IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 June 2024, accepted 6 July 2024, date of publication 12 July 2024, date of current version 23 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3427388

RESEARCH ARTICLE

Employing Hybrid AI Systems to Trace and Document Bias in ML Pipelines

MAYRA RUSSO[1,2], **YASHARAJSINH CHUDASAMA**[2,3], **DISHA PUROHIT**[2,3], **SAMMY SAWISCHA**[2], **AND MARIA-ESTHER VIDAL**[1,2,3]

[1]L3S Research Center, 30167 Hannover, Germany
[2]Institute of Data Science-Scientific Data Management, Leibniz University Hannover, 30167 Hannover, Germany
[3]TIB—Leibniz Information Centre for Science and Technology, 30167 Hannover, Germany

Corresponding author: Mayra Russo (mrusso@l3s.de)

This work was supported by the Leibniz Association's Open Access Publishing Fund. The work of Mayra Russo was supported by the European Union (EU)-Horizon 2020 Research and Innovation Program under the Marie Skłodowska-Curie Actions (MSCA) through the Project NoBias under Grant 860630. The work of Yasharajsinh Chudasama, Disha Purohit, and Maria-Esther Vidal was supported in part by the ''Leibniz Best Minds: Program for Women Professors'' through funding of the ''TrustKG-Transforming Data in Trustable Insights'' Project under Grant P99/2020.

ABSTRACT Artificial Intelligence (AI) systems can introduce biases that lead to unreliable outcomes and, in the worst-case scenarios, perpetuate systemic and discriminatory results when deployed in the real world. While significant efforts have been made to create bias detection methods, developing reliable and comprehensive documentation artifacts also makes for valuable resources that address bias and aid in minimizing the harms associated with AI systems. Based on compositional design patterns, this paper introduces a documentation approach using a hybrid AI system to prompt the identification and traceability of bias in datasets and predictive AI models. To demonstrate the effectiveness of our approach, we instantiate our pattern in two implementations of a hybrid AI system. One follows an integrated approach and performs fine-grained tracing and documentation of the AI model. In contrast, the other hybrid system follows a principled approach and enables the documentation and comparison of bias in the input data and the predictions generated by the model. Through a use-case based on Fake News detection and an empirical evaluation, we show how biases detected during data ingestion steps (e.g., label, over-representation, activity bias) affect the training and predictions of the classification models. Concretely, we report a stark skewness in the distribution of input variables towards the Fake News label, we uncover how a predictive variable leads to more constraints in the learning process, and highlight open challenges of training models with unbalanced datasets. A video summarizing this work is available online (https://youtu.be/v2GfIQPAy_4?si=BXtWOf97cLiZavyu), and the implementation is publicly available on GitHub (https://github.com/SDM-TIB/DocBiasKG).

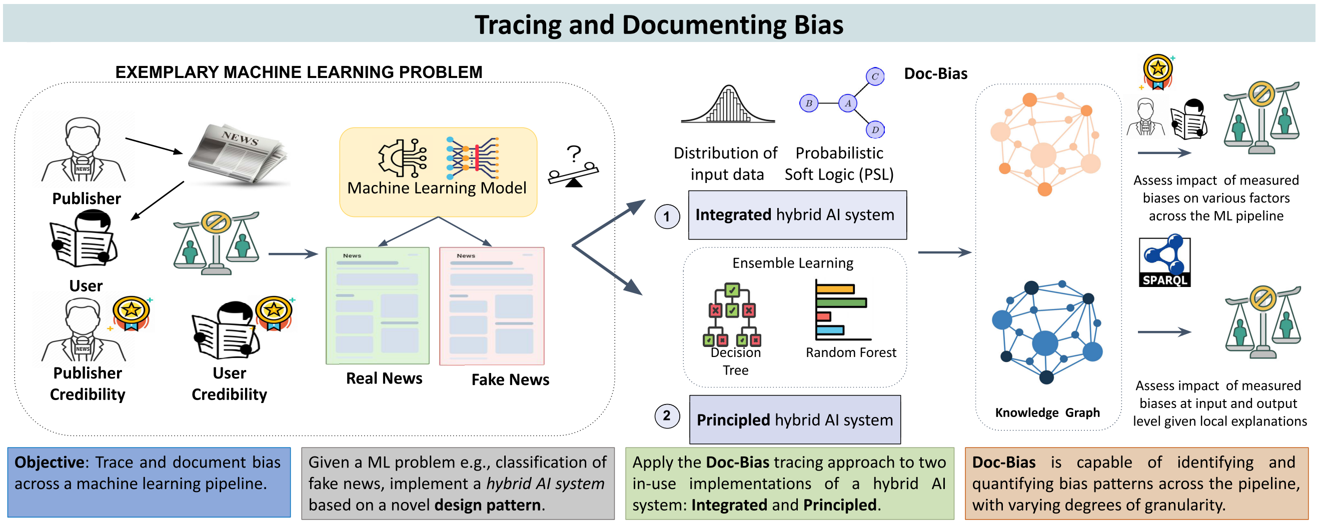
INDEX TERMS Bias, knowledge graphs, tracing, hybrid AI systems.

I. INTRODUCTION

Bias refers to a systematic and consistent deviation from the true value or objective reality in decision-making, judgment, or data analysis that can lead to detrimental or discriminatory outcomes [1]. The advent of an avalanche of available data in every domain and machine learning (ML)-powered systems has triggered a significant surge in research on this topic [2]. As the use of ML goes from seemingly trivial applications (e.g., movie recommendation algorithms to some with higher stakes and involving consequential decision-making, resource allocation in healthcare settings, or credit scoring systems in banking [3], [4]). Regardless of the perceived severity of the downstream application, all models share the ability to produce undesirable results [5], [6], [7], [8], [9], [10], [11].

While efforts in the creation of bias detection methods are most notable, the elaboration of reliable documentation can also be a valuable resource when used in combination to account for bias and to minimize harms associated with the use of ML-powered systems [7], [12], [13]. In actuality, part of the efforts by the ML community, specifically those researching fairness, accountability, and transparency in AI,

The associate editor coordinating the review of this manuscript and approving it for publication was Liangxiu Han.


Tracing and Documenting Bias
EXEMPLARY MACHINE LEARNING PROBLEM
Publisher
User
Publisher Credibility
User Credibility
Machine Learning Model
Real News
Fake News
Distribution of input data
Probabilistic Soft Logic (PSL)
Doc-Bias
1
Integrated hybrid AI system
Ensemble Learning
Decision Tree
Random Forest
2
Principled hybrid AI system
Knowledge Graph
Assess impact of measured biases on various factors across the ML pipeline
SPARQL
Assess impact of measured biases at input and output level given local explanations

Objective: Trace and document bias across a machine learning pipeline.

Given a ML problem e.g., classification of fake news, implement a *hybrid AI system* based on a novel **design pattern**.

Apply the **Doc-Bias** tracing approach to two in-use implementations of a hybrid AI system: **Integrated** and **Principled**.

Doc-Bias is capable of identifying and quantifying bias patterns across the pipeline, with varying degrees of granularity.

FIGURE 1. Graphical Abstract. Depicted is a visual representation of the proposed approach, which is demonstrated using a fake news classification use case and two implementations of a hybrid AI system, *Integrated* and *Principled*. The proposed approach demonstrates the impact of tracing and documenting biases measured across an ML pipeline.

have gone on to emphasize the importance of operating under practices that enable others to easily see the actions performed by the professionals and teams working at all steps of the ML pipeline [12], [13], [14], while also making an urgent call for data stewardship [15] and responsible data management practices [7]. This is understood to enable the study and understanding and, most importantly, boost the trust placed in the processes and systems themselves. The adequate documentation of datasets and models can produce artifacts that contribute to operationalizing best practice efforts to be seamlessly integrated into ML audit frameworks [7], [13], [14]. Additionally, promoting the elaboration of these artifacts, that at the same time are findable, accessible, interoperable, and reusable, i.e., datasets are published following the *FAIR* data principles, can contribute to the reproducibility and traceability of results [16], [17]. In this direction, documenting corresponds to the process of generating metadata represented in formats understandable by humans and by machines [16].

Existing methods for the production of documentation and benchmarks for datasets and ML models, such as [12], [18], [19], and [20], propose value-sensitive, human-readable documentation frameworks, while the works of [21] and [22], propose domain-specific and task-oriented, respectively, for dataset exploration tools. Similarly, interactive prototypes, such as [23], [24], and [25], have also been proposed for dataset exploration, visualization, and comparison. These systems and frameworks for documentation, however, do not address the comprehensive documentation of bias across ML pipelines. They also do not provide documentation artifacts available in a machine-readable format. The absence of these aspects in current documentation efforts can hinder the reliability and interpretability of ML models. Furthermore, industry-based ML practitioners prefer contextual, standardized, automated documentation frameworks to integrate with their workflows, according to recent research investigating their practical needs in terms of documentation [26].

Problem and Proposed Solution: In this work, we address the challenge of capturing interpretable knowledge about bias in ML models that support human- and machine-readability. Drawing inspiration from existing literature [27], we introduce a novel and generic documentation pattern, *Doc-Bias*, that resorts to a hybrid AI system. Our approach is first presented as a compositional *design pattern* that encompasses a generic AI system specification, two systems for tracing bias within the AI system, and one pattern for integrating the captured bias facts into a knowledge graph (KG). In order to demonstrate the implementation of our pattern in use, we present two instantiations of it that correspond to two different approaches to conceive a hybrid AI system, **principled and integrated**. In doing so, we are also able to demonstrate how the implementation of the proposed hybrid AI system generates relevant semantic metadata of a machine learning pipeline to varying degrees of tracing power. A principled integration offers coarse-grained documentation that is both comprehensive and easy to integrate with different types of ML models. In contrast, the second approach seamlessly merges the documentation subsystem into the prediction model, generating fine-grained documentation of the entire pipeline, and providing richer insights into how the model works.

Figure 1 depicts a visual summarization of the main points to be addressed throughout this work. Specifically, we implement our proposed hybrid AI system with a use-case based on Fake News classification to demonstrate the viability of *Doc-Bias* documenting and tracing bias. For the implementation of the **integrated hybrid AI system**, we use a classifier that resorts to Probabilistic Soft Logic (PSL) [28], a machine learning framework for developing probabilistic models. A model in PSL is defined through a set of weighted first-order logical rules, where the weight is learnable. Then, based on the input data, some random variables are observed, while some are unobserved, with the task of inference in PSL being to estimate the value for an unobserved random variable given the patterns learned from the observed variables. In our work, we implement the PSL classification pipeline as made available in [29], but modify its output logs settings to enable tracing, while reproducing the original setup and the reported performance results. The **principled hybrid AI system** resorts to the machine learning technique of random forests for classification. This ensemble learning method operates by constructing many decision trees at training time, with the output of the random forest being the class selected by most trees.

We conducted an empirical study to assess the capacity of the proposed methods in tracing bias patterns (i.e., label, over-representation, activity bias) in different steps of the pipeline followed to train and utilize the AI models that solve the problem of Fake News detection. The study was orchestrated over two existing benchmarks. The FakeNewsNet catalog[10] [30] is comprised of the BuzzFeed[9] dataset and the PolitiFact dataset[8]. The news collected to elaborate on the PolitiFact dataset was sampled from the fact-checking website PolitiFact and contains 120 fake and 120 real news articles. Similarly, BuzzFeed News data were sampled from news published on Facebook and fact-checked by BuzzFeed journalists. This dataset contains 91 fake, and 91 real news. The social context associated with the News was extracted from Twitter. The bias patterns, detected during dataset ingestion analysis, can be traced across the pipeline and capture their impact on the produced output. Concisely, we report a stark skewness in the distribution of input variables towards Fake News. Further, we uncover how a few users share a significant number of news articles, leading to more constraints in the learning process of one of the models, highlighting the challenges of training models with unbalanced datasets. We also demonstrate how User Credibility, as a predictive feature, overwhelmingly contributes to the classification of News, underscoring the importance of understanding and addressing biases about Users in the context of Fake News proliferation.

Contributions: This paper makes the following contributions:

- *A Design Pattern for the Hybrid AI System for Documentation.* Introducing a novel design pattern for a hybrid AI system crafted to trace a generic machine learning pipeline. This pattern not only captures the details of a model's functionality but also addresses the crucial task of identifying and documenting the effects of bias on the model's performance. The design pattern is a fundamental element in our approach.
- *Two Implementations of our Hybrid AI System for Documentation.* Instantiating the proposed pattern, we introduce two distinct implementations with a use case based on Fake News classification. The first approach features a principled integration of the hybrid AI system with the ML model, offering coarse-grained documentation. The integrated system produces fine-grained documentation of the model's behavior.
- *Evaluation of Hybrid AI Systems.* These are the results of an empirical evaluation of the two proposed hybrid AI systems with a use case based on Fake News detection. We assess performance based on bias metrics and analyze the tracing power of the two documentation approaches.
- *A Doc-Bias Knowledge Graph.* Introducing the Doc-Bias KG, a knowledge graph that seamlessly integrates traced data collected during the execution of the ML model. This additional component enhances the overall documentation capabilities of our hybrid AI system.

TABLE 1. Summary of doc-bias notations.

Notation	Description
$\mathcal{P}$	Machine Learning (ML) problem
X	Space of all possible input
Y	Space of all possible labels
$D \subseteq X \times Y$	Training dataset comprises $X \times Y$
$\mathcal{H}$	Hypothesis space of all predictive models
L	Loss function
$h(x)$	Predictive model h in hypothesis space $\mathcal{H}$
h^*	Optimal predictive model
$f(.)$	Bias measure
ϕ	Threshold for bias detection
$KG = (V, L, E)$	Directed-edged-labeled graph where V is a set of nodes, L is a set of labels and E is a set of edges
$t = \langle s, p, o \rangle$	Triple pattern where s is subject, p is predicate and o is object
$DIS = \langle O, S, M \rangle$	Data Integration System (DIS) where O represents unified schema, S represents data sources and M represents mapping assertions
$\mathcal{F}$	Fake news detection ML problem
N	Space of all possible news
U	Space of all possible users
P	Space of all possible publishers
$ActualL$	Space of all possible news labels
$U\text{-}N$	Set of all pairs of (u, n)
$U \times N$	Users posting news in (u, n)
$P\text{-}N$	Set of all pairs of (p, n)
$P \times N$	Publishers posting news in (p, n)

The rest of the paper is structured as follows: Section II summarizes the state of the art. Section III defines basic concepts and motivates our work with an example in the context of Fake News Detection. Section IV presents our problem statement and describes our proposed hybrid AI system. Results of the empirical evaluation are reported in Section VI. Finally, we close with the conclusions and future work in Section VIII.

II. RELATED WORK

A. TRACING AND ML

In recent decades, AI has made great strides, increasingly pushed by huge amounts of data and new complex algorithms.

De Bie et al. [31] report a perception of AI automation that transforms the aspects of our lives in various domains (e.g., from medical to finance) and categorized them into three stages: *Mechanization* (i.e., data engineering), *Composition* (i.e., ML model building and their hyperparameter selection) and *Assistance* (i.e., explainability and visualization). However, these algorithms are often termed as either white or black box models, i.e., the internal mechanisms are too complex to be fully understandable and explainable. The opaqueness of the underlying processes powering AI systems hinders their interpretability and, subsequently, the trust placed in them. This emphasizes the importance of operating under practices that enable the study and understanding of the intrinsic characteristics of the components that make up AI pipelines [12]. Various frameworks have evolved in the pursuit of transparency and interpretability in AI. Among the most promising approaches that relate to our research include LIME (Local Interpretable Model Agnostic Explanations) [32] and SHAP (Shapely Additive Explanations) [33]. Moreover, both frameworks have a unique way of addressing the problem of interpretability. Riberio et al. introduce a post hoc explainable framework, LIME, which provides explanations of each instance locally and lists relevant features with their contribution to an ML model's decision. Further, Lundberg et al. propose SHAP that operates over the coalition game theory, where each data instance is represented as a shapely value and provides global explanations and their feature contributions. While these post hoc explainable frameworks have the potential to understand the quantitative ML models, and in the case of LIME has been incorporated into the InterpretME framework, both of them lack the consideration or capability to detect and measure any existing bias in the dataset or predictive pipeline. Mehrabi et al. [34] observe in a survey that the AI system generates unfair outcomes in different domains. This motivates researchers to mitigate the problem of bias in AI through three aspects: *pre-processing, in-processing, and post-processing*. Existing scholarship has dedicated most of its efforts to characterizing datasets decoupled from the underlying ML task, as opposed to our holistic approach. Sun et al. [22] introduce a tool to assess fitness for using datasets. This automated data exploration tool limits its focus to three dimensions: representativeness, bias, and correctness. In a similar line, Wang et al. [21] introduce a bias visualization tool for computer vision datasets. This exploration tool narrows down its assessment to three sets of bias measures: object-based, gender-based, and geography-based dimensions. As a result, the visualization of bias generates human-understandable results for each dataset. In our case, the extensible and modular design of Doc-Bias has the functionality to allow ML researchers and practitioners to describe and trace their datasets and seamlessly incorporate additional descriptive dimensions and components of the classification pipeline as needed. Similarly, interactive tools– developed by industries– (e.g., [23], [24], [25]) have also been proposed for dataset exploration, visualization, and comparison. Our hybrid AI system provides fine-grained representations of data sources, which are semantically enriched and interlinked.

B. SEMANTIC WEB TECHNOLOGIES IN HYBRID AI

Semantic Web technologies play a crucial role in enhancing the accuracy and interpretability of AI systems, as are well positioned to support ''bias assessment, representation, and mitigation'' tasks [35]. Ristoski and Paulheim [36] highlight the potential and challenges of Semantic Web Technologies in machine learning for knowledge discovery. This comprehensive survey reports on the need for tools that mitigate bias and provide interpretability of ML model outcomes. In their work, van Bekkum et al. [27] have introduced elementary and compositional design patterns that define various types of hybrid AI systems resulting from combining AI and symbolic systems. Additionally, Breit et al. [37] have conducted an extensive survey of the state-of-the-art, highlighting the significant role of Semantic Web technologies in improving the interpretability of AI models. Based on the aforementioned hybrid AI design patterns, Russo et al. [38] introduce an analytical framework to systematically characterize knowledge graphs with regard to their structural bias properties in human- and machine-readable format. Moreover, Chudasama et al. [39] propose the InterpretME framework, which provides the interpretability of ML models over KGs. Chudasama et al. state that the InterpretME pipeline performs classification tasks based on user input and traces the metadata captured at each pipeline stage to generate the InterpretME KG. Here, InterpretME also provides a federation of KGs, which facilitates users with more contextual insights to understand the complex inner workings of the ML model. Nevertheless, these frameworks fail to consider existing bias inside the trained ML model, nor are they equipped with domain-specific semantic models to describe ML pipelines in terms of bias. Building on these recent findings from the literature, our proposed approach utilizes the characterizations by Van Bekkum et al. [27] to develop a hybrid AI system that identifies bias patterns and evaluates their impact on AI model performance. Additionally, Doc-Bias, a hybrid approach, resorts to Semantic Web Technologies to trace and document bias patterns, enhancing not only the accuracy of the ML model but also the interpretability and reliability of decisions.

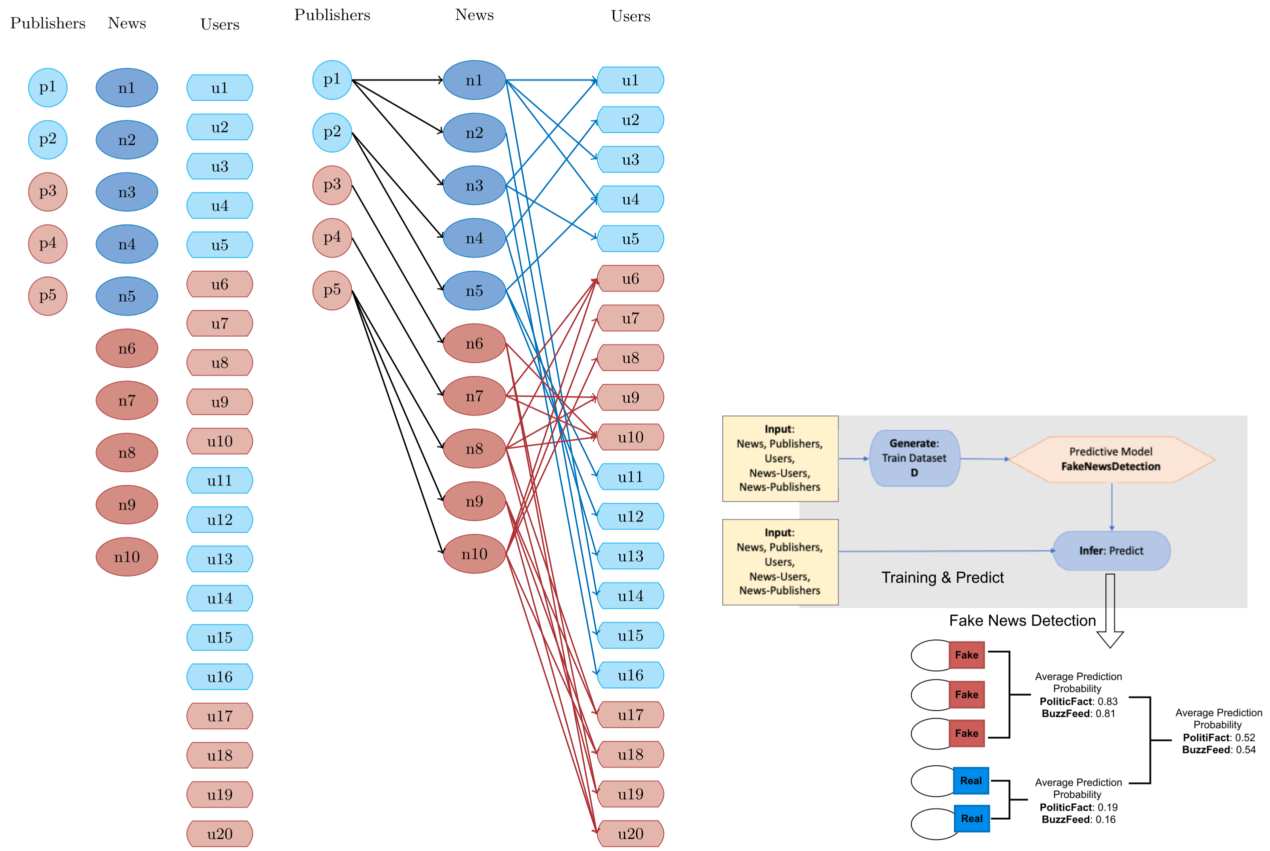
III. PRELIMINARIES AND MOTIVATION

This section presents the basic concepts required to understand the work tackled in this paper. Furthermore, we illustrate an example that puts the motivation of our work into perspective.

A. BACKGROUND

1) A MACHINE LEARNING PROBLEM

A machine learning problem $\mathcal{P}$ is defined as an optimization problem where the objective is to find an optimal predictive model that minimizes an objective function (i.e., loss function) over the training dataset. Formally, let $\mathcal{P}$ be defined as follows:


Publishers
News
Users
p1
p2
p3
p4
p5
n1
n2
n3
n4
n5
n6
n7
n8
n9
n10
u1
u2
u3
u4
u5
u6
u7
u8
u9
u10
u11
u12
u13
u14
u15
u16
u17
u18
u19
u20
Input: News, Publishers, Users, News-Users, News-Publishers
Generate: Train Dataset D
Predictive Model FakeNewsDetection
Input: News, Publishers, Users, News-Users, News-Publishers
Infer: Predict
Training & Predict
Fake News Detection
Fake
Fake
Fake
Real
Real
Average Prediction Probability PoliticFact: 0.83 BuzzFeed: 0.81
Average Prediction Probability PolitiFact: 0.52 BuzzFeed: 0.54
Average Prediction Probability PoliticFact: 0.19 BuzzFeed: 0.16

(a) Input Balanced Labelled Data

(b) Skewed data due to Unbalanced Relationships

(c) Skewed Prediction Probabilities

FIGURE 2. Motivating Example: (a) Toy-balanced dataset comprising Fake and Real News. (b) Paths represent relationships among key actors (i.e., News, Users, and Publishers). There are 32 paths relating to News and Publisher via Users; 19 involve Fake News (59.4%), while 13 paths (40.6%) are only Real News. (c) Design pattern for PSL AI model on Fake News detection. The model outputs prediction probability for labeled News. On average, probability is relatively low in PolitiFact (0.52) and BuzzFeed (0.54), but the average prediction probability is considerably higher for Fake News in PolitiFact (0.83) and BuzzFeed (0.81), and low for Real News in PolitiFact (0.19) and BuzzFeed (0.16). Skew in paths results in an accuracy loss when labeling Real News.

X- space of all the possible input data points; Y- space of all the possible labels; D- the training dataset comprising labelled examples (x, y) derived from the X and Y, i.e., $D \subseteq X \times Y$; $\mathcal{H}$- hypothesis space comprising all the candidate predictive models that map elements in X to labels in Y; **L-** loss function quantifying the discrepancy between the true y and the predicted output $h(x)$ produced by a predictive model h in the hypothesis space $\mathcal{H}$. The goal is to find an optimal model h^* in $\mathcal{H}$ that minimizes the loss function L over D.

$$h^* = \arg\min_{h \in \mathcal{H}} \frac{1}{|D|} \sum_{(x,y)\in D} \mathrm{L}(h(x), y)$$

2) MEASURING BIAS IN DATA

Emerging research has been able to demonstrate that bias can start at any point of the ML pipeline [2]. This debunks the misconception that undesired effects arising from using ML systems should be attributed only to the input dataset and widens our understanding of their vulnerabilities. For instance, data collection practices involve a series of decisions, such as determining who is the sampled population, what variables need to be measured, the definition of labeling criteria for annotations, instructions handed down, and annotators' working conditions to perform these tasks [2], [20]. More often than not, there is not an existing record of all these dimensions, making the capture of all external factors that can plague a dataset unfeasible. The same occurs during model definition tasks and during the learning process. For example, a common practice in ML is to use a random seed to preserve experiment reproducibility. Given the stochastic nature of many ML algorithms, the choice of the random seed is model-dependent and can significantly alter the outcomes, and thus can become a source of bias [40]. This knowledge provides the basis to devise effective bias mitigation strategies [2], e.g., fairness indicators and data bias measures [41], [42], to minimize harms emerging from ML systems, or artificial intelligence (AI) in broader terms.

Succinctly, a bias measure corresponds to a quantitative metric or indicator that assesses the presence and extent of bias in a particular context. They cover the following aspects [43]:

1) *Target Group:* or entities for which bias is being assessed (e.g., News, Users);
2) *Attribute(s):* that may contribute to bias (e.g., source origin or level of activity);

3) *Group Comparison:* method to compare the performance of the model across different groups based on the chosen performance metric;
4) *Thresholds or Criteria:* thresholds or criteria that indicate the presence of bias.

A *bias measure* $f(.)$ indicates bias favoring one group and provides insights into the severity of bias.

A bias pattern defines the criteria for identifying bias based on the output produced by a bias measure. It can be represented as a Boolean predicate using a bias metric f and a *threshold* ϕ, denoted as *biasPattern(f(.),* ϕ*).*

Whilst not without their challenges and limitations. i.e., what metrics to use, how to identify sensitive attributes, how to interpret algorithmic output or lack of access to domain knowledge [26], [44], [45], [46], in our work, we propose the implementation of existing bias measures to comprehend the factors influencing the manifestation of bias patterns in the implementation of an ML problem. We set out to gather this knowledge and represent it as factual statements within the Doc-Bias knowledge graph. By doing so, we can gain valuable insights into the underlying mechanisms of bias and its impact on machine learning systems.

3) HYBRID LEARNING AND REASONING SYSTEMS

Artificial Intelligence (AI) can be ramified into *symbolic* and *sub-symbolic* approaches. Symbolic approaches are expressed in explicit symbolic methods, i.e., formal logic, decision trees, and ontologies, and are often associated with human-readable and explainable processes [47]. On the other hand, sub-symbolic approaches are derived from neural connectionist notions that aim to emulate the processes the human brain performs through artificial neural networks. Sub-symbolic methods encompass different statistical learning methods, i.e., deep learning and Bayesian learning [47]. While both approaches have contributed to many real-world applications separately, their integration is greatly desired to overcome some of the existing limitations of these methods, such as scalability and elevated data-dependency correspondingly, and thus move the AI field forward [48]. Consequently, hybrid or neuro-symbolic approaches focus on integrating symbolic and sub-symbolic systems, with research interest rapidly increasing in this area [47], [49], [50]. Moreover, benefits attributed to hybrid approaches are their ability to enhance the performance and explainability of AI systems. Our approach resorts to a hybrid AI architecture. The aim is to integrate a sub-symbolic system over a symbolic system (e.g., a knowledge graph) to produce an AI system that can trace the decisions made by a sub-symbolic system and the outcomes. This architecture can be implemented by combining both systems following different strategies. One of them is that of a principled integration; here, the combination of the neural and symbolic systems are integrated but maintain a clear separation between their roles and representations. Another way to entail a fully integrated system is by integrating a symbolic reasoner into the tuning process of a sub-symbolic model. The criteria for integration will be determined by different factors, one of which can be access to the code representing the learning process or the inherent characteristics of the sub-symbolic system itself. For instance, in the case of sub-symbolic systems that are white boxes, such as models based on patterns, rules, or decision trees, set up the optimal circumstances for a principled integration. On the other hand, sub-symbolic systems that are black boxes, such as support vector machines (SVMs), neural networks, and probabilistic and combinatory logic models, can be better suited for principled integration [51].

4) KNOWLEDGE GRAPHS AS DATA INTEGRATION SYSTEMS

Knowledge graphs are data structures that can be understood as symbolic representations of knowledge [52]. Employing a graph data model [53], KGs contribute to developing a common understanding of the meaning of entities, their characteristics, and the relationships among them in a particular domain, also referred to as background knowledge. A knowledge graph is made up of metadata and taxonomies of the identified entities, relationships, and classes and can be modeled in a language such as Resource Description Framework (RDF),[1] RDF Schema (RDFs),[2] or in combination with more expressive ones such as the Web Ontology Language (OWL).[3] Formally, a knowledge graph KG is defined as a labeled directed graph, $KG = (V, L, E)$, where V is a set of nodes represented as classes and entities; L corresponds to a set of labels; and E is a set of edges such as $E \subseteq V \times L \times V$. When expressed in RDF, each triple, denoted as $t = \langle s, p, o \rangle$, adheres to specific constraints: s can be a URI or a blank node, p must be a URI, and o can take on the form of a URI, blank node, or literal [53]. Knowledge graphs can represent, as factual statements, knowledge spread across various data sources [53]. For this purpose, they can be defined as data integration systems (DIS), whose evaluation enables the transformation and integration of heterogeneous data in a knowledge graph [54]. A data integration system $DIS = \langle O, S, M \rangle$ is defined in terms of three components:

- O a unified schema or ontology that provides a uniform view to the data sources in S. The main objective of an ontology is "to make the meaning of a set of concepts, terms, and relationships explicit so that both humans and machines can understand what those concepts mean" [55]. The schema or ontology can also be understood as the conceptual representation of the KG.
- S is a set of the data sources that will compose the DIS.
- M a set of mappings between signatures of the sources in S and concepts in O. The mapping rules explicitly indicate how the source data is mapped to the schema S.

An RDF-conforming KG can be produced following the integration of these components. In order to retrieve and

[1] Resource Description Framework (RDF)
[2] RDF Vocabulary Description Language: RDF Schema (RDFS)
[3] Web Ontology Language (OWL)

manipulate data, query languages, such as SPARQL,[4] are used to analyze the KG and perform knowledge discovery. Ultimately, knowledge graphs are ideally positioned to ensure the findability, accessibility, interoperability, and reusability (FAIR) of data-centric systems [17]. These are some of the key characteristics that position knowledge graphs as an essential component for the implementation of our framework, as well as for the creation of the tracing and documentation system that is being introduced in this work.

5) INTERPRETME AS A PRINCIPLED HYBRID AI SYSTEM

As already described, hybrid AI systems integrate symbolic and sub-symbolic approaches by using different architectures (e.g., principled and integrated). Chudasama et al. propose a hybrid AI system, *InterpretME* [39], [56], that following a principled approach, combines data-dependent frameworks (i.e., a machine learning model) over knowledge graphs to yield an analytical tool capable of providing fine-grained representations of said ML models, in order to improve the interpretation of their produced outcomes. The current implementation of *InterpretME*, includes all the predictive model pipeline components, such as data preparation, sampling strategy to balance the data, and training the predictive model. *InterpretME* uses automated machine learning (AutoML) frameworks to optimize model hyperparameters to improve the performance of the ML algorithm in use. Concurrently, a post-hoc explainable framework (e.g., LIME) is utilized to construct instance-level interpretations for each instance of the resulting test data. Once the model is trained and validated, the input dataset and the metadata generated after each stage of executing the *InterpretME* pipeline are traced and semantified. The resulting process integrates the input data and ML model characteristics to produce interpretations in the form of RDF factual statements, which comprise the InterpretME KG. The knowledge graph can then be used to trace the entirety of the predictive task, and via SPARQL queries, it is possible to retrieve information about the model, such as relevant features, accuracy, precision, and LIME interpretations. The vocabulary for the trained ML model, hyperparameters, input features, and LIME explanations is publicly available as a VoCol[5] instance. Additionally, *InterpretME* enables federated query processing on top of the InterpretME KG and the input KG to provide further contextual insights for a predictive task. *InterpretME* is publicly available on PyPI,[6] and GitHub.[7] Ultimately, the results obtained by *InterpretME* show the potential of Semantic Web technologies in empowering sub-symbolic AI systems and enhancing interpretability. Given this, in our work, we implement the *InterpretME* framework as a principled hybrid AI system to trace and document an ML pipeline. In particular, we set out to uncover the implications of tracing bias over both types of hybrid AI implementation: principled and integrated.

B. MOTIVATING EXAMPLE

To motivate our work, let us consider the task of Fake News Detection $\mathcal{F}$, as an example of a machine learning problem [57] (previously defined). We chose this problem due to the surge in automated systems used in Fake News detection, driven by the increased proliferation of Fake News in online settings, which goes on to have severe social implications (i.e., interference in democratic processes, health risks, political polarization, financial losses, data leaks). From a computational point of view, the Fake News detection task can be performed by employing different methods [57]. In this work, we base all our experimentation using a method that relies on leveraging relational data relationships to better mimic the real-world spread of fake news in an online setting, i.e., the relation between news items with their publishers and social context information (i.e., social media users' activity).

$\mathcal{F}$ is defined as an optimization problem based on the following preconditions [29]: N**-** space of all the possible News; U**-** space of all the possible Users; $\boldsymbol{P}$**-** space of all the possible Publishers; ***ActualL*-** space of all the possible pairs $(n, l) \in N \times \{\text{Fake, Real}\}$ that represent News' labels; ***U-N*-** set of all pairs (u, n) in $U \times N$ representing a User u posting a News n; ***P-N*-** set of all pairs (p, n) in $P \times N$ representing a Publisher p publishing n; ***Labels*-** set of pairs (n', l) representing that the News n' is labeled as l (where $l \in$ Fake, Real); D**-** the training dataset comprising News labelled as Fake or Real $(\hat{n}, \hat{l})$, such that $\hat{n} \in N$; $\mathcal{H}$**-** hypothesis space comprising all the candidate predictive models for Fake News detection; **L-** loss function that quantifies the discrepancy between $(n, l) \in ActualL$ and the label predicted for the News n by the model h in the space $\mathcal{H}$, i.e., $h(n)$.

To illustrate the impact of relating User data to the News they share, we provide an example in Figure 2a. This example comprises a balanced Fake News dataset with five Fake News and five Real News items. Additionally, it includes data on 20 unique Users and five Publishers. Interestingly, connecting User data to the shared News leads to a change in the original balanced News distribution, introducing a skew towards Fake News. While it may be challenging to gauge the full extent of this influence at first glance, Figure 2b provides a naive illustration of the skewed distribution of paths between Publishers and Users through News. Out of the 32 paths, 19 involve Fake News (59.4%), while 13 involve Real News (40.6%). This observation highlights the importance of understanding the relationships between Users, Publishers, and News items to effectively detect Fake News. The issue of path imbalance, as illustrated in the toy example in Figure 2b, is also evident in two state-of-the-art Fake News datasets, PolitiFact[8] and BuzzFeed,[9] two datasets

[4] SPARQL Protocol and RDF Query Language

[5] http://ontology.tib.eu/InterpretME/

[6] https://pypi.org/project/InterpretME/

[7] https://github.com/SDM-TIB/InterpretME

[8] https://github.com/KaiDMML/FakeNewsNet/tree/old-version/Data/PolitiFact

[9] https://github.com/KaiDMML/FakeNewsNet/tree/old-version/Data/BuzzFeed

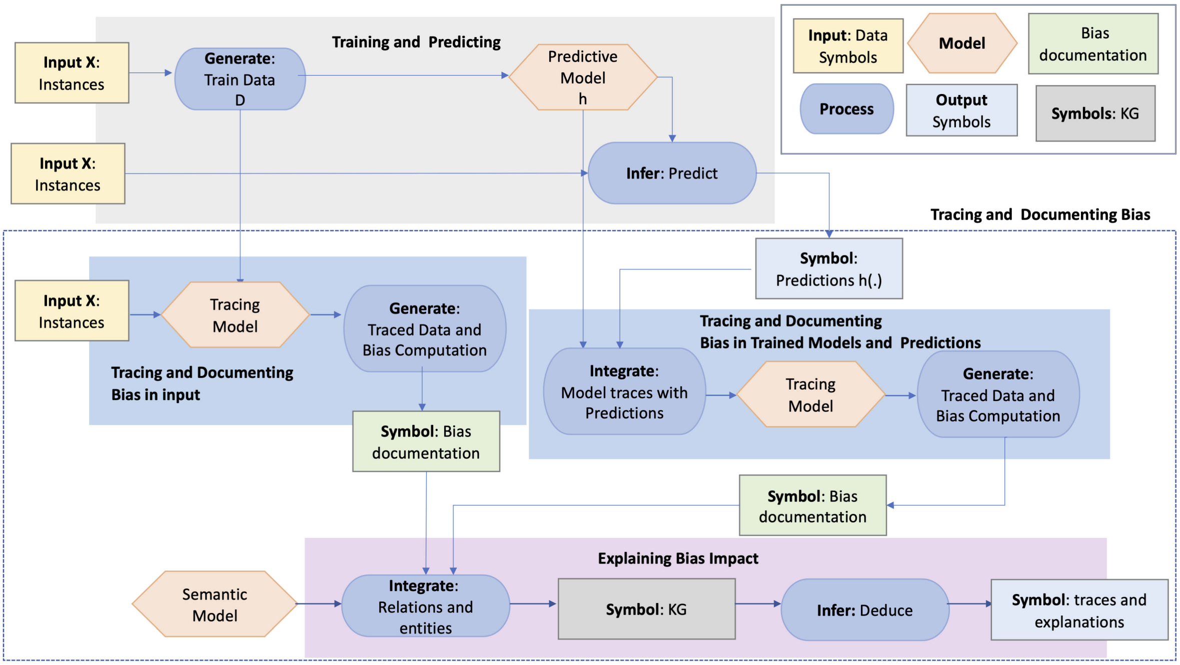

Input: Data Symbols
Model
Bias documentation
Process
Output Symbols
Symbols: KG
Training and Predicting
Input X: Instances
Generate: Train Data D
Predictive Model h
Input X: Instances
Infer: Predict
Tracing and Documenting Bias
Symbol: Predictions h(.)
Input X: Instances
Tracing Model
Generate: Traced Data and Bias Computation
Tracing and Documenting Bias in input
Symbol: Bias documentation
Tracing and Documenting Bias in Trained Models and Predictions
Integrate: Model traces with Predictions
Tracing Model
Generate: Traced Data and Bias Computation
Symbol: Bias documentation
Explaining Bias Impact
Semantic Model
Integrate: Relations and entities
Symbol: KG
Infer: Deduce
Symbol: traces and explanations

FIGURE 3. Design Pattern Hybrid AI. The Design Pattern represents a Hybrid AI System for capturing knowledge about Bias in a Machine Learning System. A Machine learning system is (Training and Predicting) conceptualized based on elementary patterns by van Bekkum et al. [27]. Two symbolic systems trace the machine learning system and quantify bias in data and in the trained predictive model and predictions. Another symbolic system integrated the traces and documentation about bias into a knowledge graph (KG). Results of analytical methods on top of the KG describe results based on bias.

from the FakeNewsNet catalog[10] [30], where the illustrated fake news detection model ($\mathcal{F}$) is evaluated [29]. Based on the foundational design patterns proposed by van Bekkum et al. [27], Figure 2c presents the design pattern that underpins the implementation of $\mathcal{F}$. This model calculates the probability of the label for each News item. On average, the prediction probability is approximately 0.52 for PolitiFact and 0.54 for BuzzFeed. Notably, when focusing on news labeled Fake, the average prediction probability substantially increases to 0.83 for PolitiFact and 0.81 for BuzzFeed. Conversely, the prediction probability for News labeled as real is relatively low, with values of 0.19 for PolitiFact and 0.16 for BuzzFeed. This unexpected behavior of the model can be attributed to the skew towards Fake News in the paths representing the relationships among Users, Publishers, and News. Thus, these skew distributions– for News based on Users and Publishers– result in a significant accuracy loss when the model labels Real News. Further evidence for this claim was obtained by reviewing the classification results from reproducing the Fake News classifier published in [29]. In the validation set for the Buzzfeed dataset, which contained 36 observations, the model misclassified 5 observations, all of which had as target variable *Real*. In this work, while we acknowledge that the scope does not include an exhaustive real-world analysis, it is imperative to highlight the classifier's behavior. The classifier not only penalizes real news by misclassifying it as fake but also places considerable importance on user behavior in making predictions. This introduces a potential vulnerability in the system, as User sharing patterns may not always align with News veracity. For instance, the model's performance could be influenced by its handling of over or underrepresented attributes, such as highly active social media users compared to those who share only one or two News stories [2]. Such attribute-focused bias may significantly impact the overall accuracy and effectiveness of the system in real-world scenarios. To address these concerns, in this paper, we propose a documentation approach that resorts to a hybrid AI system to describe both the characteristics of the input datasets and the outcomes of machine learning models in terms of biases detected. To demonstrate the viability of our approach, we characterize the fake news detection problem over two implementations of our hybrid AI system, an integrated one that traces the process of a Probabilistic Soft Logic model and a principled approach that employs Random Forests as part of its architecture. Further, in our implementation, the captured knowledge is represented as factual statements within knowledge graphs (KG). By employing this approach, ML practitioners can gain a deeper understanding of the model's vulnerabilities by traversing the KG and conducting analytical studies using SPARQL queries. This enables a comprehensive examination of the impact that hidden biases in the input data may have on the model's output. Through this KG exploration and analysis, we aim to provide a more transparent and interpretable understanding of machine learning models' behavior and their potential limitations.

[10] https://github.com/KaiDMML/FakeNewsNet/tree/old-version

IV. OUR APPROACH FOR TRACING BIAS

In this section, we formalize the problem tackled in this work and present the architecture of our proposed solution [58].

A. PROBLEM STATEMENT

Consider an ML problem $\mathcal{P}$ defined in terms of X, Y, D, $\mathcal{H}$, and L as described in Section III-A1. Let h be a model of the problem $\mathcal{P}$. Let $\mathcal{F}$ be a family of bias measures specifically designed to address various *bias aspects* of the problem. These aspects include *Target Group*, *Attributes*, *Group Comparison*, and *Thresholds or Criteria*. The *problem of documenting bias*, also known as Doc-Bias, for the model $h()$ involves collecting all the attribute values that describe the *bias aspects* present in the datasets used and generated by h. These datasets encompass the training dataset D, the testing set X', and the set of predictions produced by $h(.)$. By systematically documenting measures in $\mathcal{F}$, we can gain insights into data biases and their impact on the model's performance. In our example in Section 2, we introduce the bias measure known as *overrepresentated*. This measure is defined in relation to a given dataset *T* and a grouping attribute *gAttr*, which can have at least two categorical values, *V1* and *V2*. The function *overrepresented(T, gAttr, V1, V2)* calculates the frequency of instances in *T* based on the *gAttr* values and outputs the absolute difference between the frequencies of *V1* and *V2* normalized to the size of *T*. The set of paths in Figure 2b corresponds to dataset *T*, and *overrepresented* can be computed for attributes *News*, *Publishers*, and *Users*, e.g., *overrepresented(T, News, "Fake", "Real")* is equal to $\frac{|19-13|}{32} = 0.1875$.

B. PROPOSED SOLUTION

We propose a hybrid AI system able to trace the life cycle of datasets ingested and processed by the predictive model h, and capture knowledge about *bias aspects* to enable the production of comprehensive documentation artifacts. For instance, in our example, a bias pattern exists if the overrepresented of Fake and Real News exceeds 0.15, i.e., *biasPattern(overrepresented(T, News, "Fake", "Real"), 0.15)*. By computing this metric, we can identify a problem that was not observable in the original News dataset but in their integration.

The operationalization of the analysis of model performance allows for quantification and supports the understanding of the impact of detected biases in machine learning pipelines, i.e., identifying under-representation (or over-representation) of data or model output and loss of performance for certain sub-sets in the data. Built on design patterns proposed by van Bekkum et al. [27], Figure 3 depicts a design pattern that models this hybrid AI system. A design pattern can comprise input datasets in the form of data or symbols (yellow rectangle), models (pink hexagons), processes (blue oval rectangles), output datasets in the form of symbols (blue rectangles), symbols documenting bias (yellow rectangles), and symbols representing factual statements of a KG (gray rectangles). This hybrid AI system is defined with four basic design patterns (See Figure 3).

Training and Predicting: This pattern models a system that solves the problem $\mathcal{P}$. It receives the input X and trains a model h from the space $\mathcal{H}$ based on the training data D. An inference process allows for the generation of the predictions as symbols. The design pattern in Figure 2c corresponds to an instance of *Training and Predicting* pattern for the problem of Fake News detection.

Tracing and Documenting Bias in Input: This pattern describes a system designed to capture knowledge about bias patterns in both the input data of the AI system and the generated training set, denoted as D. A tracing model defines the family $\mathcal{F}$ of bias measures and identifies specific *bias aspects* to be traced within the input and training data. The output is a set of factual statements documenting the identified bias insights supported by the bias measures.

Tracing and Documenting Bias in Models and Predictions: A system that documents how the predictive model h works and traces bias is represented with this pattern. It comprises two processes: one to integrate traces describing the model's execution and the predictions, and another to generate integrated traces and documentation following a tracing model. As a result, this system produces factual statements documenting the observed bias patterns.

Explaining Bias Impact: This pattern represents a system that receives factual statements describing the traces and measurements of bias and, following a semantic model, creates a knowledge graph (KG). Two processes compose this system: one that performs the KG creation and another that performs deduction over the KG to infer bias patterns. The output produced by this system are traces and explanations– in the form of symbols– that can be used to capture and uncover insights about bias patterns, rendered implicit otherwise, to determine their effect on the model's performance h and the generated output.

V. TRACING THE FAKE: A USE CASE

The documentation pattern presented in Section 3 is instantiated for the problem of Fake News detection. In order to show the versatility and effectiveness of our proposed solution, we perform two implementations. First, we implement Doc-Bias as a hybrid AI system using an integrated approach; we demonstrate it with a classification model based on probabilistic soft logic. Second, we implement it following a principled approach using Random Forests and Decision Trees.

A. IMPLEMENTING AN INTEGRATED HYBRID AI SYSTEM

Chowdhury et al. [29] define a Fake News classification pipeline as a credibility score-based model (CSM). This ML model is implemented in Probabilistic Soft Logic (PSL) [28], a statistical relational learning framework that utilizes weighted first-order rules, serving as a fuzzy or continuous relaxation of Boolean first-order logic, to assess the authenticity of News. The PSL model relies on five rules that

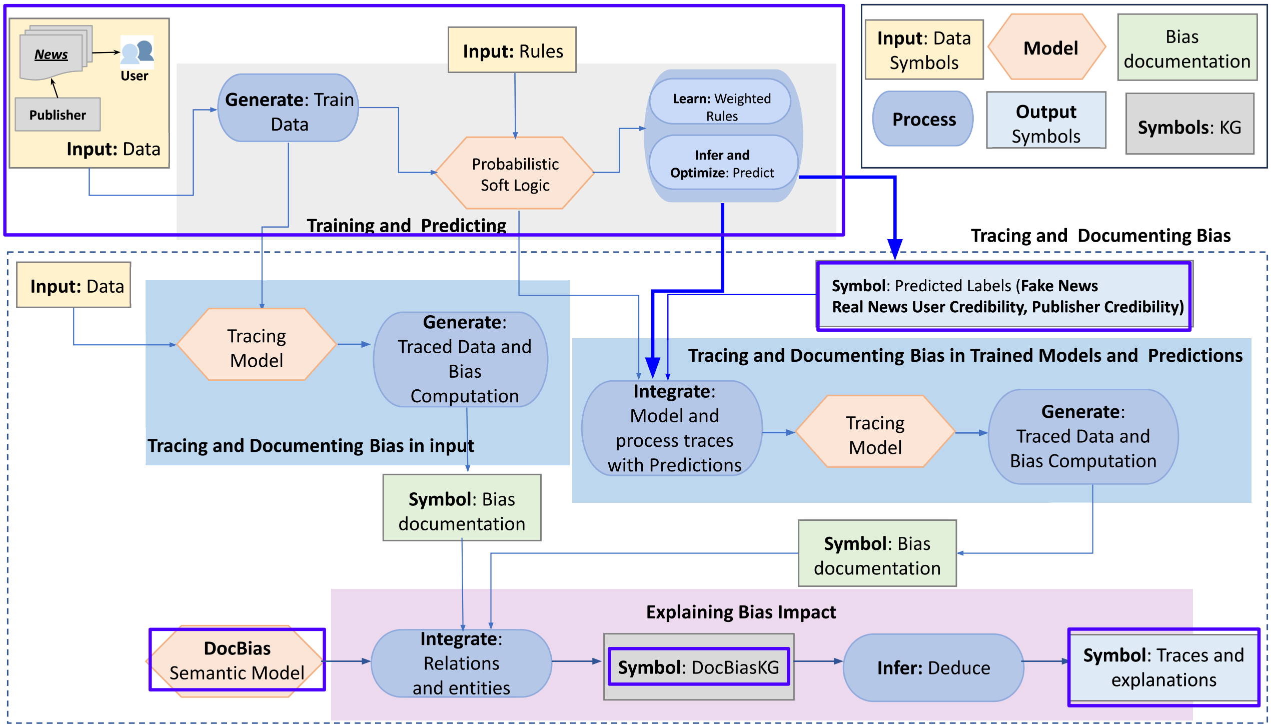

News
User
Publisher
Input: Data
Input: Rules
Generate: Train Data
Probabilistic Soft Logic
Learn: Weighted Rules
Infer and Optimize: Predict
Training and Predicting
Input: Data Symbols
Model
Bias documentation
Process
Output Symbols
Symbols: KG
Tracing and Documenting Bias
Input: Data
Tracing Model
Generate: Traced Data and Bias Computation
Tracing and Documenting Bias in input
Symbol: Predicted Labels (Fake News Real News User Credibility, Publisher Credibility)
Tracing and Documenting Bias in Trained Models and Predictions
Integrate: Model and process traces with Predictions
Tracing Model
Generate: Traced Data and Bias Computation
Symbol: Bias documentation
Symbol: Bias documentation
Explaining Bias Impact
DocBias Semantic Model
Integrate: Relations and entities
Symbol: DocBiasKG
Infer: Deduce
Symbol: Traces and explanations

FIGURE 4. Tracing as an Integrated Approach. The Design Pattern represents Doc-Bias as an Integrated Hybrid AI System, and boxed in blue are the main differences from the generic pattern. A Probabilistic Soft Logic Model is trained (Training and Predicting). Two symbolic systems trace the PSL model, learning and inference processes, and quantify bias in data and in the trained predictive model and predictions. Another symbolic system integrates the traces and documentation about bias into Doc-Bias KG. Results of analytical methods on top of the KG describe results based on bias.

play a crucial role in achieving accurate and reliable results:

$$r0w0 : UserCred(U), UserShare(N, U) \rightarrow \neg FakeNews(N)$$
$$r1w1 : MBFC(P) \rightarrow PublisherCred(P)$$
$$r2w2 : \neg MBFC(P) \rightarrow \neg PublisherCred(P)$$
$$r3w3 : PublisherCred(P), NewsPub(N, P)$$
$$\rightarrow \neg FakeNews(N)$$
$$r4w4 : FakeNews(N), NewsPub(N, P) \rightarrow \neg NewsPub(N, P)$$

Furthermore, [29] defines several key predicates denoted by *1)* to *6)* to represent various aspects of the problem. These predicates are essential to jointly learn the credibility of publishers and users and infer the authenticity of news.

1) *UserCred(U)*: This predicate indicates the credibility of User U.
2) *UserShare(N, U)*: A Boolean predicate representing that News N is shared by User U.
3) *FakeNews(N)*: Represents the label for Fake News.
4) *MBFC(P)*: Models the credibility score of Publisher P based on a website, e.g., MBFC.
5) *PublisherCred(P)*: Indicates credibility of Publisher P.
6) *NewsPub(N, P)*: A Boolean predicate representing that Publisher P has published News N.

The model aims to jointly learn *PublisherCred(P)* and *UserCred(U)* using prior knowledge captured by rules r2 and r3. Furthermore, the model infers the authenticity of News (*FakeNews(N)*) based on the available data.

Following this, we leverage the capabilities of PSL as a logic and statistical relational learning framework to present the integrated implementation of the *Doc-Bias* approach (see Figure 4). The proposed hybrid AI system first traces the PSL implementation of the AI model for Fake News detection, i.e., *Training and Predicting* pattern represents this AI system. Then, the system that uses the *Tracing and Documenting Bias in Input* pattern collects all the features that describe news, publishers, and users and their relations. Figure 5a illustrates a portion of the Doc-Bias KG that comprises the factual statements traced when the PSL model ingests the PolitiFact and BuzzFeed datasets. Note that in addition to labels and credibility scores, values representing over-representation and frequencies are also included. The execution of the SPARQL query on the bottom of Figure 5a allows determining if News is over-represented based on User interaction.

Lastly, the system specified by the *Tracing and Documenting Bias in Models and Predictions* pattern traces and integrates the weight learning, inference, and optimization processes involved in executing the PSL model to detect Fake News. Information about random variables captures the statements inferred by the model during the different applications of a rule. Figure 5b illustrates the RDF triples that represent the traces collected by this system. For example, the entity `nobias:News179` is described in its properties (e.g., User and Publisher credibility, ground truth labels) and the predictions made by the model stated by the entity representing the random variable `nobias:RVA14196`. The PSL model performance is described in terms of over-representation, prediction probability, number of ground rules, convergence point, and credibility class. The bottom of Figure 5b also depicts a SPARQL query whose evaluation retrieves per classified News, the prediction probability, bias measure, bias score, and over-representation.

1) DOC-BIAS KG IN DETAIL

The Doc-Bias KG comprises the classes, characteristics, and relationships needed to represent and trace bias across the ML pipeline whilst measuring documentation production at three steps of the ML life cycle: data ingestion, training and prediction, and bias assessment. The Doc-Bias ontology, or schema, defines classes, i.e., News, Publishers, Users, dataset, ML model, and their attributes, and the relationships between classes, i.e., shares, follows, and publishes. The data sources are then enriched by semantic representations, thus encoding information in a machine-readable format that consequently enhances interpretability. This prompts the Doc-Bias KG to produce bias-aware documentation artifacts at coarse- and fine-grain levels that observe the FAIR principles.

The Doc-Bias KG is created by the system described by the *Explaining Bias Impact* pattern. The *semantic model* is a data integration system DIS=$\langle O, S, M \rangle$ [59], where O corresponds to the Doc-Bias ontology defining concepts and properties of the traced entities, S is the set of data sources collected by the tracing system, and M comprises mapping rules expressing correspondences between O and S specified in the RDF Mapping Language (RML) [60]. 211 RML rules define the Doc-Bias KG; they are available in GitHub.[11]

A set of bias metrics captures bias patterns related to a target entity across the ML pipeline. The ontology that is integrated with the data sources consists of the necessary vocabulary to describe and contextualize the implementation of our use case. Moreover, the ontology used here is partly built by extending and re-using existing vocabularies to describe datasets and the characteristics of ML models. Concretely, the PROV-O Ontology [61], DCAT [62], ML schema [63], Description of a Model (DOAM)[12] ontology, FOAF Vocabulary [64], Data Quality Vocabulary [65]. The rest of the classes and characteristics in the Doc-Bias schema have been defined to account for actors associated with the Fake News domain, i.e., News, social media Users, and fact-checkers, as well as the vocabulary needed to describe the characteristics of PSL, i.e., rules, weights, atoms. Finally, we define classes to describe bias assessments, i.e., bias measure, bias detection method, and bias type. Table 2 summarizes the number of instances by classes in the Doc-Bias KG segmented by its corresponding *Documentation Step* according to the ML pipeline life cycle. In Appendix A, we show some of the queries used to retrieve information from the Doc-BiasKG in relation to the Documentation Steps stated here.

TABLE 2. Summary of relevant classes in the doc-bias knowledge graph.

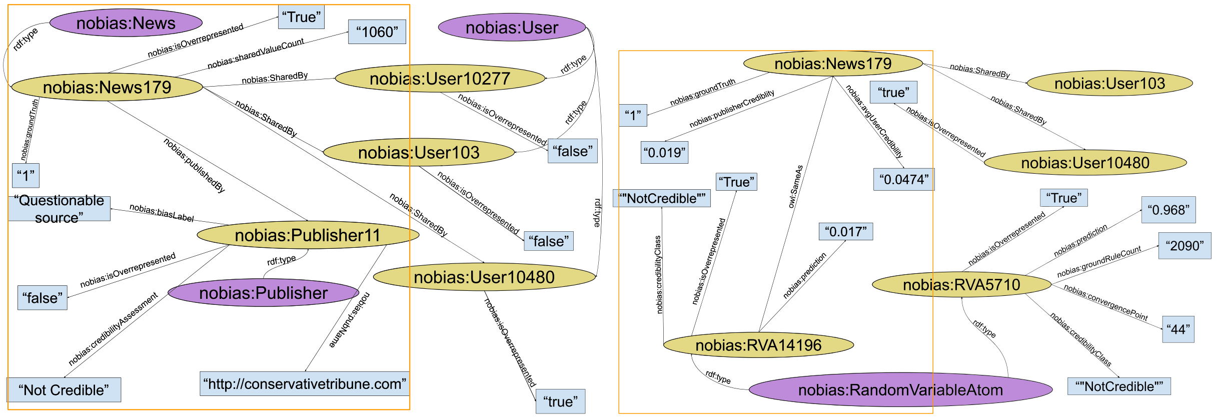
Documentation Step	Doc-Bias KG	Instances
Data Ingestion	News	422
	Publisher	103
	User	39,122
	SharingBehavior	55,570
	SocialInteraction	8,267
	FactCheckResource	3
	dcat:Dataset	3
Training Model and Prediction	Atom	94,723
	Rule	66,084
	GroundRule	66,078
	MaxTerm	66,078
	ObservedAtom	55,792
	InstantiatedArgument	39,261
	RandomVariableAtom	38,923
	Weight	720
	mls:EvaluationMeasure	360
	mls:ModelEvaluation	180
	AtomArgument	10
	Predicate	8
	mls:Model	4
Bias Assessment	BiasDetectionMethod	4
	MeasuresBiasIn	54,644

B. IMPLEMENTING A PRINCIPLED HYBRID AI SYSTEM

This section describes implementing the Doc-Bias approach over a hybrid AI framework, *InterpretME* [39], where the prediction task is to classify News based on the characteristics of Users and Publishers. Figure 6a demonstrates the workflow of the *InterpretME* pipeline. In the *Training* layer of *InterpretME*, the pipeline as an input accepts raw data about the News, Users, and Publishers from two different benchmark datasets, Dataset 1: BuzzFeed News and Dataset 2: PolitiFact News. Table 4 depicts the statistics of the datasets utilized for training the classification model. The dataset includes features, e.g., News, User, User credibility, Publisher credibility, and Publisher source. *InterpretME* as a pipeline offers an ensemble learning technique, such as random forests and decision trees. The pipeline performs data curation to handle categorical values for the classification model. Based on the pre-processed data, the optimized model and their hyperparameters are obtained through AutoML, e.g., the random forest model with the *depth of the tree: 5* and *entropy: gini*. The prediction problem involves the binary classification of News published by Users or Publishers as Fake. Henceforth, the target classes are obtained as Class 0: Real News (0) and Class 1: Fake News (1). Here, the trained ML model, i.e., decision trees, performs the prediction task with the list of best features generated from the random forest model. Additionally, the pipeline provides LIME interpretations, i.e., the interpretation of each News locally involved in the prediction task. Further, these interpretations generate a list of the top 10 relevant features with their weights, as well as prediction probabilities of classifying each news into target classes, such as real (0) and fake (1) news. In the *Tracing and Documenting* layer, *InterpretME* traces all the metadata, such as input features, model selection, hyperparameters, predictions, and results of LIME interpretations. For documentation, the pipeline utilizes the RML mappings to semantify the results obtained

[11] https://github.com/SDM-TIB/DocBiasKG/

[12] https://www.openriskmanual.org/ns/doam/index-en.html



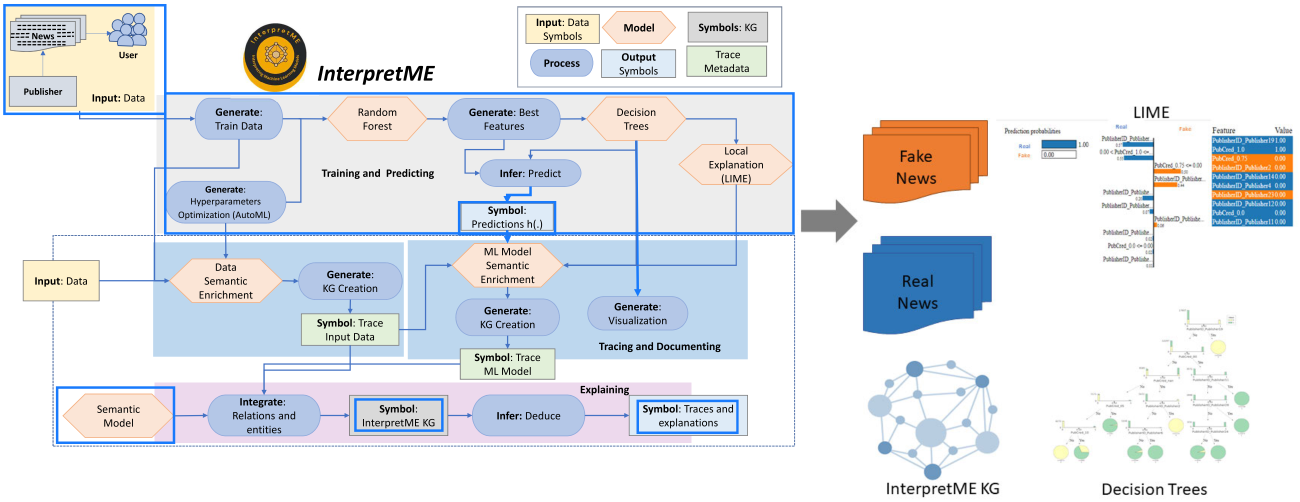
```
SELECT ?news ?groundTruth ?publisher ?biasMeasure
?biasScore ?overrepresented
WHERE {
         ?news a <News> ;
              <hasGroundTruth> ?groundTruth ;
              <hasSource> ?publisher;
              <associatedTo> <Data1>  .
        ?biasMeasure <measuresBiasInNews> ?news.
        ?biasMeasure <biasScore>  ?biasScore .
        ?news <isOverrepresented>  ?overrepresented .}
```

(a) Tracing and Documenting Bias in Input

```
SELECT ?classifiedNews ?predictedProbability ?predictedLabel
?biasMeasure ?biasScore ?overrepresented
WHERE {
    ?classifiedNews a <RandomVariableAtom> ;
                    <value> ?predictedProbability ;
               <hasCredibilityClass> ?predictedLabel;
                    <hasDataset> <Data1>  .
   ?biasMeasure <measuresBiasInRVA> ?classifiedNews.
   ?biasMeasure <biasScore>  ?biasScore .
   ?classifiedNews <isOverrepresented>  ?overrepresented .}
```

(b) Tracing and Documenting Bias in Models and Predictions

FIGURE 5. Running Example. Figure (a) depicts classes (News, Publisher, User), relationships (published by, shared by), properties (ground truth, bias label, overrepresented, credibility), and entities (News179, Publisher11, User10277) found in the Doc-Bias KG. The SPARQL query retrieves over-representation information of News calculated based on User interaction. Figure (b) depicts RDF triples that represent the traces collected by the system. Entities belonging to the class Random Variable item are described based on properties such as prediction, credibility class, convergence speed, ground rule count, and over-representation. The SPARQL query retrieves over-representation information of the selected News calculated based on its training process.



(a) Workflow of InterpretME

(b) Results

FIGURE 6. Tracing as a Principled Approach. Figure 6a demonstrates the use-case employing a principled Hybrid AI approach. The InterpretME pipeline preprocesses the input datasets of News collections and performs hyperparameter optimization for the ML classification task of Fake News detection. The trained model generates predictions, and LIME generates local explanations for each prediction. In the tracing layer, InterpretME traces and semantifies traces to generate the InterpretME KG. Figure 6b depicts the results produced by InterpretME (e.g., LIME and Decision Trees).

and generate the InterpretME KG. Moreover, *InterpretME* facilitates the visualization of decision trees and feature importance plots to understand the trained predictive model. An example of executing the use case of Doc-Bias over *InterpretME* is publicly available in the GitHub repository.

C. COMPARING TRACING POWER

Given the intrinsic characteristics of the implementations of our Hybrid AI system (i.e., integrated and principled), we provide here a comparative of the tracing power for each approach. Primarily, we want to emphasize and elaborate on

TABLE 3. Comparative overview of the systems.

Properties	Implementation	
	Integrated Hybrid AI System	Principled Hybrid AI System
Input	Dataset, First-order Rules	Dataset
Classifier	Credibility Score-based Model (CSM) [30]	Decision Trees, Random Forests
Learning Approach	Probabilistic Soft Logic (PSL)	Ensemble Learning
Output	KG with Traces and Explanations of Detected Biases	KG with Traces and Explanations of Predictive Models
Bias Metrics	Overrepresented Metric, Similarity Metric, Frequency Metric	
Granularity of Tracing Power	Fine-Grained	Coarse-Grained
Pros	Bias Detection, Interpreatability, Traceability	Interpretability, Traceability
Cons	Higher Computation, Longer Execution Times, More Data Storage	No Integrated Bias Detection, More Abstract Insights

the level of granularity (fine-grain vs. coarse-grain) we can attain with each type of implementation.

- **Integrated Approach**: The tracing power obtained through this approach is high, which provides fine-grained, human-interpretable, and machine-readable insights into the decision-making processes of the ML pipeline. The drawbacks of this approach are related to how much more expensive it is to achieve that. Higher computing and longer execution times, as well as more data storage space, are required. Further, access to the learning process and higher expertise and knowledge of how the model works are also required. This is crucial in determining what parts of the process should be integrated to obtain useful insights.
- **Principled Approach**: The tracing power obtained through this approach is lower than those of an integrated approach, which implies coarse-grained insights into the decision-making processes of the ML pipeline. That does not necessarily translate into lower quality, and yes, into a less expensive approach. The main drawbacks here have to do with the level of abstraction obtained with regard to how ML processes work. These subjects describe the input and output levels and the results of local searches.

Table 3 presents an overview of both implementations based on a selection of properties to facilitate further comparison.

VI. EMPIRICAL STUDY

We empirically assess the effectiveness and versatility of our approach over two implementations of a hybrid AI system in the context of Fake News detection. In particular, we aim at answering the following research questions: **RQ1)** What types of bias patterns are observed across the ML pipeline? **RQ2)** Do bias patterns impact the model performance? **RQ3)** Are biases traced at input present in the output? We set up the following configuration to assess our research questions.

A. BENCHMARKS

We conduct our evaluation over the FakeNewsNet catalog [30], which includes data from two fact-checking platforms: BuzzFeed and PolitiFact. The catalog includes News ID, textual content, labels, publishing websites (*a.k.a., Publishers*), and social engagement information. The news collected for the elaboration of the PolitiFact dataset were sampled from the fact-checking website PolitiFact and corresponded to News published in the time leading up to the 2016 U.S. Presidential Election. PolitiFact dataset comprises 86 publishers, 23806 users, 120 Fake news, and 120 Real news. Similarly, the BuzzFeed news data were sampled from news stories published on Facebook one week before the same Election and were fact-checked by BuzzFeed journalists. The BuzzFeed dataset includes 28 publishers, 15253 users, 90 Fake news, and 90 Real news. In our experiments, we do not resort to the content of the news, however, on contextual data (i.e., Publishers' credibility and Users' interaction).

Publishers are also identified through a unique ID and some corresponding metadata such as publishing websites. For instance, *Publisher9* is used to identify CNN, *http://cnn.it*, and *Publisher25* is used to identify the Washington Post, with their website being accessible via *http://washingtonpost.com*. Additional information on Publishers includes their trustworthiness scores obtained from Media Bias Fact Check.[13] The values of these scores range from 0.0 to 1.0. Moreover, the social context associated with the News on both datasets was extracted from Twitter. This refers to the engagement between a particular News item and social media Users, and is understood as the number of times a News item is shared by a given User in a particular social media platform. To summarize, the variables used in our experiments for the Fake News classification task are News, Publishers, Users, and the relations derived from News-Publisher and News-User interactions. The target variable is the credibility of the News: Fake (1) or Real (0). The breakdown of this information is summarized in Table 4.

TABLE 4. Input variables description across datasets.

Dataset	Real News	Fake News	Publisher	User
Buzzfeed	90	90	28	15,116
PolitiFact	120	120	90	23,605

TABLE 5. Variables description across different scenarios.

Dataset	Scenario	#Real News	#Fake News	#Publisher	#User
BuzzFeed	Scenario 1	90	90	28	15253
	Scenario 2	90	90	18695	18695
	Scenario 3	90	90	22752	15253
PolitiFact	Scenario 1	120	120	86	23806
	Scenario 2	120	120	29708	29708
	Scenario 3	120	120	32669	23806

B. EXPERIMENTAL SETTINGS

1) SETTINGS FOR INTEGRATED APPROACH

To implement the integrated Hybrid AI system, we reproduce the classification pipeline based on Probabilistic Soft Logic and define the credibility score-based model (CSM) as stated in Section V-A. All experimental settings are those reported in [29]. We highlight the following settings: We use

[13] https://mediabiasfactcheck.com/

TABLE 6. Evaluation results for the News classification using the PolitiFact and BuzzFeed datasets in various scenarios. The table shows the results of implementing Random Forests (RF) considering publisher credibility regarding precision, recall, and f1-score. Bold indicates the best performance in each scenario.

RF Model Scenarios	Classes	Publisher credibility					
		BuzzFeed			PolitiFact		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score
Scenario 1	Real	**0.94**	0.87	0.91	0.00	0.00	0.00
	Fake	0.88	**0.95**	**0.92**	**0.76**	1.00	0.87
Scenario 2	Real	**0.94**	0.87	0.91	0.00	0.00	0.00
	Fake	0.89	**0.95**	**0.92**	0.76	**1.00**	0.86
Scenario 3	Real	**1.00**	0.87	0.93	0.00	0.00	0.00
	Fake	0.89	**1.00**	**0.94**	0.77	1.00	0.88

TABLE 7. Evaluation results for News classification using the PolitiFact and BuzzFeed datasets in various scenarios with *oversampling* technique. The table shows the results of implementing Random Forests (RF) considering publisher credibility regarding precision, recall, and f1-score. Bold indicates the best performance in each scenario.

RF Model Scenarios	Classes	Publisher credibility					
		BuzzFeed			PolitiFact		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score
Scenario 1	Real	**0.94**	0.87	0.91	0.85	**0.97**	**0.90**
	Fake	0.88	**0.95**	**0.92**	**0.96**	0.83	0.89
Scenario 2	Real	**0.94**	0.87	0.91	**0.98**	0.82	0.89
	Fake	0.89	**0.95**	**0.92**	0.84	**0.97**	**0.90**
Scenario 3	Real	**1.00**	0.87	0.93	0.97	**0.98**	0.97
	Fake	0.89	**1.00**	**0.94**	**0.98**	0.97	**0.98**

continuous random grid search to learn the weights of the rules. We randomly choose 80% of data for model training and to learn hyperparameters and 20% of the data for testing. We repeat the experiments 30 times. We also evaluate the models with accuracy, precision, recall, and F1 measures.

2) SETTINGS FOR PRINCIPLED APPROACH

In our empirical study, the principled approach utilizes ensemble learning techniques such as random forests and decision trees. The evaluation of these predictive models shows the efficacy of news classification. A random forest model consists of numerous decision trees; each tree branch is created on the subset of features, and the outcome is generated by aggregating the predictions from all trees. However, the decision tree model utilizes the random forest traits to perform predictions. The method facilitates the interpretation and visualization of predictions. We evaluate the performance of the predictive models in terms of Precision, Recall, and F1-score. Here, the predictive pipeline utilizes AutoML[14] recommendations for optimal hyperparameters (e.g., max depth of the tree is 5). Additionally, for the robust performance of the predictive models, the approach employs *5-fold* cross-validation split-method. After each fold, the relevant features are traced from the predictive models and used to train the decision tree classifier. We randomly choose 70% of data for training and 30% for testing. We repeat the experiments 5 times and report the average metrics values.

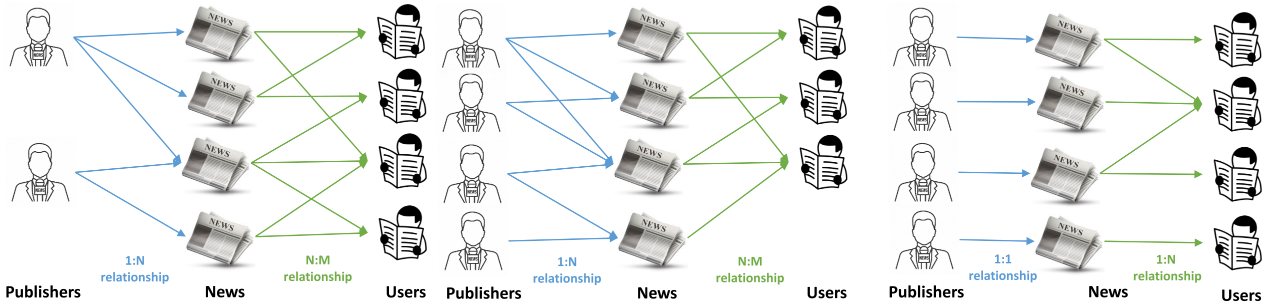
3) UNDERSTANDING THE IMPACT OF DOCUMENTING BIAS

In the scope of documenting bias, we take our original datasets and design three scenarios for both of them (*i.e., Scenario 1, Scenario 2, and Scenario 3*) to reveal the impact of the two predictive variables *Publisher* and *User* on the Fake News classification task, implementing the use of oversampling techniques (i.e., a technique to handle imbalanced datasets) over them (see Figure 7). Moreover, the *MIN*, *MAX*, and *AVG* counts, for example, *Publisher - News* represents MIN, MAX, and AVG counts of news per publisher; *User - News* represents MIN, MAX, and AVG counts of news per user; and similarly for *News - Users* counts of users per news, and *News - Publisher* counts of publisher per news. We describe each scenario below:

- Scenario 1 involves the proportion of users in the benchmark higher than that of publishers for sharing news. Figure 7 depicts the interaction between Publisher-News ($1 : N$) and User-News ($N : M$). In BuzzFeed, the count of N in Publisher-News interaction ranges from 1 to 32. Moreover, in User-News interaction, the count of N ranges from 1 to 40, and M varies from 1 to 1196. However, in the PolitiFact dataset, the maximum publisher count per News is 43 and the user's count in sharing the news is 40. Furthermore, BuzzFeed involves 28 publishers and 15253 users for sharing political news, and the PolitiFact dataset includes 86 publishers and 15253 users. This demonstrates the real-world scenario where users actively share news, highlighting the potential bias in the classification of Real news.
- Scenario 2 consists of more publishers than the user. The dataset comprises an imbalance distribution to investigate publisher bias in classifying news as Fake. The BuzzFeed benchmark includes 22752 publishers and 15253 users. Whereas, in the PolitiFact dataset, 32669 publishers and 23806 users share news. Here, the count of N for Publisher-News interaction is 1 to 29. Additionally, in BuzzFeed, the maximum count of News shared by users is 29. In PolitiFact, the count of users sharing news is 1 to 26. In both datasets, the average user interaction with news is 126.4 and 137.2, respectively.
- Scenario 3 comprises the equal distribution of users and publishers count. This distribution provides a fair comparison for evaluating the ML model's performance between the Publisher-News ($1 : 1$) and User-News ($1 : N$) relationship. Further, in both datasets, the maximum count of users interacting with news is 1196 and 2969. Moreover, the scenario represents the ideal situation where the variable distribution is balanced. Here, the distribution of users and publishers is 29708 in PolitiFact and 18695 in the BuzzFeed dataset, respectively.

Table 5 summarizes the descriptive statistics for the resulting datasets under all three scenarios. We evaluate the performance of the predictive models in the aforementioned scenarios to detect potential biases creeping into the output generated by the ML model. Here, we also follow the same criteria as reported in [29] and consider the credibility scores of publishers and users for the classification task. Table 6, and Table 7 shows the observed results for the RF model with and

[14] https://www.automl.org/



Dataset	MIN	MAX	AVG
Publisher - News			
BuzzFeed	1	32	6.4
PolitiFact	1	43	2.6
Users - News			
BuzzFeed	1	48	1.5
PolitiFact	1	40	1.3
News - Users			
BuzzFeed	1	1196	126.4
PolitiFact	1	2969	137.2
News - Publisher			
BussFeed	1	1	1.0
PolitiFact	1	1	1.0

Scenario 1

Dataset	MIN	MAX	AVG
Publisher - News			
BuzzFeed	1	29	1.2
PolitiFact	1	26	1.1
Users - News			
BuzzFeed	1	29	1.2
PolitiFact	1	26	1.1
News - Users			
BuzzFeed	1	1196	126.4
PolitiFact	1	2969	137.2
News - Publisher			
BuzzFeed	1	1196	126.4
PolitiFact	1	2969	137.2

Scenario 2

Dataset	MIN	MAX	AVG
Publisher - News			
BuzzFeed	1	1	1.0
PolitiFact	1	1	1.0
Users - News			
BuzzFeed	1	48	1.4
PolitiFact	1	40	1.3
News - Users			
BuzzFeed	1	1196	126.4
PolitiFact	1	2969	137.2
News - Publisher			
BuzzFeed	1	1196	126.4
PolitiFact	1	2969	137.2

Scenario 3

FIGURE 7. Datasets' Description. Three scenarios are used to evaluate the impact of Users and Publishers on the outcomes of predictive models. Scenario 1 represents the original distribution of the studied two state-of-the-art benchmarks. Scenario 2: extends the dataset Scenario 1 and comprises an equal number of users and publishers in the dataset. Scenario 3 also extends datasets in Scenario 1 but is composed of more publishers than users in the original benchmarks. Values of MIN, MAX, and AVG for out-degrees from Class$_i$ to Class$_j$ are reported.

without oversampling technique in the respective scenarios. In BuzzFeed, the random forest model accurately classifies Real news, with precision scores ranging from 0.94 to 1.00 in various scenarios. Moreover, when the oversampling technique is applied, the performance of the RF model remains stagnant. However, in the PolitiFact dataset, the RF model struggles to recognize Real news due to an imbalance distribution of input variables. Without oversampling, the value of the precision metric is 0.00. The lower performance in the PolitiFact dataset is attributable to a higher number of publishers disseminating Fake News than Real. Furthermore, the RF model with oversampling reveals the improved performance of the RF model with values ranging from 0.85 to 0.97 in Real News classification. Thus, the results underline that potential distribution biases in the dataset can impact the model's output, with a bias towards Fake News. Table 8 and Table 9 indicate the RF model performance based on the User credibility score used in Fake News classification. The RF model on the BuzzFeed dataset demonstrates its best performance when classifying Real News. In all scenarios except Scenario 2, the precision score is 1.00. However, in the PolitiFact dataset, the performance remains consistent with previous experiments and highlights the model-biased behavior toward classifying news as Fake. Moreover, in both datasets with oversampling, the model outperforms all scenarios. Here, the predictive model accurately classified news as Real, with a precision score of 0.99. The evaluation results elucidate the need for assistance in understanding, analyzing, and interpreting the biases present in the output generated by predictive models. The reported evaluation results summarized in Table 6,7,8 and 9 are generated using SPARQL queries over the InterpretME KG. Appendix B shows an exemplar query to retrieve the RF model performance in Fake News classification.

TABLE 8. Evaluation results for the News classification using the PolitiFact and BuzzFeed datasets in various scenarios. The table shows the results of implementing Random Forests (RF) considering user credibility in terms of precision, recall, and f1-score. Bold indicates the best performance in each scenario.

RF Model		User credibility					
Scenarios	Classes	BuzzFeed			PolitiFact		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score
Scenario 1	Real	**1.00**	0.96	**0.98**	0.00	0.00	0.00
	Fake	0.97	**1.00**	**0.98**	0.76	**1.00**	0.87
Scenario 2	Real	**1.00**	0.97	**0.98**	0.00	0.00	0.00
	Fake	0.97	**1.00**	**0.98**	0.76	**1.00**	0.87
Scenario 3	Real	**0.94**	0.87	0.91	0.00	0.00	0.00
	Fake	0.88	**0.95**	**0.92**	0.77	**1.00**	0.88

C. BIAS MEASURES

Our implementation uses the following bias measures: overrepresentation, similarity metric for Users, and frequency measure. Overrepresentation(O), as a relaxation of functional enrichment analysis [66], enables us to analyze a dataset (T) by assessing the distribution of a grouping attribute ($gAttr$), based on its frequency for each $x \in T$ ($freq(x)$). O quantifies the degree to which an instance of ($gAttr$) is overrepresented

TABLE 9. Evaluation results for the News classification using the PolitiFact and BuzzFeed datasets in various scenarios with *oversampling* technique. The table shows the results of implementing Random Forests (RF) considering user credibility regarding precision, recall, and f1-score. Bold indicates the best performance in each scenario.

RF Model Scenarios	Classes	User credibility					
		BuzzFeed			PolitiFact		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score
Scenario 1	Real	**1.00**	0.98	**0.98**	**0.99**	0.94	**0.97**
	Fake	0.98	**1.00**	**0.99**	0.94	**0.99**	0.97
Scenario 2	Real	**1.00**	0.97	**0.99**	**0.97**	0.90	**0.94**
	Fake	0.97	**1.00**	**0.99**	0.90	**0.98**	**0.94**
Scenario 3	Real	**1.00**	0.96	0.98	**0.99**	0.92	**0.96**
	Fake	0.97	**1.00**	**0.98**	0.93	**0.99**	**0.96**

at the top of a ranked list.

$$\boxed{O(x, P_t, gAttr, T) = 100 \cdot \frac{freq(x, T, gAttr) - P_t}{P_t}}$$

The degree of over-representation is determined following a specified threshold (t). In our implementation, the threshold is a percentile (P) in the context of the distribution of the values of ($gAttr$). For P_t corresponding to 90% all entities that fall over that threshold are flagged as overrepresented. Further, we apply Fisher's exact test to determine whether the distinction of being overrepresented is significantly associated with the label. We also resort to the Jaccard index [67] to define three measures to quantify the similarity of Users. The first one, named *UNews(.,.)*, receives the News posted by a pair of Users, denoted by *News(u_i)* and *News(u_j)*, and returns value between 0.0 and 1.0 representing the commonalities between the News posted by Users u_i and u_j.

$$\boxed{UNews(u_i, u_j) = \frac{|News(u_i) \cap News(u_j)|}{|News(u_i) \cup News(u_j)|}}$$

Using Jaccard index, we also define *URealNews(.,.)*) and *UFakeNews(.,.)*, where *RealNews(u_i)* and *FakeNews(u_i)* denote the real and Fake posted by User u_i, respectively. And assume a threshold of 0.75 to identify bias patterns in both cases. Additionally, we compute the similarity between Users based on their sharing behavior of Real News (a.k.a. *URealNews(.,.)*) and also according to Fake News (a.k.a. *UFakeNews(.,.)*).

$$\boxed{URealNews(u_1, u_2) = \frac{|RealNews(u_i) \cap RealNews(u_j)|}{|RealNews(u_i) \cup RealNews(u_j)|}}$$

$$\boxed{UFakeNews(u_1, u_2) = \frac{|FakeNews(u_i) \cap FakeNews(u_j)|}{|FakeNews(u_i) \cup FakeNews(u_j)|}}$$

We use the latter metric to measure interaction overlap between Users and the label of the News they share to emulate in-group bias by observing similar news sharing behavior. The *Jindex* value ranges from 0 to 1, with a higher value indicating higher similarity regarding this behavior; spotting this pattern is enabled by setting a high *Jindex* value. Finally, the frequency measure (F) receives a multi-set and returns the frequency of each element in the multi-set; we set a threshold (F>10) to identify prominent reoccurring entities.

Informed by the literature, Table 10 summarizes the types of bias we have identified to be relevant within the context of our classification problem. The measures associated with them are also specified. While far from comprehensive, this compendium of metrics lays the foundation to support the interpretation and understanding of bias in our case study at domain-agnostic and domain-specific levels.

D. DOC-BIAS AS AN INTEGRATED HYBRID AI SYSTEM

1) BIAS IN INPUT DATA

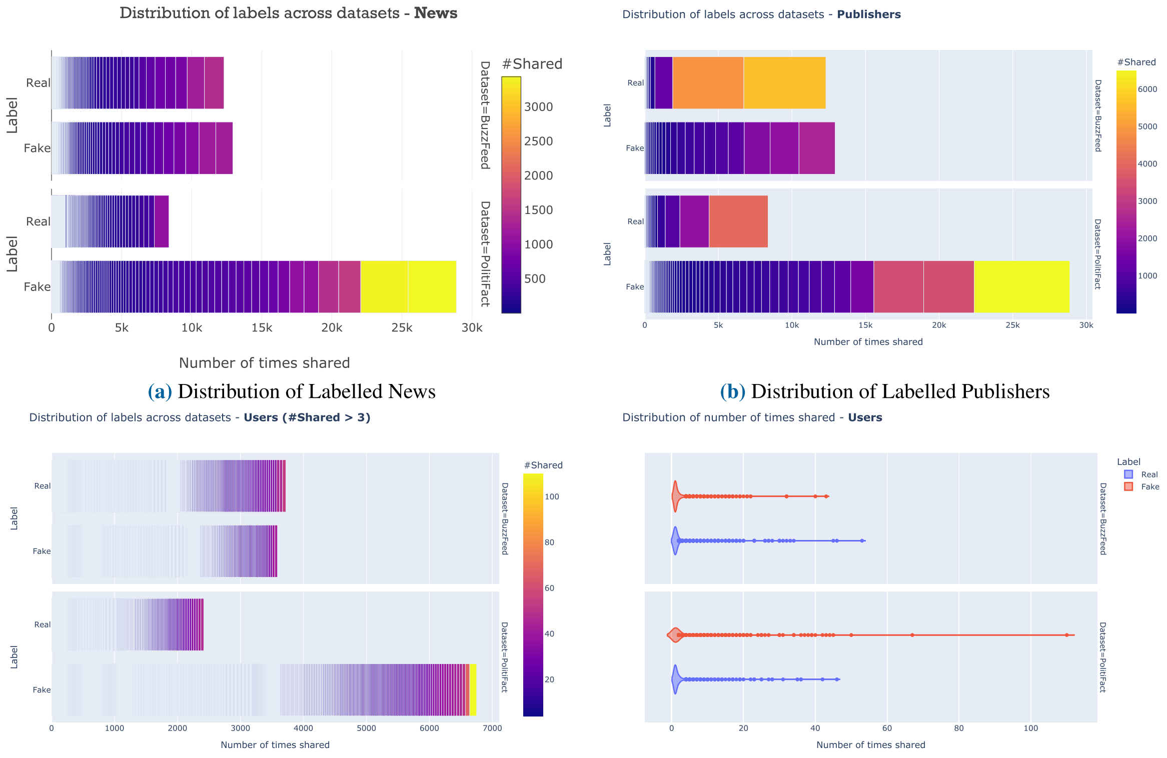
We start our analysis by further characterizing the input datasets, Dataset 1: BuzzFeed News and Dataset 2: PolitiFact News. We have three input variables: News, Publishers, and Users. News and Publishers' information is obtained from the same tabular dataset and originates in the News source, i.e., a fact-checking website, while the information about Users is mined from Twitter to be added during data pre-processing. Prior to this step, both datasets are balanced in terms of News labels, Fake (1) and Real (0). Upon pre-processing, the inclusion of User engagement data significantly impacts the initial distribution of the labels across two dimensions. Figures 8a, 8b, and 8c illustrate how the distribution of all three input variables becomes skewed towards the Fake News, exacerbated in the PolitiFact dataset. Not only that, but another skew that we can already perceive related to users' interaction when coupled with News entities is the creation of overrepresented entities across the variables, as well as a stark skew in the distribution of the Users and the number of News they share. Concisely, Figure 8d shows how 90% of Users share between 1 and 2 News, while there are a few numbers of Users that share a larger amount. Regarding the News and Publishers, we observe a slight reverse effect where it is easier to depict fewer actors capturing most of the shares. This is the case in BuzzFeed, where most of the Real News is associated with two Publishers. Fake News displays a higher diversity of publishing sources. In order to quantify the number of overrepresented entities at the input level, we implement the overrepresentated measure to detect them in all three of the variables and across both datasets. We then apply Fisher's exact test to determine whether the distinction of overrepresented entity is significantly associated with the label. The test results are significant for the User-Label association in both datasets. This implies that the distribution of overrepresented users in each dataset is different from the ones that are not, illustrating skew in the input data at the level. Additionally, the News-Label association in the PolitiFact dataset also yields significant results for the same test. This could imply a higher bias in this dataset with respect to the BuzzFeed one regarding this dimension. Due to the results obtained here, we monitor this pattern as we move along the classification pipeline.

a: USER'S BEHAVIOR

Motivated by the significance of the User's behavior in our initial assessments, we opted to take a fine-grained

TABLE 10. Bias measures.

Type of Bias	Bias Description	Bias Measure
Overrepresentation	This refers to observations in a particular sub-set, i.e., a variable in a dataset, that are statistically overrepresented within that corresponding sub-set.	Overrepresentation Metric
In-group Bias	This refers to how people tend to favor people who exist in similar groups as them. These groups could be formed by gender, race, ethnicity, favorite sports team, political convictions.	Similarity Metric
Activity Bias	This refers to how not all people are active all the time in social media, i.e., 90% are lurkers; and how a select few are responsible for creating more than 50% of content.	Frequency Measure



(a) Distribution of Labelled News

(b) Distribution of Labelled Publishers

(c) Distribution of Labelled Users (With #Shares >3)

(d) Distribution of News Shared By Users

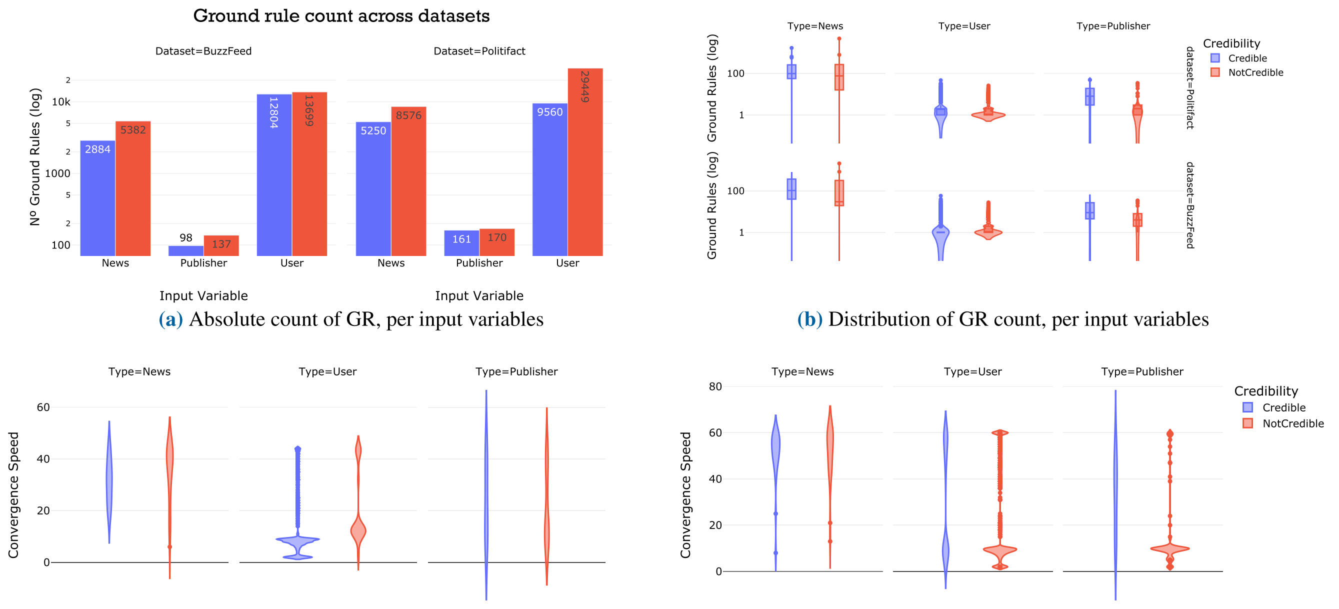
FIGURE 8. Distribution of labels per input variable across datasets and distribution of User-News combination.

look at their characteristics. We aim to detect any potential behavioral patterns that could emulate real-world phenomena associated with Fake News dissemination, i.e., in-group bias or confirmation bias, in our data sources. To analyze User-News relationships, we implement the similarity metric *UNews* and visualize the results as shown in the heatmaps in Figure 10 (a) and (b). Dendrograms are also depicted, and the label (Fake or Real) is used for the corresponding news. Unsurprisingly, few patterns can be spotted in both datasets without sub-setting our data (we attribute this difficulty to the increased amount of noise Users-News combinations that only share one News produce). Despite this, some of the identified patterns allow us to determine that there is a long range of Users in the BuzzFeed dataset (e.g., from User5865 to User13222) that show a similar behavior in News sharing. More interestingly, most of the news shared by these clustered users is real. A similar cluster of users is also visible on the lower right of BuzzFeed and in the PolitiFact heat maps.

We perform a more in-depth analysis to characterize social interactions among Users. Upon implementing similarity metrics,*UReal* and *UFake*, we can uncover a hidden pattern in relation to User behavior and the News' credibility. Figure 10 (c) denotes the expected bimodal distribution of the predictions (values comprehended in the range [0.0, 1.0]). Nevertheless, a notable concentration around values representing Real News (0) is observed, depicting that Users with a higher similarity score share more Real News than Fake.

2) BIAS IN MODEL

We now assess the model's handling of the skewed distribution of User-News interactions that was discovered at the input level, as seen in Figure 8 (d). The frequency measure allows us to observe how the classification process prioritizes popular Users (News share counts > 10) over less popular Users (News share counts < 10). Figure 12 depicts a correlation plot between several shares and the prediction error for the User's



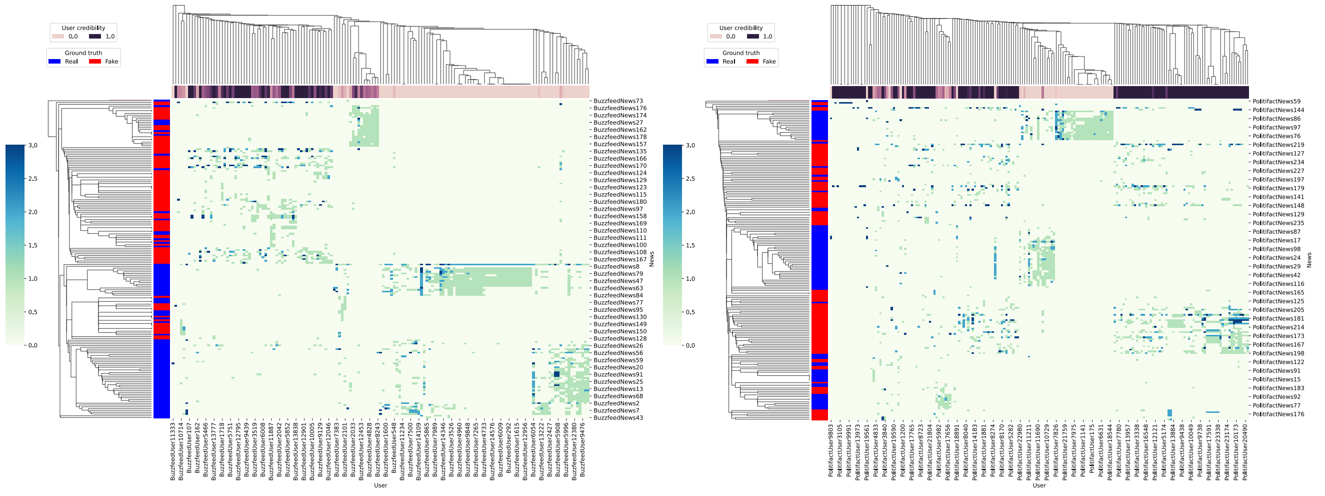
(a) Absolute count of GR, per input variables

(b) Distribution of GR count, per input variables

(c) Convergence speed of RVAs per type, BuzzFeed

(d) Convergence speed of RVAs per type, PolitiFact

FIGURE 9. **Ground rules characterization. (a) Ground rules count representing input variables (absolute terms), (b) Distribution of ground rules linked to instances representing input variables, (c - d) Distribution of RVA convergence speed. Skews in distributions impact the model's convergence; it learns faster labels for Fake News and no-credible Users and Publishers.**



(a) BuzzFeed Dataset

(b) PolitiFact Dataset

FIGURE 10. **User sharing behavior patterns. In sub-figures (a) and (b), similarity metric *UNews* is implemented to detect clusters of Users and assess News sharing patterns.**

predicted credibility. We characterized this as potential activity bias, which aligns with how PSL favors random variable atoms of louder Users due to higher significance (or higher ground rule count). While this is not necessarily problematic for the model's performance, it is interesting to further our understanding of how it works.

To trace the irregularities in User-News interactions we observed in the input data, it is necessary to dive deeper into the distribution of ground rules. Ultimately, it is the ground rules that connect the different entities - represented by ground atoms - with each other and, therefore, emulate relations between them. The intuition is that News-User and News-Publisher interactions are translated into ground rules in the classification process.

One of the irregularities observed in the data was how a few Users shared many news stories, while others shared only a few. This leads to more constraints— manifested by ground rules— for the random variable atoms (RVAs) representing the credibility of users who are overrepresented. This has an effect on the training process because the classifier favors those User RVAs with higher ground rule count, as small value changes in those RVAs can

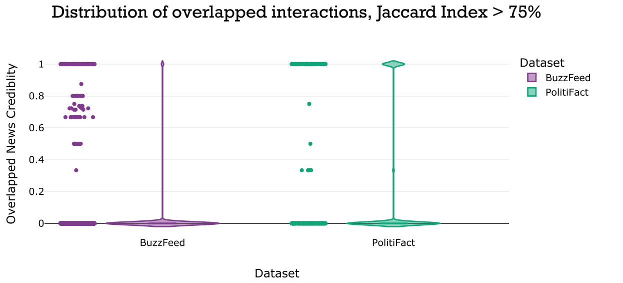


FIGURE 11. User sharing behavior patterns 2. In sub-figure (c), a fine-grain analysis is performed by implementing similarity metrics, *UReal* and *UFake*. Here, social interactions with a high similarity score correspond with users sharing real news (0) over those who share fake news (1).

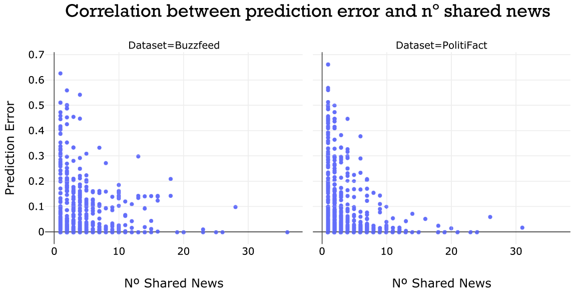


FIGURE 12. Activity Bias pattern.

have a higher impact on the total incompatibility of constraints.

We can see how RVAs with more ground rules (i.e., News) are highly favored in Figure 9 (a) and (b). Their values also take longer to reach convergence. Figure 9 (c) and (d) shows the convergence speed for News, Users, and Publisher atoms, respectively, and across both datasets. Figure 13 illustrates how rules associated with the credibility of the Users overwhelmingly contribute (90 percent) to the final classification to the detriment of the credibility of the Publishers (around 10 percent). This can lead us to imply that the User's credibility is a predictive feature in the PSL model that heavily impacts the model's inference process. Interestingly, this happens for both datasets, and the distribution followed is quite similar.

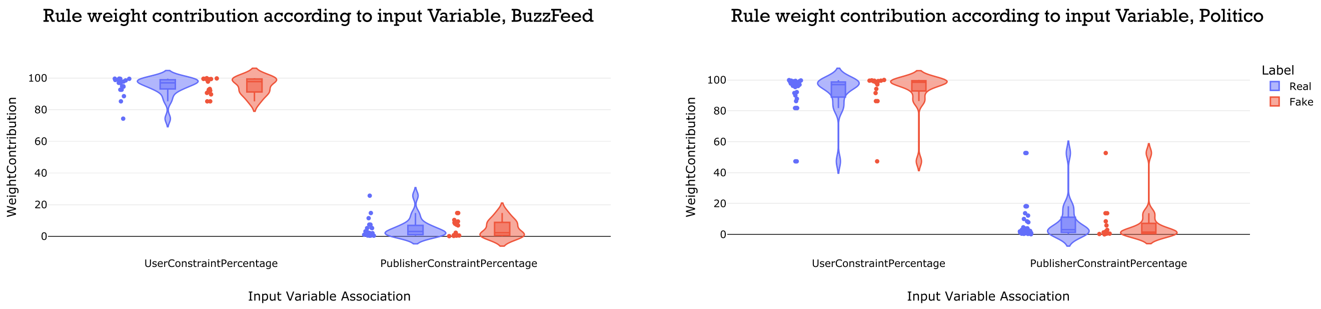
3) BIAS IN PREDICTIONS

As a last step in our study, we evaluate the output or predictions generated by the CSM model. Figure 14 (a - b) illustrates data points representing classified instances by type, News, Users, and Publishers. They are colored by the inferred truth value. Blue points represent entities that the model classified as credible, in case of News this implies real; the red points represent those that the model classified as not credible, for News entities this implies Fake. Here, we determine that the CSM model predicts more ‘‘not credible’’ entities than ‘‘credible’’ entities for all variables. Further, and to better interpret the output produced for the entities representing News, we perform a correlation analysis. The results show there is a strong correlation between News credibility and the User's credibility, less so with the Publisher's credibility. This is reflected in the analysis of false positive rates and how the common characteristic was an inferred low User credibility across all misclassified entities. There is also a correlation between the credibility of the News and the number of times it was shared. This could be indicative of a slight skew that denotes high sharing counts associated with Fake News.

E. DOC-BIAS OVER INTERPRETME

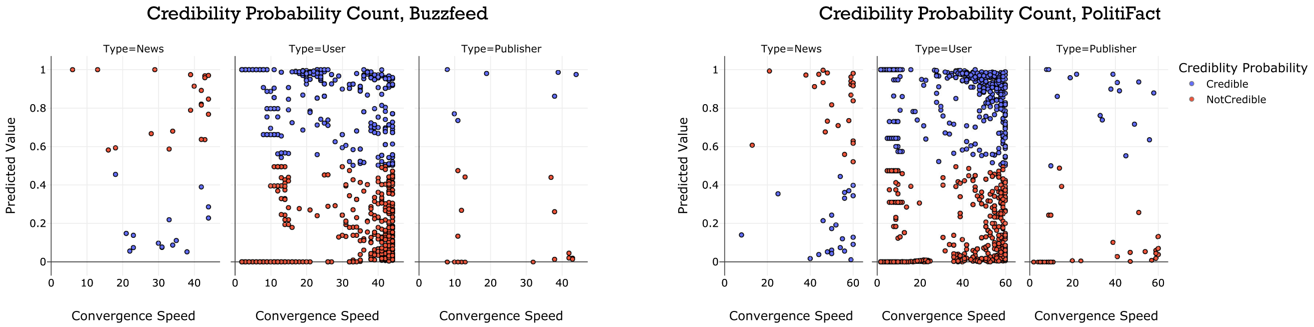
1) MODEL'S BEHAVIOR

We analyze the impact the model's behavior has on the produced output, as the input dataset characterization is the same for both implementations. The pipeline of *InterpretME* is evaluated with 8 test beds. Figure 15a, Figure 15c, Figure 15b and Figure 15d represent the results of decision trees with the User and Publisher credibility role in the classification problem over the PolitiFact and BuzzFeed datasets. Figure 15a and Figure 15b illustrate the decision tree considering the impact of User credibility on the classification of Fake News. Here, the visualization of the PolitiFact dataset states that Users with a credibility score of 1.0 (23966 data samples in Class 1) are more likely to publish Fake News, whereas Users with a credibility of 0.0 (6694 data samples in Class 0) and Publisher source -(Publisher17- 253 data samples in Class 0) are more likely to publish Real News. Users with credibility scores between 0.0 and 1.0, 70% of the population publish Fake News. Further, in the BuzzFeed dataset, the decision tree exemplifies that a User credibility score of 0.0 (10344 data samples in Class0) and with Publisher sources -(Publisher19 (391 data samples in Class 0), (Publisher2 (122 data samples in Class 0), and (Publisher9 (580 data samples)- are more likely to publish Real News. Conversely, a User with a credibility score of 1.0 (10154 data samples in Class 1) is more likely to publish Fake News. The trained classification model reports an accuracy of 0.97 for both datasets. Further, in PolitiFact, the precision of 0.99 for Fake News and 0.95 for Real News is reported, while in BuzzFeed, the precision reported is 1.0 for Real News and 0.97 for Fake News. Figure 15c and Figure 15d illustrate the decision tree considering the impact of Publisher credibility in the classification of Fake News. In the PolitiFact dataset, the decision tree depicts that a Publisher with a credibility score of 0.75 (3661 data samples in Class 0) and 1.0 (23560 data samples) is more likely to publish Real News, whereas a Publisher with credibility of 0.0, and Publisher source -Publisher28- (6013 data samples in Class 1) is more likely to publish Fake News. Other important publishing sources are -Publisher2 and Publisher56- which publish Real News. Further, in the BuzzFeed dataset, the decision tree demonstrates that a Publisher source, i.e., Publisher19 (5145 data samples in Class 0), is more likely to publish Real News. Conversely, a News item with a Publisher credibility score of 1.0 and -Publisher11, Publisher28, and Publisher24- are more likely to publish Fake News. Furthermore, the trained ML model reports an accuracy of 0.97 for PolitiFact and 0.98 for BuzzFeed. Additionally, for PolitiFact, the precision



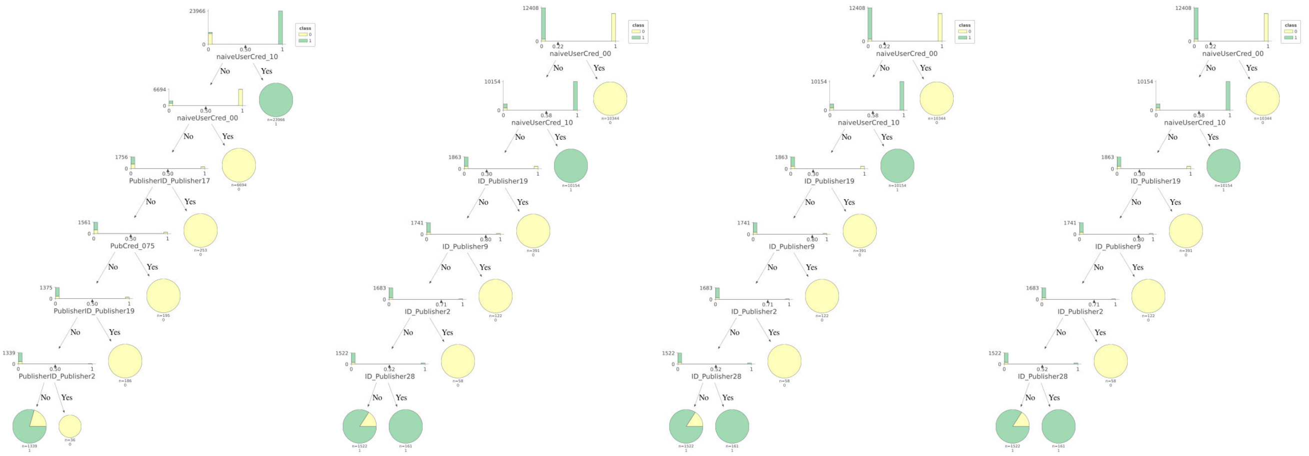
(a) The BuzzFeed dataset **(b)** The PolitiFact dataset

FIGURE 13. Rule's weight contribution to News classification.



(a) Overview output generated by CSM, BuzzFeed **(b)** Overview output generated by CSM, PolitiFact

FIGURE 14. Output generated by CSM model per dataset. Truth values for predicted instances are "credible" (blue) and "not credible" (red). The CSM model predicts more "not credible" entities than "credible" entities for all variables.

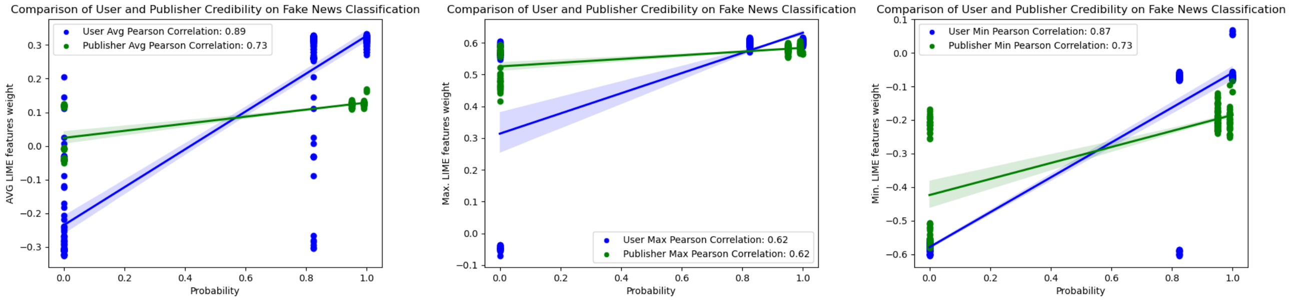


(a) PolitiFact User Credibility **(b)** BuzzFeed User Credibility **(c)** PolitiFact Publisher Credibility **(d)** BuzzFeed Publisher Credibility

FIGURE 15. InterpretME generates decision trees representing interpretations for the News classification task based on PolitiFact and BuzzFeed datasets. Figure 15a and Figure 15b demonstrate the output of decision trees considering User credibility. Figure 15c and Figure 15d reveal the output of decision trees considering Publisher credibility. Here, 0 in yellow color represents Real News, and 1 in green color represents Fake News.

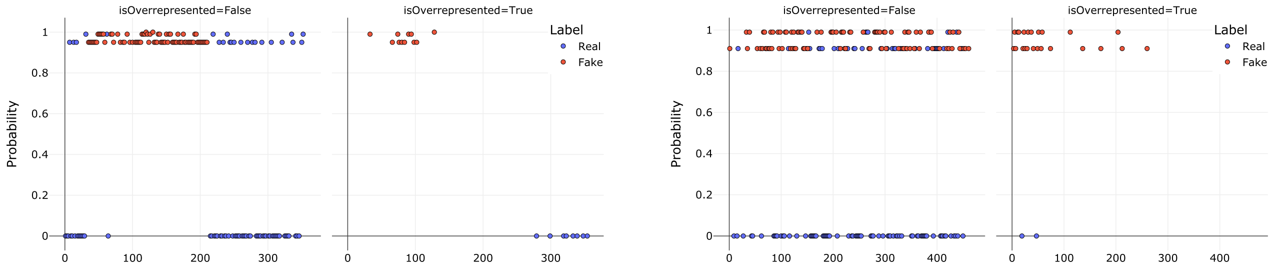
of 0.95 for Fake News and 1.0 for Real News is reported, while in the BuzzFeed dataset, the precision score reported is 1.0 for Real News and 0.97 for Fake News. Figure 17 illustrates how the classifier is more confident at classifying Fake News across both datasets, Moreover, by calculating the Overrepresentation measure over News items, we can observe how the model correctly classifies all these data points.

Pearson Correlation and Linear Regression Analysis. Figure 16 depicts the statistical analysis performed over the InterpretME KG based on the BuzzFeed dataset. The execution of SPARQL queries over the InterpretME KG retrieves the results regarding the impact of LIME features (e.g., publisher credibility) and the probability of predicting fake news. The linear regression analysis allows us to observe



(a) Average features impact (b) Maximum features impact (c) Minimum features impact

FIGURE 16. Pearson Correlation and Linear Regression Analysis plots illustrate the statistical and comparative analysis of Fake News classification based on the BuzzFeed dataset. Figure 16a, Figure 16b, and Figure 16c represent the comparison of User and Publisher credibility with average, maximum, and minimum impact on the prediction probability of Fake News classification. The analysis shows that User credibility has *more impact* on the classification of Fake News. Here, the blue color represents User credibility, and the green color is Publisher credibility. A straight line depicts the regression line, summarizing the relationship between feature impact and prediction probability and scatter plot, with each data point representing an impact value and their corresponding prediction probability.



(a) Buzzfeed (b) PolitiFact

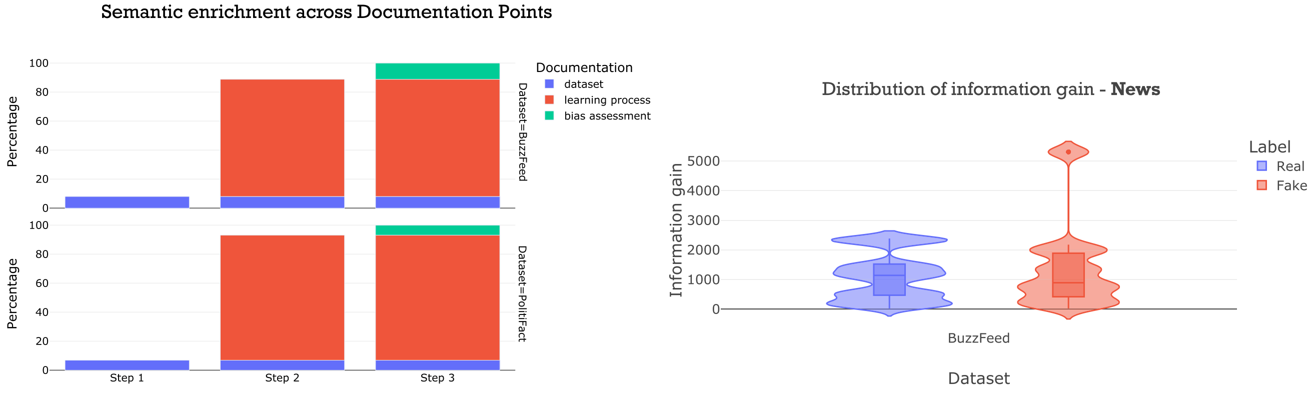
FIGURE 17. Output generated by the predictive model integrated into the InterpretME system. Truth values for predicted instances are "Real" (blue) and "Fake" (red). The model predicts more "Fake" entities than "Real" entities across both datasets. Overrepresented entities are least likely to be misclassified.

the relationship between probability and average feature weight generated by LIME. Figure 16a shows the average Pearson correlation to be positive for *User credibility* score as compared to *Publisher credibility*. This reveals that an increase in feature weights predicted by LIME also increases the probability of the predictive models for Fake News classification, considering the *User credibility* as one of the features in the BuzzFeed dataset. Therefore showing a strong positive correlation. On the other hand, Figure 16b studies the maximum impact of feature weights predicted by LIME considering *User credibility* and *Publisher credibility* on the prediction probability. The maximum impact in terms of the Pearson correlation coefficient was found to be the same for both *User credibility* and *Publisher credibility*. Similarly, in Figure 16c, when considering the minimum impact features, the *User credibility* impacts more in comparison to *Publisher credibility*. To summarize the Pearson correlation and linear regression study, we can infer that *User credibility* always has a significant positive correlation compared to *Publisher credibility*. These investigations demonstrate useful insights into the elements that influence Fake News detection tasks.

VII. DISCUSSION

A. INSIGHTS FROM DOCUMENTING BIAS WITH A HYBRID AI SYSTEM

Here, we discuss how implementing the Doc-Bias framework as a Hybrid AI system to a use-case of Fake News detection provides us with insights at varying degrees of granularity depending on the type of implementation. Based on these insights, we set out to answer the research questions guiding this work. Regarding **RQ1, RQ2**), our assessment enables us to determine the importance of User entities. We can also prove its role as a predictive feature in this implementation of PSL and when employing Random Forests. Specifically for the latter, Figure 15 shows the decision trees generated by *InterpretME* that represent Real and Fake News classification for both datasets. From the attributes examined, the one that contributed the most to the classification task was User credibility; in the *PolitiFact* dataset, in particular, higher User credibility makes it more likely that a News item will be classified as *Fake*. At the dataset level, we have quantifiably more data about Users than we do for News and Publishers. Upon pre-processing, we have more information about User and News interactions than we do about Publishers and News. Employing our integrated system, we were able to show how this translates into an over-representation of ground atoms that have more constraints (ground rules) that pertain to Users (in absolute terms), in comparison to those that pertain to Publishers. Thus, overwhelmingly contributing more to the final prediction of the News truth value. We deduce that the huge difference in the number of instances can force the model to deprioritize Publishers and News interactions and exclusively focus on Users to assign credibility scores. We see this manifest again in the accuracy of the model,



(a) Semantic Enrichment across pipeline stages

(b) Semantic Enrichment for "News"

FIGURE 18. Semantic Enrichment results On the left is the percentage of semantic enrichment across Documentation Steps for both datasets. On the right, the distribution of semantic enrichment for the target entity, News, across datasets.

as all false positives are explained by low User credibility over Publisher credibility. Further, Figure 13's revelation that User credibility overwhelmingly contributes to the final classification, underscores the importance of understanding and addressing biases associated with predictive features; in our case this pertains to the User's interactions with News and their undeniable impact on model predictions. In relation to the label unbalance and the skewness towards Fake News, we are able to demonstrate how the exacerbation of labels at input is reflected in how both models have an affinity to favor the "Not Credible/Fake" label in its classifications. When it comes to **RQ2, RQ3**), using our integrated model, we were able to uncover interesting patterns related to the User's emulation of real-world behavior: in-group bias and activity bias. Regarding the latter, News and Users relations with higher frequency at input level prompted their over-representation during optimization. In turn, the model gives higher significance to these combinations of user and news relations and adheres to their characteristics better than in comparison to those with lower frequencies. This highlights the challenges that persist in training models with imbalanced datasets. Ultimately, what started as an observational study reported in Section III-B, through our methods, we are able to empirically demonstrate how bias patterns arise across the pipeline, and their implications. Hence, we advocate for a comprehensive assessment of data-driven pipelines and not neglect "debugging" [68] tasks, especially during the data preprocessing stage [69].

TABLE 11. Characteristics of Outlier - BuzzFeed.

Property	Description
ID	BuzzfeedNews179
Dataset	BuzzFeed
PublisherID	Publisher11
Publisher Name	Conservative Tribune
Publisher Credibility	Not Credible (0)
User Share Count	1060
Label	Fake
Target Value	1
Predicted Probability	0.968
Overrepresented Input Data	True
Overrepresented Train & Predict	True

TABLE 12. Characteristics of Outlier - PolitiFact.

Property	Description
ID	PolitiFact195
Dataset	PolitiFact
PublisherID	Publisher59
Publisher Name	Neon Nettle
Publisher Credibility	Not Credible (0)
User Share Count	2548
Label	Fake
Target Value	1
Predicted Probability	0.922
Overrepresented Input Data	True
Overrepresented Train & Predict	True

B. IMPLICATIONS OF FINE-GRAINED BIAS DOCUMENTATION AND OPEN CHALLENGES

1) IMPLICATIONS

To better explicate the implications of using our integrated AI system to trace bias, we illustrate the degree of semantic enrichment gained across the *Documentation Steps* of the classification pipeline. Figure 18 (a) depicts the amount of instances (in relative terms) attributed to each of the steps. Unsurprisingly, the metadata generated for the learning process is significantly bigger than the other components. This accounts for integrating the whole sub-symbolic system into our tracing pipeline and the degree of affinity we achieved with our model description. Figure 18 (b) shows the distribution of semantic enrichment gain for the target entities, News. We can observe a relatively evened-out distribution across labels and datasets. However, the outliers on the upper fence represent entities with richer profiles due to their characteristics. Particularly, they are News entities with a higher User Share count and, given the calculation of our bias metrics, have been identified as Overrepresented at the input and during the learning and prediction process. We report on the profile generated.[15] for two entities with the maximum value with regard to information gain, one from each dataset, *BuzzfeedNews179*, and *PolitiFact195*, in Table 11 and 12 respectively.

2) CHALLENGES

While our fine-grained documentation approach has allowed us to capture knowledge on biases found in data to further elucidate how these affect the behavior of the model and the generated output, there are still several open challenges from a technical and practical side when it comes to tackling this problem. Some examples of technical challenges are mostly related to how the behavior of sub-symbolic systems is highly constrained by the characteristics of the data they are trained on [70]. Additionally, ML pipelines, in general, are subjected to a particular context upon deployment despite the great strides made in transfer learning techniques. This means that deploying these systems inattentively can lead to them not performing as expected or producing outcomes unsuitable for a particular decision-making scenario [71]. In terms of performing documentation tasks, best practices dissuade decoupling training datasets from the model, as it can make the traceability of the pipeline more challenging and thus hinder efforts to perform thorough evaluation exercises [14]. For this reason, we advocate for the production of compressive end-to-end documentation artifacts. By doing so, it is possible to improve the overall interpretability of automated decision-making systems, and is fundamental in efforts to understand bias in data, and to mitigate instances of algorithmic harm.

From a modeling perspective, our particular approach is also constrained to the characteristics of the ML pipeline under assessment, which increases the complexity of the process when it comes to determining the right level of generalization to be achieved, and not only that, but it requires technical and domain knowledge that can capture pipelines intricacies. In our case, documentation efforts reached a fine-grained level of detail in terms of generated descriptions; however, this also comes at a cost, in terms of compute and data storage.

Further, and as already discussed, bias assessments are highly complex tasks, which implies that our modeling cannot account for all potential biases that might be present in an ML pipeline. However, this is true for all types of bias analysis and documentation frameworks. This highlights the importance to disclose the limitations of any audits performed upon deployment, establish guidelines to enable constant monitoring of the systems, and update existing documentation regularly. From a practical point of view, the remaining open challenge is working towards the adoption of our documentation approach outside a research environment. Lessons from qualitative research highlight the frictions that arise from practitioners integrating documentation frameworks into their existing workflows [26]. Some of the influencing factors include time constraints, business objectives, lack of training (i.e., what metrics to use, how to identify sensitive attributes, how to interpret algorithmic output), and lack of access to domain knowledge [26], [46].

Concerning neuro-symbolic systems in particular, we adhere to common knowledge that despite the AI and Semantic Web communities making enormous advances, integrating these approaches in transparent frameworks remains an open challenge. They still require elucidating on the requirements to develop hybrid neuro-symbolic systems that remain human- and machine-understandable, but that also ensure the transparency required for humans to control decision-making.

VIII. CONCLUSION AND FUTURE WORK

We have presented a documentation pattern that resorts to a hybrid AI system to enable the trace of machine learning pipelines. Our objective is to support ML practitioners and researchers in the interpretation of these pipelines in terms of the biases captured across them. We empirically assessed our approach through two implementations of a hybrid AI system (integrated and principled), and presented a use-case based on Fake News classification. The results derived from our evaluation demonstrate the ability of the Doc-Bias approach to capture bias patterns across the pipeline to varying degrees of granularity, particularly regarding over-representation, activity, and label unbalance. Albeit, all these results, our work still faces open challenges associated with the complexities of modelling and describing intricate pipelines in terms of measured biases; however, we see them all as an opportunity for future work. In particular, we will re-use our approach to trace bias in other predictive tasks (i.e., link prediction), assess the requirements to model more complex problems and liaise with ML practitioners to evaluate the suitability of our framework in real-world scenarios. Lastly, we will continue the development of a controlled vocabulary for bias. We believe these resources can facilitate effective communication among actors involved with or impacted by AI systems, thereby enhancing their understanding of bias patterns at various stages of the AI pipeline.

CREDIT AUTHORSHIP CONTRIBUTION STATEMENT

Mayra Russo: Conceptualization, formal analysis, investigation, resources, software, writing–original draft, writing–review and editing. **Yasharajsinh Chudasama:** Conceptualization, formal analysis, investigation, resources,

[15] All documentation generated by the Doc-Bias KG is available as a FAIR resource and is human- and machine-readable: https://github.com/SDM-TIB/DocBiasKG.

software, writing–original draft, writing–review & editing. **Disha Purohit:** Conceptualization, formal analysis, investigation, resources, software, writing–original draft, writing-review and editing. **Sammy Sawischa:** Resources, software. **Maria-Esther Vidal:** Conceptualization, Formal analysis, funding acquisition, investigation, methodology, project administration, supervision, writing–original draft, writing–review and editing.

APPENDIX A

Appendix A demonstrates two queries used to retrieve interpretable information from the Doc-BiasKG. Particularly, these queries are used to compute information gain given semantic enrichment across two Documentation Steps (see Figure 18): Listing 1 is used for Step 1 - Data Ingestion and Listing 2 is used for Step 3 - Bias Assessment.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-
    syntax-ns#>
PREFIX nobias: <https://nobias-project.eu>

SELECT ?dataset ?metaInformation (COUNT(*) AS ?
    entityInformationCount) (?
    entityInformationCount + ?metaInformation AS ?
    totalInformationCount) {
    {
      SELECT ?dataset (COUNT(*) AS ?
          metaInformation) WHERE {
            ?dataset a <http://www.w3.org/ns/dcat#
                Dataset> .
            ?dataset ?r ?metaInformation .
            FILTER(?r != <https://nobias-project.
                eu/composedOf>) .
            FILTER (!isBlank(?metaInformation)) .
        }
        GROUP BY ?dataset
    }

    {
        ?entity <https://nobias-project.eu/
            associatedTo> ?dataset .
        ?entity <https://nobias-project.eu/
            hasDocumentationType> <https://nobias-
            project.eu/DatasetDocumentation> .
        ?entity ?r ?t .
        FILTER (!isBlank(?t)) .
    }
    UNION
    {
        ?entity <https://nobias-project.eu/
            associatedTo> ?dataset .
        ?entity <https://nobias-project.eu/
            hasDocumentationType> <https://nobias-
            project.eu/DatasetDocumentation> .
        ?h ?r ?entity .
        FILTER (!isBlank(?h)) .
        # Filter bias info
        FILTER NOT EXISTS { ?h <https://nobias-
            project.eu/hasDocumentationType> <
            https://nobias-project.eu/
            BiasDocumentation> } .
    }

}
GROUP BY ?dataset ?metaInformation
```

Listing 1: SPARQL Query to retrieve Information Gain through semantic enrichment at Dataset Ingestion Step over a particular dataset.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-
    syntax-ns#>
PREFIX nobias: <https://nobias-project.eu>

select ?dataset (COUNT(*) AS ?biasInformation)
    where {
     {
        SELECT ?dataset WHERE {?dataset a <http://
            www.w3.org/ns/dcat#Dataset> .}
    }

    {
        ?biasAssessment <https://nobias-project.eu
            /hasDocumentationType> <https://nobias
            -project.eu/BiasDocumentation> .
        ?biasAssessment ?r ?t .
    FILTER (!isBlank(?t))
    FILTER EXISTS { ?biasAssessment ?asseses ?
        entity . ?entity ?associatedTo ?dataset .
        } .


    }
      UNION
    {
        ?biasAssessment <https://nobias-project.eu
            /hasDocumentationType> <https://nobias
            -project.eu/BiasDocumentation> .
        ?h ?r ?biasAssessment .
        FILTER (!isBlank(?h))
        FILTER EXISTS { ?biasAssessment ?asseses ?
            entity . ?entity ?associatedTo ?
            dataset . } .
    }

}
GROUP BY ?dataset
```

Listing 2: SPARQL Query to retrieve Information Gain through semantic enrichment following Bias Assessment Step performed over a particular dataset.

APPENDIX B

Appendix B to represent different statistical and comparative queries employed over the InterpretME KG. These queries highlight the importance of traceability and documenting the ML model characteristics. Listing 3 demonstrates SPARQL query utilized over the InterpretME KG to retrieve the trained predictive model characteristics for Table 6, 7, 8, 9 in

```
PREFIX intr: <http://interpretme.org/vocab/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-
    syntax-ns#>

SELECT DISTINCT ?RunID ?precision ?recall
                ?f1score
WHERE {
?s <http://interpretme.org/vocab/hasRun> ?RunID.
?s <http://interpretme.org/vocab/hasPrecision> ?
    precision.
?s <http://interpretme.org/vocab/hasRecall> ?
    recall.
?s <http://interpretme.org/vocab/hasF1score> ?
    f1score.
?s <http://interpretme.org/vocab/hasClasses> ?
    classes.
}
```

Listing 3: SPARQL Query to retrieve the evaluation results of random forest model.

terms of precision, recall, and f1-score. Listing 4 illustrates the comparative analysis of key features such as publisher credibility influence on the outcomes of Fake news in terms of average, maximum, and minimum impact in target class Fake. Lastly, the statistical analysis depicting the counts of news and their prediction probability in class Fake is presented in Listing 5.

```
PREFIX intr: <http://interpretme.org/vocab/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-
    syntax-ns#>

#Queries for probabilityAVG, MAX, MIN

SELECT DISTINCT ?news  ?probability (AVG(?value)
    as ?num1)  (MAX(?value) as ?num2)  (MIN(?value
    ) as ?num3)
WHERE {
        ?news a <http://interpretme.org/vocab/
            TargetEntity> .
        ?news <http://interpretme.org/vocab/
            hasInterpretedFeature> ?
            interpretedFeature .
        ?interpretedFeature  <http://interpretme.
            org/vocab/hasFeatureWeight> ?
            featureWeight .
         ?featureWeight <http://interpretme.org/
             vocab/hasFeature> ?feature .
        ?featureWeight <http://interpretme.org/
            vocab/hasWeight> ?value .
        ?news <http://interpretme.org/vocab/
            hasEntityClassProbability> ?classProb
            .
        ?classProb <http://interpretme.org/vocab/
            hasPredictionProbability> ?probability
            .
        ?classProb <http://interpretme.org/vocab/
            hasClass> ?targetClass .
       FILTER(?targetClass = <http://interpretme.
           org/entity/Fake>)
       FILTER(regex(?feature,"Publisher"))
} GROUP BY ?news  ?probability
Order by Desc(?probability) ?news
}
```

Listing 4: A statistical SPARQL Query to retrieve the impact of publisher credibility over the Fake news classification.

```
PREFIX intr: <http://interpretme.org/vocab/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-
    syntax-ns#>

SELECT DISTINCT ?probability (count(distinct ?news
    ) as ?num)
WHERE {
        ?s <http://interpretme.org/vocab/hasRun> ?
            run .
        ?s <http://interpretme.org/vocab/hasEntity
            > ?news.
        ?s <http://interpretme.org/vocab/hasClass
            ><http://interpretme.org/entity/Fake>.
       ?s <http://interpretme.org/vocab/
           hasPredictionProbability> ?probability.
} GROUP BY ?probability
}
```

Listing 5: SPARQL Query to retrieve the count of news and their prediction probability in class Fake.

ACKNOWLEDGMENT

This work reflects only the authors' views and the European Research Executive Agency (REA) is not responsible for any use that may be made of the information it contains. The author Mayra Russo thanks Guillermo Climent-Gargallo and Yukti Sharma for their support in completing this research.

(Mayra Russo, Yasharajsinh Chudasama, and Disha Purohit are co-first authors.)

REFERENCES

[1] B. Friedman and H. Nissenbaum, "Bias in computer systems," *ACM Trans. Inf. Syst.*, vol. 14, no. 3, pp. 330–347, Jul. 1996.

[2] H. Suresh and J. Guttag, "A framework for understanding sources of harm throughout the machine learning life cycle," in *Equity and Access in Algorithms, Mechanisms, and Optimization*. New York, NY, USA: Association for Computing Machinery (ACM), 1145.

[3] C. Ross and B. Herman. (2023). *Denied By AI: How Medicare Advantage Plans Use Algorithms to Cut Off Care for Seniors in Need*. [Online]. Available: https://www.statnews.com/2023/03/13/medicare-advantage-plans-denial-artificial-intelligence/

[4] W. D. Heaven. (2021). *Bias Isn't the Only Problem With Credit Scores-and No, AI Can't Help*. [Online]. Available: https://www.technologyreview.com/2021/06/17/1026519/racial-bias-noisy-data-credit-scores-mortgage-loans-fairness-machine-learning/

[5] S. Barocas, M. Hardt, and A. Narayanan. (2019). *Fairness in Machine Learning*. [Online]. Available: http://www.fairmlbook.org

[6] S. Barocas and A. D. Selbst, "Big data's disparate impact," *SSRN Electron. J.*, p. 671, 2016.

[7] J. Stoyanovich, S. Abiteboul, B. Howe, H. V. Jagadish, and S. Schelter, "Responsible data management," *Commun. ACM*, vol. 65, no. 6, pp. 64–74, May 2022.

[8] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Proc. Conf. Fairness, Accountability Transparency*, 2018, pp. 1–12.

[9] S. U. Noble, *Algorithms of Oppression: How Search Engines Reinforce Racism*. New York, NY, USA: New York Univ. Press, 2018.

[10] R. Baeza-Yates, "Bias on the web and beyond: An accessibility point of view," in *Proc. 17th Int. Web Conf.*, Apr. 2020, pp. 1–10.

[11] R. Baeza-Yates, "Biases on social media data: (Keynote extended abstract)," in *Companion Proc. Web Conf.*, Apr. 2020, pp. 1–27.

[12] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. D. Iii, and K. Crawford, "Datasheets for datasets," *Commun. ACM*, vol. 64, no. 12, pp. 86–92, Dec. 2021.

[13] I. D. Raji and J. Yang, "About ml: Annotation and benchmarking on understanding and transparency of machine learning lifecycles," 2020, *arXiv:1912.06166*.

[14] I. D. Raji, A. Smart, R. N. White, M. Mitchell, T. Gebru, B. Hutchinson, J. Smith-Loud, D. Theron, and P. Barnes, "Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing," in *Proc. Conf. Fairness, Accountability, Transparency*, New York, NY, USA, Jan. 2020, pp. 33–44.

[15] K. Peng, A. Mathur, and A. Narayanan, "Mitigating dataset harms requires stewardship: Lessons from 1000 papers," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, 2021, pp. 1–18.

[16] N. Noy and C. Goble, "Are we cobblers without shoes? Making computer science data FAIR," *Commun. ACM*, vol. 66, no. 1, pp. 36–38, Jan. 2023.

[17] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J. W. Boiten, L. B. da Silva Santos, P. E. Bourne, and J. Bouwman, "The FAIR guiding principles for scientific data management and stewardship," *Sci. Data*, vol. 3, no. 1, pp. 1–9, Mar. 2016.

[18] E. M. Bender and B. Friedman, "Data statements for natural language processing: Toward mitigating system bias and enabling better science," *Trans. Assoc. Comput. Linguistics*, vol. 6, pp. 587–604, Dec. 2018.

[19] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, and T. Gebru, "Model cards for model reporting," in *Proc. Conf. Fairness, Accountability, Transparency*, Jan. 2019, pp. 220–229.

[20] M. Miceli, T. Yang, L. Naudts, M. Schuessler, D. Serbanescu, and A. Hanna, "Documenting computer vision datasets: An invitation to reflexive data practices," in *Proc. ACM Conf. Fairness, Accountability, Transparency*, Mar. 2021, pp. 161–172.

[21] A. Wang, A. Liu, R. Zhang, A. Kleiman, L. Kim, D. Zhao, I. Shirai, A. Narayanan, and O. Russakovsky, ''REVISE: A tool for measuring and mitigating bias in visual datasets,'' *Int. J. Comput. Vis.*, vol. 130, no. 7, pp. 1790–1810, Jul. 2022.

[22] C. Sun, A. Asudeh, H. V. Jagadish, B. Howe, and J. Stoyanovich, ''MithraLabel: Flexible dataset nutritional labels for responsible data science,'' in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 2893–2896.

[23] Google People + AI Res. (2022). *Know Your Data*. [Online]. Available: https://knowyourdata.withgoogle.com/docs/

[24] Hugging Face Res. (2022). *Data Measurements Too*. [Online]. Available: https://huggingface.co/spaces/huggingface/data-measurements-tool

[25] Facebook AI. (2021). *Fairness Flow*. [Online]. Available: https://ai.facebook.com/blog/how-were-using-fairness-flow-to-help-build-ai-that-works-better-for-everyone/

[26] A. K. Heger, L. B. Marquis, M. Vorvoreanu, H. Wallach, and J. Wortman Vaughan, ''Understanding machine learning practitioners' data documentation perceptions, needs, challenges, and desiderata,'' *Proc. ACM Hum.-Comput. Interact.*, vol. 6, no. CSCW2, pp. 1–29, Nov. 2022.

[27] M. van Bekkum, M. de Boer, F. van Harmelen, A. Meyer-Vitali, and A. T. Teije, ''Modular design patterns for hybrid learning and reasoning systems,'' *Int. J. Speech Technol.*, vol. 51, no. 9, pp. 6528–6546, Sep. 2021.

[28] M. Bröcheler, L. Mihalkova, and L. Getoor, ''Probabilistic similarity logic,'' in *Proc. 26th Conf. Uncertainty Artif. Intell.*, 2010, pp. 73–82.

[29] R. Chowdhury, S. Srinivasan, and L. Getoor, ''Joint estimation of user and publisher credibility for fake news detection,'' in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Virtual Event, Ireland, Oct. 2020, pp. 1993–1996.

[30] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, ''FakeNewsNet: A data repository with news content, social context and spatialtemporal information for studying fake news on social media,'' 2018, *arXiv:1809.01286*.

[31] T. D. Bie, L. D. Raedt, J. Hernández-Orallo, H. H. Hoos, P. Smyth, and C. K. I. Williams, ''Automating data science,'' *Commun. ACM*, vol. 65, no. 3, pp. 76–87, 2022.

[32] M. T. Ribeiro, S. Singh, and C. Guestrin, '''Why should I trust you?': Explaining the predictions of any classifier,'' in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1–20.

[33] S. M. Lundberg and S.-I. Lee, ''A unified approach to interpreting model predictions,'' in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1–23.

[34] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, ''A survey on bias and fairness in machine learning,'' 2019, *arXiv:1908.09635*.

[35] P. Reyero-Lobo, E. Daga, H. Alani, and M. Fernández, ''Semantic web technologies and bias in artifcial intelligence: A systematic literature review,'' *Semantic Web J.*, vol. 1, no. 2, pp. 1–28, 2022.

[36] P. Ristoski and H. Paulheim, ''Semantic web in data mining and knowledge discovery: A comprehensive survey,'' *J. Web Semantics*, vol. 36, pp. 1–22, Jan. 2016.

[37] A. Breit, L. Waltersdorfer, F. J. Ekaputra, M. Sabou, A. Ekelhart, A. Iana, H. Paulheim, J. Portisch, A. Revenko, F. van Harmelen, and A. T. Teije, ''Combining machine learning and semantic web: A systematic mapping study,'' in *Proc. Workshop Neural-Symbolic Learn. Reasoning*, 2023, pp. 1–11.

[38] M. Russo, S. F. Sawischa, and M.-E. Vidal, ''Tracing the impact of bias in link prediction,'' in *Proc. 39th ACM/SIGAPP Symp. Appl. Comput.*, New York, NY, USA, Apr. 2024, pp. 1626–1633.

[39] Y. Chudasama, D. Purohit, P. D. Rohde, J. Gercke, and M.-E. Vidal, ''InterpretME: A tool for interpretations of machine learning models over knowledge graphs,'' *Semantic Web*, vol. 1, pp. 1–21, Jan. 2024.

[40] C. Reddy, D. Sharma, S. Mehri, A. R. Soriano, S. Shabanian, and S. Honari, ''Benchmarking bias mitigation algorithms in representation learning through fairness metrics,'' in *Proc. Neural Inf. Process. Syst. Track Datasets Benchmarks*, 2021, pp. 1–28.

[41] S. Verma and J. Rubin, ''Fairness definitions explained,'' in *Proc. IEEE/ACM Int. Workshop Softw. Fairness*, New York, NY, USA, May 2018, pp. 1–7.

[42] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, ''A survey on bias and fairness in machine learning,'' *ACM Comput. Surveys*, vol. 54, no. 6, pp. 1–35, Jul. 2022.

[43] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. S. Zemel, ''Fairness through awareness,'' in *Innovations in Theoretical Computer Science*. Cambridge, CA, USA: ACM, 2012, pp. 214–226.

[44] S. L. Blodgett, S. Barocas, H. Daumé, and H. Wallach, ''Language (Technology) is power: A critical survey of 'Bia' in NLP,'' in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 5454–5476.

[45] A. Olteanu, C. Castillo, F. Diaz, and E. Kıcıman, ''Social data: Biases, methodological pitfalls, and ethical boundaries,'' *Frontiers Big Data*, vol. 2, pp. 1–28, Jul. 2019.

[46] W. H. Deng, M. Nagireddy, M. S. A. Lee, J. Singh, Z. S. Wu, K. Holstein, and H. Zhu, ''Exploring how machine learning practitioners (Try To) use fairness toolkits,'' in *Proc. ACM Conf. Fairness, Accountability, Transparency*, New York, NY, USA, Jun. 2022, pp. 473–484.

[47] E. Ilkou and M. Koutraki, ''Symbolic vs sub-symbolic AI methods: Friends or enemies?'' in *Proc. Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1–10.

[48] A. D. Garcez and L. C. Lamb, ''Neurosymbolic AI: The 3rd wave,'' *Artif. Intell. Rev.*, vol. 56, no. 11, pp. 12387–12406, Nov. 2023.

[49] A. Breit, L. Waltersdorfer, F. J. Ekaputra, M. Sabou, A. Ekelhart, A. Iana, H. Paulheim, J. Portisch, A. Revenko, A. T. Teije, and F. Van Harmelen, ''Combining machine learning and semantic Web: A systematic mapping study,'' *ACM Comput. Surveys*, vol. 55, no. 14s, pp. 1–41, Dec. 2023.

[50] A. Rivas, D. Collarana, M. Torrente, and M.-E. Vidal, ''A neuro-symbolic system over knowledge graphs for link prediction,'' in *Accepted at the Special Issue on Neuro-Symbolic Artificial Intelligence and the Semantic Web*. New York, NY, USA: Association for Computing Machinery (ACM), 2023.

[51] O. Loyola-González, ''Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view,'' *IEEE Access*, vol. 7, pp. 154096–154113, 2019.

[52] J. Zhang, B. Chen, L. Zhang, X. Ke, and H. Ding, ''Neural, symbolic and neural-symbolic reasoning on knowledge graphs,'' *AI Open*, vol. 2, pp. 14–35, Jul. 2021.

[53] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. De Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann, ''Knowledge graphs,'' *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1–30, Jul. 2021.

[54] J. Sreemathy, K. Naveen Durai, E. Lakshmi Priya, R. Deebika, K. Suganthi, and P. Aisshwarya, ''Data integration and ETL: A theoretical perspective,'' in *Proc. 7th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, vol. 1, Mar. 2021, pp. 1655–1660.

[55] E. F. Kendall and D. L. McGuinness, *Ontology Engineering*. San Rafael, CA, USA: Morgan & Claypool, 2019.

[56] Y. Chudasama, D. Purohit, P. D. Rohde, and M.-E. Vidal, ''Enhancing interpretability of machine learning models over knowledge graphs,'' in *Proc. 19th Int. Conf. Semantic Syst.*, 2023, pp. 21–25.

[57] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, ''Fake news detection on social media: A data mining perspective,'' *ACM SIGKDD Explor. Newslett.*, vol. 19, no. 1, pp. 22–36, 2017.

[58] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, ''Certifying and removing disparate impact,'' in *Proc. ACM SIGKDD*, 2015, pp. 259–268.

[59] E. Iglesias, S. Jozashoori, and M.-E. Vidal, ''Scaling up knowledge graph creation to large and heterogeneous data sources,'' *J. Web Semantics*, vol. 75, Jan. 2023, Art. no. 100755.

[60] A. Dimou, T. D. Nies, R. Verborgh, E. Mannens, and R. Van de Walle, ''Automated metadata generation for linked data generation and publishing workflows,'' in *Proc. Workshop Linked Data Web*, vol. 1593, 2016, pp. 1–36.

[61] T. Lebo, S. Sahoo, D. McGuinness, K. Belhajjame, J. Cheney, D. Corsar, D. Garijo, S. Soiland-Reyes, S. Zednik, and J. Zhao, *PROV-O: The PROV Ontology*. Cambridge, MA, USA: W3C Recommendation, Apr. 2013.

[62] R. Albertoni, D. Browning, S. Cox, A. N. Gonzalez-Beltran, A. Perego, and P. Winstanley, ''The W3C data catalog vocabulary, version 2: Rationale, design principles, and uptake,'' *Data Intell.*, vol. 6, no. 2, pp. 457–487, May 2024.

[63] G. C. Publio, D. Esteves, A. Lawrynowicz, P. Panov, L. Soldatova, T. Soru, J. Vanschoren, and H. Zafar, ''Ml-schema: Exposing the semantics of machine learning with schemas and ontologies,'' 2018, *arXiv:1807.05351*.

[64] D. Brickley and L. Miller, *The Friend of a Friend(FOAF) Project*, FOAF Vocabulary Specification, 2004. [Online]. Available: http://www.foaf-project.org/

[65] R. Albertoni and A. Isaac, ''Introducing the data quality vocabulary (DQV),'' *Semantic Web*, vol. 12, no. 1, pp. 81–97, Nov. 2020.

[66] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov, ''Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles,'' *Proc. Nat. Acad. Sci. USA*, vol. 102, no. 43, pp. 15545–15550, Oct. 2005, doi: 10.1073/pnas.0506580102.

[67] L. da F. Costa, ''Further generalizations of the Jaccard index,'' 2021, *arXiv:2110.09619*.

[68] S. Grafberger, P. Groth, J. Stoyanovich, and S. Schelter, "Data distribution debugging in machine learning pipelines," *VLDB J.*, vol. 31, no. 5, pp. 1103–1126, Jan. 2022.

[69] L. R. Lucchesi, P. M. Kuhnert, J. L. Davis, and L. Xie, "Smallset timelines: A visual representation of data preprocessing decisions," in *Proc. ACM Conf. Fairness, Accountability, Transparency*, New York, NY, USA, Jun. 2022, pp. 1136–1153.

[70] I. D. Raji, I. E. Kumar, A. Horowitz, and A. Selbst, "The fallacy of AI functionality," in *Proc. ACM Conf. Fairness, Accountability, Transparency*, Jun. 2022, pp. 959–972.

[71] J. M. Alvarez, A. B. Colmenarejo, A. Elobaid, S. Fabbrizzi, M. Fahimi, A. Ferrara, S. Ghodsi, C. Mougan, I. Papageorgiou, P. Reyero, M. Russo, K. M. Scott, L. State, X. Zhao, and S. Ruggieri, "Policy advice and best practices on bias and fairness in AI," *Ethics Inf. Technol.*, vol. 26, no. 2, pp. 1–32, Jun. 2024.

MAYRA RUSSO received the bachelor's degree in accounting and finance and the master's degree in data science from the University of Valencia, Spain. She is currently pursuing the Ph.D. degree in computer science with Leibniz University Hannover under the supervision of Dr. Prof. Maria-Esther Vidal. She is an Early Stage Researcher with the Marie Skłodowska-Curie ITN NoBIAS and a Research Assistant with the L3S Research Center. Her research interests include algorithmic bias and harm, documentation for machine learning datasets and pipelines, employing the use of ontologies, and other semantic web technologies. Outside the scope of her Ph.D. research, her interests include investigating the social implications associated with datafication.

YASHARAJSINH CHUDASAMA received the B.E. degree in mechatronics in Gujarat, India, and the M.Sc. degree in mechatronics from Leibniz University Hannover (LUH), where he is currently pursuing the Ph.D. degree under the supervision of Prof. Dr. Maria-Esther Vidal. He is a Research Associate with the Scientific Data Management (SDM) Group, TIB—Leibniz Information Centre for Science and Technology. He is working on a TrustKG project funded by the Leibniz Association. His research interests include data management, sub-symbolic learning, explainability, and neuro-symbolic AI over knowledge graphs.

DISHA PUROHIT received the B.E. degree in computer engineering in Mumbai, Maharashtra, India, and the M.Sc. degree in internet technologies and information systems (ITIS) from Leibniz University Hannover (LUH), where she is currently pursuing the Ph.D. degree under the supervision of Prof. Dr. Maria-Esther Vidal. She is a Research Associate with the Scientific Data Management (SDM) Group, TIB—Leibniz Information Centre for Science and Technology. She is working on a TrustKG (a project funded by the Leibniz Association) Project and the P4-LUCAT Project funded by ERAMed. Her research interests include data management, inductive learning, specifically symbolic learning, and discovering causal patterns from knowledge graphs.

SAMMY SAWISCHA received the degree in computer science from Leibniz University Hannover. During his studies, he was with Microsoft Research Ireland and the Leibniz Information Centre for Science and Technology (TIB). His research interests include machine learning (ML), network analysis, and how to assess bias and fairness in ML systems.

MARIA-ESTHER VIDAL is currently a Full Professor with Leibniz University Hannover and leads the Scientific Data Management (SDM) Group, TIB—Leibniz Information Centre for Science and Technology. She is also a member of the L3S Research Centre and a Full Professor (retired) with the Universidad Simón Bolívar (USB), Venezuela. Under her direction, her team has developed technologies of predominant relevance in the whole process of knowledge graph creation from heterogeneous data and query processing. She has advised more than 28 Ph.D. students and more than 120 master's and bachelor's students in computer science. She is the co-author of more than 240 peer-reviewed articles in *Semantic Web*, *Databases*, and *Artificial Intelligence*. Her research interests include data management, semantic data integration, and machine learning over knowledge graphs. She has been a Doctoral Committee Member in France, Italy, Sweden, Spain, The Netherlands, Germany, Ireland, Argentina, Uruguay, and Venezuela. She has been awarded the Science Award on Responsible Research by Stifterverband with the recommendation of the Leibniz Association, Germany, and the Program "Leibniz Best Minds: Program for Women Professors" supported by the Leibniz Association. She is also actively shaping her research communities. She has been an Editorial Board Member of renowned journals, such as JWS and JDIQ, and the General Chair, the Co-Chair, and a Senior Reviewer of major scientific events, such as ESWC, WWW, ISWC, and AAAI. She serves as an expert in several advisory boards, summer schools, and Ph.D. consortiums.