models. In addition, the complexity of the flowcharts generated can increase sig-

nificantly with the growth in the number of layers and units in the model, which may hinder interpretation for very large models.

IMAT proved to be a valuable tool for interpreting MLP models, offering detailed visualizations that make it easier to understand the model's decision-making process. Compared with other explainability tools, IMAT provides an intuitive and comprehensive view of the model's internal operations, enabling users to understand not only the final predictions but also the complete path of the data from input to output. This characteristic is important in scenarios where transparency is critical, such as in the medical, financial, and legal domains. Although there are still opportunities to expand its capabilities—such as supporting model types beyond MLP—and to improve usability for extremely large models, IMAT already stands out for its visual and accessible focus. Thus, the results of this case study highlight the importance of interpretability in the adoption of AI technologies, suggesting future directions for the development of XAI tools that can serve a broader range of models and applications, promoting a more ethical and responsible adoption of artificial intelligence in diverse areas.

6. Conclusions

IMAT is a tool developed to facilitate the interpretation and analysis of machine learning models, specifically the MLP. Through a detailed case study, we demonstrated IMAT's applicability in the context of social networks by means of sentiment analysis of tweets, illustrating how the tool generates summarized flowcharts that clearly represent the model's decision-making process, showing how the model identifies and classifies the sentiment present in the analyzed message.

IMAT's goal is to transform complex models into comprehensible visual representations, making it easier to understand and analyze the models' internal processes in detail. The figures generated by the tool, as demonstrated in the example, highlight the main processing steps—from data input to the final prediction—promoting greater transparency and trust in machine learning models. The comparison with other explainability tools, such as LIME and SHAP, underscored IMAT's advantage in providing an intuitive and comprehensive view of the internal operations of models. Thus, IMAT's importance for the analysis of interpretable models is reaffirmed by its ability to provide detailed visualizations that not only facilitate understanding of the models' decision-making process, but also allow the identification of relevant features and an understanding of their contributions to predictions—an important capability in applications where transparency and interpretability are essential, such as in the medical, financial, and legal domains, where trust in models is fundamental for informed and responsible decision-making.

Although IMAT already presents itself as a robust and effective tool, there are opportunities to expand its capabilities. Future work may explore support for other types of machine learning models beyond MLP, as well as the integration of additional functionalities that can improve the tool's usability for extremely complex models. In addition, expanding IMAT to support other deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), aims to broaden its reach and applicability. Additionally, we plan to develop automated interpretation assistants to manage the complexity of large-scale models.

References

[Adadi and Berrada 2018] Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160.

[Agarwal and Das 2020] Agarwal, N. and Das, S. (2020). Interpretable machine learning tools: A survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1528–1534. IEEE.

[Alves and ANDRADE 2022] Alves, M. A. S. and ANDRADE, O. d. (2022). Da "caixa-preta" à "caixa de vidro": o uso da explainable artificial intelligence (xai) para reduzir a opacidade e enfrentar o enviesamento em modelos algorítmicos. *Direito Público*, 18(100).

[Angelov et al. 2021] Angelov, P. P., Soares, E. A., Jiang, R., Arnold, N. I., and Atkinson, P. M. (2021). Explainable artificial intelligence: an analytical review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 11(5):e1424.

[Arrieta et al. 2020] Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115.

[Arya et al. 2022] Arya, V., Bellamy, R. K., Chen, P.-Y., Dhurandhar, A., Hind, M., Hoffman, S. C., Houde, S., Liao, Q. V., Luss, R., Mojsilović, A., et al. (2022). Ai explainability 360: Impact and design. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12651–12657.

[Assis et al. 2023] Assis, A., Véras, D., and Andrade, E. (2023). Explainable artificial intelligence-an analysis of the trade-offs between performance and explainability. In *2023 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pages 1–6. IEEE.

[Bhattacharya 2022] Bhattacharya, A. (2022). *Applied Machine Learning Explainability Techniques: Make ML models explainable and trustworthy for practical applications using LIME, SHAP, and more*. Packt Publishing Ltd.

[Biecek 2018] Biecek, P. (2018). Dalex: Explainers for complex predictive models in r. *Journal of Machine Learning Research*, 19(84):1–5.

[Cortiz 2021] Cortiz, D. (2021). Inteligência artificial: conceitos fundamentais. *VAINZOF, Rony; GUTIERREZ, Adriei. Inteligência artificial: sociedade, economia e Estado. São Paulo: Thomson Reuters*, pages 45–60.

[Ellson et al. 2001] Ellson, J., Gansner, E. R., Koutsofios, E., North, S. C., and Woodhull, G. (2001). *Graphviz - Graph Visualization Software*. Software available from http://graphviz.org/.

[et al. 2015] et al., M. A. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org.

[Go et al. 2009] Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision.

[Gregor and Benbasat 1999] Gregor, S. and Benbasat, I. (1999). Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS quarterly*, pages 497–530.

[Grinberg 2018] Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. Sebastopol, CA. ISBN 978-1-491-95791-3.

[Kim et al. 2016] Kim, B., Khanna, R., and Koyejo, O. O. (2016). Examples are not enough, learn to criticize! criticism for interpretability. *Advances in neural information processing systems*, 29.

[Lundberg and Lee 2017] Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

[Meske and Bunde 2020] Meske, C. and Bunde, E. (2020). Transparency and trust in human-ai-interaction: The role of model-agnostic explanations in computer vision-based decision support. In *Artificial Intelligence in HCI: First International Conference, AI-HCI 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings 22*, pages 54–69. Springer.

[Miller 2019] Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38.

[Molnar 2020] Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.

[Nori et al. 2019] Nori, H., Jenkins, S., Koch, P., and Caruana, R. (2019). Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*.

[Paes et al. 2022] Paes, V., Araújo, D., Brito, K., and Andrade, E. (2022). Análise de sentimento em tweets relacionados ao desmatamento da floresta amazônica. In *Anais do XI Brazilian Workshop on Social Network Analysis and Mining*, pages 61–72, Porto Alegre, RS, Brasil. SBC.

[Ribeiro et al. 2016] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

[Salih et al. 2023] Salih, A., Raisi-Estabragh, Z., Galazzo, I. B., Radeva, P., Petersen, S. E., Menegaz, G., and Lekadir, K. (2023). Commentary on explainable artificial intelligence methods: Shap and lime. *arXiv preprint arXiv:2305.02012*.

[Samek and Müller 2019] Samek, W. and Müller, K.-R. (2019). Towards explainable artificial intelligence. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 5–22.

[Silva et al. 2021] Silva, H., Andrade, E., Araújo, D., and Dantas, J. (2021). Sentiment analysis of tweets related to sus before and during covid-19 pandemic. *IEEE Latin America Transactions*, 20(1):6–13.

[Søgaard 2023] Søgaard, A. (2023). On the opacity of deep neural networks. *Canadian Journal of Philosophy*, 53(3):224–239.

[Sundar 2020] Sundar, S. S. (2020). Rise of machine agency: A framework for studying the psychology of human–ai interaction (haii). *Journal of Computer-Mediated Communication*, 25(1):74–88.

[Taud and Mas 2017] Taud, H. and Mas, J.-F. (2017). Multilayer perceptron (mlp). In *Geomatic approaches for modeling land change scenarios*, pages 451–455. Springer.

IMAT: Uma Ferramenta para Análise de Modelos de Aprendizado de Máquina Interpretáveis

André Assis[1], **Jamilson Dantas** [2], **Ermeson Andrade**[1]

[1]Departamento de Computação – Universidade Federal Rural de Pernambuco (UFRPE)
Recife, PE – Brasil

{andre.assis, ermeson.andrade}@ufrpe.br

[2]Centro de Informática – Universidade Federal de Pernambuco
Recife, PE – Brasil

jrd@cin.ufpe.br

Abstract. *Transparency and interpretability of Artificial Intelligence (AI) and Machine Learning (ML) models are increasingly relevant in applications involving social network analysis and data mining. While advanced models, such as deep learning, provide robust solutions to complex problems, their growing complexity makes it difficult to understand the decisions they make. The lack of transparency can undermine user trust and hinder the widespread adoption of these technologies. To address this challenge, this paper presents IMAT (Interpretable Models Analysis Tool), a tool designed to generate flowcharts that map each stage of data processing in deep learning models. IMAT aims to provide a clear and accessible visualization of the data flow and internal operations of models, from data input to output generation, facilitating their interpretation. Additionally, this work discusses the architecture and functionalities of IMAT and demonstrates its application in sentiment analysis of tweets, using the MLP (MultiLayer Perceptron) algorithm, evaluating the implications and limitations of the obtained results.*

Resumo. *A transparência e interpretabilidade dos modelos de Inteligência Artificial (IA) e Machine Learning (ML) são cada vez mais relevantes em aplicações que envolvem análise de redes sociais e mineração de dados. Embora modelos avançados, como os de deep learning, ofereçam soluções robustas para problemas complexos, sua crescente complexidade dificulta a compreensão das decisões tomadas. A falta de transparência pode comprometer a confiança dos usuários e limitar a adoção dessas tecnologias. Para enfrentar esse desafio, este artigo apresenta a IMAT (Interpretable Models Analysis Tool), uma ferramenta desenvolvida para gerar fluxogramas que mapeiam cada etapa do processamento de dados em modelos de deep learning. A IMAT visa oferecer uma visualização clara e acessível do fluxo de dados e das operações internas dos modelos, desde a entrada até a geração da resposta, facilitando sua interpretação. Além disso, este trabalho discute a arquitetura e funcionalidades da IMAT e demonstra sua aplicação na análise de sentimentos em tweets, utilizando o algoritmo MLP (MultiLayer Perceptron), avaliando as implicações e limitações dos resultados obtidos.*

1. Introdução

A Inteligência Artificial (IA) e a aprendizagem de máquina têm se tornado fundamentais em diversas indústrias e tecnologia da informação em geral. Visando solucionar problemas complexos do mundo real, os modelos robustos de *deep learning* destacam-se por sua capacidade de realizar previsões precisas a partir de grandes volumes de dados. No entanto, a crescente complexidade desses modelos levanta questões críticas sobre a interpretabilidade e transparência das decisões tomadas, aspectos que são essenciais para a aceitação e confiabilidade da tecnologia, especialmente em áreas sensíveis como diagnósticos médicos e decisões financeiras. A falta de transparência nos modelos pode resultar em desconfiança por parte dos usuários finais, comprometendo a adoção generalizada dessas tecnologias. Além disso, a interpretabilidade é importante para identificar e corrigir vieses e erros nos modelos, garantindo a justiça e a precisão das previsões [Assis et al. 2023]. Essa necessidade de explicações que facilitem a compreensão do modelo se torna ainda mais evidente em aplicações que lidam com dados não estruturados e altamente subjetivos, como os encontrados em redes sociais.

As redes sociais, especialmente o Twitter (atualmente denominado X), tornaram-se ambientes ricos para a análise de dados textuais, possibilitando a exploração de opiniões públicas sobre uma ampla variedade de temas [Silva et al. 2021, Paes et al. 2022]. A análise de sentimentos com modelos de IA, que identifica e classifica emoções expressas em textos, tem se mostrado particularmente valiosa para compreender percepções sociais em tempo real. Aplicável a grandes volumes de *tweets*, essa abordagem fornece informações relevantes para a tomada de decisões em áreas como marketing, política e saúde pública. No entanto, a correta interpretação dos resultados desses modelos exige atenção à transparência e confiabilidade. Compreender os critérios que levaram a uma determinada classificação é fundamental para validar os resultados, detectar vieses e garantir a qualidade das inferências produzidas.

Um dos principais desafios enfrentados é a dificuldade em compreender e explicar os processos internos de modelos de *deep learning*, que frequentemente operam como "caixas-pretas", tornando os caminhos que levam a determinadas previsões difíceis de entender, até mesmo para especialistas [Alves and ANDRADE 2022]. Essa falta de transparência impede a validação e a confiança nas previsões, além de dificultar a identificação de vieses e erros potenciais. Portanto, é fundamental desenvolver ferramentas que proporcionem uma compreensão clara e acessível do processo decisório desses modelos. Tais ferramentas interpretativas desempenham um papel crucial na validação e melhoria contínua dos modelos de ML, fornecendo *feedback* essencial para ajustes e refinamentos [Cortiz 2021]. Ao oferecer uma visão detalhada dos mecanismos internos dos modelos, elas possibilitam que desenvolvedores identifiquem padrões inadequados, corrijam vieses e realizem ajustes precisos [Agarwal and Das 2020]. Além disso, a integração de ferramentas de interpretabilidade em ciclos regulares de revisão e atualização contribui para a criação de sistemas mais justos, confiáveis e alinhados às expectativas e necessidades dos usuários.

Diversas abordagens têm sido propostas para enfrentar os desafios da interpretabilidade em modelos de aprendizado de máquina, com destaque para técnicas como LIME (*Local Interpretable Model-agnostic Explanations*), SHAP (*SHapley Additive exPlanations*) e DALEX (*Descriptive mAchine Learning EXplanations*). Essas técnicas ofer-

ecem explicações locais que ajudam a entender o comportamento dos modelos em instâncias específicas, tornando visíveis seus mecanismos internos e contribuindo para maior transparência em diferentes contextos de aplicação [Salih et al. 2023]. Apesar desses avanços, ainda existe um descompasso entre essas soluções e as necessidades de usuários finais não técnicos, como analistas de negócio e profissionais de domínio, que frequentemente encontram dificuldade na execução de *scripts* ou na interpretação de métricas estatísticas. Para esse público, explicações em forma de tabelas de importância ou gráficos abstratos não são suficientes, tornando essencial uma visualização clara que comunique o raciocínio do modelo de forma imediata e acessível. Além disso, persiste uma lacuna significativa na integração de ferramentas que, além de fornecerem explicações, visualizem todo o processo de decisão de maneira compreensível para os usuários finais, facilitando a interpretação em ambientes industriais e acadêmicos [Bhattacharya 2022].

Para preencher essa lacuna, este trabalho propõe o IMAT *(Interpretable Models Analysis Tool)*, uma ferramenta desenvolvida para construir fluxogramas detalhados e resumidos que mapeiam cada etapa do processamento de dados em modelos de *deep learning*. O IMAT busca oferecer uma visualização clara e acessível do fluxo de dados e das operações internas dos modelos, desde a entrada dos dados até a geração da resposta, facilitando a compreensão e aumentando a transparência dos modelos. Ao fornecer uma representação gráfica e detalhada do funcionamento interno dos modelos, o IMAT visa não apenas aumentar a confiança dos usuários, mas também fornecer uma ferramenta poderosa para a análise e melhoria contínua dos modelos de ML, atendendo às necessidades de diferentes usuários. Além disso, este trabalho tem como objetivos avaliar a aplicabilidade do IMAT por meio de um estudo de caso, no qual analisamos o funcionamento de um modelo de rede neural do tipo MLP aplicado à análise de sentimentos de *tweets*, utilizando o dataset *Sentiment140*.

A estrutura do artigo é organizada da seguinte forma: a Seção 2 introduz o conceito de Inteligência Artificial Explicável, destacando sua importância e técnicas principais. A Seção 3 revisa os principais trabalhos relacionados à interpretabilidade em ML, justificando a necessidade da IMAT. A Seção 4 descreve as principais funcionalidades da IMAT. A Seção 5 apresenta um estudo de caso aplicado à análise de sentimentos de *tweets*. Por fim, a Seção 6 apresenta considerações finais e sugestões para pesquisas futuras.

2. Inteligência Artificial Explicável

A XAI, do inglês *eXplainable Artificial Intelligence* é um campo de pesquisa que busca desenvolver métodos e técnicas que permitam a compreensão e interpretação dos modelos de inteligência artificial por seres humanos. O seu principal objetivo é reduzir a opacidade dos modelos de IA, frequentemente referidos como "caixas-pretas", e torná-los mais transparentes e confiáveis para os usuários finais [Adadi and Berrada 2018]. A opacidade dos modelos de IA impede que os humanos entendam os processos decisórios subjacentes, o que pode levar à desconfiança e à relutância em adotar essas tecnologias [Meske and Bunde 2020]. A transparência, por outro lado, refere-se à capacidade de um sistema de IA de ser compreendido por seus usuários, enquanto a explicabilidade envolve a capacidade do sistema de fornecer justificativas compreensíveis para suas decisões e previsões [Gregor and Benbasat 1999].

A interpretabilidade e a explicabilidade em Aprendizado de Máquina, segundo autores como Miller, Kim e Molnar, referem-se à capacidade de um humano entender e prever consistentemente as decisões de um modelo, tornando-o transparente e compreensível [Miller 2019, Kim et al. 2016, Molnar 2020]. Essa característica é fundamental para aumentar a confiança dos usuários em sistemas de IA, especialmente em domínios críticos como saúde, onde decisões algorítmicas podem ter consequências significativas. A XAI permite identificar e corrigir vieses, melhorar o desempenho dos modelos e garantir que eles operem de maneira justa e ética, contribuindo para a construção de sistemas de IA mais confiáveis e responsáveis [Sundar 2020].

Os modelos de aprendizado de máquina variam em sua complexidade e, consequentemente, em sua capacidade de explicação. Modelos mais simples são geralmente mais transparentes, permitindo uma compreensão mais intuitiva de seu funcionamento. Em contraste, modelos mais complexos tendem a ser opacos, dificultando a interpretação direta de seus processos internos [Angelov et al. 2021]. A pesquisa em XAI busca desenvolver métodos que aumentem a transparência e a interpretabilidade de ambos os tipos de modelos, fornecendo informações detalhadas sobre suas operações internas e decisões [Arrieta et al. 2020]. Promovendo assim, a confiança dos usuários e facilitando a adoção de sistemas de IA, ao mesmo tempo que assegura a responsabilidade e a transparência em aplicações críticas [Adadi and Berrada 2018].

No contexto das redes sociais e da mineração de dados, a XAI desempenha um papel crucial ao garantir transparência e confiabilidade nas análises. Plataformas como X geram grandes volumes de dados textuais, utilizados para identificar tendências e compreender opiniões públicas. A análise de sentimentos, por exemplo, é amplamente aplicada em marketing, política e saúde pública, mas sua eficácia depende da interpretabilidade dos modelos utilizados. A opacidade dos modelos de deep learning pode comprometer a aceitação de suas inferências, especialmente quando impactam políticas públicas ou estratégias empresariais [Søgaard 2023]. Métodos de XAI permitem compreender os critérios que levam a determinadas classificações, identificar vieses e garantir que os modelos operem de maneira justa e alinhada às expectativas dos usuários.

Além de melhorar a qualidade das análises, a integração de técnicas explicáveis na mineração de dados em redes sociais amplia a adoção dessas tecnologias. Ferramentas que oferecem visualizações claras do processo decisório dos modelos possibilitam que especialistas e não especialistas interpretem melhor os resultados e tomem decisões mais embasadas [Arrieta et al. 2020]. Assim, ao promover maior transparência e compreensibilidade, a XAI fortalece o uso ético e responsável da inteligência artificial em ambientes altamente dinâmicos.

3. Trabalhos Relacionados

A crescente complexidade dos modelos de aprendizado de máquina gerou uma demanda significativa por ferramentas que facilitem a compreensão de suas operações internas. Considerando que ferramentas de XAI são essenciais para garantir transparência, confiança e adoção ética da IA em diversos setores, esta seção revisa os principais trabalhos e ferramentas relacionadas à interpretabilidade em modelos de aprendizado de máquina, destacando as contribuições e as justificativas para o desenvolvimento da ferramenta IMAT.

Em [Angelov et al. 2021], é apresentada uma revisão abrangente sobre a explicabilidade da inteligência artificial no contexto de aprendizagem profunda. O estudo aborda uma taxonomia detalhada e os principais desafios relacionados à explicabilidade, baseando-se nos princípios do *National Institute of Standards* sobre explicação, significado, precisão e limites do conhecimento. A contribuição significativa desse artigo é a classificação das técnicas de XAI em cinco categorias: explicações*post-hoc*, explicações intrínsecas, explicações transparentes, explicações interativas e explicações híbridas. No contexto da aplicação prática, [Samek and Müller 2019] discutem a importância da interpretabilidade e explicabilidade na IA, especialmente em aplicações críticas como a medicina. Eles argumentam que a falta de interpretabilidade é um problema significativo para as técnicas de aprendizado profundo e também explora a ideia de que a interpretabilidade é um processo contínuo e dinâmico, enfatizando a importância da colaboração entre especialistas em IA e usuários finais para criar modelos precisos e interpretáveis.

Ao longo dos anos, várias técnicas foram propostas para abordar a questão da explicabilidade em modelos de aprendizado de máquina. Por exemplo, o LIME, desenvolvido por [Ribeiro et al. 2016], é uma técnica que visa tornar as previsões de modelos de Aprendizado de Máquina mais compreensíveis. Ele funciona criando uma versão simplificada do modelo complexo para explicar cada previsão individual. Essa versão simplificada, geralmente um modelo linear, permite identificar as características mais importantes que influenciaram a decisão do modelo. O LIME se destaca por sua versatilidade, podendo ser aplicado a uma ampla variedade de modelos. Semelhantemente, [Lundberg and Lee 2017] desenvolveram o SHAP, que utiliza conceitos da teoria dos jogos para atribuir a cada característica um valor que representa sua contribuição para a previsão final. Essa técnica oferece explicações tanto locais (para uma única previsão) quanto globais (para o modelo como um todo), permitindo uma análise mais profunda e consistente.

Nos últimos anos, diversas ferramentas foram desenvolvidas para lidar com a explicabilidade em modelos de aprendizado de máquina, incluindo a IBM *AI Explainability 360* [Arya et al. 2022]. Essa ferramenta oferece uma coleção de algoritmos, interpretadores e visualizações para auxiliar os usuários na compreensão e confiança em modelos de IA. Integrando técnicas de explicabilidade, como *LIME* e *SHAP*, com o objetivo de atender a diferentes necessidades e contextos, proporciona uma plataforma para a análise de modelos de aprendizado de máquina. A InterpretML [Nori et al. 2019] segue uma proposta semelhante, desenvolvida para fornecer explicações transparentes para modelos de aprendizado de máquina. Suportando técnicas como GAM (Generalized Additive Models) e SHAP, o InterpretML permite uma análise detalhada das previsões dos modelos, ajudando os desenvolvedores a compreender melhor os modelos que estão construindo e utilizando. A ferramenta *DALEX (Descriptive mAchine Learning EXplanations)* [Biecek 2018] permite a análise de modelos de aprendizado de máquina, oferecendo abordagens como *Partial Dependence Plots* (PDPs), *Accumulated Local Effects* (ALE) *Plots*, *Break Down Plots*, *Ceteris Paribus Plots* e *Permutational Variable Importance*. Destaca-se por sua capacidade de fornecer explicações detalhadas e acessíveis, facilitando a compreensão das decisões tomadas pelos modelos de aprendizado de máquina.

Ao contrário de abordagens como *LIME* e *SHAP*, que oferecem explicações locais e globais das previsões dos modelos, focando em explicar previsões específicas ou a

contribuição de variáveis individuais, o IMAT se destaca por fornecer uma visualização intuitiva e abrangente de todo o processo de decisão. Ele mapeia detalhadamente cada etapa do processamento de dados dentro dos modelos, permitindo que os desenvolvedores compreendam melhor o funcionamento interno dos modelos e identifiquem possíveis melhorias e ajustes necessários. De acordo com o princípio "overview first, zoom & filter, details-on-demand" [Shneiderman 2003], considera-se uma visualização intuitiva aquela que permite ao usuário compreender a estrutura geral antes de explorar detalhes específicos, sem exigir esforço cognitivo excessivo. No IMAT, esse princípio se materializa da seguinte forma: (i) as camadas da rede são codificadas por elipses cinzas; (ii) cada neurônio é representado por uma elipse azul contendo o cálculo dos pesos e elipses amarelas contendo a adição do *bias*; (iii) a ativação (cor verde) ou inativação (cor cinza) do neurônio; (iv) retângulo vermelho contendo o resultado final. Essa combinação de codificação visual permite ao usuário obter uma visão global imediata e, em seguida, investigar seletivamente as partes mais relevantes do fluxo de informações.

A importância do IMAT reside em sua capacidade de fornecer uma visão completa e compreensível do processo de tomada de decisão dos modelos de aprendizado de máquina, promovendo confiança e uma adoção mais ampla dessas tecnologias em diversos setores. Ferramentas como *IBM AI Explainability 360* e *DALEX* oferecem funcionalidades de interpretabilidade, mas nenhuma combina a capacidade de detalhar visualmente o fluxo de dados com a facilidade de uso que o IMAT proporciona. O IMAT foi desenvolvido não apenas para preencher lacunas identificadas na literatura de XAI, mas também para oferecer uma solução prática e acessível que pode ser aplicada em diversos setores. Ao tornar os modelos de aprendizado de máquina mais interpretáveis, o IMAT facilita a adoção ética e transparente da IA.

4. IMAT

O IMAT[1] é uma ferramenta desenvolvida para oferecer uma análise compreensível e detalhada de modelos de aprendizado de máquina. Atualmente, a ferramenta oferece suporte exclusivo ao algoritmo MLP, focando em fornecer uma visão clara e detalhada do processo de decisão deste tipo de modelo através de uma arquitetura composta por um *backend* implementado em *Flask* e um *frontend* desenvolvido em HTML, CSS e JavaScript.

O *Flask* é um *framework* de *micro web* em Python, que fornece as ferramentas necessárias para criar aplicativos web de forma modular e extensível [Grinberg 2018]. No IMAT, o *Flask* é responsável pelo processamento dos dados, treinamento do modelo e geração dos fluxogramas. O *frontend* oferece uma interface amigável para interação com a ferramenta, permitindo aos usuários carregar conjuntos de dados em formato CSV, configurar parâmetros do modelo, como número de camadas, funções de ativação, otimizadores e funções de perda, além de visualizar os fluxogramas gerados. Utilizando *TensorFlow* que é uma plataforma de código aberto para aprendizado de máquina e *Keras* que é uma API de alto nível que facilita a construção e treinamento de modelos de aprendizado profundo [et al. 2015], o módulo de treinamento de modelos permite o treinamento de modelos MLP com configurações personalizáveis. Assim, o modelo pode ser configurado com diferentes números de camadas e unidades em cada camada, funções de ativação como *ReLU (Rectified Linear Unit)*, *Sigmoid* e *Tanh (Hyperbolic Tangent)*, otimizadores

[1]O código-fonte está disponível em nosso repositório.

como *Adam* e *RMSprop (Root Mean Square Propagation)*, e funções de perda como *MSE (Mean Squared Error)* e *MAE (Mean Absolute Error)*. Durante o treinamento, o modelo é ajustado para minimizar a função de perda escolhida, utilizando o conjunto de dados fornecido pelo usuário.

Para cada camada do MLP, a IMAT extrai as saídas intermediárias, permitindo uma visualização detalhada de como os dados são transformados ao longo da rede. Esse processo envolve a criação de um modelo intermediário que fornece as saídas de cada camada para os dados de entrada, ajudando a compreender o fluxo de informação e as transformações aplicadas a cada etapa. Utilizando a biblioteca de código aberto *Graphviz* para gerar gráficos estruturais e representações em diagramas baseados em descrições textuais [Ellson et al. 2001], a ferramenta gera dois tipos de fluxogramas: detalhado e resumido. O fluxograma detalhado mostra todas as operações matemáticas, pesos, *bias* e ativações em cada camada, enquanto o fluxograma resumido fornece uma visão geral do fluxo de dados pelo modelo, sem incluir todos os cálculos detalhados. Esses fluxogramas são fundamentais para entender o comportamento interno do modelo e as contribuições de cada variável de entrada.

O custo de geração de fluxogramas na IMAT cresce linearmente com o número de pesos representados. Para um MLP com L camadas e N_i neurônios na i-ésima camada, a versão detalhada percorre todos os pesos do modelo, dados por:

$$W = \sum_{i=1}^{L} N_{i-1} N_i,$$

sendo que, para cada peso, são criados nós e arestas correspondentes às operações de multiplicação, soma do viés e aplicação da função de ativação. Esse processo resulta em um custo de tempo e memória proporcional a $O(W)$.

Na versão resumida, a visualização é limitada a K_d neurônios por camada, o que reduz significativamente a complexidade para:

$$O(L \cdot K_d),$$

que se simplifica para $O(L)$ quando K_d é constante, uma vez que o número total de elementos no grafo deixa de depender do tamanho real da rede.

O processo de desenvolvimento da ferramenta seguiu várias etapas, desde o design inicial até a implementação e testes realizados. A Figura 1 apresenta o *design* intuitivo e acessível da IMAT, desenvolvida para ser utilizada por usuários com diferentes níveis de conhecimento em aprendizado de máquina através de uma interface projetada para facilitar a carga de dados, configuração do modelo e visualização dos resultados, garantindo que os usuários possam facilmente configurar e interpretar os modelos gerados pela ferramenta. A ferramenta foi submetida a uma série de testes para confirmar a precisão dos modelos gerados e a clareza dos fluxogramas. Os testes incluíram verificações para confirmar a consistência dos resultados, testes de carga para observar o desempenho do sistema com grandes volumes de dados, e avaliações de usabilidade para certificar que a interface é intuitiva e fácil de navegar. Além disso, foram feitos testes específicos para garantir que a ferramenta pudesse processar grandes conjuntos de dados de maneira eficiente, assegurando a capacidade de resposta do sistema em diferentes condições de uso.

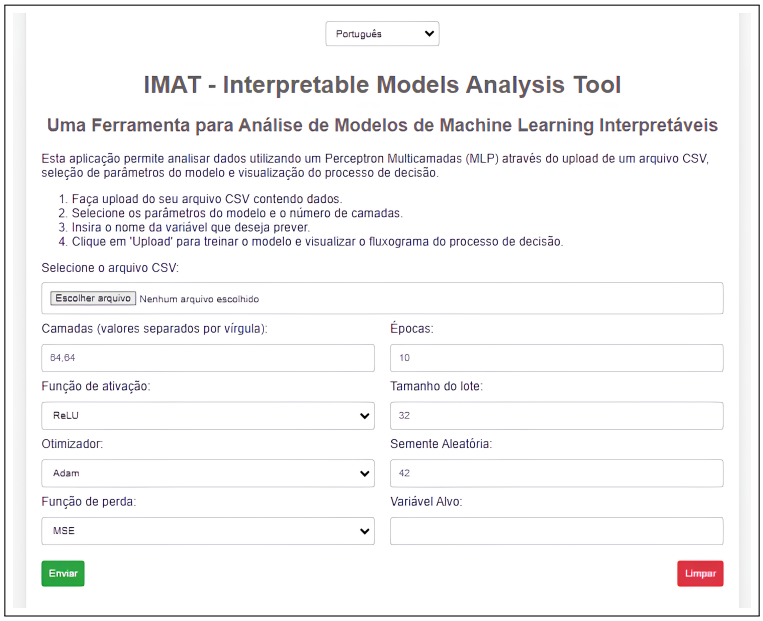

Figure 1. Design da IMAT.

Por fim, o fluxo operacional da IMAT organiza-se em seis etapas sucessivas: (i) o usuário realiza o upload do conjunto de dados em formato CSV; (ii) define, por meio da interface, os hiperparâmetros do MLP (número de camadas e neurônios, funções de ativação, otimizador, taxa de aprendizagem etc.); (iii) a ferramenta executa o treinamento do modelo, acompanhando as métricas e a validação; (iv) ao término do treinamento, são extraídas as ativações e os pesos de cada camada; (v) essas informações são então enviadas ao módulo de visualização, que gera automaticamente um fluxograma detalhado e uma versão resumida com agrupamentos.

5. Estudo de Caso

Nesta seção, apresentamos um estudo de caso que demonstra o uso prático da IMAT através de uma análise detalhada do funcionamento de um modelo de rede neural do tipo MLP aplicado à análise de sentimentos de *tweets*, utilizando o dataset *Sentiment140*. Analisamos exemplos de aplicação da IMAT, comparamos a ferramenta com outras soluções existentes e interpretamos os resultados obtidos, discutindo as implicações das descobertas e as limitações do estudo.

O algoritmo MLP possui uma arquitetura de rede neural *feedforward* composta por multiplas camadas de neurônios. Em termos gerais, o MLP consiste em uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Cada neurônio realiza uma operação linear, multiplicando as entradas por pesos e somando um viés, e em seguida aplica uma função de ativação nao linear, como por exemplo a *ReLU*, *Sigmoid* ou *Tanh*, para introduzir complexidade e permitir a modelagem de relações não lineares entre os dados. Durante o treinamento, o algoritmo utiliza o método de retropropagação para ajustar os pesos e vieses, minimizando uma função de perda, o que permite que a rede aprenda representações úteis dos dados de entrada [Taud and Mas 2017].

O conjunto de dados *Sentiment140* utilizado nesse estudo de caso, consiste em *tweets* rotulados, onde os sentimentos são convertidos para uma tarefa de classificação

binária (positivo ou negativo) [Go et al. 2009]. Os textos passaram por um processo de pré-processamento que incluiu limpeza (remoção de URLs, menções, *emojis* e *stopwords*) e a transformação em vetores numéricos utilizando TF-IDF (*Term Frequency-Inverse Document Frequency*) com 1000 atributos. Devido a limitação de processamento, foi utilizada uma amostra reduzida de 1000 registros para a etapa de treinamento. O modelo foi configurado com uma camada oculta de 50 neurônios com função de ativação *ReLU* e uma camada de saída com 1 neurônio e função *Sigmoid*, que gera uma probabilidade representando a confianca na classificação do sentimento. A Figura 2 representa o fluxograma resumido do processo decisório do modelo, onde é possível visualizar as principais etapas de processamento dos dados até a obtenção da previsão final. O modelo completo contêm 50 neurônios na primeira camada, devido a limitação de páginas foi utilizado apenas o fluxograma resumido com 5 neurônios.[2]

O fluxograma ilustra o processo decisório do modelo para identificar o sentimento contido na frase *"Happy birthdayy! You are my hero, keep up the good work!"*. Além da frase original, o nó de entrada exibe a frase vetorizada através das features TF-IDF. Na camada oculta (Hidden Layer 1), que contém 50 neurônios com ativação *ReLU*, o fluxograma detalha os cálculos de alguns neurônios. Por exemplo, para o neurônio 0, o valor linear z_0 é calculado como 1.8391, a partir da combinação dos pesos (como -0.0528, -0.0123, 0.0310, -0.0326 e 0.0508) com a entrada e a soma do viés (-0.0055). Após a aplicação da função *ReLU*, o neurônio gera uma ativação de 1.8391, indicando forte contribuição para o processamento. De forma semelhante, outros neurônios, como o neurônio 1 (com $z_1 = 0.0287$) e o neurônio 2 (com $z_2 = 0.6505$), mostram como pequenas variações nos cálculos podem resultar em ativações baixas ou moderadas. Assim, o fluxograma contribui para a visualização da heterogeneidade na resposta da camada oculta ao receber o mesmo input, permitindo identificar quais neurônios são efetivamente ativados e quais permanecem inativos.

Na camada de saída, composta por um único neurônio com ativação *Sigmoid*, as informações extraídas são consolidadas para gerar a predição final do sentimento. No exemplo apresentado, o fluxograma indica que o modelo classificou o *tweet* como *Positive* com uma confiança de 0.5322. Esse resultado é obtido após o processamento dos dados pela camada oculta, onde os cálculos de multiplicação, soma de viés e ativação são propagados para a camada final. A visualização fornecida pela IMAT contribui significativamente para a compreensão do modelo, pois permite visualizar o fluxo de dados e os cálculos intermediários de forma detalhada. Ao mostrar, por exemplo, como a mesma entrada gera respostas variadas entre os neurônios da camada oculta (com alguns neurônios apresentando ativações robustas e outros resultando em zero), o fluxograma torna evidente a heterogeneidade interna e possibilita a identificação de pontos de melhoria, além de facilitar o diagnóstico de possíveis falhas, assim como a avaliação dos pesos e vieses e a realização de ajustes para aprimorar a performance do modelo, facilitando a interpretação das nuances presentes nas decisões finais.

Para comparar a IMAT com outras ferramentas de explicabilidade, analisamos suas funcionalidades em relação ao LIME e ao SHAP. Embora LIME e SHAP forneçam explicações úteis sobre as previsões de modelos de aprendizado de máquina, a IMAT se destaca por sua capacidade de gerar fluxogramas detalhados que representam o processo

[2]Todas as versões estão disponíveis em nosso repositório.

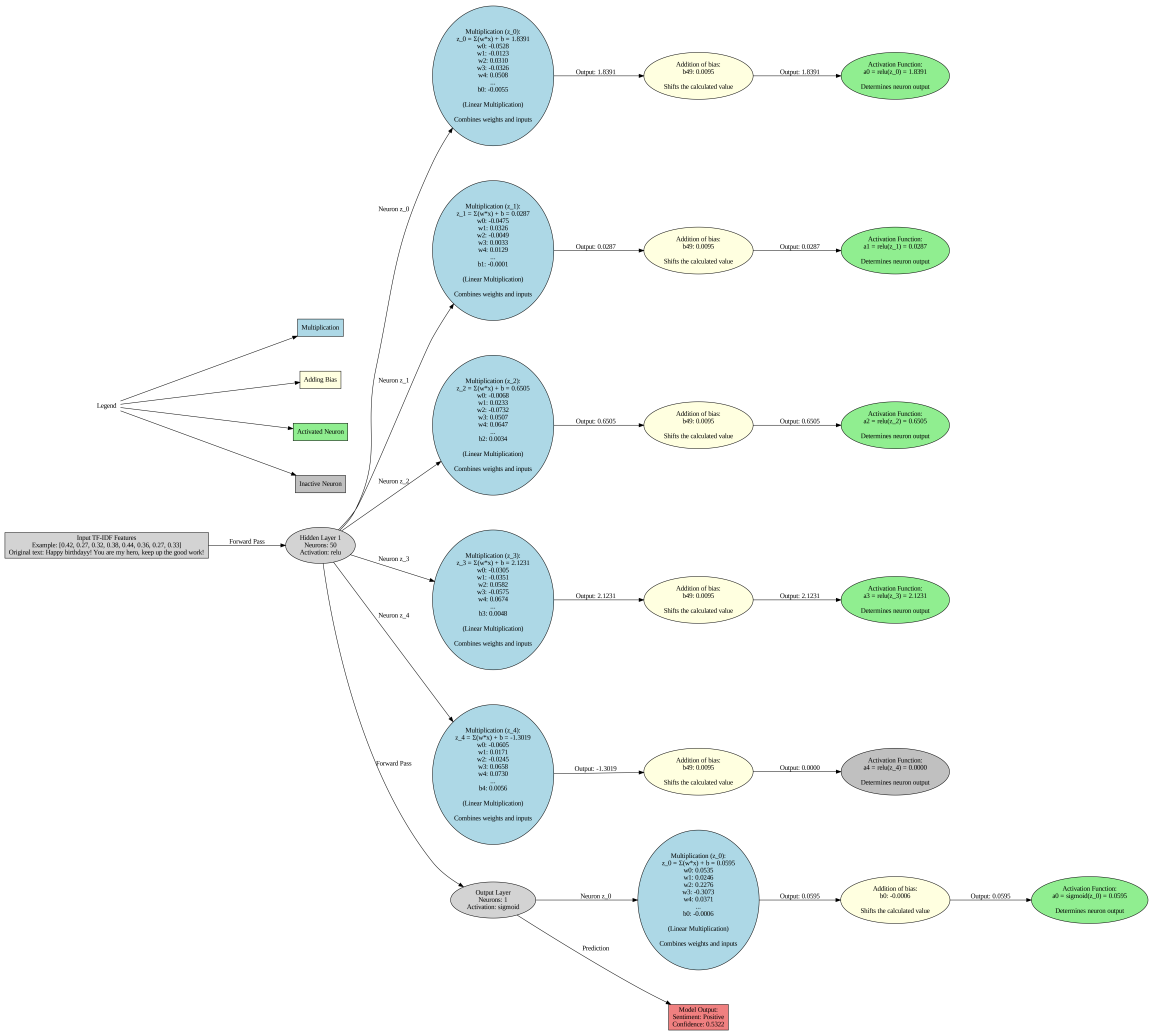

Figure 2. Fluxograma resumido para o conjunto de dados *Sentiment140*.

de decisão completo do modelo. Enquanto LIME e SHAP se concentram em explicações locais e globais das previsões, a IMAT fornece uma visualização intuitiva e abrangente das operações internas do modelo, facilitando a compreensão do fluxo de dados e das transformações aplicadas em cada camada do MLP.

Ao interpretar os resultados obtidos com a IMAT, observamos que a ferramenta facilitou a identificação de características relevantes e o entendimento das contribuições dessas características para as previsões do modelo. Em plataformas como o X, onde as opiniões públicas e as tendências se formam rapidamente, a transparência e a interpretabilidade dos modelos são essenciais para garantir a confiança nas decisões baseadas em dados, como estratégias de marketing, políticas públicas ou campanhas eleitorais. No entanto, o estudo também revelou algumas limitações, pois a IMAT, na versão atual, suporta apenas o algoritmo MLP, limitando sua aplicabilidade a outros tipos de modelos de aprendizado de máquina. Além disso, a complexidade dos fluxogramas gerados pode aumentar significativamente com o crescimento do número de camadas e unidades no modelo, o que pode dificultar a interpretação em modelos muito grandes.

A IMAT demonstrou ser uma ferramenta valiosa para a interpretação de modelos

de MLP, oferecendo visualizações detalhadas que facilitam a compreensão do processo de decisão do modelo, pois quando comparada a outras ferramentas de explicabilidade, a IMAT proporciona uma visão intuitiva e completa das operações internas do modelo, permitindo que os usuários compreendam não apenas as previsões finais, mas também o caminho completo dos dados desde a entrada até a saída. Essa característica é importante em cenários onde a transparência é fundamental, como na área médica, financeira e jurídica. Embora ainda existam oportunidades para expandir suas capacidades, como o suporte a outros tipos de modelos além do MLP, e melhorar a usabilidade em modelos extremamente complexos, a IMAT já se destaca por seu enfoque visual e acessível. Assim, os resultados deste estudo de caso destacam a importância da interpretabilidade na adoção de tecnologias de IA, sugerindo direções futuras para o desenvolvimento de ferramentas de XAI que possam atender a uma gama mais ampla de modelos e aplicações, promovendo uma adoção mais ética e responsável da inteligência artificial em diversas áreas.

6. Conclusões

A IMAT é uma ferramenta desenvolvida para facilitar a interpretação e análise de modelos de aprendizado de máquina, especificamente o MLP. Através de um estudo de caso detalhado, demonstramos a aplicabilidade da IMAT no contexto das redes sociais, através da análise de sentimentos de *tweets*, ilustrando como a ferramenta gera fluxogramas resumidos que representam claramente o processo de tomada de decisão do modelo, mostrando como o modelo identifica e classifica o sentimento presente na mensagem analisada.

O objetivo da IMAT é transformar modelos complexos em representações visuais compreensíveis, facilitando a compreensão e a análise detalhada dos processos internos dos modelos. As figuras geradas pela ferramenta, como demonstrado no exemplo, destacam as principais etapas de processamento, desde a entrada dos dados até a previsão final, promovendo uma maior transparência e confiança nos modelos de aprendizado de máquina. A comparação com outras ferramentas de explicabilidade, como LIME e SHAP, ressaltou a vantagem da IMAT em proporcionar uma visão abrangente e intuitiva das operações internas dos modelos. Assim, a importância da IMAT para a análise de modelos interpretáveis é reafirmada pela sua capacidade de fornecer visualizações detalhadas que não apenas facilitam a compreensão do processo de decisão dos modelos, mas também permitem identificar características relevantes e entender suas contribuições para as previsões, sendo importante em aplicações onde a transparência e a interpretabilidade são essenciais, como nas áreas médica, financeira e jurídica, onde a confiança nos modelos é fundamental para a tomada de decisões informadas e responsáveis.

Embora a IMAT já se apresente como uma ferramenta robusta e eficaz, existem oportunidades para expandir suas capacidades. Trabalhos futuros podem explorar testes com grandes volumes de dados, suporte a outros tipos de modelos de aprendizado de máquina além do MLP, bem como a integração de funcionalidades adicionais que possam melhorar a usabilidade da ferramenta em modelos extremamente complexos. Além disso, a expansão da IMAT para suportar outros modelos de aprendizado profundo, como redes neurais convolucionais (CNNs) e redes neurais recorrentes (RNNs), visa ampliar seu alcance e aplicabilidade. Adicionalmente, planeja-se desenvolver assistentes automáticos de interpretação para gerenciar a complexidade dos modelos em larga escala.

References

[Adadi and Berrada 2018] Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160.

[Agarwal and Das 2020] Agarwal, N. and Das, S. (2020). Interpretable machine learning tools: A survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1528–1534. IEEE.

[Alves and ANDRADE 2022] Alves, M. A. S. and ANDRADE, O. d. (2022). Da "caixa-preta" à "caixa de vidro": o uso da explainable artificial intelligence (xai) para reduzir a opacidade e enfrentar o enviesamento em modelos algorítmicos. *Direito Público*, 18(100).

[Angelov et al. 2021] Angelov, P. P., Soares, E. A., Jiang, R., Arnold, N. I., and Atkinson, P. M. (2021). Explainable artificial intelligence: an analytical review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 11(5):e1424.

[Arrieta et al. 2020] Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115.

[Arya et al. 2022] Arya, V., Bellamy, R. K., Chen, P.-Y., Dhurandhar, A., Hind, M., Hoffman, S. C., Houde, S., Liao, Q. V., Luss, R., Mojsilović, A., et al. (2022). Ai explainability 360: Impact and design. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12651–12657.

[Assis et al. 2023] Assis, A., Véras, D., and Andrade, E. (2023). Explainable artificial intelligence-an analysis of the trade-offs between performance and explainability. In *2023 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pages 1–6. IEEE.

[Bhattacharya 2022] Bhattacharya, A. (2022). *Applied Machine Learning Explainability Techniques: Make ML models explainable and trustworthy for practical applications using LIME, SHAP, and more*. Packt Publishing Ltd.

[Biecek 2018] Biecek, P. (2018). Dalex: Explainers for complex predictive models in r. *Journal of Machine Learning Research*, 19(84):1–5.

[Cortiz 2021] Cortiz, D. (2021). Inteligência artificial: conceitos fundamentais. *VAINZOF, Rony; GUTIERREZ, Adriei. Inteligência artificial: sociedade, economia e Estado. São Paulo: Thomson Reuters*, pages 45–60.

[Ellson et al. 2001] Ellson, J., Gansner, E. R., Koutsofios, E., North, S. C., and Woodhull, G. (2001). *Graphviz - Graph Visualization Software*. Software available from http://graphviz.org/.

[et al. 2015] et al., M. A. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org.

[Go et al. 2009] Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision.

[Gregor and Benbasat 1999] Gregor, S. and Benbasat, I. (1999). Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS quarterly*, pages 497–530.

[Grinberg 2018] Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. Sebastopol, CA. ISBN 978-1-491-95791-3.

[Kim et al. 2016] Kim, B., Khanna, R., and Koyejo, O. O. (2016). Examples are not enough, learn to criticize! criticism for interpretability. *Advances in neural information processing systems*, 29.

[Lundberg and Lee 2017] Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

[Meske and Bunde 2020] Meske, C. and Bunde, E. (2020). Transparency and trust in human-ai-interaction: The role of model-agnostic explanations in computer vision-based decision support. In *Artificial Intelligence in HCI: First International Conference, AI-HCI 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings 22*, pages 54–69. Springer.

[Miller 2019] Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38.

[Molnar 2020] Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.

[Nori et al. 2019] Nori, H., Jenkins, S., Koch, P., and Caruana, R. (2019). Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*.

[Paes et al. 2022] Paes, V., Araújo, D., Brito, K., and Andrade, E. (2022). Análise de sentimento em tweets relacionados ao desmatamento da floresta amazônica. In *Anais do XI Brazilian Workshop on Social Network Analysis and Mining*, pages 61–72, Porto Alegre, RS, Brasil. SBC.

[Ribeiro et al. 2016] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

[Salih et al. 2023] Salih, A., Raisi-Estabragh, Z., Galazzo, I. B., Radeva, P., Petersen, S. E., Menegaz, G., and Lekadir, K. (2023). Commentary on explainable artificial intelligence methods: Shap and lime. *arXiv preprint arXiv:2305.02012*.

[Samek and Müller 2019] Samek, W. and Müller, K.-R. (2019). Towards explainable artificial intelligence. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 5–22.

[Shneiderman 2003] Shneiderman, B. (2003). The eyes have it: A task by data type taxonomy for information visualizations. In *The craft of information visualization*, pages 364–371. Elsevier.

[Silva et al. 2021] Silva, H., Andrade, E., Araújo, D., and Dantas, J. (2021). Sentiment analysis of tweets related to sus before and during covid-19 pandemic. *IEEE Latin America Transactions*, 20(1):6–13.

[Søgaard 2023] Søgaard, A. (2023). On the opacity of deep neural networks. *Canadian Journal of Philosophy*, 53(3):224–239.

[Sundar 2020] Sundar, S. S. (2020). Rise of machine agency: A framework for studying the psychology of human–ai interaction (haii). *Journal of Computer-Mediated Communication*, 25(1):74–88.

[Taud and Mas 2017] Taud, H. and Mas, J.-F. (2017). Multilayer perceptron (mlp). In *Geomatic approaches for modeling land change scenarios*, pages 451–455. Springer.

APPENDIX C – Exploring Explainability in Machine Learning Models for Software Aging Detection

Assis et al. «Exploring Explainability in Machine Learning Models for Software Aging Detection». In: *Workshop on Software Aging and Rejuvenation (WoSAR)*, 2025.

Author's contribution: The author was responsible for implementing the ISAAT tool, designing the running the experiments, processing the collected data, generating the visual and textual artifacts (including tables, charts, and explanation outputs), and drafting the original manuscript.

Exploring Explainability in Machine Learning Models for Software Aging Detection

André Assis
Federal Rural University of Pernambuco
Recife, Brazil
andre.assis@ufrpe.br

Fumio Machida
University of Tsukuba
Tsukuba, Japan
machida@cs.tsukuba.ac.jp

Ermeson Andrade
Federal Rural University of Pernambuco
Recife, Brazil
ermeson.andrade@ufrpe.br

***Abstract*—Software aging, a phenomenon where long-running software systems degrade in performance or become prone to failures over time, presents a critical challenge in maintaining the reliability of software systems. In recent years, machine learning models have shown promise in detecting software aging early, enabling proactive measures to mitigate its effects. However, the black-box nature of many machine learning techniques raises concerns regarding their interpretability, particularly in critical environments like healthcare systems, financial systems, and autonomous vehicles. This paper highlights the importance of explainability in deep learning models, which are a type of machine learning, for software aging detection, emphasizing that interpretability is essential for building trust in their predictions. We introduce the Interpretable Software Aging Analysis Tool (ISAAT) as a solution. ISAAT offers comprehensive analysis and visualization of a Multilayer Perceptron (MLP) by using flowcharts to illustrate model structure and operations, and large language models (LLMs) to generate detailed, human-readable explanations of model decisions. The results demonstrate that ISAAT enhances the transparency of machine learning models, making them more interpretable and promoting greater trust in their application for software aging detection.**

***Index Terms*—Explainability, LLMs, Machine Learning, Software Aging.**

I. Introduction

Software aging is a phenomenon that adversely affects software reliability over time, leading to performance degradation and even system failures [1]. When software systems operate for extended periods, errors caused by aging-related bugs can accumulate in the execution environment. Common software aging issues include memory leaks and round-off errors, which both contribute to performance degradation and heighten the risk of system failures. Thus, early detection of software aging is essential to ensure operational reliability and efficiency, particularly in critical applications such as healthcare, finance, and transportation systems, where failures can have severe consequences [2].

To detect software aging in a timely manner, machine learning models have shown significant potential in previous studies [3]–[5]. However, the black-box nature of many machine learning models raises concerns about the reasoning behind their aging detection decisions. For instance, in a healthcare application, a machine learning model might predict the likelihood of system failure due to aging in a patient monitoring system based on historical data. However, if system administrators cannot understand how the model makes the prediction, they may hesitate to take proactive action based on its recommendation, potentially missing the opportunity to recover the system and compromising patient safety. In contexts where trust and transparency are important, such as in the maintenance of critical systems, the lack of clear explanations of machine learning-based decisions can limit the adoption and effectiveness of these solutions. Addressing the explainability of machine learning models in aging detection is a crucial issue, yet it has not been widely investigated in the literature.

In recent years, various approaches have been proposed to enhance the explainability of machine learning models [6], [7]. Techniques such as LIME (Local Interpretable Model-agnostic Explanations) [6] and SHAP (SHapley Additive exPlanations) [7] aim to provide insights into model predictions by approximating complex models with simpler interpretable ones. Additionally, methods that utilize visualizations, such as scatter radar diagram [8] and flowcharts [9], have gained attention for their ability to illustrate model behavior and highlight important features affecting predictions. Despite these advancements, there remains a gap in the application of explainability specifically in the context of software aging detection.

This paper explores the role of explainability in machine learning models applied to software aging detection. We present ISAAT, a tool designed to provide a comprehensive analysis and visualization of an MLP using flowchart representations and an LLM to generate concise, high-level explanations that support system administrators in strategic decision-making. To demonstrate the tool's efficacy, we conducted experiments on a real dataset related to software aging in a database environment. The experimental results show ISAAT's ability to provide detailed visualizations and reasonable explanations that not only facilitate understanding of the decision-making processes of the models but also enable the identification of relevant features and their contributions to the predictions. Rather than proposing a new aging detector, the contribution of this work is to complement existing ones by providing transparency and interpretability. ISAAT addresses a gap scarcely explored in the literature by explaining how and why machine learning models generate their predictions in software aging scenarios. This increases operational trust

and enables proactive decision-making, supported by verifiable artifacts such as flowcharts and structured reports. Furthermore, ISAAT is openly available, encouraging its adoption and extension by the research community.

The remainder of this paper is organized as follows. Section II presents the related works. Section III details ISAAT. Section IV presents the experimental plan. Section V details the obtained results. Finally, Section VI presents the conclusions and briefly describes future work.

II. Related Work

In recent years, software aging has attracted significant attention from both academia and industry due to its impact on the reliability and availability of long-running software systems [10]. Various approaches have been proposed to detect, mitigate, or prevent software aging, ranging from traditional time-based models to advanced machine learning techniques. However, despite these advances, there is a lack of studies focusing on the explainability of software aging detection models. The ability to interpret and understand how and why a model predicts software aging is crucial, especially in critical systems where trust and transparency are important.

A substantial number of works have explored the use of machine learning techniques to detect software aging in computational systems. These approaches can generally be categorized into linear and non-linear models. For linear models, studies such as [11] and [12] employed techniques like ARIMA, linear regression, and M5P to predict software aging. In contrast, non-linear models encompass a broader range of techniques used to detect software aging or resource exhaustion. For example, in [3], a MLP was applied to predict memory exhaustion, with the authors creating an artificial environment simulating memory leaks to provoke system crashes. Similarly, Yan [4] adopted a comparable method, using an MLP to predict available and heap memory, but this approach was applied to a real-world commercial application. In addition, Qiao et al. [5] utilized Long Short-Term Memory (LSTM) networks to predict aging-related metrics for Android systems.

In recent years, various works have been proposed to address explainability in ML models. Permutation Importance, explored by Altmann et al. [13], evaluates feature importance by analyzing the impact on model performance when feature values are randomly permuted, offering a simple and effective way to identify relevant features. Partial Dependence Plots (PDPs), introduced by Friedman [14], visualize the isolated effect of one or two features on predictions, providing insights into feature-response relationships while holding others constant. Counterfactual explanations, as explored by Mothilal et al. [15], determine the minimal changes required to alter model predictions, offering actionable and personalized insights. Layer-wise Relevance Propagation (LRP) [16] identifies the most influential parts of an input for deep neural networks by propagating relevance from outputs back to inputs, delivering intuitive visual explanations but posing challenges for implementation in very deep models. Lastly, attention-based methods, such as those proposed by Vaswani [17], enhance interpretability by highlighting input elements the model focused on during predictions through attention weights.

Table I: Comparison with related works.

Work	Explainability Addressed	Aging Detection	ML Model
[13]	Yes	No	Random Forest
[14]	Yes	No	Gradient Boosting Machine
[15]	Yes	No	Logistic Regression, Neural Network
[16]	Yes	No	Multilayer Neural Networks
[17]	Yes	No	Transformer
[11]	No	Yes	Linear regression, ARIMA, and others
[12]	No	Yes	M5P
[3]	No	Yes	MLP
[4]	No	Yes	MLP
[5]	No	Yes	LSTM
Ours	Yes	Yes	MLP

Table I provides a summary of the related works. In contrast to prior studies, our research aims to investigate explainability in machine learning models specifically applied to software aging detection. While existing approaches focus either on detecting aging symptoms or on improving explainability in general machine learning applications, our work combines both aspects and stresses the importance of understanding model predictions to support decision-making in critical systems. Unlike general-purpose explainability methods, ISAAT provides domain-oriented explanations that explicitly connect model reasoning to operational risks such as memory exhaustion and system crashes. By linking model outputs to actionable strategies and delivering visual and textual artifacts aligned with software aging scenarios, ISAAT supports more transparent and informed operational decisions.

III. ISAAT (Interpretable Software Aging Analysis Tool).

ISAAT[1] was developed to address the transparency in machine learning models, particularly those based on deep learning. Its primary objective is to provide an intuitive and accessible platform for analyzing the decision-making process of models, which enhances interpretability.

A. Structure and Functionalities

ISAAT was designed to provide an efficient and accessible experience to users, offering functionalities that simplify the use and maximize the interpretative value of the models. The tool allows the configuration of detailed parameters of machine learning models, such as the number of layers in a neural network, the activation functions (e.g., ReLU or Tanh) [18] and the optimization algorithms (e.g., Adam or RMSprop) [19]. In addition, users can upload datasets directly through the interface, defining specific characteristics for training.

After training the models, ISAAT presents detailed performance metrics to assess model quality. Key metrics include Mean Absolute Error (MAE), which highlights small deviations; Mean Squared Error (MSE), which gives more weight to large errors; and the coefficient of determination (R^2), which measures the proportion of variance explained

[1]The tool is available for download at https://github.com/andrecarlos26/ISAAT.

by the model [20]. Beyond numerical evaluations, ISAAT generates interpretive flowcharts that visually represent the model's internal behavior, making its decision-making process more transparent.

Another key feature is the ability to export results and visualizations, enabling users to document and share analyses easily. This flexibility makes the tool useful for both technical experts and those who are less familiar with machine learning, promoting the democratization of the use of interpretable models.

B. Tool Architecture

The ISAAT architecture is designed to ensure modularity, scalability, and a clear separation of responsibilities, as illustrated in Figure 1. At the core of the tool is a Graphical User Interface (GUI), which serves as an interface between users and the tool's main functionalities. Through the GUI, users can upload datasets, configure model parameters, and access visualizations, enabling an intuitive workflow for data analysis and interpretation. Beneath the GUI, the system is divided into two main modules: the Data Preprocessing Module and the Model Analysis Module. These modules operate collaboratively to process data and generate insights. The Data Preprocessing Module handles both data validation and model training. It includes a Training Submodule, which uses the input data to train interpretable models, and a Validation Submodule, which ensures the integrity and consistency of the data.

The Model Analysis Module handles the evaluation and visualization of trained models. It is composed of four submodules, each with a specific role. The Metric Calculation Submodule computes essential performance metrics, providing quantitative assessments of the models. The Visualization Submodule provides graphical representations of the model's outputs, such as performance metrics and feature importance, to enhance interpretability. The Flowchart Generation Submodule creates structured visual representations of the decision-making process within the AI model using the Graphviz Library [21], a robust tool for generating diagrams programmatically.

The Strategic Explanation Submodule utilizes the .txt files generated by the Flowchart Generation Submodule as input to a LLM, specifically *DeepSeek v3*, to produce a concise, high-level explanation aimed at supporting system administrators in strategic decision-making. These files, containing the model's internal decision-making data, are transmitted via a POST request to the OpenRouter API, which is preconfigured to interface with the selected LLM. Once the flowchart is completed, the system automatically triggers the request, sending both the decision data and a structured prompt. This prompt instructs the LLM to produce a concise, high-level explanation tailored for system administrators. The generated explanation addresses key aspects such as the model's decision-making process, predictive accuracy, potential biases, confidence levels, and strategic implications for system management and resource planning. The use of natural language and minimal technical jargon ensures that the output is both accessible and actionable for operational decision-makers.

The data flow follows a structured hierarchy. First, the process begins with the Data Preprocessing Module or the Model Analysis Module, depending on the user's objective. When user inputs are submitted through the GUI to the Data Preprocessing Module, the data is first used to train interpretable models and then validated to ensure consistency and integrity. Once validated, the trained model is passed to the Model Analysis Module for evaluation and visualization. Within this module, the Metric Calculation Submodule assesses performance, the Visualization Submodule generates graphical insights, the Flowchart Generation Submodule uses the Graphviz Library to create structured diagrams, and the Strategic Explanation Submodule consults the LLM to provide concise strategic summaries that clarify the model's decision-making process and analytical workflow.

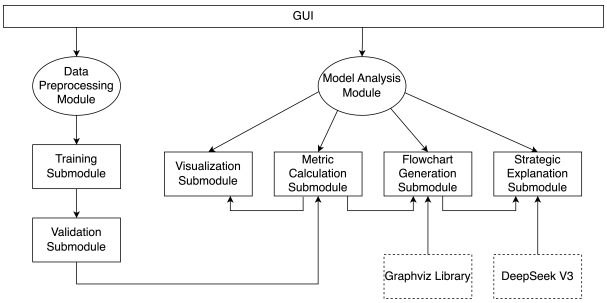

Figure 1: Architecture of the ISAAT tool.

Designing a tool for interpretability in software aging detection presents challenges that go beyond the use of generic ML visualization techniques. Explanations must be aligned with the semantics of software aging, translating low-level model operations into domain-related concepts, such as resource depletion and the need for rejuvenation. In addition, outputs need to be presented as actionable insights that inform administrators when and how to intervene. Finally, the model's reasoning must be exposed in a structured and verifiable way to establish trust in critical environments. ISAAT addresses these challenges by combining validation mechanisms, LLM-generated textual explanations, and visual artifacts such as flowcharts, which explicitly link the model's behavior to operational risks and support proactive decision-making.

C. User Interface

The user interface of ISAAT is designed to be intuitive and accessible, enabling users to configure and analyze machine learning models in a simple and straightforward manner. The main interface, illustrated in Figure 2, is divided into different sections that allow users to upload data, configure model parameters, and visualize the analysis results.

The interface includes specific fields for model configuration, providing full customization options. Users can specify the neural network architecture, including the number of layers and the number of neurons per layer. Other adjustable

parameters include the activation function, optimizer, loss function, batch size, number of epochs, and random seed for controlling randomness. A specific field is also available to define the target variable that the model will predict. Once the model parameters are configured, users can upload the required dataset in CSV format for training and validation. The training process begins when users click the "Upload" button, while the "Clear" button allows them to reset the configuration. Designed for flexibility and efficiency, the interface provides an ideal platform for building and analyzing machine learning models.

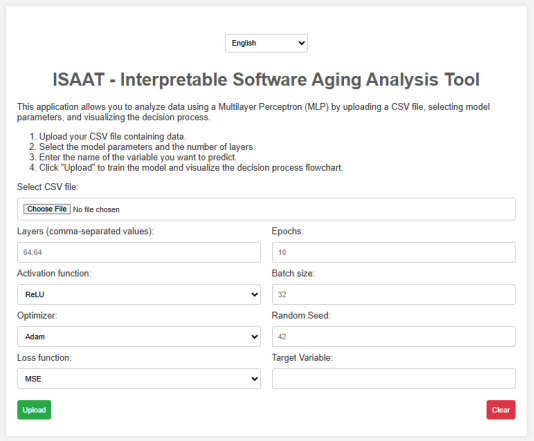

Figure 2: User interaction of ISAAT, designed to easily configure models, upload data, and view results.

Visualization is an import feature of ISAAT, enabling users to interpret and understand the behavior of machine learning models through dynamic flowcharts. Utilizing the *Graphviz* library, ISAAT automatically generates two types of flowcharts that visually represent the model's decision logic, as outlined in Table II. The summarized flowchart offers a broad view of the data flow through the model, suitable for users who want to quickly understand the basic structure, including the sequence of layers and the operations performed. On the other hand, the detailed flowchart presents an exhaustive view, including every mathematical operation, the values of weights and biases, and the activations at each layer. This level of detail enables experts to analyze the internal workings of the model, helping them identify how each input influences the final output. The dual flowchart approach ensures both beginners and experts can gain insights suited to their level of expertise, making the visualization tool accessible and informative for a wide range of users.

Table II: Levels of detail of the flowcharts generated by ISAAT.

Level	**Description**
Summarized	Provides a high-level view of the data flow in the model, offering an overview of its operating structure.
Detailed	Includes all mathematical operations, weights, biases, and activations in each layer, allowing for a comprehensive, in-depth analysis.

Figure 3 shows a summarized flowchart generated by ISAAT that illustrates the internal processing of the model for a given input. The diagram uses colors to indicate different stages: blue ellipses represent multiplications (inputs combined with weights), yellow ellipses indicate bias addition (shifting the computed value), green ellipses show activated neurons (after the activation function), and grey ellipses depict either inactive neurons or grouped layers. The flow starts with the input "Memory Used: 2.0 GiB" and proceeds through a hidden layer with 64 neurons using the ReLU function. One sample neuron computes $z_63 = 0.1400$ from a weight of 0.2226 and bias of -0.0826, followed by ReLU activation. The output layer aggregates five inputs with weights (e.g., 0.2988, 0.0976) and bias 0.3733 to produce $z_0 = 0.6174$, which passes through a linear activation. The red ellipse highlights the predicted result: "Predicted Time: 104.0h", and the yellow ellipse shows the evaluation metric: "Mean Squared Error (MSE): 0.00072619".

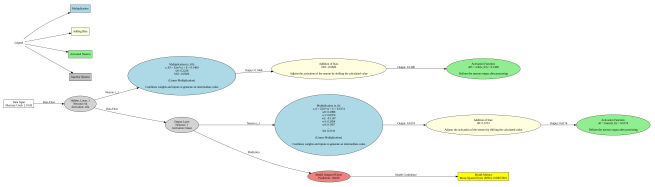

Figure 3: Summarized flowchart produced by ISAAT that illustrates the path followed by the data in a neural-network model. A high-resolution version can be downloaded from repository.

IV. Experimental Plan

This section briefly outlines the experimental plan for collecting and analyzing the software aging phenomenon in SQL Server. The experimental setup was designed to gather aging-related data for use with the ISAAT tool in the context of exploring explainability in machine learning models for software aging detection. A more detailed explanation is available in our previous work [22]. The experimental setup involved two computers: one acting as a client and the other as a server. The client sent queries to the server, which responded promptly. To minimize network interference and ensure precise results, the two machines were connected via a wired local network with isolated traffic. The client machine was equipped with an Intel Core i7-4790 processor running at 3.6 GHz, a 1 TB HDD, and 8 GB of RAM. The server used an AMD A10 Pro 7800B R7 processor, a 250 GB HDD, and 7 GB of RAM. Both machines operated on the Linux Ubuntu 20.04 64-bit operating system, and SQL Server Developer 2019 was installed on the server. Throughout the experiments, default settings were maintained for SQL Server, serving as a baseline for our evaluations.

To evaluate the behavior of the server system under various workloads, we conducted a series of 48-hour experiments simulating realistic database usage scenarios. Each query, executed from the client to the server, returned four thousand tuples and was designed to represent a complex workload involving join operations between tables. To accelerate the

manifestation of potential symptoms of software aging, three different workload levels were applied: a low workload with 5 concurrent users and a 1-second interval between requests, a medium workload with 50 concurrent users and a 0.5-second interval, and a high workload with 100 concurrent users and a 0.2-second interval. Each experiment began with a reboot of the server to ensure that all resources were refreshed, allowing us to isolate and effectively analyze the effects of software aging.

RAM usage was collected as the primary indicator of software aging. A Shell script was utilized to monitor RAM usage every five seconds. For trend analysis, the R language and the "trend" package were employed, utilizing the Mann-Kendall test [23] to evaluate whether continuous read requests led to degradation in the server's performance. The Mann-Kendall test evaluated two hypotheses: the null hypothesis (no trend) and the alternative hypothesis (the presence of a trend). A p-value of less than 0.05 indicated acceptance of the alternative hypothesis, suggesting a trend in the data. Additionally, we calculated the Sen's slope to measure the rate of change over time. For memory exhaustion prediction, we employed deep learning models, specifically a Multi-Layer Perceptron (MLP) with two dense layers and the rectified linear unit (ReLU) activation function. The model was trained for twenty epochs, and its performance was evaluated using mean squared error (MSE) to assess prediction accuracy.

V. Results

In this section, we first present the analysis of software aging suspicions related to RAM usage. Next, we use unsupervised multilayer perceptron models to predict memory exhaustion in the adopted environment. Finally, we demonstrate how ISAAT can enhance explainability in the prediction of memory exhaustion.

A. Analysis of RAM Memory Consumption

Figure 4 presents a comparison of RAM usage in the evaluated environment under low, medium, and high workloads. The x-axis represents the elapsed time (in hours), while the y-axis indicates memory consumption as a percentage. For the low workload (orange line), RAM usage starts near 19% and gradually increases over time, exhibiting periodic peaks and progressive increments. Under the medium workload (green line), memory consumption is consistently higher than in the low workload scenario, starting slightly above 20% and following a similar increasing pattern with noticeable peaks and variations. In the high workload scenario (blue line), RAM usage starts above 21% and shows a steady rising trend, with a more pronounced increase and fluctuations compared to the lower workloads. Overall, all workloads demonstrate a gradual increase in RAM consumption over time, suggesting a potential long-term accumulation of memory usage. Statistical tests are necessary to confirm potential signs of software aging.

As a part of the analysis to confirm suspicions of software aging in RAM usage, the Mann-Kendall test and Sen's slope estimator were applied. Table III presents the computed values,

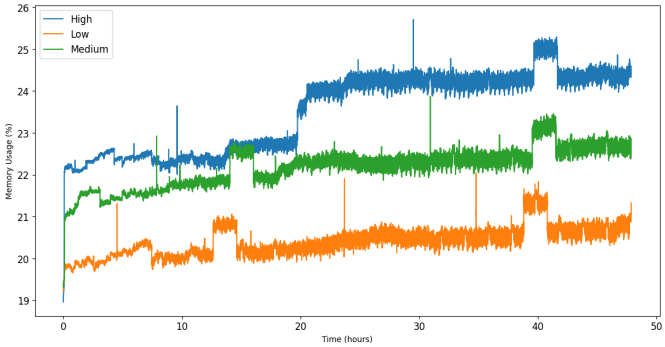

Figure 4: RAM Memory Consumption.

which confirm a consistent trend of increasing RAM usage across all scenarios. The results indicate that the SQL Server environment experienced a degradation in memory performance over time under all the applied workloads. Moreover, the high workload scenario exhibited the most pronounced signs of aging, with Sen's slope revealing greater magnitudes of growth in RAM usage over time.

Table III: Mann-Kendall and Sen's slope test for RAM memory usage.

Workload	***p-value***	**S**	**Slope (%/h)**
Low	2.98e-14	8.56e+02	0.02
Medium	4.22e-15	8.84e+02	0.032
High	2.2e-16	9.26e+02	0.06

B. Memory exhaustion analysis

To assess the impact of software aging in SQL Server environments, this study employed MLP models to predict memory exhaustion under varying workload conditions. The primary objective is to estimate the time required for memory consumption to reach exhaustion, defined as the moment when RAM usage exceeds 7GB, which corresponds to the total memory available in the experimental setup. Surpassing this threshold would lead to a system crash in the adopted environment. Understanding this behavior allows for better system management by preventing performance degradation, increased latency, and potential service failures in critical database environments.

Figures 5, 6 and 7 illustrate the projected memory consumption trends for SQL Server under high, medium, and low workloads, respectively. The x-axis represents the elapsed time (in hours), while the y-axis corresponds to memory consumption (in GB). The continuous orange line denotes the predicted memory consumption trajectory, while the dashed red line marks the exhaustion threshold at 7GB. The green dashed line represents the estimated time at which memory exhaustion is expected to occur, providing an estimate of system sustainability under different load conditions.

The results presented in these figures indicate that workload intensity has a significant impact on the rate of memory consumption over time, where higher query execution rates accelerate resource depletion. Under high workload conditions,

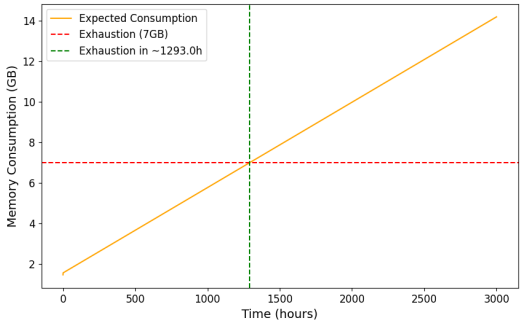

Figure 5: Memory exhaustion prediction for SQL Server under high workload.

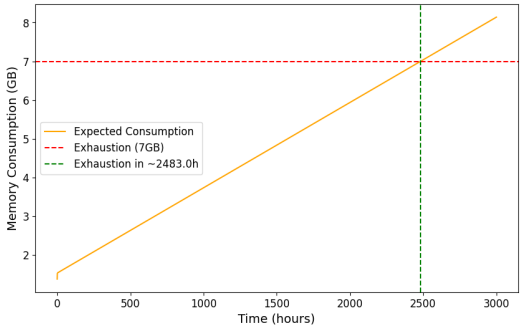

Figure 6: Memory exhaustion prediction for SQL Server under medium workload.

the memory usage exhibits a pronounced upward trend, reaching exhaustion considerably faster than in medium and low workload scenarios. On the other hand, in the low workload scenario, memory consumption increases at a much slower rate, demonstrating that SQL Server can sustain operations for a significantly longer period before reaching the exhaustion threshold.

Under high workload conditions, SQL Server is expected to deplete its available memory within approximately 1,293 hours, while medium and low workloads extend this period to approximately 2,483 and 2,515 hours, respectively. Increased query execution and data processing demands place greater stress on memory resources, leading to faster depletion. The ability to estimate these exhaustion times helps in planning workload distributions and resource allocation, reducing the likelihood of unexpected failures due to memory shortages.

The accuracy of the predictive models was assessed using the MSE, which quantifies the deviation between predicted and actual memory consumption trends. The MLP model was trained over twenty epochs with a batch size of 32, using 20% of the dataset for validation, which was crucial for evaluating both accuracy and generalization performance. The results indicate that the medium workload model achieved the highest accuracy, with an MSE of 0.00051710, showing a reliable forecast of memory depletion patterns. The high workload model exhibited a slightly higher MSE of 0.00044323, still demonstrating a strong predictive capability. Despite having the highest MSE at 0.000751, the low workload model effectively captured long-term memory consumption trends. The gradual nature of memory usage under low workloads likely contributed to slight deviations in prediction accuracy.

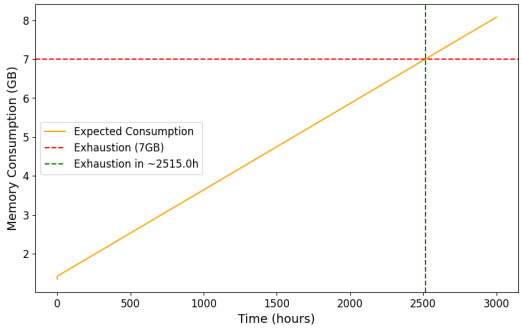

Figure 7: Memory exhaustion prediction for SQL Server under low workload.

C. Analysis of explainability results

Once memory aging and exhaustion were detected and predicted, ISAAT was employed to explore the explainability of the data across different workload levels. The model was configured with hidden layer of 64 neurons, using the ReLU activation function, Adam optimizer, MSE as the loss function, 20 training epochs, a batch size of 32, and a fixed random seed of 42. The target variable was memory usage, with an input value of 7GB. After filling in these parameters, uploading the CSV data file, and submitting the form, a loading indicator signaled that processing had started. Once completed, the interface displayed the model's performance metrics, flowchart visualizations, and a summary generated by the LLM. In the following analysis, we focus on the flowchart and LLM-generated summaries for the low workload scenario due to space restrictions. Figure 8 shows the flowchart that illustrates the model structure and operations related to memory usage under low workload. For clarity, this diagram uses a simplified version with only 5 neurons per hidden layer, while the detailed view includes 64 neurons per layer.[2]

In this scenario, the model achieved an MSE of 0.000751 and predicted a time to memory exhaustion of 2515.0 hours. This result is derived from processing the input data (memory usage: 7GB) through the hidden layers and output layer. The model's predictive capability comes from its optimized parameters, where each neuron's weights w and biases b were initialized with small random values and subsequently adjusted during training to minimize prediction error. The computational process within each neuron follows a well-defined mathematical operation: for every input, the neuron calculates the weighted sum z of inputs multiplied by their respective connection weights, adds the bias term, and then applies the ReLU activation function. This transformation enables the network to learn meaningful representations while maintaining interpretability through these measurable parameters.

For instance, in the low workload scenario for the hidden layer, Neuron 1 has $z_1 = 0.5101$, $w_1 = 0.1719$, and $b_1 =$

[2]All flowcharts are available at our repository.

0.3382, with an activation of $\text{ReLU}(0.5101) = 0.5101$. This moderately high activation suggests that the weight and bias capture gradual memory usage changes effectively. Neuron 2 has $z_2 = 0.4324$, $w_2 = 0.0335$, and $b_2 = 0.3988$, with an activation of $\text{ReLU}(0.4324) = 0.4324$. In this case, a large bias compensates for the relatively small weight, resulting in significant activation. Neuron 3 has $z_3 = -0.0841$, $w_3 = -0.0841$, and $b_3 = 0.0$, with an activation of $\text{ReLU}(-0.0841) = 0$. The ReLU function produces a zero output, indicating that certain inputs are not strongly emphasized in low workload conditions. In the output layer, the final linear combination produces $z_0 = -0.0743$, $w_0 = 0.1093$, $w_1 = 0.1547$, ..., and $b_0 = 0.3346$. Despite producing a negative raw value, the bias ensures that the predicted exhaustion time aligns with the observed trend of slower memory consumption.

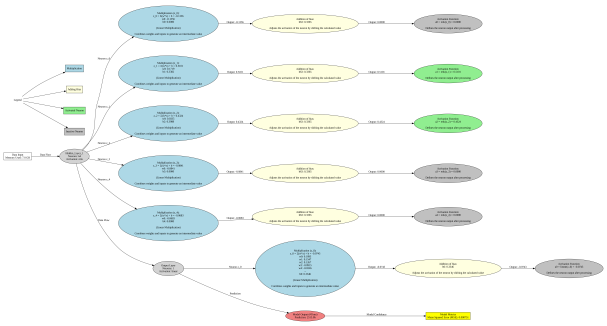

Figure 8: SQL Server low workload model flowchart. A high-resolution version can be downloaded from our repository.

Although the generated graph is valuable, it may still require some effort to interpret. To enhance explainability and improve accessibility, ISAAT produces a structured text file that can be processed by DeepSeek V3 via OpenRouter, generating a detailed and human-readable explanation. The example below, corresponding to a low workload scenario, illustrates how ISAAT can enhance explainability. The output is organized into clear sections, including High-Level Overview, Model Accuracy and Confidence, Potential Biases or Errors, Strategic Implications for System Management, and Final Recommendation. These sections present critical information including the model input (e.g., 7.0 GB), neuron activation behavior, the predicted memory exhaustion time (e.g., 2515 hours), confidence metrics (e.g., MSE = 0.000751), and practical guidance for system management. This structured and annotated output significantly enhances the model's interpretability, making its internal reasoning and limitations more transparent for both technical teams and decision-makers.

The Output Generated by ISAAT (low workload)

High-Level Overview

The model takes the current **memory usage (7.0 GB)** as input and processes it through a series of mathematical steps to predict when the system will run out of memory. Here's how it works:

1) **Input Processing**
   The model starts with the current memory usage (7.0 GB) and passes it through a **hidden layer** with **64 neurons**. Each neuron applies a weighted calculation (`weight × input + bias`) and then an **activation function (ReLU)**, which determines whether the neuron "fires" (outputs a positive value) or stays inactive (outputs zero).
2) **Neuron Activation Example**
   Some neurons activate (e.g., `Neuron z_1` outputs `0.5101`), while others stay inactive (e.g., `Neuron z_0` outputs `0.0000`). The activated neurons contribute to the final prediction, while inactive ones are ignored.
3) **Final Prediction**
   The outputs from the hidden layer are combined in the **output layer**, which produces a single value (`-0.0743`). This value is scaled to predict the time until memory exhaustion: **2515 hours (~105 days)**.

Model Accuracy & Confidence

- **Mean Squared Error (MSE): 0.000751** – This suggests the model's predictions are **very close** to actual observed values in training.
- **Confidence**: The low MSE indicates **high confidence** in predictions, but real-world variability (e.g., sudden memory spikes) could affect accuracy.

Potential Biases or Errors

- **Systematic Errors**: If the model was trained mostly on stable memory usage patterns, it might **underestimate** exhaustion time during irregular workloads (e.g., sudden surges in usage).
- **Bias Toward Inaction**: Some neurons frequently output zero (e.g., `z_0`, `z_3`, `z_4`), meaning they don't contribute to predictions. This could make the model **less sensitive** to certain memory patterns.

Strategic Implications for System Management

1) **Proactive Monitoring**
   Treat the **2515-hour prediction** as a **baseline**, but monitor for deviations (e.g., unexpected memory leaks). If memory usage trends upward faster than predicted, investigate applications or processes consuming extra resources.

2) **Resource Planning**
If the model consistently predicts exhaustion in 105 days, plan upgrades or optimizations **well before** that point (e.g., at 60–80 days). Use the model's output to justify budget requests for additional memory.
3) **Model Limitations**
The model **does not account for sudden workload changes**. Pair it with real-time alerts for abnormal memory spikes. If historical data lacks extreme cases (e.g., crashes due to memory exhaustion), the model may be **overly optimistic**.

Final Recommendation

This model provides a **data-driven estimate** of memory exhaustion, but administrators should:

- **Verify predictions** with real-world monitoring.
- **Combine with thresholds** (e.g., alert at 80% memory usage).
- **Retrain periodically** with new data to improve accuracy.

ISAAT advances software aging management by adding interpretability to conventional ML predictions. Unlike traditional black-box models that provide only numerical outputs, ISAAT enhances transparency through visual flowcharts, neuron activation maps, and high-level natural language explanations. In the case study, while a typical MLP simply predicted time to memory exhaustion, ISAAT enabled system administrators to trace the reasoning behind the 2515-hour estimate, understand feature influence, and assess prediction confidence. This combination of quantitative output and interpretable insights supports proactive, evidence-based maintenance decisions.

VI. Conclusions

This study explored explainability of machine learning models in software aging. The proposed ISAAT tool bridges the gap between black-box neural network operations and practical system maintenance by combining intuitive flowchart visualizations with DeepSeek v3, which generates concise, high-level explanations to support strategic decision-making in mitigating potential software aging risks. By making complex aging patterns accessible and actionable for operations teams, this approach enables a shift from reactive to proactive system management, ultimately enhancing service reliability and resource utilization. Future research will explore alternative deep learning models and LLMs, broaden the range of monitored metrics, and assess the scalability of the approach in more complex and dynamic environments.

References

[1] K. S. Trivedi, M. Grottke, and E. Andrade, "Software fault mitigation and availability assurance techniques," *International Journal of System Assurance Engineering and Management*, vol. 1, pp. 340–350, 2010.

[2] E. Andrade, R. Pietrantuono, F. Machida, and D. Cotroneo, "A comparative analysis of software aging in image classifiers on cloud and edge," *IEEE Transactions on Dependable and Secure Computing*, 2021.

[3] M. Yakhchi, J. Alonso, M. Fazeli, A. A. Bitaraf, and A. Patooqhy, "Neural network based approach for time to crash prediction to cope with software aging," *Journal of Systems Engineering and Electronics*, vol. 26, no. 2, pp. 407–414, 2015.

[4] Y. Yan, "Software ageing prediction using neural network with ridge," *IET Software*, vol. 14, no. 5, pp. 517–524, 2020.

[5] Y. Qiao, Z. Zheng, and Y. Fang, "An empirical study on software aging indicators prediction in android mobile," in *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. IEEE, 2018, pp. 271–277.

[6] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[7] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[8] T. Chen and Y.-C. Wang, "A two-stage explainable artificial intelligence approach for classification-based job cycle time prediction," *The International Journal of Advanced Manufacturing Technology*, vol. 123, no. 5, pp. 2031–2042, 2022.

[9] T. Chen, "A self-adaptive agent-based fuzzy-neural scheduling system for a wafer fabrication factory," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7158–7168, 2011.

[10] D. Cotroneo, R. Natella, R. Pietrantuono, and S. Russo, "A survey of software aging and rejuvenation studies," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 10, no. 1, pp. 1–34, 2014.

[11] P. Pereira, J. Araujo, R. Matos, N. Preguiça, and P. Maciel, "Software rejuvenation in computer systems: An automatic forecasting approach based on time series," in *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 2018, pp. 1–8.

[12] J. Alonso, J. Torres, J. L. Berral, and R. Gavalda, "Adaptive on-line software aging prediction based on machine learning," in *2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*. IEEE, 2010, pp. 507–516.

[13] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer, "Permutation importance: a corrected feature importance measure," *Bioinformatics*, vol. 26, no. 10, pp. 1340–1347, 2010.

[14] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[15] R. K. Mothilal, A. Sharma, and C. Tan, "Explaining machine learning classifiers through diverse counterfactual explanations," in *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 2020, pp. 607–617.

[16] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[17] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[18] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1, no. 2.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] G. James, D. Witten, T. Hastie, R. Tibshirani *et al.*, *An introduction to statistical learning*. Springer, 2013, vol. 112, no. 1.

[21] J. Ellson, E. Gansner, L. Koutsofios, S. C. North, and G. Woodhull, "Graphviz—open source graph drawing tools," in *Graph Drawing: 9th International Symposium, GD 2001 Vienna, Austria, September 23–26, 2001 Revised Papers 9*. Springer, 2002, pp. 483–484.

[22] H. Couto, G. R. A. Callou, F. Machida, and E. C. Andrade, "A comparative analysis of software aging in relational database system environments," *IEEE Transactions on Emerging Topics in Computing*, vol. V, p. 1, 2024.

[23] H. B. Mann, "Nonparametric tests against trend," *Econometrica: Journal of the econometric society*, pp. 245–259, 1945.