CENTRO ALTI STUDI DIFESA
SCUOLA SUPERIORE UNIVERSITARIA

UNIVERSITÀ DEGLI STUDI DI SALERNO

Dottorato di Ricerca in

Scienze dell'Innovazione per la Difesa e la Sicurezza

XXXVII CICLO

TITOLO DELLA TESI

EXPLAINABLE AI FOR DEFENCE SYSTEMS

SETTORE SCIENTIFICO-DISCIPLINARE: **INF/01**

PRESENTATA DA: **Claudio STANZIONE**

COORDINATRICE DEL DOTTORATO: **Prof.ssa Paola ADINOLFI**

Tutor:
Prof. Nicola CAPUANO

ANNI ACCADEMICI: 2021/2024

HIGHER DEFENCE STUDIES/
UNIVERSITY OF SALERNO

Defence Analysis & Research Institute Center

XXXVII Cycle of the PhD Program in "Innovation Sciences for Defense and Security"

TITLE

EXPLAINABLE ARTIFICIAL INTELLIGENCE METHODS FOR DEFENCE SYSTEMS

Scientific-Disciplinary Sector: INF/01

PhD Student: Claudio STANZIONE

Tutor: Prof. Nicola CAPUANO

Doctoral Coordinator: Prof. Paola ADINOLFI

Academic Years: 2021/2024

Abstract

Artificial intelligence (AI) has increasingly become an integral part of daily life, evolving into a necessity for a large portion of the population. The advent of large language models, such as GPT, has deepened our dependence on these technologies, making them seem indispensable. However, this growing reliance on AI has introduced challenges, particularly when these systems function as "black boxes"—producing outputs without transparent reasoning. Therefore, ongoing research is essential to ensure that AI-generated content aligns with human values and adheres to security standards.

This doctoral thesis explores the application of Explainable Artificial Intelligence (xAI) within the realm of cyber and cognitive security. The latter has been investigated focusing on the threats related to information disorder phenomenon. The integration of xAI in these threats is double-edged sword; while it offers new opportunities for understanding model decisions, it simultaneously exposes models to potential adversarial attacks by revealing their vulnerabilities. One significant challenge identified through a preliminary literature review is applying xAI to mitigate these vulnerabilities. Addressing this, the thesis examines three main challenges leveraging xAI: enhancing the reliability of machine learning models, counter Information Disorder and last but not least, strengthen models against adversarial attacks.

Firstly, the reliability of machine learning models has become a critical concern, especially with the impending implementation of the AI ACT by the European Union, which mandates higher transparency and reliability. This thesis presents two experiments utilizing Formal Concept Analysis to improve model reliability.

Secondly, the thesis focuses into the largely unexplored area of using xAI to combat Information Disorder, a phenomenon within Cognitive Security that encompasses the spread of misleading, false, or propagandistic content. Three studies are presented, each leveraging xAI to enhance defenses against Information Disorder.

Finally, the work addresses the use of xAI to detect and mitigate the vulnerabilities of machine learning models. Given the rise of adversarial attacks, where small, imperceptible changes to input data can deceive models—this aspect of the research is crucial. Three papers are discussed that utilize xAI to identify potential vulnerabilities in models and propose solutions to fortify them.

Table of contents

List of figures

List of tables

List of original publications

This thesis is based on a series of peer-reviewed original research papers published in different international scientific forums. The following papers are referenced in the text using their Roman Numerals (I-XIII).

International Journals

I Capuano, N., Fenza, G., Loia, V., & Stanzione, C. (2022). Explainable artificial intelligence in cybersecurity: A survey. IEEE Access, 10, 93575-93600. *doi:https://doi.org/10.1109/ACCESS.2022.3204171*

II De Maio, C., Fenza, G., Gallo, M., Loia, V., & Stanzione, C. (2023). Toward reliable machine learning with Congruity: a quality measure based on formal concept analysis. Neural Computing and Applications, 35(2), 1899-1913. *doi:https://doi.org/10.1007/s00521-022-07853-7*

III Fenza, G., Gallo, M., Loia, V., Petrone, A., & Stanzione, C. (2023). Concept-drift detection index based on fuzzy formal concept analysis for fake news classifiers. Technological Forecasting and Social Change, 194, 122640. *doi:https://doi.org/10.1016/j.techfore.2023.122640*

IV Bangerter, M. L., Fenza, G., Furno, D., Gallo, M., Loia, V., Stanzione, C., & You, I. (2024). A Hybrid Framework Integrating LLM and ANFIS for Explainable Fact-Checking. IEEE Transactions on Fuzzy Systems. *doi:https://doi.org/10.1109/TFUZZ.2024.3431710*

V Fenza, G., Loia, V., Stanzione, C., & Di Gisi, M. (2024). Robustness of models addressing Information Disorder: A comprehensive review and benchmarking study. Neurocomputing, 127951. *doi:https://doi.org/10.1016/j.neucom.2024.127951*

International Conferences

VI Bangerter, M., Fenza, G., Gallo, M., Loia, V., Volpe, A., De Maio, C., & Stanzione, C. (2023, July). Unisa at SemEval-2023 task 3: a shap-based method for propaganda detection. In Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023) (pp. 885-891). *doi:https://aclanthology.org/2023.semeval-1.122*

VII Capuano, N., Fenza, G., Gallo, M., Loia, V., & Stanzione, C. (2023, December). Unfolding the Misinformation Spread: An In-Depth Analysis Through Explainable Link Predictions and Data Mining. In International Conference on Intelligent Systems Design and Applications (pp. 137-146). Cham: Springer Nature Switzerland. *doi:https://doi.org/10.1007/978-3-031-64779-6_13*

VIII Cavaliere, D., Gallo, M., & Stanzione, C. (2023, July). Propaganda Detection Robustness Through Adversarial Attacks Driven by eXplainable AI. In World Conference on Explainable Artificial Intelligence (pp. 405-419). Cham: Springer Nature Switzerland. *doi:https://doi.org/10.1007/978-3-031-44067-0_21*

IX Fenza, G., Gallo, M., Loia, V., Nicolosi, A., & Stanzione, C. (2024). Detecting Persuasive Prompts: A Framework for Secure LLMs (***Accepted - In press***)

X Fenza, G., Gallo, M., Loia, V., & Stanzione, C. (2024, May). Evaluating Web Domain Credibility: A Multifactorial Score for Analyzing Online Reliability. In 2024 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS) (pp. 1-8). IEEE. *doi:https://doi.org/10.1109/EAIS58494.2024.10570009*

XI Fenza, G., Froncillo, S., & Stanzione, C. (2024). Enhancing Fraud Detection through Cascading Machine Learning Models and Clustering Techniques. *ref:https://ceur-ws.org/Vol-3731/paper19.pdf*

XII Fenza, G., Loia, V., Mainardi, P. M., & Stanzione, C. (2023). OSINT Knowledge Graph for Fact-Checking: Google Map Hacks Debunking (No. 9975). EasyChair. *ref:https://ceur-ws.org/Vol-3488/paper18.pdf*

XIII Bangerter, M. L., Fenza, G., Gallo, M., Genovese, A., Nota, F. D., Stanzione, C., & Zanfardino, G. (2021, June). Unmask inflated product reviews through Machine Learning. In 2021 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA) (pp. 1-6). IEEE. *doi:https://doi.org/10.1109/CIVEMSA52099.2021.9493576*

The publications included in the thesis discussion are I through IX.

Chapter 1

Introduction

In this chapter, the foundation for the entire thesis is given by exploring the broad context in which this research is situated. The discussion begins by introducing the growing importance of Artificial Intelligence (AI) in various domains, emphasizing its pervasive impact on modern society. As AI technologies continue to evolve, so do the challenges associated with their adoption, particularly in terms of transparency and security. This chapter introduces the concept of xAI as a promising approach to addressing these challenges, specifically within the field of Cyber Security. The chapter also outlines the key research questions that drive this thesis and provide an overview of the structure of the subsequent chapters.

1.1 Research Area

AI is a field within computer science dedicated to developing machines capable of carrying out tasks that typically need human intelligence, including perception, reasoning, and learning. These tasks include learning from experience, understanding natural language, recognizing patterns, making decisions, and solving complex problems. Driven by the convergence of vast computational power, sophisticated algorithms, and enormous datasets, AI is not just an abstract concept but a tangible force reshaping industries, economies, and everyday life. To quantify this phenomenon numerically, AI market is anticipated to experience substantial growth over the next decade, according to various sources. Data from Statista indicates that the market size is expected to increase from 241.8 billion U.S. dollars in 2023 to nearly 740 billion U.S. dollars by 2030, reflecting a compound annual growth rate of 17.3%[1]. Similarly, Next Move Strategy Consulting predicts that the market, valued at about 208 billion U.S. dollars in 2023, will expand ninefold to approximately 1.85 trillion U.S. dollars by 2030[2]. The AI market spans numerous industries, such as healthcare, education, finance, media and marketing.

The literature in this field has also become increasingly important, as shown by the Figure 1.1. Referring to Figure 1.1, it is possible to see that in 2000, published papers that

[1]https://www.statista.com/outlook/tmo/artificial-intelligence/worldwide
[2]https://www.nextmsc.com/report/artificial-intelligence-market

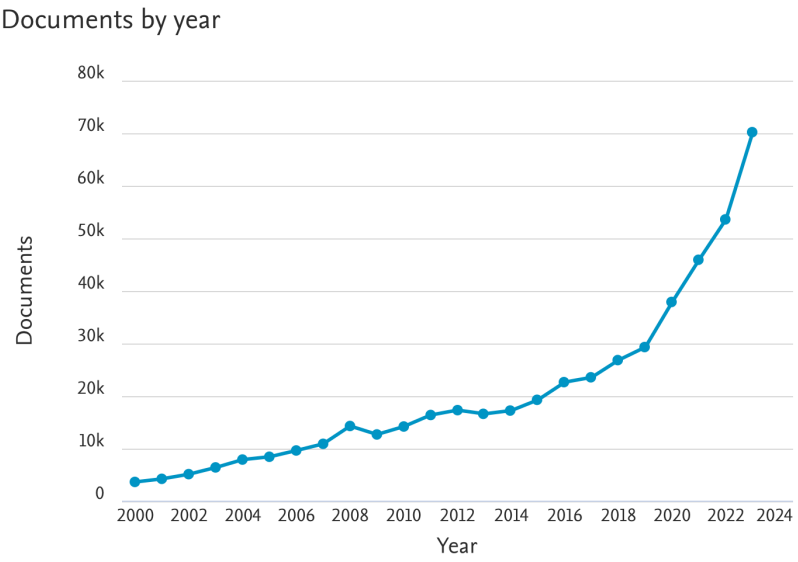

Fig. 1.1 Evolution of the number of total publications whose title, abstract and/or keywords refer to the field of AI from 2000 to 2023. Data retrieved from Scopus using as search key [TITLE-ABS-KEY (Artificial AND Intelligence)].

had the words Artificial Intelligence in the title, abstract or keywords were just 3678; in 2010, they had reached 14219 and then exploded in 2020 when they reached 37911 to close at the end of 2023 with 70211 published in a single year that touched on this topic.

The ongoing evolution of AI technologies presents not only vast opportunities but also significant challenges. Issues such as model interpretability, models vulnerabilites, data bias, and the necessity for substantial computational resources are persistent hurdles that need to be addressed.

This manuscript focuses on the xAI.

xAI is an emerging field of artificial intelligence that focuses on making the decision-making processes of AI systems transparent and understandable to humans. As AI technologies increasingly integrate into various aspects of society, explainability becomes crucial to ensure trust, accountability and fairness. Explainable AI aims to explain how AI models reach their conclusions, enabling users to understand, trust, and effectively manage these systems.

The importance of xAI is manifold. First, it increases trust and adoption. For AI to be widely accepted and used, especially in critical areas such as healthcare, finance and autonomous driving, stakeholders must understand and trust its decisions. xAI helps build trust among users, regulators, and the general public by making AI systems more transparent.

Second, xAI is essential for accountability and compliance. In many industries, organizations must comply with strict regulatory standards that require them to explain their decision-making processes. For example, the European Union's General Data Protection Regulation (GDPR) requires that individuals have the right to obtain an explanation of decisions made by automated systems. Explainable AI provides the tools necessary to meet these regulatory requirements.

Third, xAI plays a pivotal role in promoting fairness and equity by identifying and mitigating biases in AI models. AI systems are trained on data that may contain historical biases, which can lead to unfair or discriminatory results. Explainability enables developers to identify and understand these biases, reassuring the audience about the ethical implications of AI systems and facilitating the development of more fair and equitable AI systems.

1.2 Application Domain

The digital age has ushered in unprecedented connectivity and convenience, but it has also brought about significant vulnerabilities and threats, making cybersecurity an essential field of study and practice. Cyber Security encompasses various techniques and strategies to protect computers, networks, and data from unauthorized access, attacks, and damage. As cyber threats evolve in complexity and scale, the need for robust security measures becomes increasingly critical for governments, businesses, and individuals.

Within this dynamic landscape, Cognitive Security represents a cutting-edge approach that leverages AI and ML to enhance the detection, prevention, and response to cyber threats.

One of the most pressing challenges in Cognitive Security today is the counteraction of Information Disorder, which encompasses the spread of false, misleading, or manipulated information across digital platforms. Information disorder can take many forms, including misinformation (false information spread without harmful intent), disinformation (false information spread with malicious intent), and malinformation (true information shared to cause harm). These phenomena can undermine public trust, influence political processes, and incite social unrest.

Explainable AI is a powerful ally in the battle against information disorder. It can help identify the origins and spread of misinformation by clarifying the patterns and logic used by AI to flag suspicious content. This transparency not only helps to effectively counter false narratives but also plays a crucial role in maintaining the integrity of information, making the audience feel more secure and confident in the face of misinformation. Application of xAI in Cyber Security is a double-edged sword, make more transparent the models' decision to the analysts but also open the door to the attackers that are able to understand how models work, bringing out the vulnerabilities.

Models' vulnerabilities are a significant concern in AI, particularly regarding their robustness against adversarial attacks. Adversarial attacks involve manipulating input data in subtle ways that cause AI models to make incorrect predictions or classifications. These attacks can undermine the reliability and security of AI systems, posing serious risks, especially in sensitive applications like information integrity and cybersecurity. In the realm of Information Disorder, where AI models are crucially deployed to detect and combat the spread of false or misleading information, the need for robustness against adversarial attacks is not just important, it's paramount. Adversarial actors can exploit vulnerabilities in these models to bypass detection systems, propagate disinformation,

and disrupt the trust in digital information ecosystems. Consider a real-world scenario where an AI model is deployed to flag misleading news articles on social media platforms. Adversarial actors could subtly modify the content of these articles to evade detection by the AI model. By strategically altering certain keywords or introducing subtle grammatical changes, they could trick the model into classifying the articles as genuine news rather than misinformation. This could lead to the unchecked spread of false information, eroding public trust in the reliability of digital information sources. Moreover, adversarial attacks in Information Disorder are not limited to textual content. Adversarial actors can also manipulate images, videos, and audio recordings to create convincing but false narratives. Deepfake technology, for example, allows adversaries to generate highly realistic videos that depict individuals saying or doing things they never actually did. These manipulated media can spread false information, manipulate public opinion, and sow discord in society.

1.3 Objectives & Scope

This manuscript aims to answer the research questions presented in this section that focus mainly on the trust of machine learning models, the application of xAI to counter Information Disorder, and the use of xAI to discover and mitigate the vulnerabilities of ML models countering Information Disorder. The research is driven by the following research questions emerged after the answer to the first research question. In the following the four research questions:

- **RQ1. Where, how, and for what purposes is xAI applied in Cyber Security according to the state of the art?**

  This question emphasizes the literature in the field of explainable AI in general and the application of xAI in Cyber Security with a comparison, in different fields of application, between transparent and non-transparent methods. From this analysis emerged three fundamental research questions that were explored with different works, these research questions are presented in the following and can be resumed in three points:

  - xAI 4 Reliability
  - xAI 4 Information Disorder
  - xAI 4 ML Robustness

- **RQ2. How to improve model's reliability?**

  The research is dedicated to providing a viable solution to support confidence in the results of ML models. It is underpinned by Formal Concept Analysis (FCA), a mathematically based theory for data analysis and classification. FCA's unique feature of constructing a lattice of concepts and a system of dependencies (implications and association rules) reassures the identification and visualization of hidden

relationships and structures in the data, thereby improving model understanding and transparency.

- **RQ3. How xAI can help to improve systems countering Information Disorder?**

  This research question takes a novel approach, exploring how xAI can be harnessed to combat Information Disorders. It aims to provide support by shedding light on the tools being fielded. Three papers are leveraged to answer this question. First one focuses on the innovative application of an xAI method to enhance a model used in text classification, thereby countering the spread of propaganda. The second work, on the other hand, provides insights into the predictions of a link prediction algorithm in a GNN for the potential dissemination of news containing false information. The third work presents an automated explainable Fact-Checking framework integrating LLM and ANFIS. The framework leverages the Retrieval Augmented Generation technology to explain which sources support or refute a fact's truth.

- **RQ4. How can xAI be utilized to identify and mitigate vulnerabilities in models?**

  xAI can identify and mitigate models' vulnerabilities by revealing how models make decisions, thus highlighting potential weaknesses. In practice, xAI techniques have been used to pinpoint key terms in sentence classification that can be targeted in adversarial attacks. Understanding which terms are crucial for the model's decision-making process makes it possible to test and improve the model's robustness against such attacks. The xAI has also been tested in identifying keywords and subsequently neutralizing them in prompts that lead LLMs to lose alignment by producing fake news.

To summarize, **RQ1** reviews the state-of-the-art in xAI for Cyber Security. **RQ2** explores improving model reliability FCA, **RQ3** investigates how xAI can counteract Information Disorder, highlighting applications using xAI to combat propaganda, for understanding predictions in Graph Neural Networks (GNN) for false information dissemination and for provide an explanation of fact-checking decision. In the end, **RQ4** examines using xAI to identify and mitigate vulnerabilities in ML models, revealing decision-making processes and potential weaknesses to improve model robustness against adversarial attacks and misinformation prompts.

Table 1.1 shows how each publication contributes to each research question.

1.4 The publications and their contribution

A summary abstract of each publication contributing to this work is presented in this section.

Publication I offers a thorough investigation into the applications of xAI in Cybersecurity, drawing upon an extensive review of over 300 articles. This analysis systematically explores the key domains within Cybersecurity where xAI has been applied while also

Research Questions	Publications
RQ1. Where, how, and for what purposes is xAI applied in Cyber Security according to the state of the art?	I
RQ2. How to improve model's reliability?	II, III
RQ3. How xAI can help to improve systems countering Information Disorder?	IV, VI , VII
RQ4. How can xAI be utilized to identify and mitigate vulnerabilities in models?	V, VIII, IX

Table 1.1 Contributions of the publications toward each research question.

identifying major limitations in the current literature. The most extensively studied areas include intrusion detection systems, malware detection, phishing and spam detection, botnet detection, fraud detection, Zero-Day vulnerabilities, Digital Forensics, and Crypto-Jacking. Particular attention is given to the explicability methods employed or proposed in these fields, emphasizing noteworthy advancements and highlighting emerging challenges in areas where xAI remains a promising but underdeveloped approach.

Publication II establishes a measure, namely Congruity, to provide information about the reliability of ML/DL model results. Congruity is defined by the lattice extracted through the Formal Concept Analysis built on the training data. It measures how much the incoming data items are close to the ones used at the training stage of the ML and DL models. The general idea is that the reliability of trained model results is highly correlated with the similarity of input data and the training set. In the experimentation, a correlation greater than 80% between the Congruity and the well-known Accuracy by varying ML models was demonstrated.

Publication III focuses on defining the Concept-Drift Detection Index to predict when the ML model performance for text stream classifiers goes low. It proposes an index that relies on the Fuzzy Formal Concept Analysis theory. The index exploits the formal lattice to understand whether new incoming facts (e.g., news) are well represented in the training data used to build the Machine Learning model. Fake News was deemed ideal for testing this new measure since its typical application scenario requires handling a stream of unstructured content and concept drift awareness. Experimentation revealed a relevant correlation between the Accuracies of different ML models and the proposed index.

Publication IV presents the Hybrid Fact-Checking Framework (HFCF) as a Deep Neural Fuzzy Sistem (DNFS) tailored to address the uncertainty inherent in fact verification tasks and enhance the reliability of model responses. The proposed DNFS integrates an LLM and an Adaptive Neuro-Fuzzy Inference System (ANFIS) for automated fact verification. The framework utilizes relevant evidence from open-world and closed-world sources to generate and justify verdicts by leveraging deep language models and employing few-shot prompting without additional training. Including fuzzy rules and considering the trustworthiness and relevance of retrieved evidence enhances response reliability, thereby improving overall effectiveness and outcome interpretability. Experimental results

demonstrated that the proposed DNFS could ensure better results, transparency of outcomes and mindful decision-making.

Publication VIII provides a comprehensive assessment of the robustness of the models against such attacks in key tasks well evaluated in the information disorder literature: Toxic Speech Detection, Sentiment Analysis, Propaganda Detection, and Hate Speech Detection. Rigorous experiments conducted on 13 models and 12 different datasets highlight significant vulnerabilities. The methodological framework implements adversarial attacks that strategically manipulate model inputs based on the meaning of keywords identified using the LIME method, an advanced explainable AI technique. The experiments reveal that the tested models show inconsistent resistance to adversary manipulations, underscoring the urgent need to develop more sophisticated defensive strategies. Finally, the study sheds light on critical weaknesses in existing models and charts a path for future research to strengthen AI resilience against evolving cyber threats.

Publication VI presents proposed solutions for addressing two tasks: "News Genre Categorisation" where the goal was to classify a news article as an opinion, a report, or a satire and "Detection of Persuasion Technique", where the goal was to identify persuasion techniques used in each news article paragraph choosing among 23 different. Solutions leverage the application of xAI method, Shapley Additive Explanations (SHAP), and in particular in the first task was used to understand what was driving the model to fail so that it could be improved accordingly. In contrast, in the task of Detection of Persuasion Technique, a re-calibration of the Attention Mechanism was realized by extracting critical tokens for each persuasion technique. The underlying idea is the exploitation of xAI for countering the overfitting of the resulting model and attempting to improve the performance when there are few samples in the training data.

Publication VII exploits a GNN to make link predictions on a graph representing information about misinformation tweets, their authors, and their spread. The objective was to comprehensively investigate the specific attributes of online pathways that compel users to share and amplify inaccurate information. In this sense, starting from an existing dataset of misinformation tweets, the proposed approach first applies an explainability method to each prediction, then, through frequent itemset mining, tried to detect patterns among collected explanations. Results of qualitative and quantitative research questions mainly demonstrated the contribution of interpersonal aspects to misinformation tweets spreading.

Publication VIII focuses on the role of the malicious use of xAI in increasing the effectiveness of adversarial text attacks or, dually, the aid its correct use may provide in measuring the robustness of propaganda detection models. The approach proposed leverages xAI and Adversarial Text Generation techniques to simulate malicious attacks and measure the robustness of a propaganda detection model based on BERT. The attacks generate a new dataset by perturbing critical words in the original one identified with SHAP and LIME. The goal is to quantify the impact of disrupted instances on learning model performance. Experiments revealed that modifying words detected by xAI methods significantly affects classification performance by reducing accuracy by 30%.

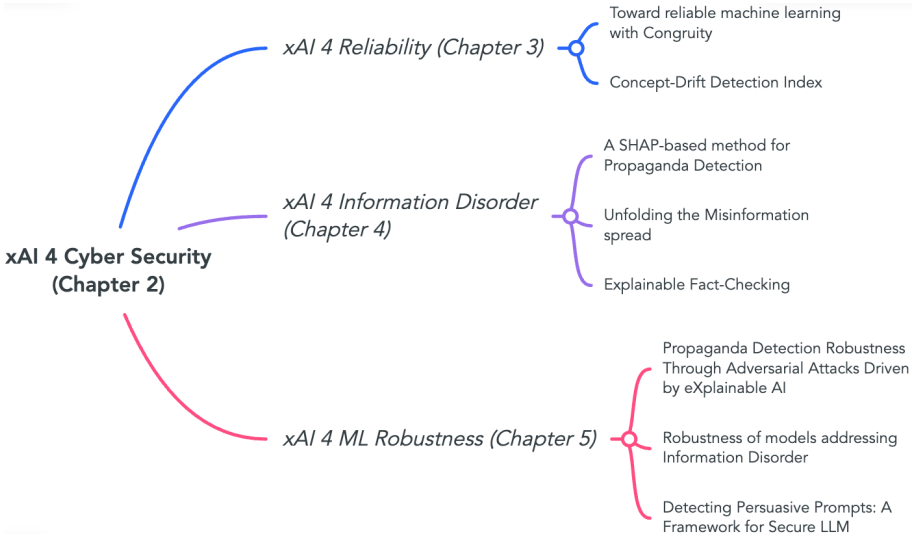

Fig. 1.2 Thesis Roadmap. Chapter 2 posed three research questions after a literature review on xAI 4 Cyber Security. The three research questions are explored in each chapter with different research work.

Publication IX proposes a mechanism for identifying jailbreaking prompts for manipulating Large Language Models (LLMs). The designed process involves the interaction between an LLM Attacker and an LLM Victim. LLM Attacker generates potential jailbreaking prompts to induce the LLM Victim to generate unethical content. The prompts and their corresponding persuasion success are collected during their interaction, creating a new synthetic dataset. Such a dataset is exploited to train a new model for detecting hidden persuasion in prompts that can induce an LLM to produce deviating content. This new model, assisted by algorithms for xAI, works as an anti-persuasion filter interposed between the input prompt and the victim model. It identifies attempts to mislead LLM and tries to neutralize them by modifying words recognized as crucial by xAI algorithms like SHAP and LIME. Experimentation reveals that adopting SHAP and removing the first ten most important words in the original prompt allows for neutralizing 80% of persuasive prompts.

1.5 Structure of the thesis

The thesis is structured in six chapters. After this chapter of Introduction, in Chapter 2 is conducted a literature review in the State-of-the-art of Explainable Artificial Intelligence in CyberSecurity. From this study emerged several challenges, three of these are explored in the chapters that follows, as depicted also in the Figure 1.2. Chapter 3 presents two works on how to improve models' reliability, Chapter 4 explores the possible use of xAI in order to improve the countermeasures for contrasting the Information Disorder. Chapter 5 focus, instead, on how mitigate models' vulnerabilities, in particular against adversarial attacks, leveraging xAI frameworks. Chapter 6 end the manuscript with the conclusions and future works.

Chapter 2

Literature Review

The following section describes research proposals addressing the first question: " What is the State-of-the-art of xAI and what are the challenges?". The goal is to provide an overview of xAI and applications in Cyber Security and the challenges that emerged. Published contributions are the following:

- Capuano, N., Fenza, G., Loia, V., & Stanzione, C. (2022). Explainable artificial intelligence in cybersecurity: A survey. IEEE Access, 10, 93575-93600.

2.1 Explainable Artificial Intelligence

DARPA, the Defense Advanced Research Projects Agency, financed the "Explainable AI Program" at the beginning of 2017. xAI aims to develop more understandable models while maintaining a high degree of learning performance (prediction accuracy); and enable human users to comprehend, adequately trust, and manage the future generation of artificially intelligent partners.
After the launch of the program, the scientific contribution in the Explainable Artificial Intelligence field has grown significantly, as shown in Figure 2.1.

2.1.1 xAI Taxonomy

Throughout the presented literature, various terms have been adopted, trying to cover all possible fields of application. Following are just a few of the wide variety used:
Trasparency: Do users grasp the format and language choices made by the model?
Fairness: Can it be proven that model judgments are fair to protected groups?
Trust: How comfortable are human users with using the system?
Usability: How well-equipped is the system to give users a secure and productive environment in which to complete their tasks?
Reliability: How resistant is the system to changes in parameters and inputs?
Causality: Do the predicted changes in the output, resulting from input perturbation, occur in the actual system?

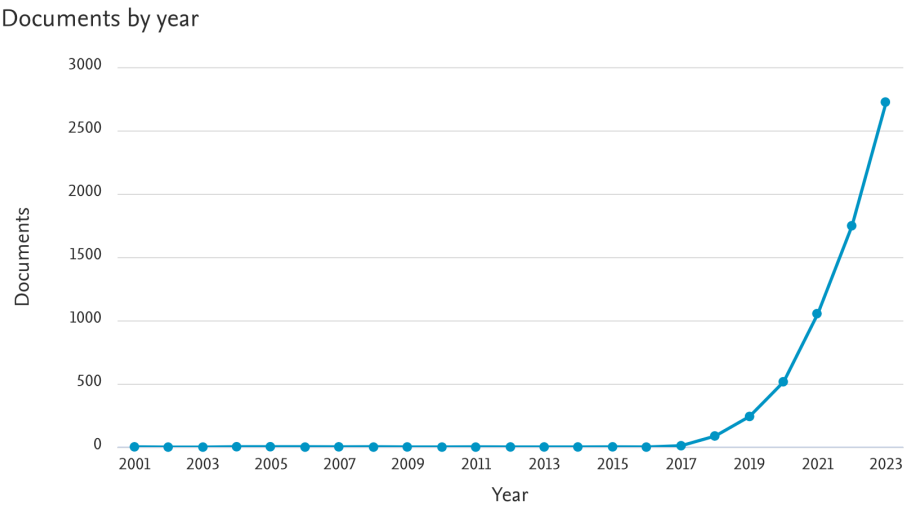

Fig. 2.1 Evolution of the number of total publications whose title, abstract and/or keywords refer to the field of xAI until 2023. Data retrieved from Scopus using as search key [TITLE-ABS-KEY (Explainable AND Artificial AND Intelligence)].

In the middle of 2020, the National Institute of Standards and Technology (NIST) presented four fundamental principles for explainable AI systems [153] as shown in Figure 2.2. The *Explanation* principle obligates AI systems to supply evidence, support, or reasoning for each output. A system fulfils the *Meaningful* principle if the recipient understands the system's explanations. The *Explanation Accuracy* principle imposes accuracy on a system's explanations and in the end *Knowledge Limits* principle states that systems identify cases they were not designed or approved to operate, or their answers are not reliable [153].
Over the years, a vast taxonomy has been developed on the various ways and methods that can make an AI model explainable. The first distinction needed is between *Interpretability* and *Explainability*. *Interpretability* is all about understanding the cause and effect within an AI system. On the other hand, *Explainability* goes beyond interpretability in that it helps us understand how and why a model came up with a prediction in a human-readable form. Figure 2.3 presents the current taxonomy and makes a crucial distinction between true transparency (interpretable models) and post-hoc interpretations (additional techniques used to shed transparency on complex black-box models). These techniques include producing local explanations for specific inputs or the entire model globally. Following a quick overview:

- **Model Specific or Model Agnostic:** This determines whether or not the interpretation method is restricted to a specific model. Model-specific methods and tools are those that are specific to a model. Model agnostic methods can be applied to any ML model to gain interpretability. Internal model data such as weights and structural details are not accessible to these models.

- **Intrinsic or Extrinsic (post-hoc) :** This indicates whether the model is interpretable on its own or whether interpretability requires using methods that examine models

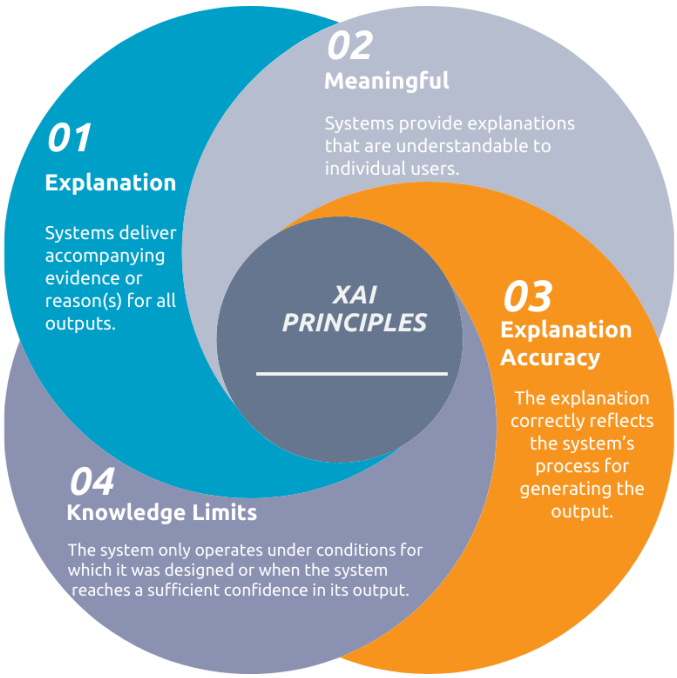

Fig. 2.2 xAI Principles presented by NIST in [153] .

after training. Simple, comprehensible models, like decision trees, are intrinsic. Utilizing an interpretation strategy after training to achieve interpretability is extrinsic.

- **Local or Global:** Whether the interpretation method describes a single data record or all of a model's behaviour depends on whether it is local or global. Global methods and tools interpret the entire model, whereas Local methods and tools only explain a single prediction.

2.1.2 xAI Frameworks

An xAI framework is a tool that creates reports on model activity and tries to explain how it works. The following are the main ones adopted in the SOTA.

LIME. Local Interpretable Model-agnostic Explanations (LIME) is a framework that seeks to provide an individual-level explanation of individual predictions (*Local*) in an extrinsic (*Post-hoc*) manner and is able to explain any model without needing to 'peak' into it (*Model-Agnostic*) [163]. In order to figure out what parts of the interpretable input are contributing to the prediction, it perturbs the input around its neighbourhood and see how the model's predictions behave. Then it generates a new dataset consisting of perturbed samples and the corresponding predictions of the black box model. On this new dataset, LIME then trains an interpretable model, which is weighted by the proximity of the sampled instances to the instance of interest.

SHAP. SHapley Additive exPlanations (SHAP) [129] is a framework with a clear goal, explaining the prediction of an instance x by computing the contribution of each feature to

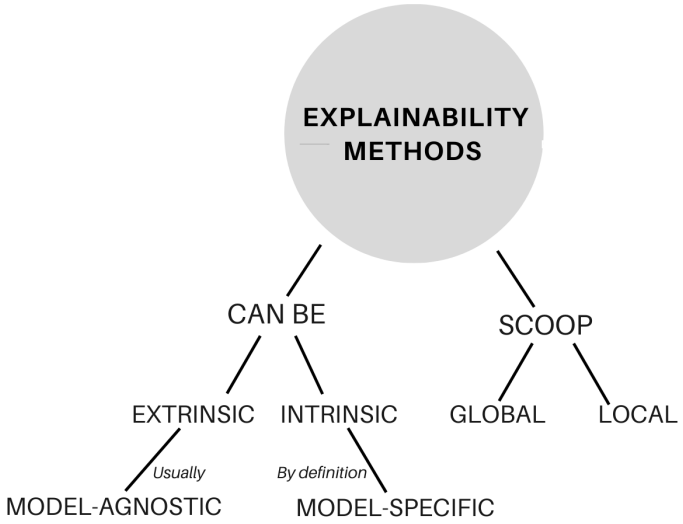

Fig. 2.3 A visual representation of xAI taxonomy.

the prediction. Like LIME, it is a *Local*-based, *Post-hoc*, and *Model-Agnostic* paradigm. The SHAP explanation technique uses coalitional game theory to compute Shapley values. A data instance's feature values operate as coalition members. Shapley values inform how fairly distributed the prediction is across the characteristics. A player might be a single feature value or a collection of feature values. It is not necessary to establish a local model in SHAP (as opposed to LIME), but rather the same function is used to calculate the Shapley values for each dimension.

Anchors. The Anchors approach [164] locates a decision rule that "anchors" the prediction adequately and uses it to explain specific predictions of any black box classification model. If changes in other feature values do not affect the prediction, a rule anchors it. Anchors reduces the number of model calls by combining reinforcement learning techniques with a graph search algorithm. The ensuing explanations are expressed as simple IF-THEN rules known as anchors. This framework is *Local*-based, *Post-hoc* and then *Model-Agnostic*.

LORE. LOcal Rule-based Explanations (LORE) [81] creates an interpretable predictor for a given black box instance. A decision tree is used to train the local interpretable predictor on a dense set of artificial cases. The decision tree allows for the extraction of a local explanation, which consists of a single choice rule and a collection of counterfactual rules for the reversed decision. This framework is *Local*-based, *Post-hoc* and then *Model-Agnostic*.

GRAD-CAM. Gradient-weighted Class Activation Mapping (GRAD-CAM) [175] is a technique for producing a class-specific heat map from a single image. Grad-CAM produces a class discriminative localization map as a result. The framework makes use of the feature maps generated by a CNN's final convolutional layer. This is *Local*-based,

Post-hoc but *Model-Specific*.

CEM. Contrastive Explanation Method (CEM) [59] provides explanations for classification models. More in detail, it retrieves the features that should be sufficiently present to predict the same class for the input instance. It also identifies minimal features to change for associating the input instance to a different class. This is *Local*-based, *Post-hoc* but *Model-Agnostic*.

2.2 Explainable Artificial Intelligence in Cyber Security

The following section reviews works that attempt to achieve Explainability in Cyber Security. In particular, the discussion examined the main areas of application that have been of most interest in Explainability at the state of the art. The chapter is then divided into two sections, with the first section addressing and analyzing the literature in the fields that have received the most attention from the world of Explainability, and these include Intrusion Detection Systems, Malware Detection, Phishing and Spam Detection, and BotNet Detection. Fields that have seen less interest and are analyzed in the second section include Fraud detection, Zero-day vulnerabilities, Digital Forensics and Cyber-Physical Systems. Section 2.3 then explored the challenges that emerged from this study that gave rise to the research questions addressed in this thesis.

2.2.1 Major Cybersecurity Threats Addressed by xAI

The following subsection analyzes the Cyber Security categories in which the most effort has been expended to apply Explaining Artificial Intelligence.

Intrusion Detection Systems

An Intrusion Detection System (IDS) is a cybersecurity solution designed to monitor network traffic and system activities for malicious activities or policy violations. An IDS can detect and alert administrators about potential intrusions, helping to prevent data breaches and mitigate damage. These systems work by analyzing the flow of data within a network, looking for patterns or behaviors that match known threats or deviate from normal usage patterns.

IDS can be broadly categorized into Network Intrusion Detection Systems (NIDS) and Host-based Intrusion Detection Systems (HIDS). NIDS monitors the network traffic for suspicious activity, while HIDS focuses on individual devices or hosts, inspecting system logs and activities for signs of compromise.

Organizations can bolster their defensive strategies by integrating an IDS, ensuring real-time detection and response to security incidents. This proactive approach is essential in safeguarding sensitive information, maintaining business continuity, and upholding the trust of customers and stakeholders in an increasingly hostile cyber environment.

xAI approaches in Intrusion Detection Systems. In [60], a system is proposed that is based on rules dictated by experts. It is *Hybrid* in the sense that it is a combination of human work and ML. The Explainability comes from *Rule-based*; the model behind it is a Decision Tree, a white-box model.

Szczepanski et al. in [186] propose a combination of oracle (ML model, in this case, tested ANN with a PCA) and an explainer module that would explain why a given classification is made. In the explainer module, one compares the distance from the clusters created on the training data. Then, the cluster closest to the test set instance is used for explanation.

In [135], the idea is to use an adversarial approach in order to be able to account for the minimal changes necessary for a classifier to arrive at an incorrect classification. The method thus makes it possible to visualize the features responsible for misclassification. For example, regular connections with low duration and low login success are misclassified as attacks. In contrast, attack connections with a low error rate and higher login success are misclassified as regular, demonstrating that relevant features significantly affect the final result.

A new way of interpreting an Intrusion Detection System is presented in [200]. The authors propose the use of SHAP for both local and global explanations. SHAP, by its nature, is a local method; they propose combining all local explanations to obtain a global explanation of the model. Almost equal work, with some less experimentation, is proposed in [203]. Le et al. [117] propose similar work through SHAP with an ensemble Tree model given a Decision Tree and a Random Forest model. Specifically, at the global level, they use a Heatmap for visualizing the impact of individual features on the classification of the overall model. At the local level, they use a Decision Plot to explain decisions on individual instances of the datasets. Another similar work is the framework proposed by [199], consisting of a Random Forest model using SHAP. The model can assess the credibility of the predicted results and ensure a high level of accuracy in detecting modern Cyber threats. The strategy adopted makes the final decision after cross-validation of the local explanation of the predicted outcome with the global explanation of SHAP.

The general idea proposed in [189] against adversarial attacks is divided into two parts, initialization and detection. During initialization, the model is trained with an SVM and features and characteristics that make a Normal classification are deduced via LIME. During detection, the Intrusion Detection System goes to compare. If it does not find the data as Normal, it classifies as an attack. On the other hand, if it is classified as Normal, there is a risk of an adversarial attack that is fooling the model. So a further check is done by reusing LIME. After that, the final result is reached.

FAIXID [124] is a new proposed framework that uses data cleaning techniques. They used four algorithms in the experiment to make the results explainable. They use the Boolean Rule Column Generation (BRCG) algorithm [49], which provides a directly interpretable supervised learning method for binary classification. Logistic Rule Regression (LogRR) [205] is a directly interpretable supervised learning method that can perform logistic regression on rule-based functions. The ProtoDash algorithm [85] pro-

vides example-based explanations to summarize datasets and explain the predictions of an AI model. Finally, the Contrastive Explanations Method (CEM) is used to compute explanations that highlight both relevant positives (PP) and relevant negatives (NP). Their proposal is not static but involves the use of algorithms depending on the specific case.
The work proposed in [120] defines a method to make rules for accessing the network dynamically and not statically as, for example, the rules set in a firewall may be. Thus, Explainability is the focus of the proposal. The explanation of the results consists of two main steps: i) training a model to approximate the local decision boundary of the target predictive model, and ii) reasoning about the trained model and the given input based on an explanation logic. The explanation is Local-based. They are inspired by LEMNA [82].
The aim in [4] is to increase transparency in an IDS based on a Deep Neural Network. Feedback is presented by computing the input features most relevant to the predictions made by the system. The model adopted is an MLP. Two forms of feedback are generated: 1) offline feedback (after training, before deployment) and 2) online feedback (during deployment). In offline feedback, the user is given the most relevant input features for each concept learned from the system. This information allows the user to evaluate whether the input characteristics that guide the IDS's decision toward a particular class (i.e., the type of attack) align with the domain experts' knowledge. On the other hand, the user is given the most relevant input characteristics for each prediction in the online feedback.
In [215], the authors focus on the possibilities of analyzing encrypted traffic, particularly for accurate detection of DoH (DNS Over HTTPS) attacks. They implement an explainable AI through the use of SHAP that allows visualizing the contribution of individual features to the model classification decision. Similarly, EXPLAIN-IT [139] is applied to the YouTube video quality classification problem in encrypted traffic scenarios. The work is based on a methodology that deals with unlabeled data, create meaningful clusters and proposes an explanation of the clustering results to the end-user. They use LIME interpreting clusters that are associated with a Local-based strategy then. Alike, ROULETTE [7] focuses on Network traffic. Specifically, attention is coupled with a multi-output DL strategy that helps better discriminate between network intrusions categories. As Post-hoc explanations, they consider visual explanation maps produced through Grad-CAM.
A two-stage ML-based Wireless Network IDS (WNIDS) is implemented in [1] to improve the detection of impersonation and injection attacks in a Wi-Fi network. The xAI was implemented to gain insight into the decisions made by the first-stage ML model, especially for cases where records were predicted as impersonation or injection. The features that contribute significantly to their prediction were determined. This set of features almost corresponds to those identified by the feature selection method for the second-stage ML model. They use SHAP.
In [134], the authors create a framework with a Deep Neural Network at its base and apply an xAI method depending on who benefits from it. For data scientists, SHAP and BRCG [49] are proposed, while for analysts Protodash is used. For end-users where an explanation on the single instance is required, they suggest SHAP, LIME, and CEM. Saran

et al. [172] propose a comparison between the NetFlow-based feature set[1] and the feature set designed by the CICFlowMeter tool[2]. This reliable comparison demonstrates the importance and need for standard feature sets among NIDS datasets, such as evaluating the generalizability of ML model performance in different network environments and attack scenarios. The SHAP method is used to explain the prediction results of ML models by measuring the importance of features. For each dataset, key features that influence model predictions were identified.
In conclusion, this work mentions [141], where an explainable automotive intrusion detection system is proposed, and [227] where a new general method is presented and tested on an IDS dataset. In [131] instead, the authors emphasize the importance of trust but do not use xAI methods.

Malware Detection

The term malware refers to programs potentially harmful to the user, which are aimed at stealing sensitive data, controlling the PC, or stealing user identity. The term malware originates from the contraction of the words "malicious software" and stands for a program (an executable, a dynamic library, a script, an HTML page, a document with macros, etc.) having unwanted and potentially dangerous effects on the user such as stealing sensitive data, controlling activity at the PC, identity theft, encrypting the hard disk with subsequent ransom demands, and so on.
Malware is usually classified according to its behaviour as Botnet, Backdoor, Information Stealer, Downloaders, Scareware, Rootkit, Worm, Virus, Ransomware or Trojans.
Some of the most common methods an attacker uses are Spam, Phishing, Hacking, Banner advertising, Search page rank, Expired domains or Domain Name Server (DNS) hijacking.
Malware detection techniques can be classified into three main categories (although other classifications exist) : (i) *Signature-based*, (ii) *Anomaly-based*, and (iii) *Heuristic-based*.
When using a *Signature-based* approach, programmers scan a file for malware, compare the information with a database of virus signatures, and then verify the results. If the information matches the information in the database, the file is infected with viruses. This approach limits the detection of unknown malware, but its main advantage is that it works well for known malware.
Anomaly-based methods mitigate the limitations of signature-based techniques, allowing detection of any known or unknown malware by applying classification techniques to the actions of a system for malware detection. Detection of malware activity is improved by moving from pattern-based to classification-based detection to identify normal or anomalous behaviour. Applying AI to *Signature-based* and *Anomaly-based* detection systems improves the efficiency of malware detection. *Heuristic-based* method use data mining and ML techniques to learn the behavior of an executable file.

[1]https://en.wikipedia.org/wiki/NetFlow
[2]https://github.com/CanadianInstituteForCyberSecurity/CICFlowMeter

xAI approaches in Malware Detection. One of the main works in this area is Drebin [8]; however, for consistency, it will not be analyzed in-depth as it is a pre-2018 work. Drebin explains his decisions by reporting, for each application, the most influential features, i.e., those present in the application and to which the classifier assigns the highest absolute weights. Melis et al. [137] provide an approach for the Explainability of malware detection in Android systems with an extension of the conceptual approach provided by Drebin on non linear models. Staying focused on Mobile, the authors of [105] use LIME in a method to identify locations deemed important by CNN in the opcode sequence of an Android application to help detect malware, while Kumar et al. [113] propose a static methodology for malware detection in Android where Feature Extraction provides transparency.
XMal [208] is an MLP-based approach with an attention mechanism to detect when an Android App is malware. The interpretation phase aims to automatically produce neural language descriptions to interpret key malicious behaviours within apps. Although the method is not so clear, the authors say they achieve better performance in interpretation than LIME and DREBIN.
The authors in [225] propose a backtracking method to provide a high-fidelity explanation of the DL detection method. The backtracking method selects the most important features contributing to the classification decision, thus resulting in a transparent and multimodal framework.
Feichtner et al. [68] designed a Convolutional Neural Network (CNN) to identify sample-based correlations between parts of the description text and the permission groups an app requests. They employ LIME to calculate a score for each word that shows the output's significance and visualize it as a heatmap.
As analyzed in the previous section, several methods focus on malware detection as an image; in [98], the authors propose a method relying on application representation in terms of images used to input an Explainable Deep Learning model. They represent a mobile application in terms of image and localize the salient parts useful to the model to output a certain precision by exploiting the Grad-CAM algorithm. In this way, the analyst can acquire knowledge about the areas of the image symptomatic of a specific prediction.
Shifting the focus from mobile applications to more general ones, LEMNA [82] is one of the main methods in the landscape of Explainability techniques. It was developed specifically for DL-Based Security Applications and is, therefore, one of the references in the general field of CyberSecurity. It was included in this section because the authors' primary experimentation is conducted on a Malware Detection Dataset. Given a sample of input data, LEMNA generates a small set of interpretable features to explain how the input sample is classified. The central idea is to approximate a local area of the complex DL decision boundary using a simple interpretable model. LEMNA uses a fused lasso-enhanced mixed regression model to generate high-fidelity explanation results for a range of DL models, including RNN.
DENAS [36] is a rule generation approach that extracts knowledge from software-based DNNs. It approximates the nonlinear decision boundary of DNNs, iteratively superimposing a linearized optimization function.

CADE [213] is designed to detect drifting samples that deviate from the original training distribution and provide the corresponding explanations to reason the meaning of the drift. The authors derive explanations based on distance changes, i.e., features that cause the most significant changes to the distance between the drifting sample and its nearest class. It was included in this paragraph because it is tested on a Malware detection dataset.
Pan et al. [149, 150] in two related works propose a hardware-assisted malware detection framework developing a regression-based Explainable Machine Learning algorithm. They apply a Decision Tree or Linear Regression to interpret the final result.
In order to understand how a Deep Network architecture generalizes to samples that are not in the training set and explains the outcomes of deep networks in real-world testing, the authors of [22] propose a framework that interpolates between samples of different classes at different layers. By examining the weights and gradients of various levels in the MalConv architecture [160] and figuring out what the architecture discovers by examining raw bytes from the binary, they try to use this framework to demystify the workings of the MalConv architecture. As a result, they can better explain the workings of ML algorithms and the decisions they make using the proposed framework. Additionally, the analysis will enable network inspection without starting from scratch.
Hsupeng et al. [96] introduce an explainable flow-data classification model for hacker attacks and malware detection. The flow data used for training the model is converted from packets by CICFlowMeter. This process significantly shrank the data size, reducing the requirement for data storage. For Explainability, they utilize SHAP further to investigate the relation between cyberattacks and network flow features.
MalDAE [87] is a framework that explores the difference and relation between the dynamic and static API call sequences, which are correlated and fused by semantics mapping. MalDAE provides a practical and explainable framework for detecting and understanding malware based on correlation and fusion of the static and dynamic characteristics. The explainable theoretical framework divides all API calls into several types of malicious behaviours according to their impact on security and builds a hierarchical malware explanation architecture.
Several works in the literature attempt to interpret malware detection by generating *Adversarial* attacks. The authors in [56] discovered that MalConv neural network does not learn any useful characteristics for malware detection from the data and text sections of executable files but instead has a tendency to learn to distinguish between benign and malicious samples based on the characteristics found in the file header. Based on this discovery, they devised a novel attack method that creates adversarial malware binaries by altering a small number of file header bytes. For the explanation, they use Feature Attribution to identify the most influential input features contributing to each decision and adapt it to provide meaningful explanations for classifying malware binaries. Other such works are [166, 176] employing SHAP and [179] proposing a new explanation algorithm to identify the root cause of evasive samples. It identifies the minimum number of features that must be modified to alter the decision of a malware detector, using Action Sequence Minimizer and Feature Interpreter.

To conclude the section, it is necessary to analyze the work of Fan et al. [67]. They designed principled guidelines to assess the quality of five explanation approaches by designing three critical quantitative metrics to measure their *Stability*, *Robustness*, and *Effectiveness*. The five explanation approaches are SHAP, LIME, Anchors , LEMNA and LORE. Based on the generated explanation results, they conducted a sanity check of such explanation approaches in terms of the three metrics mentioned. Based on their analysis, the ranking of the five explaining approaches in terms of the *Stability* metric is LIME $\geq$ SHAP > Anchors > LORE > LEMNA. The ranking of the five explaining approaches in the *Robustness* metric is LIME > SHAP > Anchors > LORE > LEMNA. In the *Effectiveness* metric is LIME > LORE > Anchors $\geq$ SHAP > LEMNA.

Phishing and Spam Detection

Phishing refers to a particular type of Internet fraud; the purpose of the malicious attackers, in this circumstance, is to get hold of users' personal and confidential data. More specifically, phishers practice the theft of logins and passwords, credit card and bank account numbers, and additional confidential data.
Spam is also called junk mail. It has existed almost as long as the internet as a means of selling products or services to a larger market of buyers than have ever expressed interest in those products or services. After obtaining the email addresses of a considerable number of individuals, spammers bulk send their offers hundreds or thousands at a time. Spam can be very dangerous if it is part of a phishing attempt.

xAI approaches in Phishing and Spam Detection. The current state of the art for Phishing and Spam detection with explainable methodologies is relatively poor. Therefore, techniques that are not created on-demand for Phishing and Spam Detection but use datasets targeted at these application domains were also considered.

Phishing. Phishpedia [122] is a *Hybrid* DL system that addresses two prominent technical challenges in phishing identification, (i) accurate recognition of identity logos on webpage screenshots and (ii) matching logo variants of the same brand. The authors compare the identity logo and input box providing Explainable annotations on webpage screenshots for the Phishing report.
Two works where the goal is not Phishing detection, but a dataset of this type is used for tests are [132, 31]. The first is based on a Deep embedded Neural Network expert system (DeNNeS) with a rule extraction algorithm for Explainability. The second is based on the Multi-Modal Hierarchical Attention mechanism (MMHAM) that permits the Explainability thanks to the hierarchical system.
Kluge et al. [107] propose a framework to convey to the user which words and phrases in an e-mail influenced a Phishing detector's classification of the e-mail as suspicious. They do it by locally perturbing inspiring to Anchors. The last analyzed work is [92], where the

authors use LIME and Explainable Boosting Machine (EBM) [145].

Spam. The authors of [183] looked into how different ML explanations, ML model's accuracy, and user confidence in the ML model affect user performance in a simulated Spam detection task. According to their findings, a user's confidence level in the model significantly influences the decision process. Users performed better when using an accurate model. Participants were more likely to spot false alarms generated by the more accurate model and more willing to follow through on a model "miss" when an additional model explanation was given.
FreshGraph [217] is a two-step system for recommending new products to target people that is Spam-aware. First, use item-user Meta-Path similarity and then entropy encoding measurements on a heterogeneous information network structure to identify false positives from candidate lists and avoid potential Spam. The suggested approach takes advantage of the semantic data stored within the graph structure, which considers user activity in addition to item content aspects for more precise audience targeting. Graph structure provides Explainability.
Gu et al. [79] examine the use of DL models to predict the effectiveness of outbound telemarketing for insurance policy loans to decrease Spam problems created by phoning non-potential customers. They propose an Explainable multiple-filter Convolutional Neural Network (XmCNN) to reduce overfitting. Explainability is calculated using feature importance by including a CancelOut layer after the input layer.
These two methods avoid getting into spam and are not spam detector methods. However, they still use Explainable methods of AI to avoid spam; that is why they were analyzed in this section.
The following analysis will focus on techniques that were not created to avoid Spam but instead use Spam datasets as testing. GRACE [116] generates contrastive samples that are concise, informative and faithful to the neural network model's specific prediction. SLISEMAP [18] finds local Explanations for all data items and builds a (typically) two-dimensional global visualization of the black box model such that data items with similar Local Explanations are projected nearby. [147, 26] are two works focused on text classification that use Spam datasets.

Bot (Net) Detection

A "Bot" or Robot, is a software program that performs automatic, repetitive, preset operations. Bots often mimic or replace the behaviour of human users. Since they are automated, they work considerably more quickly than actual individuals [44].
Malware and Internet bots can be programmed/hacked to access users' accounts, search the Internet for contact information, transmit Spam, and execute other dangerous operations. Attackers may use malicious Bots in a Botnet, or network of Bots, to launch these attacks and conceal their source. A Botnet is a collection of online-connected devices running one or more Bots, frequently without the owners' knowledge. Since each device has a unique IP

address, Botnet activity comprises many IP addresses, making it more challenging to locate and stop the source of malicious Bot traffic. When used to infect additional computers, Spam e-mail recipients' devices can help Botnets grow larger. They are commanded by hackers known as Botmasters or Bot herders.
Botnets are hard to spot since they consume very few computer resources. This keeps them from interfering with applications' regular operation and does not make the user suspicious. However, the most sophisticated Botnets can also alter their behaviour by the CyberSecurity systems of the PCs to evade detection. Most of the time, users are unaware that their devices are part of a Botnet and are under the control of online criminals [148].

xAI approaches in Bot (Net) Detection. BotStop [3] is a *Packet-based* Botnet detection system that examines incoming and outgoing network traffic in an IoT device to prevent infections from Botnets. The proposed system is founded on Explainable ML algorithms thanks to SHAP use with features extracted from network packets. Once an attack is detected, the source is blocked. Always SHAP is used in [114] to determine the relevant traffic features in a framework to detect traffic generated by a Bot and then determine the type of Bots using a Convolutional Neural Network.
Suryotrisongko et al. [185] propose the xAI and OSINT combination for Cyber Threat Intelligence Sharing in preventing Botnet DGA. This research applied four existing xAI techniques: Anchors, SHAP, Counterfactual Explanation and LIME. This latter is also used in [159] and [80] where the final goal is the detection in IoT Networks.
BD-GNNExplainer [226] is a Botnet Detection Model based on Graph Neural Network. The explanation is attributable to subgraph decomposition theory [130], where it is feasible to determine whether the learned model is interpretable by identifying the subgraph with the most significant influence on prediction and judging whether the subgraph is faithful to general knowledge.
[214, 62, 16], three explainable studies focused on DGA-based botnet detection, are also worth mentioning, as is [142], in which the authors created a Gradient-based Explainable Variational Autoencoder for *Network Anomaly Detection* utilizing a BotNet dataset as a test.

Bot-Detective [108] is an explainable Twitter bot detection service with crowdsourcing functionalities that uses LIME. LIME is also used in JITBot [102], An Explainable Just-In-Time Defect Prediction Bot, and in [61], a bot-type classification schema.
SHAP and LIME are used in [151] for game BOT detection, while in [123], the authors used a Decision Tree model, Explainable by definition, for automatic detection on Twitter with a particular case study on posts about COVID-19.

2.2.2 Minor Cybersecurity Threats Addressed by xAI

The categories considered up to this point are those in which the most effort has been expended to apply Explaining Artificial Intelligence in CyberSecurity. The categories that

have seen attention but minimal effort are considered below.

Fraud Detection

The financial sector is one of the ones most frequently targeted by cyberattacks. Frauds are frequent Cyber-attacks linked to money and reputation issues in this field. Data leaks and illegal credit losses may be the root of such attacks.
xFraud, an Explainable fraud transaction detection framework based on GNN, is presented in [162]. The authors designed a *Learnable Hybrid Explainer* that leverages GNNExplainer and centrality measures to learn node- and edge-level Explanations simultaneously. Srinath et al. [182] present an Explainable Machine Learning framework for identifying credit card defaulters using DALEX [17].

Zero-Day Vulnerabilities

The term "Zero-day" refers to recently identified security flaws that hackers utilize to attack systems. The expression "Zero-day" alludes to the notion that the vendor or developer has "Zero days" to repair the defect because they have just become aware of it. When hackers use a vulnerability before developers have a chance to fix it, a Zero-day assault is launched. The authors of [194] propose a new visualization technique using similarity matrices of features depicting behaviour patterns of malware and displaying them in image form for faster analysis for detection of Zero-day malware. Kumar et al. [112] use Shapley Ensemble Boosting and Bagging Approach instead for the same goal.
The authors in [174] propose a method for Zero-Day Web Attacks delivering outlier explanations. The method shows that Explanations can be backwards transformed through n-gram encoding and dimensionality reduction.
In [224], Zhou et al. define a Zero-day artificial immune system driven by xAI for intrusion detection in telecommunications. The central part of the artificial immune system is extracting strict rules for benign traffic. It uses a Decision Tree that is, by definition, a white-box model.

Digital Forensics

Digital Forensics, also known as Computer Forensics, is a specialized field within Forensic Science or Criminalistics. It focuses on the investigation, analysis, and retrieval of digital evidence from electronic devices. This branch of forensic science plays a crucial role in criminal investigations and legal proceedings by examining digital data to uncover, preserve, and present evidence that can be used in court. The collected data are identified, acquired, analyzed, and a technical report is written.
Hall et al. [86] assert that the application of AI in digital/network forensics is still a "Black box" at this time, requiring verification by digital/network Forensic investigators, and is

therefore unlikely to be justified in court. Furthermore, the admissibility of digital/network analysis performed by xAI in court is still debatable as it would necessitate a review of applicable laws (e.g., evidence law). However, xAI can be used efficiently and legally in the future to support the digital/network forensic profession if it is not viewed as a replacement for a digital/network forensic examiner but rather as a reliable tool to aid in investigations.
ATLE2FC [152] is a model for IoT Forensics using Ensemble Classification with an Explainable layer consisting of FPGrowth with GRU-based RNN classifier for rule estimation and severity classification.
For media forensic investigations focusing on media forensic object modification detection, such as DeepFake detection, a domain-adapted forensic data model is introduced in [109, 57].

Cyber Physical Systems

When an adversary gains access to a computer system that controls equipment in a manufacturing facility, oil pipeline, refinery, electric generating plant, or other similar infrastructure, they can control the operations of that equipment to harm those assets or other property. This is known as a Cyber-Physical attack on critical infrastructure. Cyber-Physical attacks pose a risk not only to the owners and operators of those assets but also to their suppliers, clients, enterprises, and people nearby the targeted asset, as well as to any individual or entity they could negatively impact. For example, a Cyber-Physical attacker may take down cameras, switch off the lights in a building, cause a car to wander off the road, or make a drone land in the hands of adversaries.
Wickramasinghe et al. [207] propose a Desiderata on Explainability of unsupervised approaches in Cyber-Physical Systems since they generate a large amount of unlabeled data. These are potential solutions for meaningfully mining these data, maintaining and improving desired functions, and improving the safety of these systems.
An Explainable Cyber-Physical Systems based on Knowledge Graph is proposed in [9] for Energy Systems while in [19] the authors propose a framework to build Self-Explainable Cyber-Physical System.

2.3 Discussion and Challenges

Due to the broad spectrum of XAI approaches, analyzing the different surveys involving these works were preferred to better orient the reader. Table 2.1 summarizes the principal works of XAI for each CyberSecurity application analyzed with a focus on the ML/DL model, the type of explanation and a summary concerning the taxonomy presented in section 2.1.1. In the following the main challenges emerged after the review conducted.

Table 2.1 Summary of Methods.

APPLICATION	REF.	YEAR	AI MODEL	EXPLANATION METHOD	SUMMARY
INTRUSION DETECTION SYSTEMS	[135]	2018	LINEAR CLASSIFIER/ MLP	ADVERSARIAL	H-L-AG
	[4]	2018	MLP	KEY FEATURE EXTRACTION	H-L-AG
	[120]	2019	RNN	REGRESSION + FUSED LASSO	I-L-SP
	[139]	2019	SVM	LIME	H-L-AG
	[186]	2020	ANN/DT	CLUSTERING	H-L-AG
	[1]	2020	RF/ NB/ XGB	SHAP	H-L (G) -AG
	[200, 203]	2020, 2021	MULTICLASS/ BINARY CLASSIFIERS	SHAP	H-L (G) -AG
	[199]	2021	RF	SHAP	H-L (G) -AG
	[189]	2021	SVM	LIME	H-L-AG
	[124]	2021	BRCG + LOGRR + PROTODASH+ CEM	BRCG + LOGRR + PROTODASH+ CEM	H-L (G) -AG
	[134]	2021	DNN	SHAP/BRCG/LIME/CEM/PROTODASH	H-L (G) -AG
	[172]	2021	DFF/RF	SHAP	H-L-AG
	[60]	2022	DT	WHITE-BOX MODEL	I-G-SP
	[117]	2022	RF	SHAP	H-L (G) -AG
	[215]	2022	RF	SHAP	H-L (G) -AG
	[7]	2022	CNN	GRAD-CAM	H-G-SP
MALWARE DETECTION	[137]	2018	SVM-RBF/ RF	GRADIENT-BASED	H-L (G) -AG
	[113]	2018	SVM	KEY FEATURE EXTRACTION	H-L-AG
	[82]	2018	RNN/ MLP	REGRESSION + FUSED LASSO	I-L-SP
	[225]	2019	CNN	BACKTRACKING	H-L-SP
	[56]	2019	MALCONV	FEATURE ATTRIBUTION	H-L-AG
	[68]	2020	CNN	LIME	H-L-AG
	[36]	2020	DNN	RULE-BASED	H-G-AG
	[149, 150]	2020, 2022	RNN/ DT	LINEAR REGRESSION/ WHITE-BOX MODEL	H-L-SP/ I-G-SP
	[105]	2021	CNN	LIME	H-L-SP
	[208]	2021	MLP	KEY FEATURE EXTRACTION	H-L-AG
	[98]	2021	CNN	GRAD-CAM	H-L-AG
	[213]	2021	MLP	CLUSTERING	I-L-AG
	[96]	2022	XGB	SHAP	H-L-AG
PHISHING DETECTION	[122]	2021	FASTER-RCNN	COSINE SIMILARITY	H-L-SP
	[107]	2021	RNN	Similar to ANCHORS	H-L-AG
	[92]	2021	RF/ SVM/EBM	LIME / EBM	H-L (G) -AG
SPAM DETECTION	[79]	2021	CNN	XMCNN filter	H-L-SP
BOTNET DETECTION	[80]	2019	RF/ KNN/ DECISION TREE	LIME	H-L-AG
	[159]	2021	RF/ EXTRA TREES	LIME	H-L (G) -AG
	[3]	2022	XGB	SHAP	H-L (G) -AG
	[114]	2022	CNN	SHAP	H-G-AG
	[185]	2022	NB/ LR/ RF/ EXTRA TREES	SHAP/ LIME/ ANCHORS/ COUNTERFACTUAL EXPLANATION	H-L (G) -AG
	[226]	2022	GNN	SUBGRAPH DECOMPOSITION	I-L-SP
FRAUD DETECTION	[162]	2021	GNN	CENTRALITY MEASURES	I-L-SP
	[182]	2022	XGB	DALEX	H-L-AG
ZERO-DAY VULNERABILITIES	[194]	2018	CNN	SIMILARITY MATRICES	H-G-AH
	[112]	2022	RF/ XGB/ EXTRA TREES	SHAPLEY VALUES	H-L-AG
	[224]	2022	DT	WHITE-BOX MODEL	I-G-SP
DIGITAL FORENSICS	[152]	2021	RNN	FPGROWHT	I-L-SP
CYBER PHYSICAL SYSTEMS	[9]	2021	KNOWLEDGE GRAPH	EXPLANATION GENERATION ALGORITHM	H-L-SP
Legend Summary: I: Intrinsic, H: Post-hoc, G: Global, L: Local, SP: Model-specific, AG: Model-agnostic					

More formalism is needed. xAI is a multidimensional target that a single theoretical approach cannot achieve. However, the synergistic employment of techniques from diverse study horizons must be done in a well-integrated manner. In other words, for the area to advance, it needs to be supported by a separate research community, which, at this point of development, should primarily focus on increased formalism. The reference is mainly to works that apply Explainable Artificial Intelligence methods in CyberSecurity without specifying in what and how, at what level, with output reported to whom (whether users, analysts or developers) and especially with what techniques. In the same field of application (e.g., Malware Detection), it would be good to unify the work in terms of

Explainability so that those in charge of analyzing and preventing cyber-attacks can have a unified and more understandable view.

How to achieve Explainability. In the current state of the art the proposed methods use post-hoc explanation in most cases. Developing models that provide an intrinsic explanation is a priority; an explanation method developed ad-hoc for that particular type of application is necessary for a field such as CyberSecurity, where one risks providing an assist to the attacker. Moreover, the problem may be precisely in terms of explanation, and the risk is to provide an untruthful output. As pointed out several times in [82], LIME, one of the most widely used methods, assumes that the decision boundary is locally linear. However, when the local decision boundary is non-linear, as it is in the majority of complex networks, those explanation approaches cause significant inaccuracies. In some cases, the linear portion is severely constrained to a relatively tiny region. The artificial data points beyond the linear zone are easily struck by standard sampling methods, making it hard for a linear model to estimate the decision boundary near x. The challenge then is not easy, the inverse correlation between model opacity and performance is well known, but an effort is needed to develop increasingly high-performing but transparent models.

Explainability for achieving Reliability. Simply explaining the model is not enough; the user must fully comprehend it to ensure strong reliability. Moreover, even with a well-crafted explanation, achieving this level of understanding may require additional clarifications and responses to anticipated user queries. Thus, explainability can only occur through human-machine interaction. In [95], the authors present an approach for creating a concept for an xAI-driven junior cyber analyst based on understanding the information needs of both humans and AI components in terms of the work context and workflow. This method may be required to design future systems that people can use, particularly for critical systems where human stakeholders cannot interact with black-box outputs from intelligent agents, as is the case in many CyberSecurity applications. Therefore, the idea and proposal are to think about and build frameworks that have human-machine interaction at their core for CyberSecurity applications, which is vital in many cases. The only way to get there is to build models understandable to humans.

Explainability for countering Information Disorder. A big gap has emerged in the subfield of Cognitive Security. Cognitive security is configured as a problem in the cyber domain, which involves a collection of strategies and approaches to protect against social engineering attacks and the deliberate or unintentional manipulation of cognitive processes and sensory disruptions. Information disorder is intrinsically connected to cognitive Security, which directly threatens the integrity of individuals' cognitive processes, impacting how they acquire knowledge, form beliefs, and make decisions. Encompassing misinformation, disinformation, and malinformation, information disorder can manipulate perceptions, distort reality, and erode trust in institutions and media. Countering Information Disorder is a huge problem that grows constantly, but the literature that leverages xAI to improve these

tools is not. It is needed, then, that these fields get in touch for constrain a growing problem.

Adversial Attacks. An in-depth investigation of how pattern explanations can provide new attack surfaces for the underlying systems is needed. A motivated attacker can use the information offered by the explanations to perform membership inference and pattern mining attacks, damaging overall system privacy. Regular adversarial attacks are predicated on the assumption that an adversary may inject a perturbation into an input sample that is undetectable to humans, and, as a result, the ground-truth class of the perturbed input does not change. The second issue is that a ML model's projected class changes. Attackers have developed several techniques to exploit weaknesses in xAI-enabled CyberSecurity frameworks. Adversary attacks circumvent authentication systems, such as the xAI-enabled facial authentication system, while poisoning attacks were used to alter or damage training data [115]. To combat these attacks, a solution could be to analyze "Desiderata for adversarial attacks in different scenarios involving explainable ML models" presented in [192]. The property of a model to withstand an adversary attack is called Robustness.

Chapter 3

xAi 4 Reliability

The following chapter is focused on two works published on how to improve model's reliability. Both works are based on the same mathematical theory, the FCA. FCA can be considered an explainable methodology due to its structure, which provides a clear and interpretable representation of data through formal concepts, which are pairs of shared sets of objects and attributes. FCA-generated structures, such as conceptual lattices, make it easy to visualize and understand relationships and hierarchies among dataset instances. This transparency and ease of interpretation help users understand the decisions made by FCA-based models, thereby making decision-making processes more explainable. This chapter presents approaches to improving machine learning models' reliability in two different ways. The first approach introduces a new index called Congruity, which measures how well the model recognizes items similar to the given input based on the dataset used during training. This index enhances the transparency of the training dataset, providing better insights into the model's awareness and decision-making process. The general idea is that the reliability of trained model results is highly correlated with the similarity of input data and the training set. The objective of the paper is to demonstrate the correlation between the Congruity and the well-known Accuracy of the whole ML/DL model. Experimental results reveal that the value of correlation between Congruity and Accuracy of ML model is greater than 80% by varying ML models. This work is presented in Section 3.1

The second work defines the Concept-Drift Detection Index to predict when the machine learning model performance for text stream classifiers goes low and then is not reliable. Fake News was deemed ideal for testing this new measure since its typical application scenario requires handling a stream of unstructured content and concept drift awareness. Experimentation on three datasets about news revealed a relevant correlation (i.e., 73.9%, 80.8%, and 81%) between the Accuracy of Random Forest (RF), Naive Bayes (NB), and Passive Aggressive (PA) models, respectively, and the proposed index. The strong correlation suggests the new index could avoid wrong classifications and help re-training decisions. This work is presented in Section 3.2

Published contributions are the following:

- De Maio, C., Fenza, G., Gallo, M., Loia, V., & Stanzione, C. (2023). Toward reliable machine learning with Congruity: a quality measure based on formal concept analysis. Neural Computing and Applications, 35(2), 1899-1913.
- Fenza, G., Gallo, M., Loia, V., Petrone, A., & Stanzione, C. (2023). Concept-drift detection index based on fuzzy formal concept analysis for fake news classifiers. Technological Forecasting and Social Change, 194, 122640.

3.1 Toward reliable machine learning with *Congruity*

What happens if you are a user of a model that is not performing well? What happens if the model you are using performs poorly? Was it trained on the wrong data? Did the data scientists choose a selective or biased dataset that does not match their reality? Did they select incorrect hyper-parameters that may work well for the programmers but not for final user?

Finding answers to these questions is almost impossible due to a lack of transparency in the ML or DL models. Since the market shifts from model builders to model users, more visibility and transparency make one able to trust the models that others built [43, 5]. Concerns related to model non-explicability can be multiple; firstly, the algorithm can be unexplainable in terms of outputs; secondly, there can be a lack of visibility of the training set. Such last aspect also determines opacity about data selection methods and bias in the training set that is also worsened by data constant changes due to drifts.

The US National Institute of Standards and Technology (NIST) [153] has developed four explainable AI principles that capture a variety of disciplines that contribute to explainable AI, including computer science, engineering, and psychology.

This work mainly impacts the following principle: *"The system should be able to explain its output and provide supporting evidence (at least)"*. To achieve greater trust in ML, it provides the following explanation types:

- Explanations that benefit the end-user;
- Explanations that are designed to gain trust in the system;
- Explanations that are expected to meet regulatory requirements;
- Explanations that can help with algorithm development and maintenance;
- Explanations that benefit the model owner, such as movie recommendation engines.

This contribution focuses on the lack of transparency enabling users to trust in predictions and decisions generated by the ML and DL models. The paper defines a *Congruity* measure to support the ML model explainability by enhancing the training dataset transparency. It is mainly devoted to providing reliability value every time we invoke an ML model after its creation. Like the Accuracy gives an idea of the performance of the model, the proposed Congruity is a value that says to the users how much the result provided by

the ML model is reliable. Such an index is mainly helpful for whoever (different from the researcher who created it) adopts the model without knowing the training details. Indeed, let us suppose we are asking to classify an email as spam or not. According to our idea, the system will provide the resulting label (spam or not spam) and the corresponding reliability measure, i.e. Congruity. The Congruity measures how much the model is aware of items similar to the input one considering the dataset used during the training stage. The proposal gets inspiration from the Out-of-Distribution detection paradigm [58]. It aims to evaluate the level of representativeness of new instances concerning the training set, to assess learning model predictivity. Despite the high overall Accuracy, the Congruity stresses that if the model is low experienced with the input, the user cannot trust the resulting classification. Moreover, the researcher who is training an ML model may use the average of the Congruity measured on the test set to know whether the training set is well-balanced and how well it represents the problem data.

More in detail, *Congruity* is a function that can be applied to each data item input of the ML model. It is evaluated by browsing a lattice built on the training dataset using FCA [73]. The result of the *Congruity* function reveals how much the data item input of the ML model is well represented in the training dataset. The objective of this work is to verify that *Congruity* is correlated to the *Accuracy* of an ML model. Once the correlation is confirmed, *Congruity* could be used twofold. Firstly, it could be used at the training stage of the ML model to select the most representative training dataset by averaging the *Congruity* value of the test set. Secondly, *Congruity* could be used later to know the trustability of the ML model results when the model is running on new real inputs. Experimentation is mainly performed on three existing datasets by varying the ML techniques. It reveals a high correlation between *Congruity* and *Accuracy*.

3.1.1 Methodology

The methodology of the work is presented below.

Congruity Measure based on Formal Concept Analysis

As outlined in the introduction, this work aims to find a correlation between Machine Learning and Deep Learning models and the *Congruity* concerning the dataset used during the training. The idea is to compare the *Accuracy* of the ML/DL model by varying the *Congruity* of items in the test set.

This section defines the *Congruity*. It measures how well a data item is represented in a data set (i.e., the training set). In the following subsections, the notation and the foundations of the theory are given before introducing the *Congruity* that is calculated by traversing the lattice-based structure extracted using FCA theory.

The motivations for using the FCA and the resulting lattice for evaluating the *Congruity* are essentially the following ones: i) coherently with the aim underlying this research work about the transparency of ML, FCA, and lattice are white-box data mining techniques; ii) the lattice gives the opportunity to summarize data with different levels of granularity

useful to empirically change the configuration of the *Congruity* evaluation according to the specific needs.

Notation. The following is the essential notation used in the next subsections for defining the *Congruity*:

- M – set of context attributes;
- G – set of context objects;
- L – lattice resulting from FCA;
- Attr(X) – attributes of an object or a concept X.
- Objs(C) – objects of a concept C.
- $C_i = (A_i, B_i)$ – Concept $i-th$ where $A_i \equiv Attrs(C_i)$ and $B_i \equiv Objs(C_i)$ are the set of attributes and the set of objects of concept C_i, respectively.
- $c(L, New)$ – congruity function that, given a lattice L and a new instance (or object) New, associates a real value in $[0-1]$.
- $G' = \{G'^1, G'^2, ..., G'^k\}$ whose intersection cardinality $|Attr(New) \cap Attr(G'_i)| > 0$ for $i = 1, 2, ..., k$;
- $S(C)$ - support of a concept C calculated as the ratio:

$$S = \frac{|Objs(C)|}{|G'|}$$

- $P(C, New)$ - given an input instance New and a concept C, the Precision P is evaluated as:

$$P = \frac{|Attr(New) \bigcap Attr(C)|}{|Attr(C)|}$$

- $R(C, New)$ - given an input instance New and a concept C, the Recall R is evaluated as:

$$R = \frac{|Attr(New) \cap Attr(C)|}{|Attr(New)|}$$

- $F-Measure(C, New)$ - the combination of Precision and Recall evaluated between instance New and concept C is given by the following equation:

$$F-Measure = 2 * \frac{P*R}{P+R}$$

Formal Concept Analysis. The formal model behind the proposed methodology is the FCA [73]. In the literature, the FCA is known as a method for knowledge representation, information management and data analysis [178]. It is able to understand relationships between a set of objects and a set of attributes represented in the formal context (through a tabular way). So, it detects concepts containing objects sharing the same attributes. In this way, the resulting lattice represents the underlying structure of the analyzed context.

From its introduction, FCA was applied for numerous purposes, for example, data mining, data analysis, information retrieval, taxonomies and ontologies building, clustering, recommendation, network analysis [156, 223, 71], etc. More recently, it was also adopted for Machine Learning explainability goals [169].

Following, some definitions about FCA are given.

Definition 1: *A* **Formal Context** *is a triple $K = (G, M, I)$, where G is a set of objects, M is a set of attributes, and $I = (G \times M)$ is a binary relation. $(g, m) \in I$ is read "object g has attribute m".*

The context is often represented as a "cross table" (see Table 3.1): the rows represent the formal objects and the columns are formal attributes; the relations between them are represented by the crosses.

Taking into account the Formal Context, FCA algorithm is able to identify Formal Concepts and subsumption relations among them. More formally, the definition of Formal Concept and order relation among them are given as follows:

Definition 2: Formal Concept. *Given a formal context $K = (G, M, I)$, for $A \subseteq G$, apply a derivation operator, $A' = \{m \in M \mid \forall g \in A : (g, m) \in I\}$ and for $B \subseteq M$, $B' = \{g \in G \mid \forall m \in B : (g, m) \in I\}$. A formal concept C is identified with a pair $C = (A, B)$, where $A \subseteq G$, $B \subseteq M$, such that $A' = B$ and $B' = A$.*

Definition 3: *Given two concepts $C_1 = (A_1, B_1)$ and $C_2 = (A_2, B_2)$, then C_1 is a subconcept of C_2 (equivalently, C_2 is a superconcept of C_1), $(A_1, b_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 (\Leftrightarrow B_2 \subseteq B_1)$. The set of all concepts of a particular context, ordered in this way, forms a complete lattice.*

Note that each node in Fig. 3.10 (i.e., a Formal Concept) comprises the objects and the associated set of attributes. In the figure, each node has a different color according to its characteristics: a half-blue colored node represents a concept with *own* attributes; a half-black colored node instead outlines the presence of *own* objects in the concept; finally, a half-white colored node represents a concept with no *own* objects (if the white-colored portion is the half below of the circle) or attributes (if the white half is up on the circle).

Given the Formal Concepts, it is easy to see that the subconcept relation $\leq$ induces a *Lattice* of Formal Concepts. As a matter of fact the lowest concept contains all attributes and the uppermost concept contains all object of the Formal Context.

FCA Example. Let us explain FCA through a practical example.

Assume to have information about age, gender, and Body Mass Index (BMI) of 10 people, as shown in Table 3.1.

Table 3.1 Example data - The table shows the attributes representing each person used to build the example.

Person	Age	Gender	BMI
Person1	25	Male	23
Person2	59	Female	20.5
Person3	68	Female	24
Person4	18	Male	25
Person5	44	Male	27
Person6	81	Male	17.5
Person7	33	Female	31
Person8	49	Male	19.5
Person9	77	Female	30
Person10	90	Female	18

The construction of the Formal Context needs information bucketing. In particular:

- For categorical attributes, such as *gender*, we define an attribute for each possible value (i.e., *Male* and *Female*).
- For numerical attributes (i.e., *age* and *BMI*), we establish some thresholds and define an attribute for each range. In particular, people are considered *Adult* if their age is lower than 60; otherwise, *Elderly*. Regarding BMI, we assume:
  - *Underweight* people have a BMI lower than 18.5;
  - *Normal* people have a BMI between 18.5 and 24.9;
  - *Overweight* people have a BMI between 25 and 29.9;
  - *Obese* people have a BMI greater than or equal to 30.

The resulting Formal Context is one in Table 3.8. As notable, for each row (i.e., person) there is a "X" at each intersection with the owned attribute.

By applying the FCA algorithm, the lattice in Figure 3.10 results. We can notice that people with the same attributes (e.g., *Person4* and *Person5*) rely on the same concept.

Congruity FCA conceptualizes input data and generates a hierarchical knowledge structure (i.e., lattice L). *Congruity* is defined by the lattice resulting from the FCA. It is a function $c(L,X)$ that takes as input the lattice L and a new data item, i.e., the instance X. Intuitively, the *Congruity* should represent a measure of representativeness of X in L, which describes qualitatively and quantitatively the modeling of the new input instance with the concepts already available in the extracted lattice. In essence, two pieces of information are evaluated:

Table 3.2 Formal Context Example - After an appropriate bucketing process, the attributes shown in Table 3.1 create the necessary context for constructing the lattice.

	Adult	**Elderly**	**Male**	**Female**	**Under-weight**	**Normal**	**Over-weight**	**Obese**
Person1	X		X			X		
Person2	X			X		X		
Person3		X		X		X		
Person4	X		X				X	
Person5	X		X				X	
Person6		X	X		X			
Person7	X			X				X
Person8	X		X			X		
Person9		X		X				X
Person10		X		X	X			

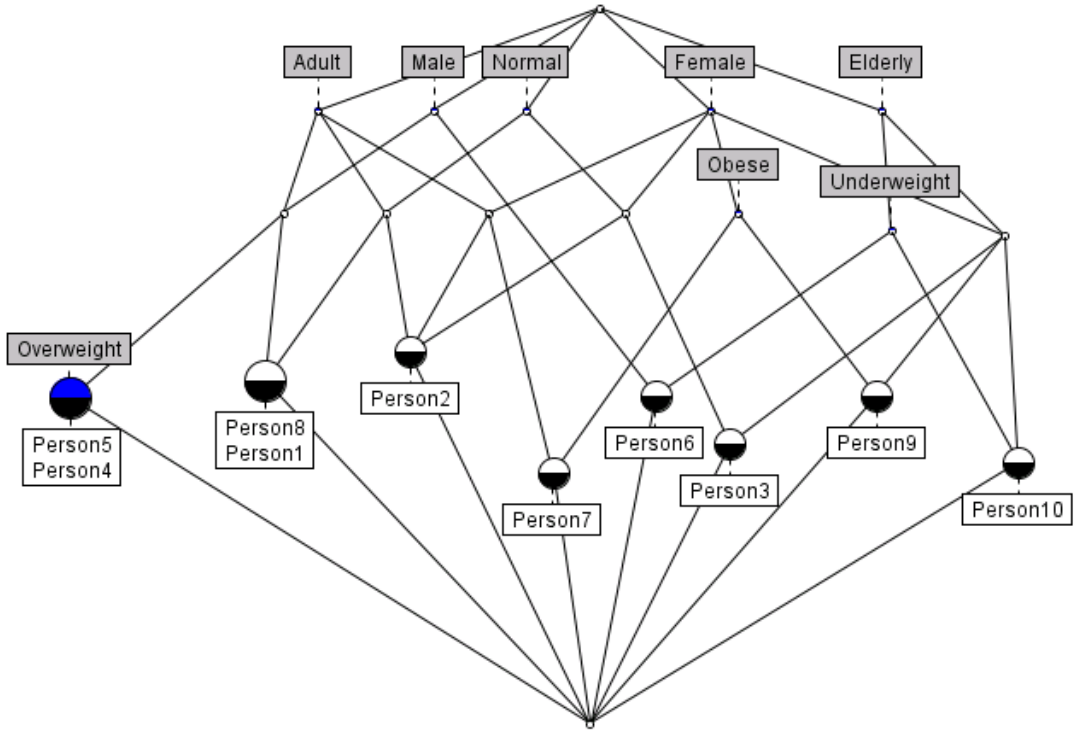

Fig. 3.1 Lattice Example - The figure shows the lattice generated from the Formal Context in Table 3.2. Each concept owns objects (i.e., people) that share the same attributes (e.g., *Person1* and *Person8* that share *Adult*, *Male*, and *Normal*).

- how many items with the same characteristics were in the sample set during the lattice construction;
- the coverage degree of characteristics of the new instance with concepts in the lattice *L*.

If we consider a Machine Learning classifier, the level of Congruity depends on how much the sample used to train the classification model is representative of new instances

to classify. That happens, on the one hand, if the features of the classifier include the characteristics of the instance to classify, and, on the other hand, if the classification of similar objects is being trained a number of times such that the classes are balanced in terms of the number of considered instances. Starting from this observation, this work aims to evaluate the correlation between the proposed *Congruity* measure and the measure of reliability of a classifier output.

Analysis of possible cases. Given the previous intuitive definition of congruity, we could say that in the presence of a new input object, there are two aspects to consider:

- the *support* of the concept(s) in which the new input instance would fall (or which are closest to that in which it would fall);
- the *coverage* of the input instance in terms of attributes in the available concepts. According to [133], the coverage could be the degree of matching calculated as the F-measure of new instance characteristics for the lattice concepts.

Thus, the *Congruity* could be a linear convex combination of the *support* and *F-measure* values between the characteristics of the concepts with a set of attributes whose intersection with the attributes of the new instance is non-empty. To assess the appropriateness of computing and combine these two values to obtain the *Congruity*, we proceed with the possible case enumeration.

Given a new instance X where $Attrs(X) = \{a_1, a_2, ..., a_n\}$, the cases that can occur with respect to the existing lattice L are the following:

1. There are one or more concepts in the lattice, $\widetilde{C} = \{\widetilde{C}^1, \widetilde{C}^2, ..., \widetilde{C}^k\}$ whose intersection cardinality $|Attr(X) \cap Attr(\widetilde{C}_i)| > \frac{|Attr(X)|}{2}$ for $i = 1, 2, ..., k$, and then we have the following further subcases:

   - Among all concepts in $\widetilde{C}$, there exists $\widetilde{C}^*$ such that Attrs($\widetilde{C}^*$) coincides with Attrs(X); in that case, the F-Measure($\widetilde{C}^*, X$) shall be maximum (i.e., equal to 1), and the congruity will only have to consider the support of the concept $\widetilde{C}^*$. Assuming congruity is a linear convex combination of the support and F-Measure values with weights α_1 and α_2, then we will have:

$$\alpha_1 S(\widetilde{C}^*) + \alpha_2 F - Measure(\widetilde{C}^*, X) = \alpha_1 S(\widetilde{C}^*) + \alpha_2 \tag{3.1}$$

   - In other cases, Congruity $c(L, X)$ can be calculated as:

$$c(L,X) = \frac{1}{|\widetilde{C}|} \sum_{\widetilde{C}^i \in \widetilde{C}} \alpha_1 S(\widetilde{C}^i) + \alpha_2 \; F - Measure(\widetilde{C}^i, X) \tag{3.2}$$

2. There is no concept C^* such that the intersection cardinality $|Attr(X) \cap Attr(\widetilde{C}_i)| > \frac{|Attr(X)|}{2}$. In this case, the congruity is $c(L, X) = 0$.

Congruity Example. Let us start from the example in Section 3.1.1 and assume to try to classify two new people characterized as in Table 3.3. First, we need to evaluate its Congruity concerning the existing Lattice (i.e., Figure 3.1) through a bottom-up visit for each new instance.

Regarding *Person11*, we found a concept (C_7 in Figure 3.2) that contains all new instance attributes. The Congruity index evaluation is as follows:

- The Support (the number of objects belonging to the retrieved concept divided to the total number of objects sharing at least one *Person11* attributes) measures $1/8 = 0.125$.
- The F-Measure is maximum (i.e., 1) because the concept attribute set corresponds with the instance attribute set.
- Let the weights $\alpha_1 = 0.1$ and $\alpha_2 = 0.9$, by following Equation 3.1, the *Congruity* value for *Person11* is: (0.125*0.1)+(1*0.9) = **0.913**.

Table 3.3 Formal Context example of new items - The table presents attributes of new objects.

	Adult	**Elderly**	**Male**	**Female**	**Under-weight**	**Normal**	**Over-weight**	**Obese**
Person11		X		X		X		
Person12	X			X			X	

The second object, *Person12*, has the attribute set {Adult, Female, Overweight}. This combination of attributes is new, and no concept like this already exists. So, the first step is to identify the concepts with the most similar set of attributes (i.e., concepts owning at least half of instance attributes). In this case, matching concepts are C_1, C_3, C_4, and C_5. Congruity evaluation starts from Support and F-Measure of these matching concepts, as shown in Table 3.4.

Table 3.4 Example Congruity Computation - The table shows for each matched concept the results of Support, Precision, Recall and then F-measure for the new Person12 instance.

Concept	**Support**	**Precision**	**Recall**	**F-Measure**
C_1	0.22	0.67	0.67	0.67
C_3	0.11	0.67	0.67	0.67
C_4	0	1	0.67	0.8
C_5	0.11	0.67	0.67	0.67

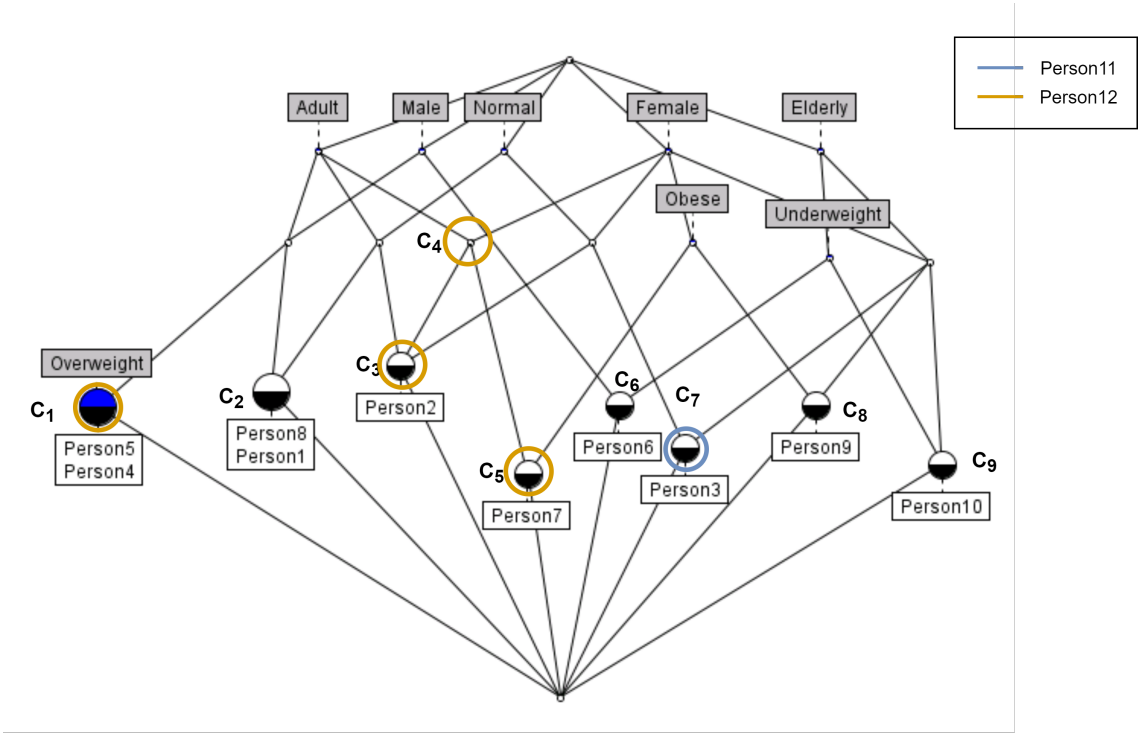

Fig. 3.2 Congruity Computation Example - In the lattice, all generated concepts are represented; note that only those where objects are present or which match the example have been numbered to make the picture clearer. Moreover, in blue is highlighted the concept that completely matches *Person11*; in yellow, there are concepts matching *Person12*'s attributes. In particular, we only consider those concepts where the intersection with the attributes of the incoming instance is greater than half of the attributes f the new instance itself.

From Table 3.4, it follows that:

$$\begin{aligned} Congruity = \frac{1}{4}[(0.1*0.22)+(0.9*0.67)+(0.1*0.11)+ \\ (0.9*0.67)+(0.1*0)+(0.9*0.8)+ \\ (0.1*0.11)+(0.9*0.67)] = \\ \frac{1}{4}[0.022+0.603+0.011+0.603+0.72+0.011+0.603] = \\ \frac{1}{4}*2.573 = \mathbf{0.643} \end{aligned}$$

The example serves to clarify the Congruity value and its calculation. It is good to consider that the example uses a minimal context and test set. After creating the lattice, we calculated the Congruity for the two new incoming instances, which in the specific case represent two people. In the case of *Person11*, the congruity calculation is straightforward. The new instance completely matches an existing object. Therefore, the F-measure is equal to 1, as specified in the formulas previously. On the other hand, the Support is equal to 0.11 since there are eight objects involved in terms of attributes intersection. The Congruity for *Person11* is, therefore, equal to 0.913. Such a high value is due to the existence of the same attribute set in the initial context.

For *Person12*, the calculation involves four concepts considering an intersection cardinality between attributes of the new instance and once of the concept greater than half of the first one. Regarding the Support evaluation, we must consider 9 objects. Summing up, in this case, the Congruity value is 0.643. A good value, but not excellent, confirming that the learning model never saw these attributes together but only partially and with low Support. Of course, as mentioned before, there is to keep in mind that the example is based on a very low number of instances and could mislead the reader. On the other hand, the lattice can present millions and give rise to a profound representation of what is the context used to create it.

Correlation between Congruity and Accuracy

Explainability in Artificial Intelligence could refer to two aspects:

1. *Explaining the AI model pedigree*: how the model was trained, which data was used, which types of bias are possible, and how to mitigate them.
2. *Explaining the overall model*: this is also called "model interpretability."

This work focuses on the first point: explaining the AI model pedigree. In particular, to explain the model pedigree means to answer the following questions:

- How was the model trained?
- What data was used?
- How was the impact of any bias in the training data measured and mitigated?

These questions are the data science equivalent of explaining what school your surgeon went to — along with who their teachers were, what they studied, and what grades they got. Of course, getting this right is more about the process and leaving a paper trail than pure AI, but it is critical to establishing trust in a model.

Fig. 3.3 shows the workflow of the proposed solution. The process starts with data pre-processing to obtain a dataset that allows the lattice extraction through the FCA. Then, the following steps are carried out:

1. Lattice extraction of the training set by applying the FCA;
2. ML/DL model training on the same dataset;
3. Calculation of the test set Congruity concerning the lattice constructed on the training set;
4. Grouping congruity values and calculating corresponding ML Accuracy;
5. Calculation of Congruity-Accuracy correlation.

The following subsections will detail each workflow step.

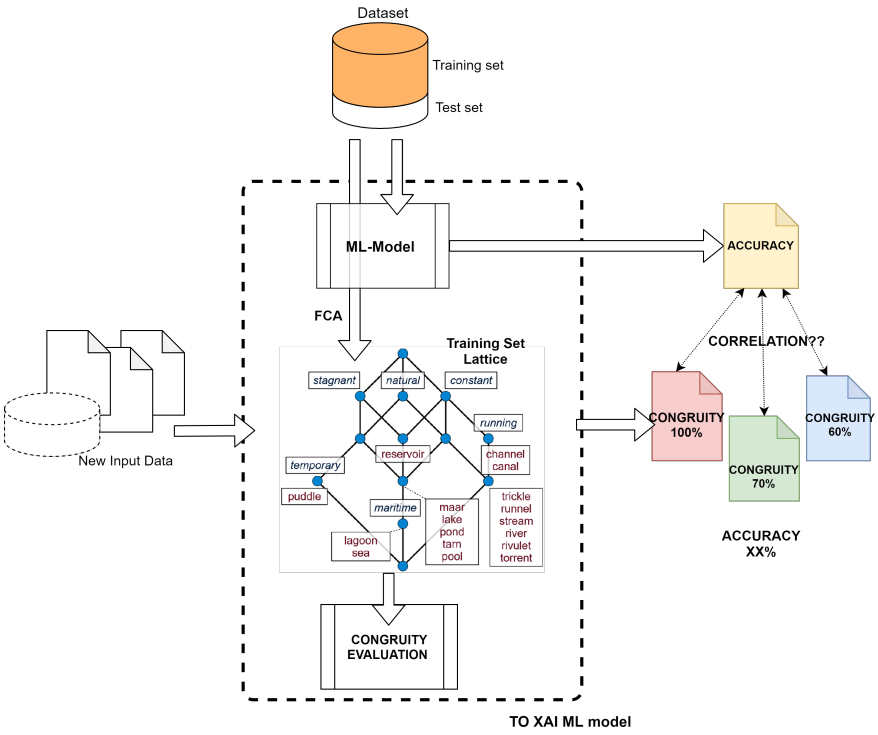

Fig. 3.3 General Workflow - The workflow reflects the general idea. First, the lattice is built using the entire training set. Then, the Congruity is calculated for each instance of the test set. The congruity values are sorted and grouped by similar values. At this point, the model Accuracy is calculated for each group. The aim is to demonstrate a correlation between the Congruity and Accuracy values: Do increasing congruity values correspond to increasing accuracy values?

Lattice extraction. This phase computes the FCA algorithm from the dataset used to train the learning model. The objective is to construct the formal lattice functional to calculate the *Congruity* indicator.

ML/DL model training. In this phase, the dataset used for lattice construction is used to train the ML/DL model. The choice of implementing a classifier as a Machine Learning model is dictated by the fact that the lattice only serves to "measure" the dataset and not to make any prediction.

Congruity computation. After creating the lattice through the training set, the Congruity value is calculated for each instance in the test set. The objective is to evaluate consistency between training and test sets.

Grouping Congruity values and Accuracy calculation. This step is fundamental for the final one, where the correlation between Congruity and Accuracy of the classifier is calculated. The idea is to sort, in ascending order, Congruity values previously evaluated. Subsequently, group them to have a similar number of instances for each group. The grouping is done for values that are very close to each other.

Calculation of Congruity-Accuracy correlation. The last step is to calculate the correlation between Congruity and Accuracy values. For this calculation, the values obtained in the previous step are used, where groups of instances with close congruity values are associated with Accuracy values obtained using the classifier. Pearson's correlation coefficient, Kendall's Tau coefficient and Spearman's rank correlation coefficient [173] are used to measure the correlation.

3.1.2 Experimentation

This section describes experimentation conducted on two health datasets and a textual one. It discusses the achieved correlation between Congruity and ML/DL model classification Accuracy.

Tools. The main tools adopted during the experimentation are: a custom developed library implementing FCA derived by extending "Colibri"[1]; Apache Solr[2], used for indexing concepts hierarchy extracted by FCA; the sci-kit learn library[3], used to construct and adopt the logistic regression model, keras[4] library for ANN and DNN models building.

Datasets. Since model interpretability plays a crucial role in decision support systems, especially in areas like health, the framework evaluation starts from two health datasets: Pima Indians Diabetes Database[5](PIDD) and Stroke Prediction Dataset[6](SPD). In addition, experimentation on a textual dataset is conducted through the Coronavirus tweets NLP dataset[7](CT).

Pima Indians Diabetes Database is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to predict whether or not a patient has diabetes diagnostically.

Stroke Prediction Dataset is used to predict whether a patient is likely to get a stroke based on his/her characteristics like gender, age, other diseases, etc.

Coronavirus tweets NLP dataset collects tweets with a common topic, the Coronavirus. The focus of the dataset is Sentiment Analysis with 5 labels ranging from "extremely positive" to "extremely negative".

Workflow. In Fig. 3.4, the experimentation workflow is presented. It summarizes the application of the proposed methodology by reporting some application examples.

Data preparation. Since the Stroke Prediction Dataset had 201 samples with absent BMI (body mass index) value, rather than imputing it naively with the mean or median,

[1]https://code.google.com/p/colibri-java/
[2]https://en.wikipedia.org/wiki/Apache_Solr
[3]https://scikit-learn.org/stable/
[4]https://keras.io
[5]https://www.kaggle.com/uciml/pima-indians-diabetes-database
[6]https://www.kaggle.com/fedesoriano/stroke-prediction-dataset
[7]https://www.kaggle.com/datasets/datatattle/covid-19-nlp-text-classification

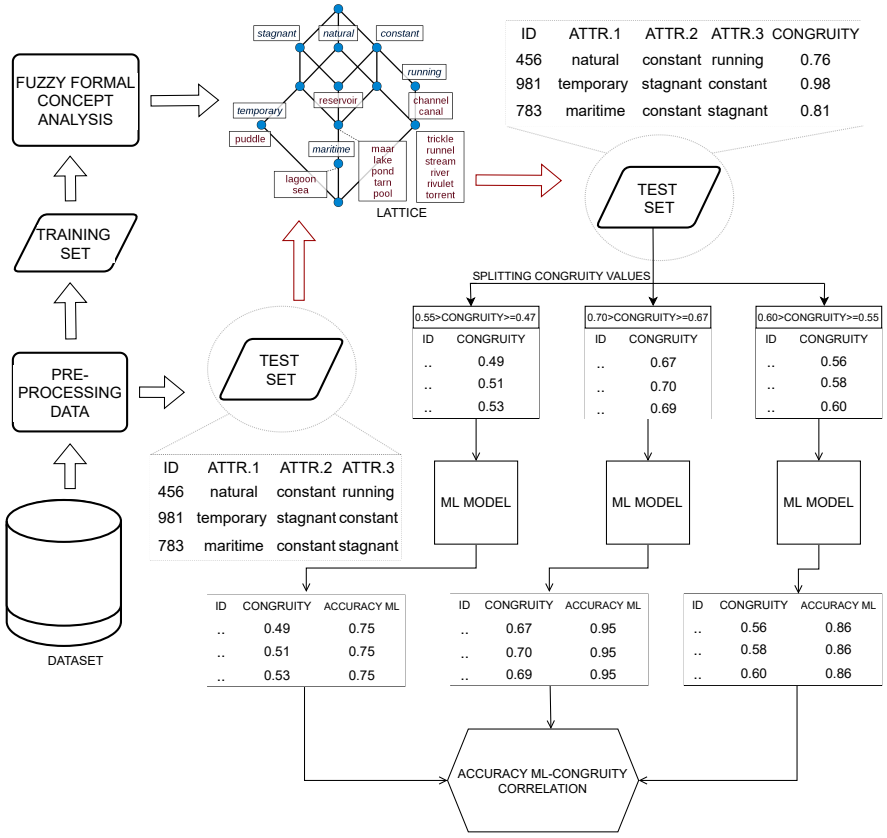

Fig. 3.4 Experimentation Workflow - The data is pre-processed and divided into training and test sets. In the case of the textual dataset, vectorisation with Tf-idf was also adopted during the pre-processing phase. The Formal Concept Analysis was applied to the training set, producing a lattice containing concepts representing data used to train the ML and DL models. The test set data is synthesised into queries to calculate the Congruity values for each incoming instance. The Congruity values are divided into equally distributed ranges. The corresponding instances are fed to the model to estimate the related Accuracy. Finally, the correlation value between the two measures is calculated at the end of the process.

we adopted a solution also suggested in the literature [104] that uses a decision tree model. The model establishes a fair decision tree model based on the age and gender of all other samples predictions for the missing values. Then, the dataset is divided into training and testing, with a percentage of $75-25\%$ with 3832 and 1278 instances, respectively. The same percentage has been adopted for the PIDD dataset.

Regarding the textual dataset (i.e., CT) that consists of approximately 45000 rows corresponding to likewise tweets, the pre-processing mainly applies a Natural Language Processing workflow. It removes unnecessary parts (e.g., links, stopwords, users' tags, etc.) and tokenizes the tweet textual content. Moreover, a vectorization was applied by exploiting the Tf-idf (Term Frequency — Inverse Document Frequency) was applied. Finally, the dataset was divided into training and test sets with a percentage assigned to the latter of 30%, thus obtaining 31352 instances in the training set and 13338 in the test set.

Experimental Results. Four different models are used for experimenting with correlation among Congruity and Accuracy values in the health datasets: a Kernel Support Vector

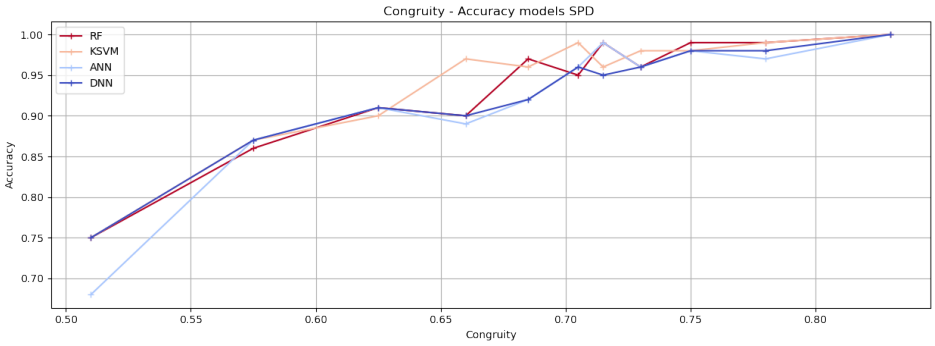

Fig. 3.5 Congruity values range - Accuracy models on SPD dataset. The chart shows the Accuracy values of the model tested with instances falling within the specific Congruity range. RF refers to Random Forest, KSVM to Kernel Support Vector Machine, ANN to Artificial Neural Network and DNN to Deep Neural Network.

Machine with a radial basis function kernel, a Random Forest with ten trees and entropy for the information gain, an Artificial Neural Network (ANN) and a Deep Neural Network (DNN), both using a Sequential model fitted in 100 epochs. In DNN, hidden layers are 3. Two models were used for experimentation for the textual dataset, Coronavirus Tweets NLP, the Random Forest and the Multilayer Perceptron (MLP). Table 3.5 shows the Accuracy values of each model on the overall test set for the three datasets.

Table 3.5 Overall Models Accuracy. - The table shows the Accuracy results of each adopted model.

Dataset	**Model**	**Accuracy**
SPD	Deep Neural Network	95.1%
	Kernel Support Vector Machine	95.2%
	Random Forest	95%
	Artificial Neural Network	94.7%
PIDD	Deep Neural Network	93.9%
	Kernel Support Vector Machine	94.3%
	Random Forest	93.6%
	Artificial Neural Network	92%
CT	Multilayer Perceptron	84.3%
	Random Forest	81.9%

Once the various *Congruity* values are calculated, the next step is adopting the selected ML/DL model with the corresponding instances. The objective is to demonstrate that a lower *Congruity* value corresponds to a lower *Accuracy* value and vice versa for the learning model.

Figures 3.5-3.7 show the *Accuracy* values achieved by the adopted learning models for every group of instances in the specific *Congruity* range for each considered dataset. It is easy to notice that the increase in *Congruity* corresponds to an *Accuracy* increase.

The final step consists of calculating the correlation between the Accuracy of the ML model and the Congruity values.

As shown in Table 3.6, in the SPD dataset, the best correlation for the **Random Forest** is **85.5%**, **86.4%** for **KSVM**, **86.1%** for **DNN**, and **84.7%** for **ANN**. Analogous results

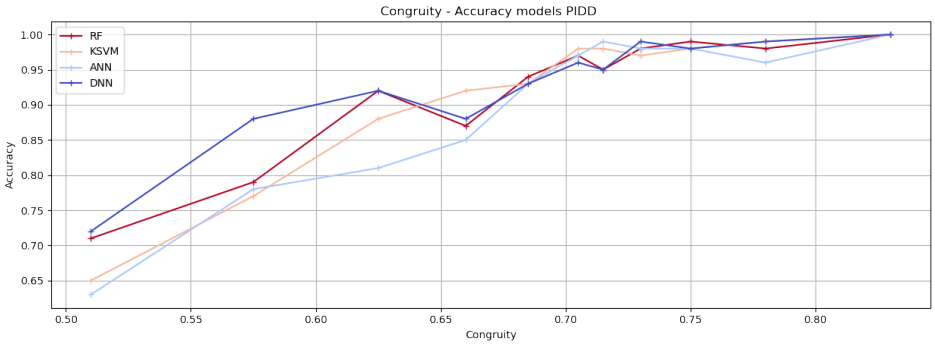

Fig. 3.6 Congruity values range - Accuracy models on PIDD datasets. The chart shows the Accuracy values of the model tested with instances falling within the specific Congruity range. RF refers to Random Forest, KSVM to Kernel Support Vector Machine, ANN to Artificial Neural Network and DNN to Deep Neural Network.

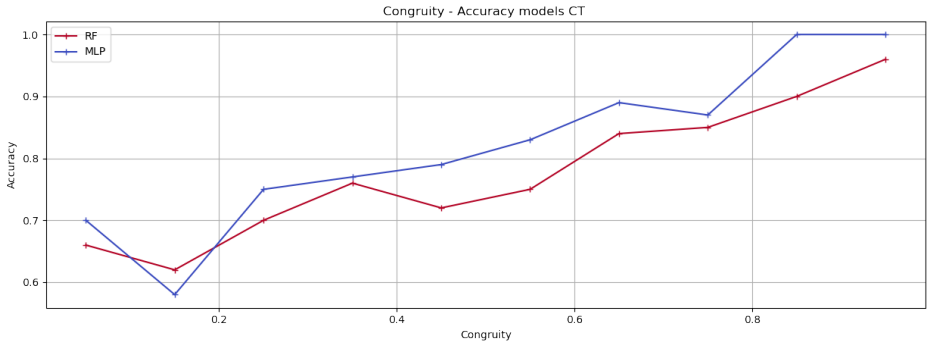

Fig. 3.7 Congruity values range - Accuracy models on CT dataset. The chart shows the Accuracy values of the model tested with instances falling within the specific Congruity range. MLP refers to Multilayer Perceptron and RF to Random Forest.

are achieved through the PIDD dataset. Better results are achieved through the Coronavirus Tweets dataset with a correlation of up to **92%**.

Table 3.6 Correlation Congruity - Accuracy models. - The table shows the Correlation between Congruity and Accuracy results of ML and DL models for respective datasets.

Dataset	Model	Pearson	Kendall	Spearman
SPD	Deep Neural Network	85.2%	83.1%	86.1%
	Kernel Support Vector Machine	85%	82%	86.4%
	Random Forest	84.2%	82.3%	85.5%
	Artificial Neural Network	83.7%	82.5%	84.7%
PIDD	Deep Neural Network	84.9%	82.1%	85.1%
	Kernel Support Vector Machine	83%	81.3%	83.2%
	Random Forest	84%	82.3%	84.5%
	Artificial Neural Network	83.5%	81.8%	84.3%
CT	Multilayer Perceptron	89.4%	87.8%	92.1%
	Random Forest	88%	85.1%	91.5%

Comparison with state of the art approaches. Although to the best of our knowledge, there not exist other similar indexes for measuring the reliability of a learning model, we tried to compare our proposal with an existing similarity measure. The objective is

to understand whether a correlation between the train and test sets similarity and model Accuracy exists. In this sense, the Cosine Similarity has been adopted. Cosine similarity is a metric for comparing two numerical sequences. Sequences are considered vectors in inner product space, and Cosine Similarity is defined as the cosine of the angle between them, defined as the dot product of the vectors divided by the product of their lengths. During experiments, we evaluate the similarity between new instances and instances of the training set. Then, we evaluate the correlation between the mean Cosine Similarity and Accuracy. Table 3.7 shows the results. The reported correlations are lower than those evaluated through our proposed Congruity index. Moreover, a significant execution time is requested.

Table 3.7 Correlation Cosine Similarity - Accuracy models. - The table shows the Correlation between Cosine Similarity and Accuracy results of ML and DL models for respective datasets.

Dataset	Model	Pearson	Kendall	Spearman
SPD	Deep Neural Network	81.3%	78.4%	82%
	Kernel Support Vector Machine	81.8%	75%	82.4%
	Random Forest	80.2%	73.3%	83.5%
	Artificial Neural Network	79.5%	70.5%	82.4%
PIDD	Deep Neural Network	79.3%	72.4%	79.5%
	Kernel Support Vector Machine	83.8%	78.1%	84.6%
	Random Forest	82.2%	76.3%	81.3%
	Artificial Neural Network	77.3%	73.7%	76.4%
CT	Multilayer Perceptron	80.3%	61%	78.2%
	Random Forest	78.4%	57.9%	78.1%

Discussion. From experimentation emerges a strong correlation between the proposed Congruity index and the performance of adopted ML and DL models. It follows that, by knowing the Congruity value, it is possible to deduce the reliability of the training set and, so, of the trained model giving more transparency to anyone who uses the model itself.

By comparing our approach with existing ones, we demonstrate that the Congruity has a higher correlation with the model Accuracy, which can guarantee more relevance during the reliability evaluation of the model. Moreover, although Congruity needs the lattice construction, it is done only the first time; subsequently, the index evaluation is converted into a query to a NoSQL database (i.e., Apache Solr) which quickly returns the best matching lattice concepts. On the contrary, a similarity-based approach (e.g., Cosine Similarity) must be evaluated for each test instance against each instance of the training set, requiring significant processing time.

Some limitations of the proposal regard the lack of experimentation on higher-dimensional datasets like images. In this sense, to reduce the FCA complexity, the literature suggests techniques like clustering or Linear Discriminant Analysis to group common characteristics and reduce the number of Formal Context attributes [94, 103].

3.2 Concept-Drift Detection Index

Machine learning is one of the most widely investigated methods for automatically detecting Fake News [111]. However, most studies ignore the dynamic aspect of the problem. The shift in mistake rates can be used to discern concept deviations. In this sense, various statistical tests, such as the Hoeffeding bound [72], can be used to quantify the change.

Text data streams are adopted to train learning models and discern between real and fabricated content. However, a text data stream differs from others due to sparsity and high dimensionality characteristics. It makes it challenging to train a good and stable classifier [15]; as a result, finding concept incompatibilities in text streams is problematic for several reasons. Classifier performance determines if a concept mismatch is recognized. The variation of classification error rates is considerable when text streams are sparse and high-dimensional, making detection based on error rates erroneous. Moreover, the sources of concept deviations vary, and not all changes in the mistake rate are immediately reflected. Because the change in error rate is gradual, concept deviations can only be identified once a sufficient number of incorrectly predicted cases have accumulated. When concept drifts happen frequently, the misclassified examples are insufficient to detect and are thus overlooked.

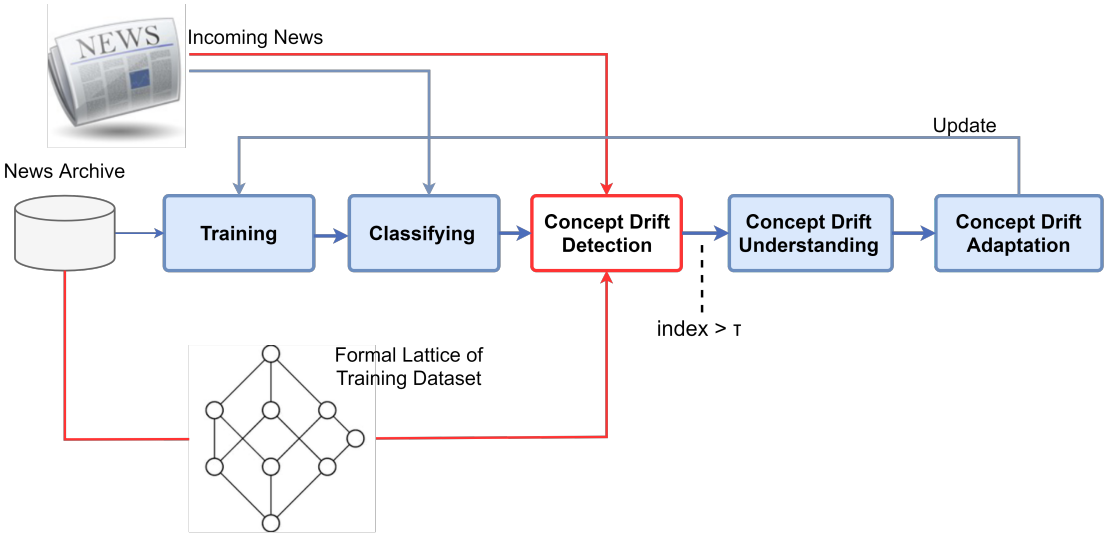

Fig. 3.8 This framework focuses on Concept Drift Detection activity. It proposes an index to detect the intensity of concept drift.

As shown in Figure 3.8 and pointed out in [128], learning under concept drift introduces three new components: Concept Drift Detection (if drift occurs), Concept Drift Understanding (when, how, and where drift occurs), and Concept Drift Adaptation (response to the presence of drift).

This work focuses on the Concept Drift Detection task (red box in Figure 3.8). Given an infinite number of continuous measurements, the question is whether a machine learning model *trained* on past events still works well to *classify* news as fake or not. To answer this question, this work defines a Concept-Drift (or, simply, Drift) Detection Index that estimates the performance of a machine learning model. This index uses the Fuzzy Formal

Concept Analysis (FFCA) technique to track the concepts associated with new incoming instances. It could be compared with a fixed threshold (τ) to establish whether new training needs (i.e., by successive phases of *Concept Drift Understanding* and *Concept Drift Adaptation)*.

Experimental results reveal that the index defined in this work is highly correlated with the machine learning model performance. Furthermore, the index tends to decrease, just like the Accuracy and F-Measure of Machine Learning models, which is lower if the model makes a prediction error because it is not used to recognize certain words, phrases and concepts.

Recently, online Machine Learning models that continuously update and refine a model by incorporating new arriving data revealed good options for avoiding concept drift. Although, they can be expensive in terms of time and resources, requiring continuous monitoring and immediate feedback about the performances. The proposed indicator has also experimented with the Passive-Aggressive classifier, an online Machine Learning model. Tests reveal a high correlation with the quality of the feedback and thus may prevent false news from going viral promptly.

3.2.1 Methodology

Fuzzy Formal Concept Analysis

The proposed measure for concept drift identification is based on the Fuzzy Formal Concept Analysis (FFCA) [74]. FFCA is a formal model that allows aggregating data based on its characteristics, creating a hierarchical structure of concepts and facilitating knowledge processing and extraction operations by leveraging Fuzzy Logic.

The decision to use FFCA to derive a concept Drift Detection Index is based on its ability to compile and conceive data comprehensibly. In particular, comparing new data with existing data (i.e., training set) is made easier regarding time and resources used by concepts with similar qualities (in our case, words). In this way, through the level of representativeness of new data concerning the existing lattice, predicting future learning model performances is straightforward. Moreover, the hierarchy produced by the FFCA allows us to focus on attribute subsets that could be more crucial or representative of the target data. On the contrary, a classical flat similarity measure can only consider the overall attribute set with a loss in data representation understanding and generalization. Finally, although the proposed index requires effort to generate the formal lattice, it refers only to the training stage, usually executed in batch mode. Therefore, at runtime, needed time and resources are considerably low with respect to a similarity measure.

Main definitions of FFCA are given following.

Definition 1: *A* **Fuzzy Formal Context** *is a triple* $K = (G, M, I)$*, where* G *is a set of objects,* M *is a set of attributes, and* $I = ((G \times M), \mu)$ *is a fuzzy set.*

The context can be seen as a "cross table" where formal objects are on the rows, and formal attributes are on the columns. The value $\mu(g,m) \in [0,1]$ in the cell identifies the

membership of the attribute to the object.

Definition 2: Fuzzy Representation of Object. *Each object O in a fuzzy formal context K can be represented by a fuzzy set* $\Phi(O)$ *as* $\Phi(O)$=$\{A_1(\mu_1), A_2(\mu_2),\ldots, A_m(\mu_m)\}$, *where* $\{A_1, A_2,\ldots, A_m\}$ *is the set of attributes in K and* μ_i *is the membership of O with attribute* A_i *in K.* $\Phi(O)$ *is called the fuzzy representation of O.*

The FFCA method may find Fuzzy Formal Concepts and subsumption relationships among them by considering the Fuzzy Formal Context. The following is a more formal definition of the Fuzzy Formal Concept and order relationship.

Definition 3: Given a fuzzy formal context $K = (G,M,I)$ and a confidence threshold χ, for $G' \subseteq G$ and $M' \subseteq M$, we define $G^* = \{m \in M \mid \forall g \in G',\ \mu_I(g,m) \geq \chi\}$ and $M^* = \{g \in G \mid \forall m \in M',\ \mu_I(g,m) \geq \chi\}$.

Definition 4: Fuzzy Formal Concept. *A fuzzy formal concept (or fuzzy concept) C of a fuzzy formal context K with a confidence threshold* χ, *is* $C = (I_{G'}, M')$, *where, for* $G' \subseteq G,\ I_{G'} = (G', \mu), M' \subseteq M, G^* = M'$ *and* $M^* = G'$. *Each object g has a membership* $\mu_{I_{G'}}$ *defined as*

$$\mu_{I_{G'}}(g) = min_{m \in M'}\left(\mu_I(g,m)\right) \tag{3.3}$$

where μ_I *is the fuzzy function of I.*

Definition 5: *Given two concepts* $C_1 = (A_1, B_1)$ *and* $C_2 = (A_2, B_2)$, *then* C_1 *is a subconcept of* C_2 *(equivalently,* C_2 *is a superconcept of* c_1*),* $(A_1, b_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 (\Leftrightarrow B_2 \subseteq B_1)$. *The set of all concepts of a particular context, ordered in this way, forms a complete lattice.*

Each lattice node (i.e., a fuzzy formal concept) keeps objects and the associated set of attributes. Furthermore, for the resulting Fuzzy Formal Concepts exists a subconcept relation $\leq$ that arises a *Fuzzy Lattice* of Fuzzy Formal Concepts.

Definition 6: ***Fuzzy Formal Concept Support***. *Let* $K = (G,M,I)$ *be a fuzzy formal context, the support of a Fuzzy Formal Concept* $C' = (I_{G'}, M')$ *is given by*

$$Supp(C') = \frac{|G'|}{|G|} \tag{3.4}$$

Support measures the frequency of objects belonging to the same lattice concept.

Concept-Drift Detection Index based on Fuzzy Formal Concept Analysis

The proposed methodology mainly consists of leveraging the Fuzzy Formal Lattice to evaluate the degree of representativeness (i.e., the Drift Detection Index) of the training set with respect to new items at the test stage. Thus, the Drift Detection Index is correlated

with learning model performances to demonstrate its validity. Summing up, as shown in Figure 3.9, two main activities are carried out:

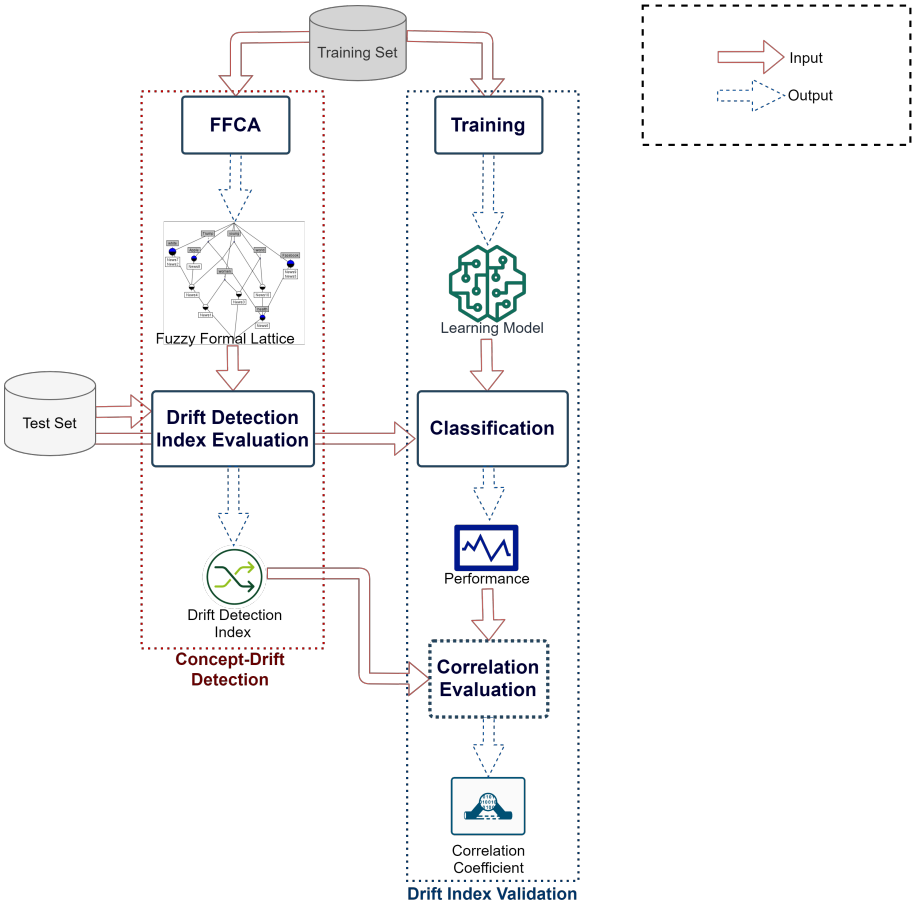

Fig. 3.9 Process of Concept-Drift Detection and Index validation. The general process consists of two main activities, Concept-Drift Detection and Drift Index Validation. In Concept Drift Detection, the Fuzzy Formal Concept analysis is used to build the training lattice to measure the Drift Index of new instances from the Test Set. In Drift Index Validation, a model is trained and evaluated with the same training and test set used previously; after this, the correlation between the new index and the model's Accuracy is calculated to obtain the correlation coefficient.

1. Concept-Drift Detection, which corresponds to the activity in the red block of Figure 3.8. In this stage, through the FFCA, a fuzzy lattice is extracted to represent training data. Then, it is adopted to measure the Drift Index associated with new instances (i.e., test set).

2. Drift Index Validation. This stage is not directly connected with the proposal but must demonstrate its validity. The objective is to evaluate whether a correlation between the introduced Drift Index and a classifier trained on the same lattice dataset exists.

So, the overall process of Drift Detection Index evaluation and validation is the following:

1. **FFCA and Training.** This phase consists of two parallel sub-phases. The first one computes the FFCA algorithm from the dataset subsequently used to train the learning model. On the one hand, the objective is to construct the fuzzy formal lattice functional to calculate the Drift Detection Index. On the other hand, one or more models (classifiers) are trained for subsequent predictions.

2. **Drift Detection Index Evaluation.** The test set is considered for Drift Detection Index evaluation through the method described in Section 3.2.1.

3. **Classification.** In this phase, the test set, consisting of a period subsequent to the training, is adopted to evaluate the adopted model(s) performance. In particular, Accuracy and F-Measure are considered.

4. **Correlation evaluation.** In the final stage, the correlation between the Drift Detection Index and the Accuracy and F-Measure values is calculated in order to validate the proposed index. In particular, the correlation is evaluated through Pearson's [42] and Spearman's [64] correlation coefficients.

The following sections detail the proposed Drift Detection Index evaluation.

Notation. The following is the basic notation used in the next subsections for defining the Concept-Drift Detection Index, as defined also in Notation of 3.1.1

- $P(C,X)$ - given an input instance X and a concept C, the Precision P is evaluated as:

$$P = \frac{|Attr(X) \cap Attr(C)|}{|Attr(C)|} \tag{3.5}$$

- $R(C,X)$ - given an input instance X and a concept C, the Recall R is evaluated as:

$$R = \frac{|Attr(X) \cap Attr(C)|}{|Attr(X)|} \tag{3.6}$$

- $F-Measure(C,X)$ - the combination of Precision and Recall evaluated between instance X and concept C is given by the following equation:

$$F-Measure = 2 * \frac{P * R}{P + R} \tag{3.7}$$

Concept-Drift Detection Method. FFCA conceptualizes input data and generates a hierarchical knowledge structure (i.e., fuzzy formal lattice L). The drift occurring is identified by an index through the function $c(L,X)$ that takes as input the lattice L and a new data item, i.e., the instance X. The index, which represents qualitatively and quantitatively the modeling of the new input instance with the ideas already existing in the extracted lattice, is conceptually a measure of the representativeness of X in L. Essentially, two pieces of information are crucial:

- how many items with the same characteristics values lie in the sample set during the lattice construction;
- the degree to which concepts in the lattice L cover features of the new instance.

The effectiveness of a classifier in machine learning depends on how representative the sample used to train the model is of the new instances that need to be classified. In this way, the suggested methodology assesses how the lattice is representative of new items. The method value is evaluated by analyzing the correlation between the Drift Detection Index and a reliability metric for the classifier output (Accuracy and F-Measure). The Drift Detection Index makes it possible to determine in advance whether the machine learning model is experiencing drift and needs to be retrained.

The method worth is evaluated by measuring the correlation between the Drift Detection Index and a reliability measure for the classifier output (i.e., Accuracy and F-Measure). The Drift Detection Index allows saying a priori whether the machine learning model is undergoing a drift process and thus must be retrained.

Starting from the previous intuitive definition of Drift Detection Index, with a new input object, there are two aspects to consider:

- the *Support* of the concept(s) ($S(C)$) in which the new input instance would fall;
- the *coverage* of the input instance with respect to attributes in the available concepts. According to [53], the coverage could be the degree of matching calculated as the F-measure of new instance characteristics for lattice concepts.

Thus, the Drift Detection Index could be a linear convex combination of the *support* and *f-measure* values between the characteristics of the concepts with a set of attributes whose intersection with the attributes of the new instance is non-empty. More formally, given a new instance characterized by the attribute set $Attrs(X) = \{a_1, a_2, ..., a_n\}$ there could be two cases concerning the existing lattice L:

1. There are one or more concepts in the lattice, $\widetilde{C} = \{\widetilde{C}^1, \widetilde{C}^2, ..., \widetilde{C}^k\}$ whose intersection $Attr(X) \cap Attr(\widetilde{C}_i) \neq \emptyset$ for $i = 1, 2, ..., k$, and then the following further subcases:
   - Among all concepts in $\widetilde{C}$, there exists $\widetilde{C}^*$ such that Attrs($\widetilde{C}^*$) coincides with Attrs(X); in that case, the F-Measure($\widetilde{C}^*, X$) shall be maximum (i.e., equal to 1), and the Drift Detection Index will only have to consider the support of the concept $\widetilde{C}^*$. Assuming Drift Detection Index is a linear convex combination of the support and f-measure values with weights α_1 and α_2, then:

$$\alpha_1 S(\widetilde{C}^*) + \alpha_2 F - Measure(\widetilde{C}^*, X) = \alpha_1 S(\widetilde{C}^*) + \alpha_2 \tag{3.8}$$

- In other cases, Drift Detection Index $c(L,X)$ can be calculated as:

$$c(L,X) = \frac{1}{|\widetilde{C}|} \sum_{\widetilde{C^i} \in \widetilde{C}} \alpha_1 S(\widetilde{C^i}) + \alpha_2 \; F - Measure(\widetilde{C^i}, X) \tag{3.9}$$

2. There is no concept C^* such that the intersection $Attr(X) \cap Attr(C_i^*) \neq \emptyset$. In this case, the Drift Detection Index is $c(L,X) = 0$.

Example of use. Suppose considering 10 news. By applying TF-IDF (Term Frequency - Inverse Document Frequency), we extract our dataset term vector and produce the fuzzy formal context like the one in Table 3.8. The corresponding fuzzy lattice is one in Figure 3.10. To each considered news, TF-IDF is applied and represented as a membership value. Then a threshold of 0.6 is used to derive the fuzzy lattice. Assume that, after lattice construction, two additional news (i.e., $News_{11}$ and $News_{12}$) arrive (see Table 3.9).

	Trump	**White**	**Young**	**Women**	**World**	**Apple**	**Facebook**	**Health**
$News_1$	**0.71**	**0.63**	**0.80**	**0.61**	0.48	**0.68**	0.32	0
$News_2$	0	**0.72**	0.15	0	0	0	0.23	0.03
$News_3$	0.22	0.21	0	**0.67**	**0.86**	0	0.21	0
$News_4$	**0.80**	**0.72**	**0.68**	0	0.10	**0.70**	0	0.21
$News_5$	0.09	0	0	0.12	0.14	0	**0.77**	0
$News_6$	**0.62**	0.33	**0.74**	0	**0.72**	0	**0.88**	**0.90**
$News_7$	0	**0.65**	0	0	0	0	0.11	0.08
$News_8$	**0.66**	0.12	0	0	0.11	**0.81**	0	0
$News_9$	0	0.25	0.45	0	0	0.12	**0.80**	0
$News_{10}$	0	0.19	**0.68**	0	**0.77**	0.27	0	0

Table 3.8 Fuzzy Context example

Objects	**Attributes**
$News_{11}$	TRUMP, WHITE, YOUNG, APPLE
$News_{12}$	COVID, PANDEMIC, YOUNG, VIRUS, HEALTH

Table 3.9 News feature vectors

In $News_{11}$, the most important words are {TRUMP, WHITE, YOUNG, APPLE}. According to the Drift Detection Index definition, there is a fuzzy concept (i.e., $C5$) whose attribute set coincides with the input one. Since the F-Measure is 1 and the Support of C_5 0.1, assuming weights of 0.9 and 0.1 for F-Measure and Support, respectively, it follows that the Drift Detection Index is equal to:

$$0.1 * 0.1 + 0.9 * 1 = 0.01 + 0.9 = 0.91 \tag{3.10}$$

Now, suppose to have a second news item, $News_{12}$, which attributes are the following: {COVID, PANDEMIC, YOUNG, VIRUS, HEALTH}. In this case, since there is not a perfectly matching concept, we must consider each concept with a no empty attribute

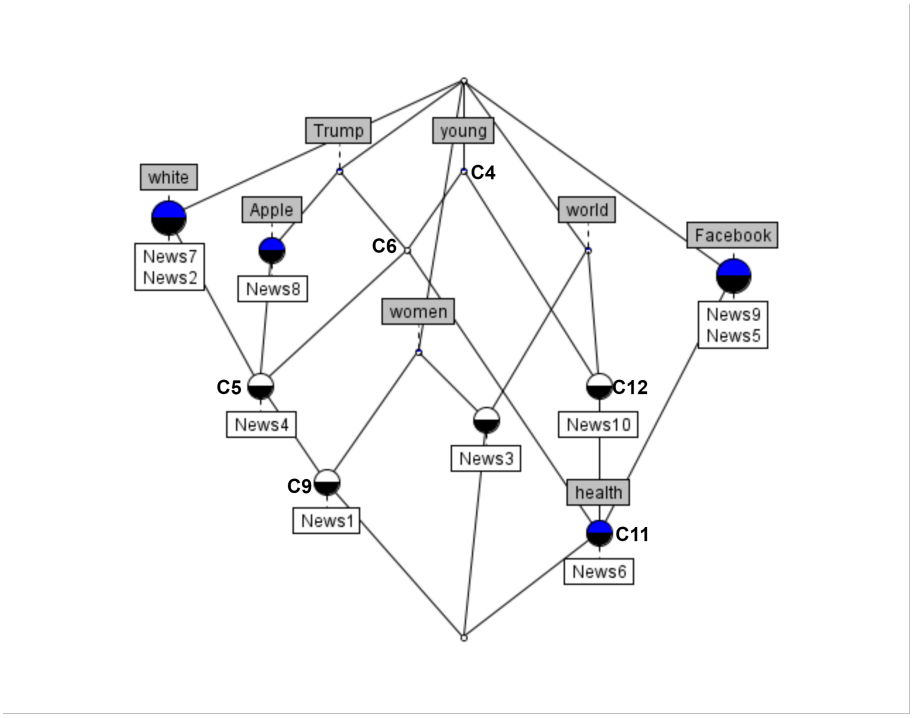

Fig. 3.10 Fuzzy Lattice corresponding to the fuzzy formal context in Table 3.8.

intersection with the input set (i.e., $\{C4, C5, C6, C9, C11, C12\}$). Table 3.10 resumes computed measures.

Concept	Precision	Recall	F-Measure	Support
C4	1	0.2	0.33	0
C5	0.25	0.2	0.22	0.4
C6	0.5	0.2	0.29	0
C9	0.2	0.2	0.2	0.5
C11	0.4	0.4	0.4	0.4
C12	0.33	0.2	0.25	0.1

Table 3.10 Measures Evaluation for $News_{12}$

By combining measures in Table 3.10, the Drift Detection Index results as follows:

$$(0.3 + 0.24 + 0.26 + 0.23 + 0.4 + 0.24)/6 = 0.28 \tag{3.11}$$

This example shows that news items represented by more frequent words in the lattice have a significantly higher Drift Detection Index value. On the contrary, a low index value indicates that the training set does not amply represent a test instance.

3.2.2 Experimentation

This section describes experimentation conducted to validate the proposed index on three public datasets and discusses the resulting correlation between the Drift Detection Index and the ML model classification Accuracy and F-Measure. In particular, three different machine learning models were used for experimentation: Random Forest (RF) with 100

trees, *entropy criterion*, and $random\_state = 0$. Naive Bayes (NB) with $alpha = 1$ and $fit\_prior = True$. The last classifier is the Passive-Aggressive (PA) with $max\_iter = 1000$, $random\_state = 5$ and $tol = 1e^{-3}$.

Tools. The main tools adopted during the experimentation are a custom-developed library implementing the Fuzzy Formal Concept Analysis (FFCA) derived by extending "Colibri"[8]; Apache Solr[9], used for indexing concepts hierarchy extracted by FFCA; the sci-kit learn library[10], used to construct and adopt the Random Forest, Naive Bayes and Passive Aggressive classifiers, and matplotlib[11] for calculating correlations.

Datasets. Considered datasets consist of large and public sets of news published over a long period:

- NELA-GT-2018 [146]: a dataset including 713k items collected between February and December 2018 to investigate misinformation. These pieces were gathered from 194 news and media outlets, including mainstream, hyper-partisan, and conspiracy sites.
- NELA-GT-2019 [77]: comprises 1.12 million media articles gathered from 260 sources between January 1st and December 31st, 2019. These sources, like NELA-GT-2018, are derived from various mainstream and alternative news sources.
- NELA-GT-2020 [78]: between January 1st, 2020 and December 31st, 2020, almost 1.8 million news stories were collected from 519 sources. These sources, like those for NELA-GT-2018 and NELA-GT-2019, originate from a variety of mainstream and alternative news sources. In addition, this dataset includes a portion of news concerning Covid-19 and a subset of articles about the 2020 US presidential election.

There is no missing data in the three datasets. News from these datasets is labeled as unreliable, mixed, or reliable. Experiments are performed only with unreliable or reliable news.

Data preparation. Three selected datasets have been merged. They were broken down monthly after labeling them as reliable or unreliable, thanks to the sources made available by the authors of the dataset themselves. All news items from 01/02/2018 to 31/08/2018 were used for the training set and, thus, for creating the lattice. News items from 01/09/2018 to 31/12/2020 were grouped monthly and used as a test set. TF-IDF [157] was applied to all data, allowing the identification of essential words for each document and creating the attributes. The least influential words were eliminated, and the lattice was created through the training set. Let us note that the various processes were done separately so that words from later periods could not influence earlier ones to calculate the TF-IDF score.

[8] https://code.google.com/p/colibri-java/
[9] https://en.wikipedia.org/wiki/Apache_Solr
[10] https://scikit-learn.org/stable/
[11] https://matplotlib.org/

Once all data was collected, it was possible to determine the Drift Detection Index for each instance of the test set and obtain a monthly score calculated as an average.

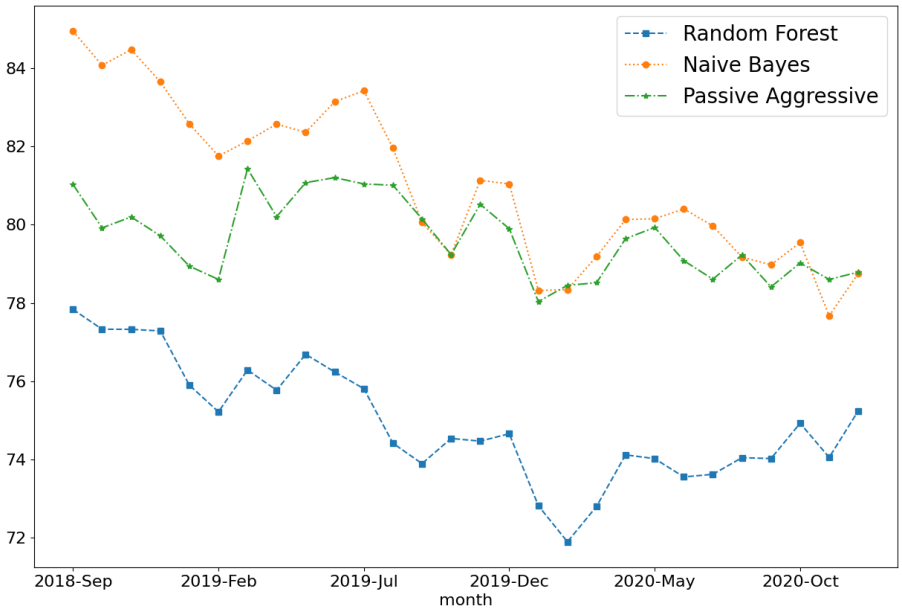

Fig. 3.11 Machine Learning Models Accuracy Trends. The accuracy performances of machine learning models decrease over time which is undoubtedly exacerbated by the arrival of news and, as a result, phrases that the classifier cannot recognize, such as “covid”, “pandemic”, and anything else unrelated to the first months of 2018.

Experimental Results. Table 3.11 shows the models Accuracy (ACC) and F-Measure (F1) and the corresponding Drift Detection Index values. As also shown in Figure 3.11, there is a decrease in the performance of the machine learning models, which is undoubtedly accentuated by the arrival of news and, therefore, words that the classifier cannot recognize, such as “covid”, “pandemic”, and everything else not related to the first months of 2018. It is worth noting that all three models hit their lowest Accuracy precisely in February 2020, when the pandemic reached popularity, demonstrating a flaw in classification performance due to a lack of training data. The Pearson’s correlation coefficient, evaluated between Drift Detection Index and Accuracy values, as depicted in Figure 3.12 and Table 3.12, reaches **80.8%** when assessed for the Naive Bayes Classifier, **81.0%** for the Passive-Aggressive, and **73.9%** for Random Forest. Moreover, Table 3.12 highlights the correlation between the F-Measure of the various models and the Drift Detection Index that, in the case of Naive Bayes, comes to be **87.3%**. On the other hand, Table 3.13 shows the correlation results using Spearman’s correlation coefficient with slightly lower but still significant results. Identified correlation is particularly evident in Figure 3.12: a decreasing Accuracy value corresponds to a decrease in the Drift Detection Index over time.

Passive Aggressive Classifier requires deeper discussion. It belongs to the family of online learning classifiers, known to not suffer from concept drift or model obsolescence and recently widely applied for fake news detection objectives. Nevertheless, one of the major drawbacks of an online learning algorithm is that the system performs poorly when trained with inaccurate data, with an immediate impact on the user. As a result, it

Month	PA		RF		NB		Drfit Indicator
	ACC	F1	ACC	F1	ACC	F1	
09/2018	83.0%	76.6%	77.8%	68.0%	84.9%	75.8%	75.2%
10/2018	81.9%	75.2%	77.3%	67.2%	84.1%	75.3%	73.1%
11/2018	82.2%	75.5%	77.3%	66.8%	84.5%	75.1%	70.1%
12/2018	81.1%	75.3%	77.3%	68.0%	83.7%	75.6%	74.6%
01/2019	80.5%	75.7%	75.9%	67.6%	82.6%	76.1%	71.9%
02/2019	80.6%	76.0%	75.2%	67.7%	81.8%	75.4%	69.2%
03/2019	82.8%	78.8%	76.3%	69.1%	82.1%	76.5%	66.0%
04/2019	81.7%	76.9%	75.8%	67.7%	82.6%	76.5%	70.4%
05/2019	82.7%	78.2%	76.7%	69.2%	82.4%	77.0%	76.0%
06/2019	82.6%	76.9%	76.2%	66.8%	83.1%	75.9%	74.0%
07/2019	81.6%	76.8%	75.8%	66.9%	83.4%	76.6%	69.1%
08/2019	81.6%	78.5%	74.4%	67.7%	82.0%	76.3%	69.4%
09/2019	81.1%	78.5%	73.9%	68.5%	80.0%	75.3%	67.3%
10/2019	80.3%	76.4%	74.5%	67.8%	79.2%	74.9%	75.6%
11/2019	81.0%	75.0%	74.5%	64.4%	81.1%	73.0%	59.6%
12/2019	80.8%	74.5%	74.7%	64.4%	81.0%	72.2%	55.7%
01/2020	78.8%	73.1%	72.8%	65.7%	78.3%	72.0%	48.4%
02/2020	79.2%	73.5%	71.9%	65.1%	78.3%	71.9%	51.7%
03/2020	78.7%	74.5%	72.8%	64.8%	79.2%	70.2%	54.6%
04/2020	79.9%	75.1%	74.1%	66.1%	80.1%	71.9%	55.1%
05/2020	80.1%	75.6%	74.0%	66.1%	80.2%	72.8%	52.5%
06/2020	79.8%	74.9%	73.6%	66.6%	80.4%	73.8%	56.6%
07/2020	79.2%	74.6%	73.6%	66.8%	80.0%	72.9%	54.4%
08/2020	79.7%	75.2%	74.0%	67.0%	79.2%	71.8%	44.0%
09/2020	79.5%	74.6%	74.0%	64.9%	79.0%	71.7%	52.1%
10/2020	79.4%	73.6%	74.9%	65.1%	79.6%	71.6%	49.3%
11/2020	79.1%	73.0%	74.1%	64.0%	77.7%	69.0%	50.2%
12/2020	79.7%	74.3%	75.2%	64.7%	78.8%	69.9%	51.3%

Table 3.11 Drift Detection Index - Machine Learning Models Performance

Indicator	PA		RF		NB	
	ACC	F1	ACC	F1	ACC	F1
Cosine Similarity	16.5%	−8.3%	16.9%	−11.2%	25.4%	10.5%
Drift Detection Index	**81.0%**	**70.3%**	**73.9%**	**73.6%**	**80.8%**	**87.3%**

Table 3.12 Pearson's Correlation coefficients

Indicator	PA		RF		NB	
	ACC	F1	ACC	F1	ACC	F1
Cosine Similarity	0.1%	0.0%	19.6%	0.1%	18.7%	15.5%
Drift Detection Index	**79.9%**	**72.8%**	**69.6%**	**70.8%**	**80.5%**	**81.1%**

Table 3.13 Spearman's Correlation coefficients

is critical to implement proper filters to ensure that the data fed is of good quality. In addition, it is crucial to monitor the performance of the machine learning system in a very close manner. In this sense, due to the demonstrated significant correlation, the proposed indicator can be combined with an online learning classifier to give immediate feedback about its performance.

Cosine Similarity. To test the efficiency of the new indicator, the same type of experimentation was conducted with the Cosine Similarity [210]. Cosine similarity is a metric used to measure how similar documents are, regardless of their size. Mathematically, it calculates the cosine of the angle between two vectors projected in a multidimensional space. The idea consists of evaluating the representativeness of a new instance with regards to the training set through their similarity and then measuring its correlation with learning models Accuracy and F-Measure.

The similarity is evaluated among the vector representation of news (i.e., words) by months. In this case, the results (see Table 3.14) showed a poor correlation between the Machine Learning models Accuracy and F-Measure. In particular, as shown in Tables 3.12 and 3.13, Pearson's correlation with the Accuracy of Naive Bayes was 25.4%, 16.9% with the Random Forest, and even 16.5% with the Passive-Aggressive. In some cases, Pearson's correlation with the F-Measure of these models is negative. Results are not better using Spearman's correlation, confirming the worth of the proposed index at the expense of the existing measure.

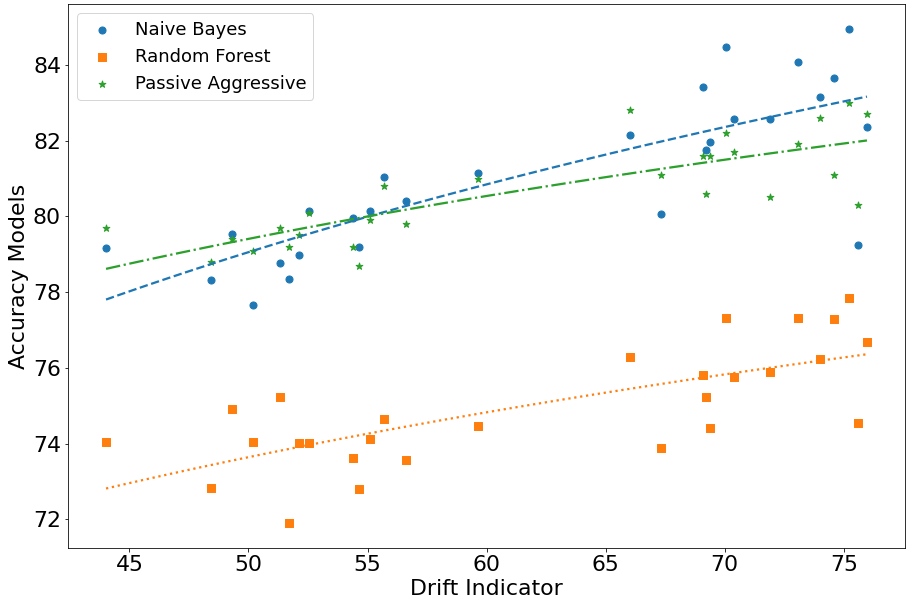

Fig. 3.12 Pearson's Correlation between Machine Learning models Accuracy and Drift Detection Index. In the figure, it is straightforward to notice that a decreasing Accuracy value corresponds to a decrease in the Drift Detection Index over time.

Month	Cosine Similarity
09/2018	45.2%
10/2018	44.3%
11/2018	48.1%
12/2018	49.6%
01/2019	45.8%
02/2019	45.1%
03/2019	41.8%
04/2019	45.0%
05/2019	46.4%
06/2019	53.2%
07/2019	48.3%
08/2019	47.3%
09/2019	42.7%
10/2019	41.3%

Month	Cosine Similarity
11/2019	43.9%
12/2019	44.1%
01/2020	46.8%
02/2020	50.3%
03/2020	44.9%
04/2020	44.6%
05/2020	45.1%
06/2020	42.6%
07/2020	44.1%
08/2020	42.2%
09/2020	46.0%
10/2020	43.7%
11/2020	46.3%
12/2020	46.4%

Table 3.14 Cosine Similarity by month.

Chapter 4

xAI 4 countering Information Disorder

In the contemporary landscape of digital communication, the proliferation of false information poses a significant threat to societal stability and trust in public discourse. This chapter focus into the potential of xAI to enhance systems designed to counteract Information Disorder. Three works are presented to support the following Research Question: "How xAI can help to improve systems countering Information Disorder?"

The first work was been developed by partecipating in the International Competition SemEval 2023 Task 3 "Detecting the Genre, the Framing, and the Persuasion techniques in online news in a multilingual setup". In subtask 1, "News Genre Categorisation", the goal was to classify a news article as an opinion, a report, or a satire. In subtask 3, "Detection of Persuasion Technique", the system must reveal persuasion techniques used in each news article paragraph choosing among 23 defined methods. Solutions leverage the application of xAI method, SHAP. In sub-task 1, SHAP was used to understand what was driving the model to fail so that it could be improved accordingly. In contrast, in subtask 3, a re-calibration of the Attention Mechanism was realized by extracting critical tokens for each persuasion technique. The underlying idea is the exploitation of xAI for countering the overfitting of the resulting model and attempting to improve the performance when there are few samples in the training data. This work is presented in Section 4.1

The second work exploits a GNN to make link predictions on a graph representing information about misinformation tweets, their authors, and their spread. The objective is to comprehensively investigate the specific attributes of online pathways that compel users to share and amplify inaccurate information. In this sense, starting from an existing dataset of misinformation tweets, the proposed approach first applies an explainability method to each prediction, then, through frequent itemset mining, tries to detect patterns among collected explanations. Results of qualitative and quantitative research questions mainly demonstrate the contribution of interpersonal aspects to misinformation tweets spreading.This work is presented in Section 4.2

The third work presents the Hybrid Fact-Checking Framework (HFCF) as a DNFS tailored to address the uncertainty inherent in fact verification tasks and enhance the reliability of model responses. The proposed DNFS integrates an LLM and an Adaptive Neuro-Fuzzy Inference System (ANFIS) for automated fact verification. The framework

utilizes relevant evidence from open-world and closed-world sources to generate and justify verdicts by leveraging deep language models and employing few-shot prompting without additional training. Including fuzzy rules and considering the trustworthiness and relevance of retrieved evidence enhances response reliability, thereby improving overall effectiveness and outcome interpretability. Experimental validations have been conducted on three publicly available datasets ranging in different domains of expertise: Climate-FEVER, SciFact, and FEVER. This work is presented in Section 4.3.

Published contributions are the following:

- Bangerter, M., Fenza, G., Gallo, M., Loia, V., Volpe, A., De Maio, C., & Stanzione, C. (2023, July). Unisa at SemEval-2023 task 3: a shap-based method for propaganda detection. In Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023) (pp. 885-891).

- Capuano, N., Fenza, G., Gallo, M., Loia, V., & Stanzione, C. (2023, December). Unfolding the Misinformation Spread: An In-Depth Analysis Through Explainable Link Predictions and Data Mining. In International Conference on Intelligent Systems Design and Applications (pp. 137-146). Cham: Springer Nature Switzerland.

- Bangerter, M. L., Fenza, G., Furno, D., Gallo, M., Loia, V., Stanzione, C., & You, I. (2024). A Hybrid Framework Integrating LLM and ANFIS for Explainable Fact-Checking. IEEE Transactions on Fuzzy Systems.

4.1 A SHAP-based method for Propaganda Detection

The main drive behind the SemEval 2023 Task 3 [154] is to foster the development of methods and tools to support the analysis of online media content to understand what makes a text persuasive: which writing style is used, what critical aspects are highlighted, and which persuasion techniques are used to influence the reader. Persuasion attempts and propaganda (through different ways), appeal to the user's sentiment to influence his/her opinions. Unfortunately, detecting propaganda and persuasion techniques are trending research topics that still perform poorly, especially when labeled data is limited [45, 37]. Our approach tries to overcome problems related to the limited number of training instances by generalizing data through the SHAP [129] method, an xAI technique. SHAP is a framework trying to explain the prediction of an instance x by computing the contribution of each feature to the prediction. The SHAP explanation technique uses coalitional game theory to compute Shapley values. In subtask 1, SHAP is employed for feature selection [65]. The idea is to leverage features to understand how to pre-process documents before sending them to the model. In subtask 3, SHAP re-calibrates the Attention Mechanism for each persuasion technique. The idea consists of identifying words characterizing each type of persuasion and helping the learning model to focus on them.

4.1.1 Methodology

The methodology of the work is presented below.

System Overview

The system adopts the DistilBert transformer [170] and the SHAP method. DistilBert is a compact version of BERT, retaining about 97% of its performance with significantly fewer parameters. It uses knowledge distillation to achieve faster, more efficient NLP processing while reducing model size. The following subsections detail how they are combined to address the subtasks 1 and 3 included in SemEval-2023 Task 3: "Detecting the Genre, the Framing, and the Persuasion techniques in online news in a multi-lingual setup".

More in detail, subsection 4.1.1 points out that for subtask 1, SHAP was used to understand what tokens drive the classification of the article. In contrast, in subsection 4.1.1, for subtask 3, SHAP was adopted to re-calibrate the Attention Mechanism by leveraging words that distinguish each persuasion technique and help the model to generalize better.

News Genre Categorisation. The English training dataset contains unbalanced classes, with minimal satire articles despite many opinion pieces. So, to improve the classification performance, the sub-task was approached as follows. First, data augmentation was made by translating articles in other languages to the target language (i.e., English) for the least represented classes (i.e., reporting and satire). Our corpus was finally the combination of the translated articles with the original English ones. These articles were first passed to a preprocessing phase, where non-redundant tokens were cleaned to reduce the article's noise and avoid losing more information for the reduced amount of tokens available to pass to the transformer. Then, the cleaned articles were passed to the DistilBert model for a sentence-level classification task taking as input article words for a maximum of 512 tokens, as allowed.

The model is fine-tuned for the specific task by adding additional classifiers on top of the pre-trained DistilBert model. After the model was trained for the first time, we used SHAP to understand what input tokens had more influence on the model for the article's final (wrong) classification. This process produces a black-list of tokens to filter out that may adjust the cleaning function and reduce the effect of the limited number of allowed tokens. Finally, once the preprocessing function is defined, we fine-tune the model again with these adjusted articles, and the final model is then used to predict new coming articles.

Figure 4.1 presents the workflow of classifying a new incoming article. First, the tokenization adds special tokens to the first 512 tokens of the input (the first tokens resulting from filtering ones identified by SHAP) and passes them to the trained model. The idea is to remove the tokens guiding the model to a wrong classification and then adopt the first 512 tokens of the input for the classification. Finally, we pass the logits made by the model to a softmax function and choose the most probable class.

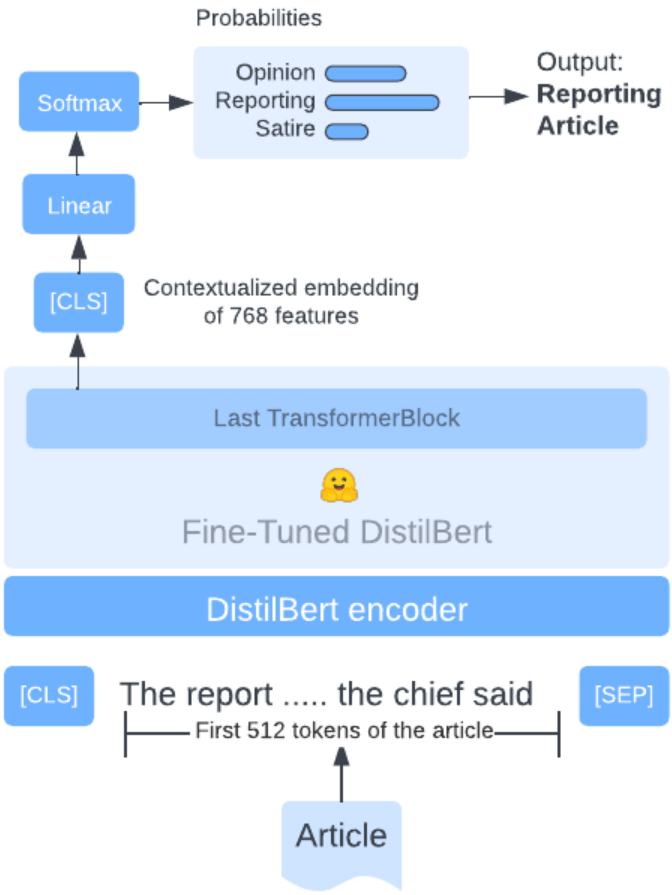

Fig. 4.1 Sub-task 1 - Prediction of new coming input

Persuasion Techniques Detection. Regarding subtask 3, the overall system is depicted in Figure 4.2. A binary model first processes a new incoming paragraph to predict whether it contains any persuasion attempt (i.e., is classifiable as "Persuasion"). If the text is predicted to be persuasion, it is compared with *SHAP Vocabularies* previously created, representing the most important words associated with each persuasion technique. Such comparison defines the additional input to pass to the final multiclass model that will establish the probability by which the input text is classified to each persuasion technique. Classes with probabilities exceeding a fixed threshold will be part of the final multilabel classification.

Following subsections detail the processes of training (i.e., learning model preparations) and testing (i.e., learning model adoptions).

Training Phase. The training process can be, in turn, divided into two main stages: the first builds SHAP Vocabularies; the second is aimed at training two DistilBert Transfomers (i.e., the Binary Persuasion Classifier and the Multiclass Persuasion Classifier).

Constructing SHAP Vocabularies consists of extracting the essential words characterizing each persuasion technique by exploiting the SHAP method. The process consists of constructing N DistilBert binary classifiers (one for each persuasion technique) and

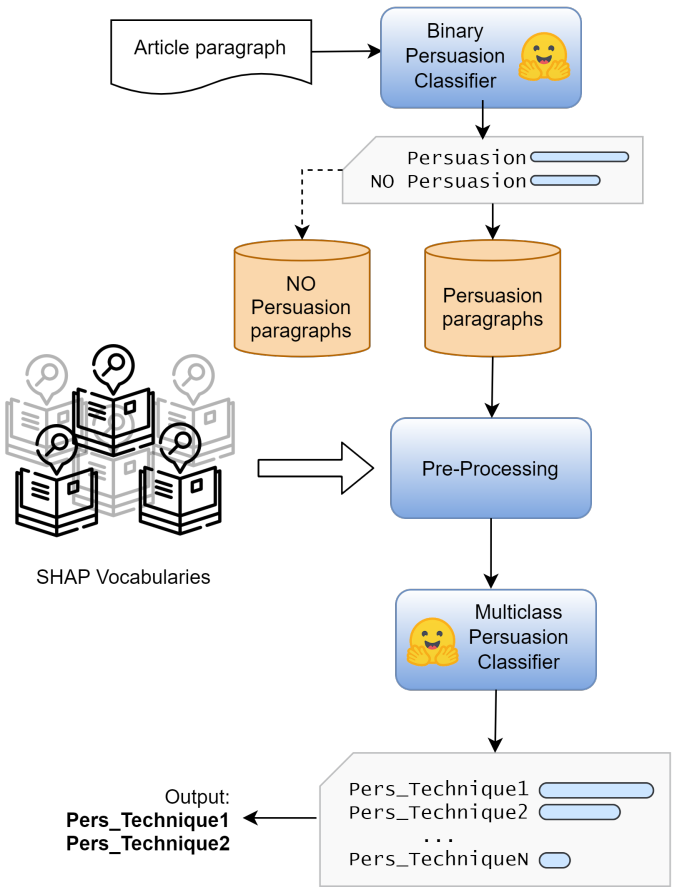

Fig. 4.2 Multi-label classification for Subtask3

exploiting SHAP to identify the most important words (i.e., ones that guided the classification decision). For dataset construction, rows with more than one label are split to obtain rows with only one label. Then, for each technique, a positive sample dataset with (M) rows belonging to specific techniques and a negative sample dataset containing M random samples belonging to other $N-1$ techniques are constructed. Validation and testing subsets were extracted from datasets created for each technique. Thus, a binary DistilBert Transformer was trained and tested for each technique. The registered Accuracy for these classifiers is about 70%, on average. Once the predictions were obtained, ones corresponding with the starting persuasion technique were filtered and applied SHAP. Based on Shaply Values, we construct N vocabularies of main words distinguishing each technique.

In the second training stage, two DistilBert Transformers are trained: a Binary Persuasion Classifier establishes if the text can be considered persuasion in general; a Multiclass Persuasion Classifier assigns to the input a probability of belonging to each type of persuasion. The choice of adding a separate classifier to first identify text containing persuasion attempts allows for improving the performance of the subsequent multi-class classifier. In fact, as described in Section 4.1.2, paragraphs not containing persuasions are more than half. To train the model, paragraphs in the Train and Dev Set that featured at least one technique, out of the available 23, were associated with the label “Persuasion”, while

paragraphs that did not feature any technique were labeled as "NO Persuasion". During the test of this classifier, registered Accuracy was 84%.

The second adopted model is a DistilBert Multiclass single-label Transformer, trained as follows. First, all rows with more than one label are split to obtain rows with only one label. The insight, in this case, was to provide the model with the part of the text on which to focus attention most, thus going on to recalibrate the attention mechanism. In particular, after the tokenization of input text, each word was compared with each word contained in the N SHAP dictionaries. Those words that were found to have a similarity of at least 90% with one of the words contained in the SHAP dictionaries were considered essential. Such words become part of the input paragraph as follows:

$$\begin{gathered}[CLS]\ token_1\ token_2\ \ldots\ token_n\ [SEP] \\ paragraph\ [SEP]\end{gathered} \tag{4.1}$$

where $token_1, token_2, \ldots\ token_n$ are words found to be fundamental after comparison with SHAP dictionaries.

Testing Phase. During the test phase, for each incoming instance classified as "Persuasion", the model returns a belonging probability for each persuasion technique. So, to move such type of classification to a multi-label result, all labels with a probability greater than 0.15 are included in the resulting classification. Such threshold is selected empirically.

4.1.2 Experimentation

The experimentation conducted is presented in the following.

Experimental setup

Methodologies described in Section 4.1.1, have been applied to the dataset provided by the SemEval-2023 Task 3 organizers, specifically for the English subtask.

The following subsections describe the experimentation setups and results for both considered subtasks. In particular, results have been evaluated through official measures: macro-F1 and micro-F1 for subtasks 1 and 3, respectively, as described in [154]. Furthermore, the number of epochs used in each subtask was defined with a maximum of 6 as we train LMs; if no improvements are made, fewer epochs are used.

News Genre Categorisation. Regarding Subtask 1, the English training dataset contains 433 news and web articles, while the development set a total of 83 instances. Finally, 54 articles were released for the test set without known gold labels.

As described previously, for subtask 1, where the aim is to classify entire articles, a preprocessing was made to clean the text in each article, removing punctuation. The Python

googletrans[1] library that implements the Google Translate API was used to translate non-English language articles. Through this translation process, the final training set contains a total of 674 articles. In particular, the following articles were added: 52 items from Italian, 54 from French, 46 from German, 40 from Poland, and 49 from Russian.

The adopted Transformers model is a DistilBert base uncased fine-tuned through the following hyperparameters: batch size of 16; learning rate of $2e^{-5}$; AdamW optimizer; 6 epochs.

A randomly sampled 15 percent of the training set is selected for validation purposes. For the training, an NVIDIA GeForce GTX 1060 with a memory of 6GB was used.

Persuasion Techniques Detection. The dataset provided by the SemEval-2023 Task 3 organizers for Subtask 3 specific for the English language contains 446 training news and web articles and 9498 paragraphs, where 5738 of these are labeled as "No Persuasion" and 3760 fall under at least one of the 23 persuasion techniques. 90 articles are provided in the Dev set, with 3127 paragraphs: 2007 as "No Persuasion" and 1120 falling under at least one of the persuasion techniques. Finally, 54 articles containing 910 paragraphs were released for the test set where no gold labels were known.

For subtask 3, where the aim is to identify the persuasion techniques in each paragraph, a preprocessing was made to clean the text in each article, removing punctuation. In the first phase, for the Binary Persuasion Classifier, the model used is a Transformer DistilBert base uncased, with the following hyperparameters: batch size of 16; learning rate of $2e^{-5}$; AdamW optimizer; 4 epochs.

A randomly sampled 15 percent of the training set is selected for validation purposes.

For the Multiclass Persuasion Classifier, the adopted model is again a Distilbert base uncased but with the same hyperparameters except for the batch size, which was set to 10.

For both models' training, an NVIDIA GeForce GTX 1060 with a memory of 6GB was used.

Results

In this section, obtained results for both considered subtasks are described.

News Genre Categorisation. Table 4.1 shows the results of our approach compared with one of the competition winner and the baseline for the English Subtask 1. Our model achieves a macro F1 of $58,621\%$, resulting in 6th place on the English leaderboard from 23 teams.

Table 5.1 reports how the model performs on the development set for each label. The percentage of correct prediction is better in Reporting class, followed by Opinion and Satire. Although we increased the number of articles to have a more balanced dataset, the minor class still performed poorly. Additionally, general performances with and without the use of SHAP are presented.

[1] https://pypi.org/project/googletrans/

Method	F1 macro	F1 micro
Best ranked (MELODI)	0.78431	0.81481
BERT-SHAP (Ours)	**0.58621**	0.61111
Baseline	0.28802	0.61111

Table 4.1 Scores for English sub-task 1 - News Genre Categorisation

Labels	Precision	Recall	F1-Score	Support
Opinion	0.46	0.60	0.52	20
Reporting	0.78	0.70	0.74	54
Satire	0.06	0.11	0.08	9
BERT-SHAP	0.43	0.47	0.44	83
BERT	0.32	0.33	0.32	83

Table 4.2 Performance per class on the development set for sub-task 1 and comparison with an approach without SHAP.

Persuasion Techniques Detection. Table 4.3 shows the results of our approach compared with one of the competition winner and the baseline for the English Subtask 3. Our model achieves a micro F1 of 29,758%, resulting in 12th place for the English subtask 3 leaderboard from 23 teams.

Method	F1 micro	F1 macro
Best ranked (APatt)	0.37562	0.12919
BERT-SHAP (Ours)	**0.29758**	**0.10871**
Baseline	0.19517	0.06925

Table 4.3 Scores for English sub-task 3 - Persuasion Techniques Detection

Table 4.4 reports the results on the Dev set for each involved persuasion technique. Analyzing the results, it is immediately apparent how so many techniques have a performance of 0%, highlighting the difficulty of constructing a multilabel classifier with unbalanced data, despite the adoption of an xAI approach. For these techniques, additional analysis must be done to identify and extract relevant patterns and further help the classifier. Moreover, another analysis should be conducted for the "Repetition" technique, which involves the repeated use of words or concepts to redundant an idea. In this and similar cases, defining additional models or rules more focused on the specific target class and lightening the multi-class responsibility could be helpful.

4.2 Unfolding the Misinformation spread

In today's digital age, the proliferation of misinformation poses a pressing challenge to our society. Often disseminated unknowingly, false or misleading information can spread like wildfire through social media and online platforms, eroding trust in credible sources

Labels	Precision	Recall	F1-Score	Support
Loaded Language	0.53	0.81	0.64	483
Name Calling-Labeling	0.62	0.58	0.60	250
Doubt	0.39	0.48	0.43	187
Repetition	0.15	0.17	0.16	141
Appeal to Fear-Prejudice	0.37	0.32	0.35	137
Exaggeration-Minimisation	0.28	0.28	0.28	115
Flag Waving	0.45	0.50	0.47	96
False Dilemma-No Choice	0.31	0.06	0.11	63
Appeal to Popularity	0.00	0.00	0.00	34
Appeal to Authority	0.21	0.14	0.17	28
Slogans	0.30	0.32	0.31	28
Conversation Killer	0.15	0.24	0.19	25
Causal Oversimplification	0.06	0.08	0.07	24
Red Herring	0.00	0.00	0.00	19
Obf-Vag-Conf	0.00	0.00	0.00	13
Straw Man	0.00	0.00	0.00	9
Appeal to Hypocrisy	0.00	0.00	0.00	8
Guilt by Association	1.00	0.25	0.40	4
Whataboutism	0.00	0.00	0.00	2
BERT-SHAP	0.40	0.48	0.44	1666
BERT	0.36	0.40	0.38	1666

Table 4.4 Performance per class on the development set for sub-task 3 and comparison with an approach without SHAP.

and distorting public perception [63]. This phenomenon threatens our collective Cognitive Security and undermines the foundations of informed decision-making, democracy, and social cohesion. Cognitive Security involves a collection of strategies and approaches aimed at protecting against social engineering attacks and the deliberate or unintentional manipulation of cognitive processes and sensory disruptions [6].

Artificial intelligence stands out as a highly promising way for threat detection and mitigation, with researchers focusing on graph-based methodologies to effectively monitor and counteract the dissemination of misinformation within social networks [184]. In particular,GNNs [12] have gained popularity across diverse domains capacity to handle complex graph-structured data [70], capture non-linear relationships between nodes and

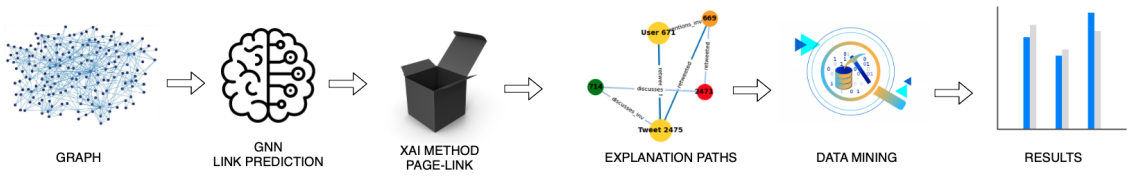

Fig. 4.3 Overall workflow

generalize to unseen data. However, like other neural network models, GNNs confront the challenge of opacity, as they often operate as black boxes, obscuring the underlying mechanisms [2].

Concerning misinformation diffusion, making analysts aware of key factors determining the propagation of false information allows them to take proactive measures to prevent or mitigate its viral spread. Going in this direction, the aim of this work is leveraging explainability on GNN to be aware of what influences some behaviors. In particular, the proposed framework first models relationships among users, tweets and their content through a GNN and, through the link prediction, focuses on retweets, which are considered both a user's interest in the tweet arguments and his/her contribution to disseminating the content. So, it applies link prediction to figure out who retweets tweets concerning misinformation, then, through an explainability method, discovers paths connecting the user and the target tweet, and finally, using frequent itemsets mining, analyzes possible emerging patterns. The analysis is guided by several research questions, such as:

- Are the paths driven more by interpersonal relationships or affinity for ideas? Does his/her subnetwork influence the user?
- Did the tweet end up in a network that is helping to spread it? Are there some users who are conveying the tweet to spread it? Does the tweet go through other tweets systematically?
- How often do different explanations for the same prediction originate from the same relationship?

4.2.1 Methodology

The proposed approach depicted in Fig.4.3 combines GNN, xAI, and Data Mining to understand mechanisms guiding the decision to retweet misinformation content. Firstly, input data is converted into a graph, and the GNN is trained and tested to make link predictions. Subsequently, the PaGE-Link xAI method [219] is adopted to extract the explanation paths for each prediction. In the last step, frequent itemset mining discovers global patterns from local explanations in the spread of tweets containing false information.

MuMiN dataset. The adopted dataset [144] is a large-scale multilingual multimodal fact-checked misinformation social network dataset. It contains 21 million tweets belonging to 26 thousand Twitter threads, each linked to 13 thousand fact-checked claims across dozens

of topics. This dataset was selected due to its wide range of content diversity, making it well-suited for constructing a heterogeneous graph and conducting a comprehensive large-scale analysis.

Graph building. While mapping the dataset into the graph, all nodes except the *Image* node were used, which was considered not very useful for the analysis work. The constructed graph then contains the following relationships: *User* → follows/mentions → *User*, *User* → has hashtag → *Hashtag*, *User*→ posted → *Reply*, *User* → has article → *Article*, *User* → retweeted → *Tweet*, *Reply* → reply to/quote of → *Tweet*, *Tweet* → has article → *Article*, *Tweet* → dicusses → *Claim*, and *Tweet* → has hashtag → *Hashtag*. As explained later, link prediction focused on the *User* → retweeted → *Tweet*. Such a relationship could indicate both the user's interest in the tweet arguments and his/her contribution to disseminating the content. The *dgl* library[2] was used to construct the graph, a Python package built for easy implementation of the graph neural network model family. The GNN was trained on 70% of the data, with the remainder 20% for the test and 10% for validation.

Link Prediction. As aforementioned, link prediction regards the *User* → retweeted → *Tweet* relationship. It was trained by adopting the dgl library for Heterogeneous link prediction model[3][4]. The optimizer utilized in this study is Adam, with a learning rate set at 0.01. The training was conducted over a span of 50 epochs, yielding the most favorable outcome with an Area Under the Curve (AUC) value of 0.9963 for the validation dataset and a slightly lower value of 0.9930 for the test dataset.

Explainability. Once the model was tested, the explainability method PaGE-Link developed in [219] was used. It is an agnostic, post-hoc approach and instance-level method. PaGE-Link takes as input both the model and the test dataset. The obtained explanations consist of various paths reconstructing why a user and a tweet are connected. The method consists of two key modules:

- k-core Pruning Module: This module eliminates extraneous neighbors within the graph, enhancing computational efficiency. It is designed to work with the k-core of a graph, where a k-core is defined as the largest subgraph containing nodes with a minimum degree of k.
- Heterogeneous Path-Enforcing Mask Learning Module: This module focuses on learning masks that identify crucial path-forming edges within the graph. The core idea here is to acquire a mask covering all edges of various types to pinpoint the significant edges essential for the analysis.

[2]https://www.dgl.ai
[3]https://docs.dgl.ai/en/0.6.x/tutorials/basics/5_hetero.html
[4]https://docs.dgl.ai/en/0.8.x/tutorials/models/1_gnn/4_rgcn.html

For the specific execution of PaGE-Link, parameters were configured as follows. The number of hops (i.e., num hops) was set to 2. The number of paths (i.e., num paths) was established at 40, indicating that a maximum of 40 distinct paths could be returned. The maximum path length (i.e., max path length) was set to 15, signifying that each explanatory path could encompass a maximum of 15 edges. The remaining parameters were retained at their default values.

Frequent Itemsets mining. In the last step, the approach applies frequent itemsets mining to analyze the acquired explanations.Frequent itemset mining is a technique to discover groups of items that co-occur frequently in transaction data. The objective is to find potential patterns. The adopted algorithm is Apriori, with a minimum support set at 0.25.

Three different executions of the algorithm are made. Results are presented and discussed in the subsequent section.

4.2.2 Experimentation

Results of frequent itemset mining have been analyzed by formulating research questions that help qualitatively and quantitatively understand motivations guiding social media users spreading misinformation.

What are the most frequent relationships? Are the paths driven more by interpersonal relationships or affinity among ideas? In the first Apriori execution, the objective is to understand the most frequent relationships; so, items consist of triples representing different relationships extracted by PaGE-Link.

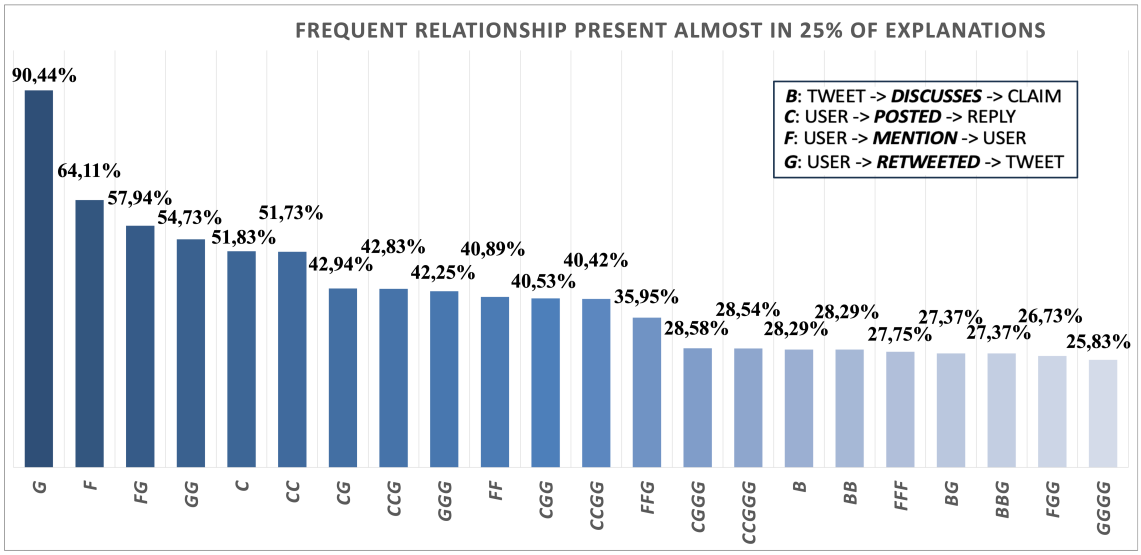

Fig. 4.4 Combinations of relationships having a support of at least 25%.

Fig.4.4 shows the emerging frequent itemsets. What stands out immediately is the *retweeted* relationship (i.e., *G*) in about 90% of the explanation paths; detached instead

is the *mentions*/*follows* relationship (i.e., F) at 64%. While the *mentions*/*follows* relationship is classified as an interpersonal relationship, the *retweeted* relationship should be analyzed individually in order to be able to understand the reason for a retweet, whether due to a condition of acquaintance or content affinity. The same consideration should be made for the *reply* relationship, which is present in more than 50% of the explanation paths, testifying how a trivial action, such as creating a reply to a user's tweet, leads him or her to connect to new nodes. In about 28% of the explanation paths, the discussion of a claim is a common point.It is left to the reader to consider the many combinations of relationships found in the same path of explanation.

Analysis on subnetworks. For analyses devoted to understanding how much explanations involve the same node instances, items are constructed by differentiating relationships by source and target nodes by their IDs. For example, $user_{15} \rightarrow$ retweeted $\rightarrow tweet_9$ and $user_{15} \rightarrow$ retweeted $\rightarrow tweet_{25}$ are considered two different items, although they are sharing the same relationship type.

Is the user influenced by his subnetwork? In the case of reply nodes, in 70% of cases, a reply to a tweet that appears in the explanation of one retweet prediction also appears in another at least. A high percentage also for user nodes where in 65% of cases, a user appears at least in two different predictions for the same user target. The percentage drops to 57% for the tweet category and the Article and Claim categories to 44% and 38%, respectively. This confirms what has already been said about the greater influence of interpersonal relationships than content affinity. To better understand, Fig.4.5 shows an example regarding the case in which it was predicted that user 356 would retweet tweets 713 and 763. Both tweets are united by user 620 retweeting them, which user 356 gets to in a different way.

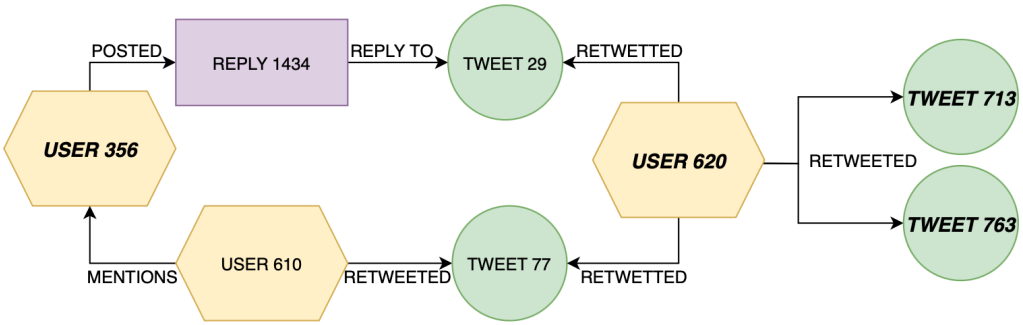

Fig. 4.5 Example of pathways of explanation passing through the same node instance.

Did the tweet end up in a network that is helping to spread it? Are there some users who are conveying the tweet to spread it? Does the tweet go through other tweets systematically? This analysis, similar to the one conducted in the previous paragraph, seeks to understand what percentage a user or a tweet appears on different predictions concerning different target users. In this case, it was decided not to consider article

and claim nodes since they are certainly present and discussed in the same tweet. For a target tweet concerning different predictions, in 55% of cases, the explainability path goes through the same tweet, 45% for the same user, and 37% for the same reply node. It is interesting to learn, in this case, how most likely to convey the spread of a tweet are other tweets, and here is needed to understand whether by affinity of content pushed by the recommendation system or because shared by influencer users. In less than half of the cases, the same user falls into different explanations of predictions, but more is needed to be able to say anything.

How often do different explanations for the same prediction originate from the same relationship? In 50% of cases, the path is always the same; in 17% of cases, the path has 2 starting options; in 13%, with 3 different options; and in 4%, with 4 different relations, the rest are below the 3% threshold. These numbers demonstrate that it takes only a few actions to bring a user closer to a retweet of false information, perhaps due to the recommender system pushing toward content or people discussing certain related content.

What kind of relationship is this starting one? How does the path develop differently from this one? In order to find paths that appear in multiple explanations, the dataset of items is constructed by ordering relationships based on their occurrence in the explanation. Assuming to have an explanation like the following one: $user_{36} \rightarrow$ mention $\rightarrow user_{14}$ and $user_{14} \rightarrow$ retweeted $\rightarrow tweet_{56}$; the corresponding items will be: $user_0 \rightarrow$ mention $\rightarrow user_1$ and $user_1 \rightarrow$ retweeted $\rightarrow tweet_0$.

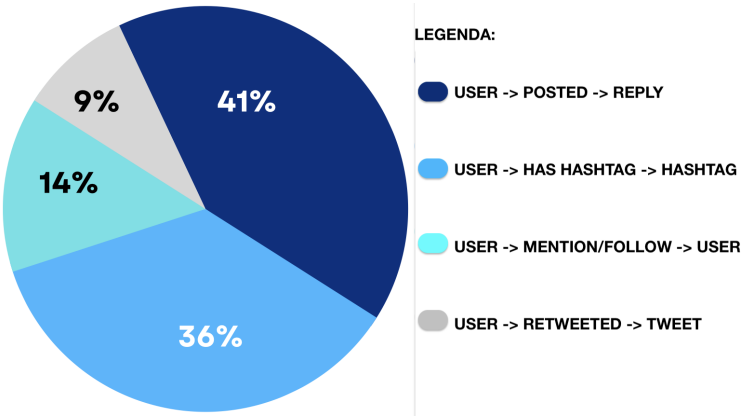

Fig. 4.6 Percentages of relationships at the starting point of found pathways.

Fig. 4.6 shows the results according to the starting relationship, which can be of 4 types. In 41% of cases, Explanatory paths start from a response to a tweet made by the user; in 36%, the path sees a mention as the starting point, while in 14% and 9%, respectively, the paths start because the user has a hashtag in his or her description and because a user retweeted a tweet, still emphasizing how the part of interpersonal relationships prevails over that of content affinity. Compared to these starting points, the development paths also have different characteristics. In particular, it is good to emphasize the length of the path; if a path starts with the user-posted reply, the average length is 8 hops; if the starting

point is a mention, the average length is 6 hops, with the user has hashtag 7 hops while the shortest average path is recorded when the starting point is a user retweet, in this case, the average length is 5 hops.

Additional Analysis. Further statistical analysis was carried out to understand the pathways of explanation better numerically.

What number of hops form the prediction explanation path? Fig. 4.7 shows the results of this analysis, with the length between 6 and 8 hops being around 18% and covering more than 55% of the total number of explanations. The most meaningful explanations are those with the shortest possible length, so explanations with 9 or more hops forming the entire explanation path may be both uninterpretable and insignificant. In the parameters of the explanation model, a maximum path length of 15 was entered solely for the sake of a broader view.

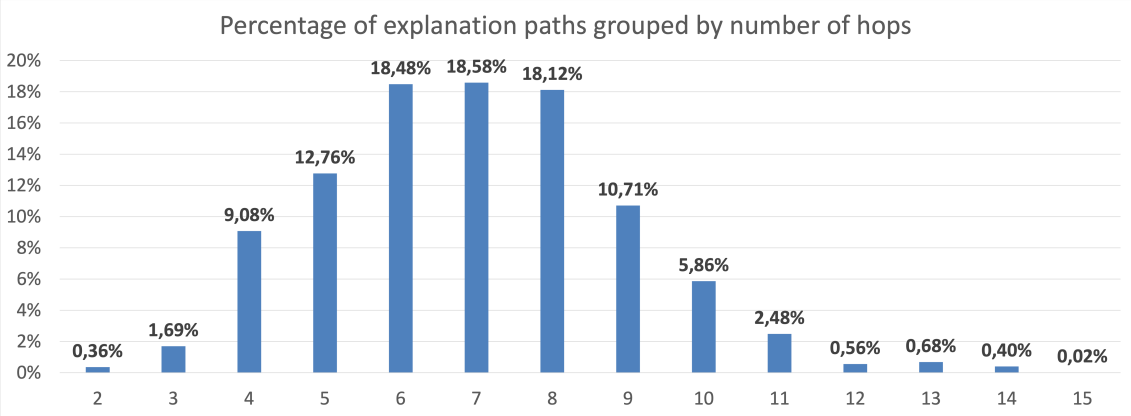

Fig. 4.7 Percentages of number of hops in explanations.

How many paths exist to explain the prediction of retweets? Assuming that a limit of 40 as a max number of paths has been set, in 67% of the cases, predictions have 40 paths available to explain the prediction, leaving the easy guess that there may be even more than this amount. In 7% of the cases, however, there are only 2 paths to explain the predictions, while in 4%, there are 6 explanations. The rest are below the 3% threshold.

4.3 Explainable Fact-Checking

The rapid evolution of the digital landscape heavily influences the spread and accessibility of information. Although advantages correlated to instant access to information, it also raises significant challenges correlated to the reliability and consistency of information.

Recognizing the gravity of risks related to information disorder, researchers, journalists, tech companies, and policymakers are actively working towards understanding and combating it [212]. Initiatives are being developed to improve media literacy, promote fact-checking, enhance digital literacy skills, and hold online platforms accountable for

their role in facilitating the spread of false information [218]. Combating disinformation requires a multidisciplinary approach [76] that spans various fields, including computer and education sciences. It is crucial to face the problem through a combination of automatic fact-checking tools and increased public awareness [83] . Effective disinformation countermeasures involve leveraging advancements in computer science, such as Natural Language Processing (NLP) and Machine Learning (ML), to develop automated fact-checking tools. These tools can analyze large volumes of information, compare it with credible sources, and provide real-time assessments of the accuracy of claims [54, 52]. Contemporary approaches predominantly rely on extensive labeled training data to fine-tune learning models. By leveraging fine-tuning techniques, these approaches can adapt pre-existing models to the specific task of fact-checking, thereby improving their accuracy and ability to handle the complexity of the task. Very recently, the spread of LLMs made it possible to interact with the model in a question-answering form in almost all domains without additional training [180].

DNFS, integrating Deep Neural Networks and Fuzzy Logic, is an active and promising research area with much potential for solving complex real-world problems with high accuracy and interoperability [187].

This research introduces the Hybrid Fact-Checking Framework (HFCF) designed to verify claims automatically by providing a reliability measure of responses based on adopted evidence.

Initially, the LLM extracts sub-facts within the input claim. These sub-facts are then used to infer the final verdict for the claim by referencing official pieces of evidence. Evidence can consist of closed-world information (e.g., domain reports) or the Web (in the absence of official documents). Finally, the ANFIS trained through the trustworthiness of sources, the relevance of retrieved evidence, and the correctness of LLM output measures the response reliability. Main idea is leveraging the inherent knowledge of LLMs, improving it by the Retrieve Augmented Generation (RAG) and measuring the result reliability through the transparency power of fuzzy systems.

HFCF has been evaluated on a domain-specific dataset about climate change (Climate-FEVER dataset), on scientific claims (SciFact dataset) and on a large-scale dataset (FEVER).

4.3.1 Methodology

The methodology of the work is presented below.

Problem and Solution Formulation

Given a claim (or fact) c, the objective is to verify c by classifying it as *Supports* or *Rejects*, meaning the availability or not of supporting evidence.

The LLM first learns a function:

$$g : c \rightarrow F \tag{4.2}$$

which decomposes c in a set of sub-facts (or sub-claims) $F = \{f_1, f_2, \ldots, f_k\}$, $k \in \mathbb{N}$. Sub-facts in F are adopted in generating search queries to retrieve the set of evidence sentences $\mathbb{E}$ useful for the claim classification. Evidence sentences can be chunks extracted from official reports and contained in a pre-established vector storage or Google snippets retrieved in real-time. In particular, $\forall\, f_i \in F$, m pieces of evidence are extracted, generating k subsets $E_i \subset \mathbb{E}$, $i = \{1, \ldots, k\}$. Each subset E_i is input for the LLM that learns a second function:

$$gg : (f_i, E_i) \rightarrow \{Supports, Rejects\} \tag{4.3}$$

that establishes a verdict for the sub-claim f_i. Finally, the verdict with the maximum frequency becomes the final system outcome. If it is impossible to extract a verdict with a maximum frequency (i.e., the same number of verdicts for both labels), the system returns $Not\_Enough\_Information$.

A similarity score s_i and a credibility score cs_i are extracted for each subset $E_i \subset \mathbb{E}$. The first one determines the average relevancy of $E_i \subset \mathbb{E}$ in terms of text similarity with the corresponding sub-fact f_{ij}; the second one identifies the average level of trustworthiness of pieces of evidence in E_i. Applying the average on all $E_i \subset \mathbb{E}$, corresponding average scores (namely, μ_s and μ_{cs}) are extracted. Two latter values together with k, the number of sub-facts, are input for the Adaptive Neuro-Fuzzy Inference System that learns the function:

$$confidence : (k, \mu_s, \mu_{cs}) \rightarrow \mathbb{R} \tag{4.4}$$

The function $confidence$ outputs the confidence associated with inputs through previously generated fuzzy rules, giving a reliability value of the classification result.

Preliminaries

The following subsections introduce the pillar concepts and components underlying the overall proposed framework.

Large Language Models. Language modeling (LM) is a crucial instrument to enhance the linguistic capabilities of machines. The main objective is to indicate the likelihood of generating word sequences, allowing for predicting upcoming or absent tokens. Pre-trained Language Models (PLMs) use a bidirectional LSTM (biLSTM) network for initial pre-training, capturing context-aware word representations, and later fine-tuning this network for specific downstream tasks. BERT is a pre-training bidirectional language model using specially designed tasks on large unlabeled datasets based on the highly parallelizable Transformer architecture with self-attention mechanisms. Researchers have observed that enhancing PLMs through methods such as scaling the model size or the amount of data leads to improved model capacity for downstream tasks. This realization gave rise to LLMs [138].

The process of pre-training plays a crucial role in LLMs. It entails embedding general knowledge acquired from large corpora into the model large parameters. Language Modeling and Denoising AutoEncoding (DAE) are the two most typical pre-training tasks.

When provided with a sequence of tokens denoted as $x = x_1, ..., x_n$, the fundamental goal of the job is to forecast the target tokens, x_i, in an autoregressive manner. This prediction is based on the information in the tokens x_i preceding it in the same sequence.

After pre-training LLMs, it is critical to use a dedicated decoding strategy to obtain the appropriate output from these models. One essential decoding strategy is *Greedy* search, which involves anticipating the most likely token at each step based on the tokens created in previous steps. Other decoding tactics are *Sampling-based* methods that add randomness and diversity to the generating process by selecting the next token randomly based on the probability distribution.

Prompt Engineering. The process of developing a suitable prompt is known as Prompt Engineering. A well-designed prompt can be highly useful in eliciting LLMs' capacities to complete specified activities. Four critical aspects represent the functionality of a prompt for evoking LLMs' capacities to execute activities. A *task description* is often a detailed instruction that LLMs must follow. To make *input data* legible by LLMs, it must be presented properly and conveniently. *Contextual or background information* is required for various tasks besides the task description and input data. Establishing a proper *prompt style* for different LLMs to elicit their ability to complete specific tasks is critical.

Several approaches improve LLM performance on difficult reasoning tasks. Instead of merely generating prompts with input-output pairs, like in *In-Context Learning (ICL)*, *Chain-of-Thought (CoT)* integrates intermediate thinking steps that can lead to the ultimate output within the prompts [206]. Another power technique is *Self-consistency*, which involves sampling numerous and diverse reasoning paths across the CoT at a few hits and using generations to find the most consistent response [202]. *Retrieval Augmented Generation (RAG)* is another strategy. RAG entails taking an initial input and collecting a group of documents that support the source. These retrieved materials are then merged with the original input prompt to form a contextual framework submitted to the text generator for final output generation. RAG's adaptability is especially useful when information changes or evolves. RAG uses this strategy to prevent the need for re-training and to provide language models with access to the most up-to-date information. This, in turn, makes it easier to generate trustworthy output with a retrieval-based strategy [119]. This work is based on the RAG idea supported by few-shot learning.

Adaptive Neuro-Fuzzy Inference Systems . ANFIS combine the principles of Fuzzy Logic (FL) and Artificial Neural Networks (ANN) to form a powerful model that not only captures the vagueness of human thought through fuzzy IF-THEN rules but also mimics the neural processes of the human brain, thereby enhancing learning capabilities.

The ANFIS model is prevalent in classification, rule-based process controls, and pattern recognition applications, showcasing its ability to manage uncertainty and complexity. This makes it particularly effective in domains such as fact-checking, where interpretability and adaptability are crucial [187]. Its self-learning capabilities and real-time updating

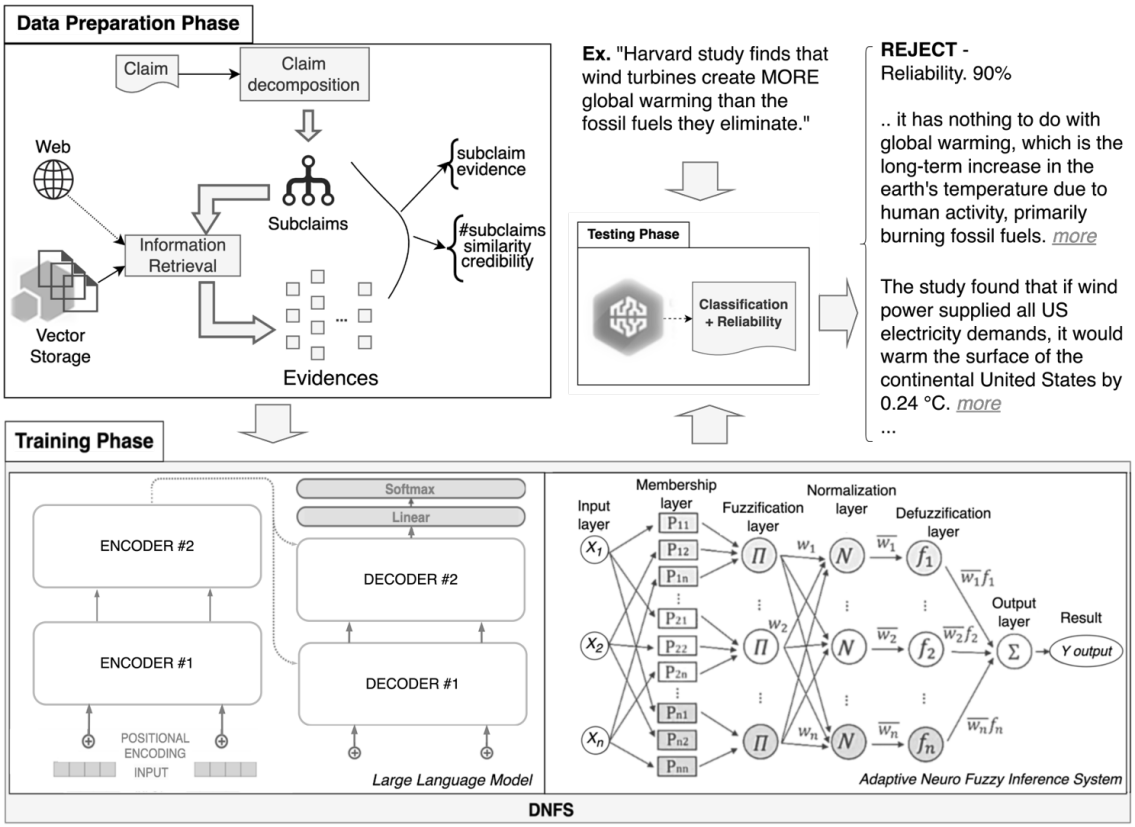

Fig. 4.8 Architecture of HFCF (partially inspired by [100])

ensure transparent and reliable decision-making, critical in managing the dynamic nature of information disorders [188, 75].

Significantly, ANFIS has been applied in various sectors: - **Healthcare:** Used for diagnostic systems where precise and adaptive modeling of clinical data is crucial. - **Financial Markets:** Employed to predict stock prices and forex movements, benefiting from its ability to handle noisy and incomplete data. - **Environmental Modeling:** Assists in predicting pollution levels and climate changes, adapting to new environmental data inputs dynamically.

The architecture of ANFIS consists of five layers: 1. **Input Layer:** Receives the input variables. 2. **Input Membership Function (MF) Layer:** Determines the degree to which these inputs belong to each of the appropriate fuzzy sets. 3. **Fuzzy Rules Layer:** Applies fuzzy logic rules. 4. **Output MF Layer:** Combines the fuzzy outputs. 5. **Defuzzification Layer:** Converts the fuzzy classification results into a crisp output.

Each layer's nodes are interconnected through directed links, forming a multilayer feed-forward network. The learning algorithm optimizes the fuzzy inference system parameters, including membership functions and consequents, using methods like Gradient Descent (GD) for backward updates and Least Square Estimator (LSE) for forward parameter tuning [188].

Hybrid Fact-Checking Framework

The proposed Hybrid Fact-Checking Framework (HFCF) leverages and improves LLMs' inherited knowledge through Retrieval Augmented Generation for interpreting the claim, finding suitable evidence, and classifying the claim. Then, it exploits evidence characteristics to train an ANFIS that gives a reliability score to classification results. More in detail, as shown in Fig. 4.8, it mainly concerns three phases: (i) *Data Preparation Phase*, the preliminary creation of the vector storage containing official evidence and the preparation of the data necessary for next phases; (ii) *Training Phase*, where the Deep Neuro-Fuzzy System is trained; (iii) *Testing Phase*, in this last phase both components are used in parallel to generate the classification with information retrieval and a reliability assessment.

In the following the detail of each aforementioned phases.

Data Preparation Phase. The first stage of HFCF consists of generating the vector storage of documents and subclaims and acquiring evidence from the sources (i.e., vector storage or web).

1. Vector storage generation: The vector storage consists of a long-term memory database containing documents for fact-checking claims. These documents comprise verified reports or domain-specific documents. These documents are partitioned into chunks of a predetermined length to facilitate efficient processing and segmented using defined splitters.

2. Generation of subclaims and evidence extraction: Once the knowledge has been loaded into the vector storage, it becomes available for claim verification. The claim verification process begins by sending the claim to a LLM with a few-shot prompting. This prompt helps in the claim decomposition, generating sub-facts (or sub-claims), simplifying the assertions within the claim and facilitating their retrieval in vector storage. The claim-checking process can be, in turn, split into two different flows regarding the source for the evidence extraction. The first flow utilizes the previously constructed storage vector containing domain-specific documents. This approach is particularly useful when dealing with claims supported by official reports or established sources. On the other hand, the second flow is tailored for situations where official reports are unavailable or insufficient. In such cases, the framework relies on the Web as a source of information to assess the claim validity and provide necessary evidence for fact-checking.

- Information retrieval through storage vector: Each sub-fact, generated by the LLM is embedded and sent to the vector store. Here, it undergoes a similarity search to retrieve the k most similar chunks, represented by vectors closely resembling the sub-claim and having the lowest L2 (Euclidean) distance. All retrieved pieces of evidence having a distance less than the predefined threshold τ are considered suitable for the subsequent step. Moreover, if suitable pieces of evidence are less

than k, an analog search is conducted on the Web for each sub-claim, as described in the following.

- Information retrieval through the Web: In cases where not all the sub-claims have at least k pieces of evidence with a distance smaller than τ, an additional flow is pursued to gather evidence. In this scenario, the given sub-claim is directly searched on Google. The framework extracts the first relevant snippets from search results and appends them to evidence from the vector storage.

Training Phase. The Training Phase of the HFCF is pivotal for preparing the DNFS to accurately classify claims based on evidence. During this phase, the LLM is invoked with a specific prompt, including claims and corresponding evidence, enhancing its capability to comprehend and interpret the nuances of claim-related discourse. This process leverages few-shot learning to enrich the LLM's understanding of the domain-specific context and evidence evaluation criteria.

Similar to the previous step, a few-shot prompting, learns the model of how to classify the sub-claim. Finally, the most frequent label for sub-claims of the given claim becomes the final verdict for the claim itself. The following linguistic variables characterize each data sample (i.e., claim):

- $Number\_of\_Sub\_facts$ (N): the number of sub-facts LLM extracts for the given claim;
- $Similarity$ (S): the average of the mean text similarities between each sub-fact and the corresponding pieces of evidence or snippets;
- $Credibility$ (C): the average of the mean trustworthiness values of pieces of evidence and snippets corresponding to each sub-claim. If the evidence comes from an official report (i.e., retrieved through the vector storage), it assumes the value of 1. Whether it is a snippet, the value is in the range $[0-1]$ and matches with Google's PageRank of the page.

Moreover, the label associated with each claim can assume a value of 0 or 1. Whether the ground-truth value matches the LLM classification result, the label is 1; otherwise, it is 0.

Subsequently, the ANFIS is trained using the outputs generated by the LLM, including the evidence reliability scores and the preliminary classification results. The ANFIS training uses a dataset annotated with evidence credibility ratings, enabling it to learn and apply fuzzy rules that quantify the reliability of the LLM's classifications.

Testing Phase. In this phase, the complete HFCF is collectively employed to formulate a classification for new input claims. ANFIS operates parallel with the LLM to generate the output, constituting the final verdict. This output is accompanied by the fragments of evidence employed by the LLM in arriving at the decision, as well as an assessment of the response's reliability. In particular, as expressed in the example in Fig. 4.8, the

system receives a claim (specifically about scientific research on wind turbines) and returns a classification value (i.e., Reject), a reliability value (i.e., 90%), and a list of evidence supporting the final verdict. This phase demonstrates the seamless integration between the training and testing stages, as the LLM and ANFIS work in concert to evaluate the validity of claims against the backdrop of accumulated knowledge and fuzzy logic rules. For each claim, the LLM first identifies relevant evidence and generates a preliminary classification. This classification, along with the evidence's features, such as source credibility and text similarity, are then processed by the ANFIS to assess the reliability of the classification outcome. The unique contribution of the Testing Phase lies in its dynamic feedback loop. Based on the performance of the DNFS in classifying new claims, adjustments are made to both the LLM and ANFIS components. This iterative refinement process, informed by real-world testing, enhances the framework's accuracy and reliability over time, embodying the integrated essence of the training and testing phases. Through this approach, the HFCF ensures that its fact-checking capabilities evolve in response to a constantly changing information landscape.

4.3.2 Experimentation

The performance of the proposed framework has been evaluated on three publicly available datasets, Climate-FEVER, SciFact, and FEVER, in the following subsections.

Datasets. **Climate-FEVER** is a dataset from Diggelmann et al. [190] for the verification of climate-related claims. The dataset consists of 1535 claims labeled with four categories: *Supports* (43%), *Refutes* (16%), *Not_Enough_Info* (31%), and *Disputed* (10%). The labels indicate whether the evidence supports, refutes, or lacks sufficient information to validate the claim. To ensure consistency with existing literature, only instances labeled as *Supports* or *Refutes* are considered, obtaining a total of 907 claims. Regarding the construction of the storage vector, evidence comes from the documents of the IPCC's Sixth Assessment Report[5].

SciFact [197] is a dataset created by researchers from the *Allen Institute for AI* and the University of Washington for verifying scientific claims. Claims originate from high-quality articles belonging to journals spanning domains from basic science to clinical medicine. The dataset contains 1400 claims labeled as *Supports* (39%), *Refutes* (37%), or *Not_Enough_Info* (24%).

FEVER (Fact Extraction and VERification) [191] is a dataset for training and evaluating fact extraction and verification systems. It consists of about 185k claims generated by altering sentences extracted from Wikipedia. Claims are categorized into three classes: *Supported*, *Refuted*, or *Not_Enough_Info*. The experimentation adopts the labeled dev version of the dataset consisting of about 20k items (i.e., 33% *Supported*, 33% *Refuted*, 33% *Not_Enough_Info*).

[5]https://www.ipcc.ch/assessment-report/ar6/

Tools, Libraries, and External APIs. The implementation of the framework heavily relies on the LangChain Python package[6], which serves as the core tool. Within this package, multiple modules were utilized, including document loading (e.g., PyMuPDFLoader for PDFs), text splitters, chains, retrievers, and LLMs.

An open-source embedding model called sentence-transformers/all-mpnet-base-v2 was incorporated for embedding the text. This model was invoked using the LangChain library embeddings, specifically leveraging HuggingFaceEmbeddings. By utilizing the embedding model, effective vectorization of all chunks of the IPCC AR6 files and the input claims was achieved, which is essential for subsequent similarity searches and evaluations.

The Facebook AI Similarity Search (FAISS)[7] library was employed as the vector store to facilitate the efficient storage and retrieval of these vector embeddings. FAISS's capabilities enable the persistence of vector representations in a database, optimizing the handling of large-scale embeddings and speeding up the retrieval process during fact-checking. For this proposal, the vector database is persisted locally.

Regarding the Large Language Model, the capabilities of two versions were harnessed. In particular, the LLMs Vicuna and Flan-T5 performances were evaluated for sub-facts generation and the final classification of claims.

Vicuna is open-source and trained by fine-tuning LLaMA on approximately 70K user-shared conversations collected from ShareGPT. It is an auto-regressive language model based on the transformer architecture [39]. T5 consists of a transformer-based architecture that uses a text-to-text approach and is the epitome of encoder-decoder excellence in Natural Language Processing (NLP). FLAN-T5 does not need large devices because its smaller checkpoints are created for the common citizen [41].

Finally, after empirical research, the following implementations were adopted:

- ggml-vicuna-7b-1.1-q5_1 (7B parameters), an implementation of Vicuna enabling large models and achieving high performance on commodity hardware. It was downloaded and called locally for the generation of sub-facts, with the following parameters: n_ctx : 1024, $temperature$: 0 and top_p : 0.9.

- google/flan-t5-xxl (11B parameters), a Large Language Model open-sourced by Google, for the final answer of the framework, using the Langchain HuggingFaceHub. The parameters used are $temperature$: 0.1 and max_length : 1024.

For the Web flow (i.e., fact-checking through the Web), the Serper API[8] was utilized to extract snippets from Google search results.

A type-1 Sugeno Fuzzy Inference System is generated by the Matlab app Fuzzy Logic Designer[9]. An ANFIS was specially used to create the fuzzy inference system. Specifically, the FIS generation used the "grid partitioning" clustering algorithm because the input dimensionality was reduced to two columns. 50 ANFIS epochs were used in the training

[6]https://python.langchain.com/docs/get_started/introduction.html
[7]https://github.com/facebookresearch/faiss
[8]https://serper.dev/
[9]https://it.mathworks.com/help/fuzzy/fuzzylogicdesigner-app.html

process. The output of each rule was expressed as a linear function of the input variables, proportionally scaled by the antecedent result value.

FIS assessment. The formulation of fuzzy rules designed for the FIS leverages an intricate interplay between its input variables, *Similarity* (S) and *Credibility* (C), to predict the output variable, Reliability. This predictive power stems from the inherent flexibility of fuzzy logic, which excels in capturing complex, non-linear relationships within data.

The FIS may be represented as follows:

$$\text{Reliability} = \text{FIS}(S, C) \tag{4.5}$$

Here, the FIS computes the Reliability based on the linguistic variables of *Similarity* and *Credibility*, with generalized bell-shaped membership functions utilized for inputs.

During the preprocessing of the dataset, several crucial aspects emerged, shedding light on the relationships between the input and output variables. These insights are critical to understanding the FIS's predictive capabilities:

- The correlation between *Similarity* (S) and the output label is positively moderate (0.56), which can be expressed as:

  $$\rho(S, \text{Label}) = 0.56$$

  This statistical measure signifies a moderate, positive relationship between *Similarity* and the label. As the similarity score increases, there is a tendency for the label value to rise correspondingly.

- The correlation between *Credibility* (C) and the output label is strongly positive (0.81), indicating a robust, positive correlation, as expressed:

  $$\rho(C, \text{Label}) = 0.81$$

  When *Credibility* increases, there is a notable increase in the label value.

- The correlation between $Number\_of\_Sub\_facts$ (N) and the output label is about -0.14. Expressed in formulas:

  $$\rho(N, \text{Label}) = -0.14$$

  This value indicates a negative correlation between the two variables, which means that when the value of "sub_facts" increases, the value of "label" tends to decrease, and vice versa. Since the correlation is quite weak, there is no discernible correlation between N and the label. Consequently, this input variable was excluded from the final experimentation phase as it did not contribute significantly to the predictive power of the model.

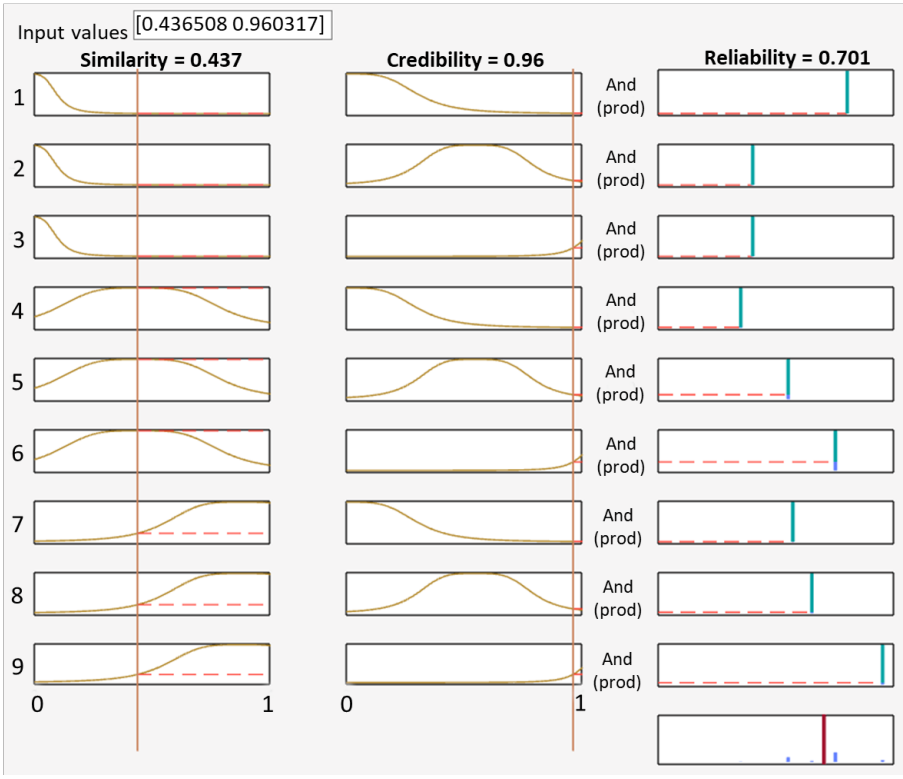

Fig. 4.9 Example of fuzzy inference rules application.

These observations align closely with the FIS model, where the rules governing the relationships between the input variables (S and C) and the output variable (Reliability) are established using fuzzy logic. The FIS utilizes these rules to make predictions based on the degrees of membership of input variables in linguistic terms (e.g., "High," "Low," and "Medium"). The FIS is configured with the following parameters:

- AND method: PROD
- OR method: PROBOR
- Implication method: PROD
- Aggregation method: SUM
- Defuzzification method: WTAVER

In the example depicted in Fig. 4.9, the Reliability score exceeds the 0.7 with a combination of a medium level of *Similarity* and a high level of *Credibility*. This example showcases the nuanced behavior of the system where the weighted significance of *Credibility* allows for a substantial *Reliability* output, even if *Similarity* is not at its highest level, underpinning the robustness of the model in scenarios with varied input profiles.

Results & Discussion. The experiments encompassed various tests, beginning with comparing the proposed methodology's performance against state-of-the-art methods. Then, an ablation study regarding the contribution of single flows is described. Furthermore,

the contribution of the Fuzzy Inference System is measured by evaluating the fluctuation in fact-checking performance by changing a *Reliability Threshold* (τ_R).

Comparisons with state-of-the-art. The baselines were carefully chosen for each considered dataset. For the Climate-FEVER dataset:

- A RoBERTa model fine-tuned on the Climate-FEVER dataset, along with data augmentation [40].
- climateGPT-2 [193]: A GPT model fine-tuned on a corpus of approximately 360 thousand abstracts related to climate change.
- CLIMATEBERT [204]: A BERT model fine-tuned on climate-related texts collected from multiple sources.

The following approaches were considered for comparison on the SciFact dataset:

- ARSJOINT (BioBERT) [222] is a fine-tuned model of pre-trained BioBERT-large on the SciFact dataset.
- MultiVers [198] implements Longformer architecture as its base encoder. It is first pre-trained on a combination of general-domain fact-checking and then fine-tuned on the SciFact dataset.
- BEVERS [55] is a fine-tuned model of pre-trained DeBERTa V2 XL MNLI on the FEVER dataset.

Finally, for FEVER dataset, the selected approaches are the following:

- DeSePtion [93]: a system for end-to-end fact-checking leveraging on (1) a document ranking model; and (2) a sequence labeling model.
- Few-Shot Fact-Checking [118] exploits the transfer learning ability of language models and the perplexity score.
- TARSA [177]: a topic-aware evidence reasoning and stance-aware aggregation approach for fact verification.

Table 4.5 presents the experiment results, showcasing the proposed framework (i.e., HFCF, in bold) superiority in terms of *F1 score* compared to state-of-the-art methods in the climate change and scientific articles domains. In order to make comparisons with the baseline, two types of *F1 averages* are presented in the table: *F1-Macro*, which calculates metrics for each label and finds their unweighted mean, and the *F1-Weighted* that alters the F1-Macro to account for label imbalance. The comparison reveals that the framework surpasses existing approaches, even without utilizing pieces of evidence or labeled datasets to fine-tune the adopted model.

Table 4.5 Performance of proposed approach with respect to baselines

Dataset	Model	F1-Macro	F1-Weighted
Climate-FEVER	RoBERTa [40]	0.691	-
	climateGPT-2 [193]	-	0.72
	CLIMATE-BERT [204]	-	0.757
	HFCF	**0.783**	**0.741**
SciFact	ARSJOINT [222]	0.667	-
	MultiVerS [198]	0.672	-
	BEVERS [55]	0.732	-
	HFCF	**0.778**	-
FEVER	DeSePtion [93]	0.69	-
	Few-Shot + Fact-Checking [118]	0.717	-
	TARSA [177]	0.779	-
	HFCF	**0.878**	-

Table 4.6 Results of ablation study applied on Climate-FEVER dataset.

Model	F1-Macro	F1-Weighted
FlanT5-XXL	0.665	0.663
FlanT5-XXL + domain reports	0.718	0.780
FlanT5-XXL + domain reports + Web search	**0.741**	**0.783**

Ablation study. Besides comparing with the state-of-the-art, an ablation study was conducted on one of the datasets to understand the significance of evidence extracted from both domain reports and the Web. Initially, the experimentation evaluated the performance of the LLM, specifically FlanT5-XXL, without the support of any evidence or additional context. In this sense, few-shot learning was adopted to address the classification task using the LLM without a domain context. The results of this experiment, shown in the first row of Table 4.6, highlight that relying solely on the inherited knowledge of the LLM is not sufficient to effectively approach the fact-checking problem. It emphasizes that evidence plays a pivotal role in verifying claims accurately. Subsequently, tests are conducted for two scenarios: one considers only domain reports as evidence, and the other considers both domain reports and information from the Web. Results are, respectively, in the second and third rows of Table 4.6. The best performance is in bold.

The ablation study effectively reveals the positive contribution of adding evidence to the input claim used by the framework to motivate rejection or acceptance of the claim. Moreover, the performance increase, including input evidence from both domain reports and the Web, underlying the importance of integrating multiple information sources.

Reliability Assessment. The reliability assessment component has been evaluated by setting different values for the *Reliability Threshold* (τ_R) and evaluating its effect on the performance of the claim-checking process. Increasing τ_R reduces the set of claims that the system can accept or reject and positively impacts the F1-Macro (i.e., its highest value corresponds to the highest τ_R). Charts in Figs. 4.10-4.12 depict the variation of F1-Macro (in the y-axis on the right) and the percentage of covered claims (in the y-axis on the left) by varying τ_R for considered datasets. The blue bars show the percentage of claims from the dataset that are processed by the model at different reliability thresholds, starting from the reliability corresponding to the 100% coverage of the dataset. As the reliability threshold is raised, the model becomes more stringent, resulting in a lower percentage of claims being considered. The grey line represents the F1-Macro score, reflecting the balance between

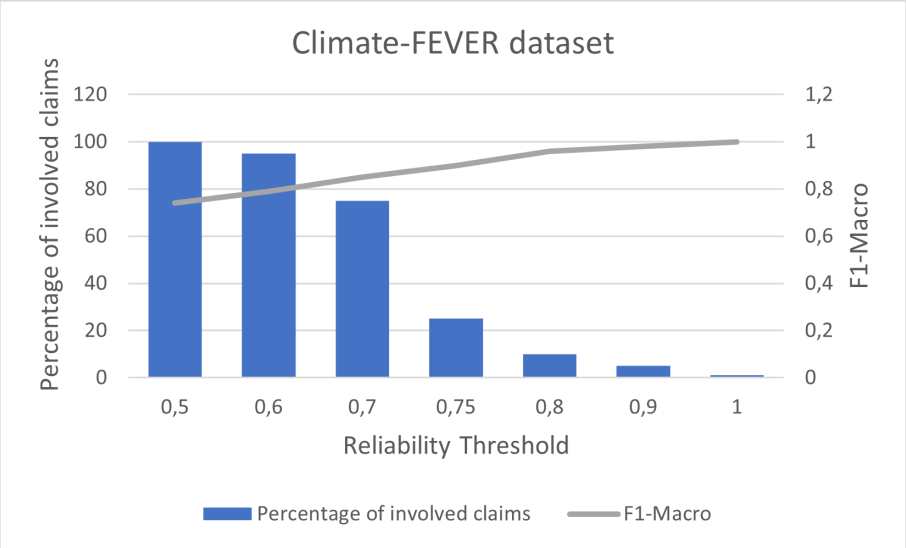

Fig. 4.10 The effect of adjusting the reliability threshold on the coverage and F1-Macro of the climate claims included in Climate-FEVER dataset.

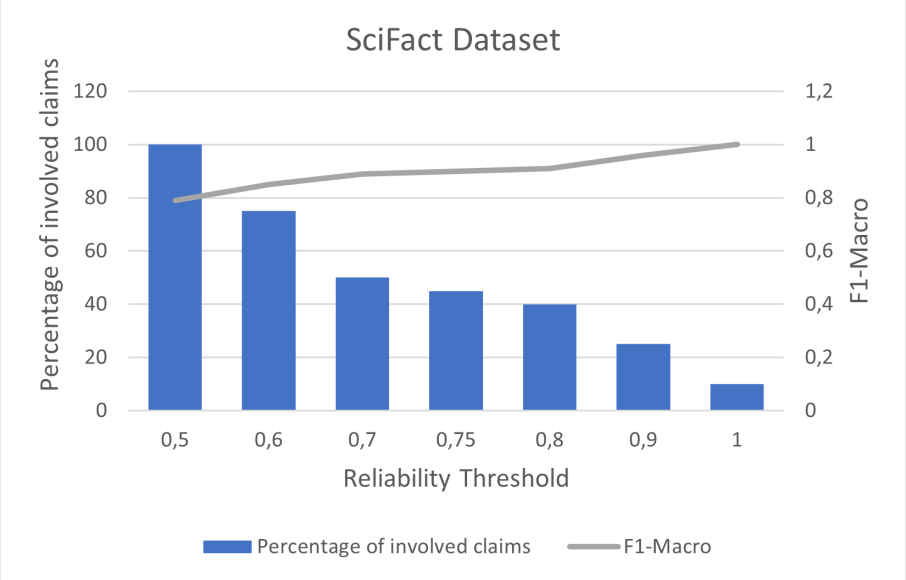

Fig. 4.11 The trade-off between scientific claims coverage and F1-Macro score on the SciFact dataset by varying the reliability threshold.

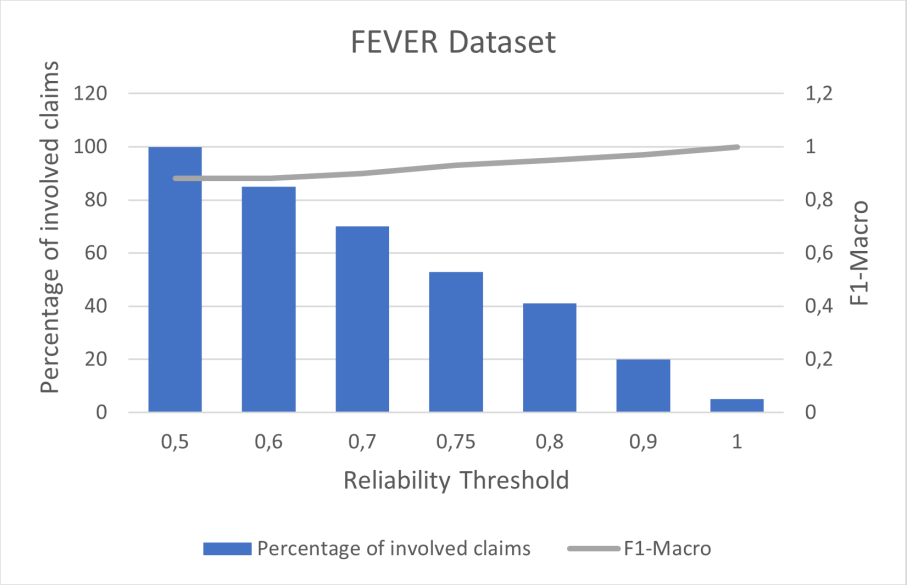

Fig. 4.12 The effect of reliability thresholds on claims coverage and F1-Macro within the FEVER dataset.

precision and recall across the claims. Notably, the F1-Macro score remains fairly stable in Fig. 4.10across different thresholds. This stability can be attributed to the high reliability of the answers given on the overall Climate-FEVER, where the evidence supporting or rejecting the claim included in the input to the framework is mainly taken from official information sources. On the other hand, there is a greater impact on the improvement of the F1-Macro score (grey line) in Fig. 4.11 and 4.12 in correspondence to the increasing reliability threshold for the datasets SciFact and FEVER, respectively.

The experiments detailed in the previous section demonstrate that the proposed framework significantly enhances the fact-checking task performances compared with the existing literature in different domains. Moreover, the ablation study supports the initial hypothesis that combining open and closed sources can improve final performance in discerning between real and false claims. The combined approach can be particularly useful when domain knowledge is insufficient. Finally, experiments regarding the reliability assessment confirm the valuable contribution of ANFIS in measuring each prediction confidence (or reliability) and the validity of a hybrid solution for fact-checking problems.

Experimental results, specifically the ablation study, also demonstrate that, thus far, existing LLMs (specifically, FlanT5-XXL) can better resolve this type of task with additional information. In this sense, presenting evidence with the prediction label and the confidence level as information motivating whether the claim was supported or refuted can be considered a valid solution for bolstering the interpretability of the approach.

Chapter 5

xAI 4 ML & DL Robustness

Machine learning (ML) models have achieved significant advances but face challenges regarding robustness or their ability to maintain performance under input perturbations, a relevant problem, for example, when fighting Information Disorders. This vulnerability is particularly concerning for LLMs, where adversarial attacks can lead to "jailbreaking"—manipulating the model to bypass restrictions and generate harmful content. Such exploits highlight serious ethical and security issues. This chapter discusses three papers that help answer the research question: "How can xAI be utilized to identify and mitigate vulnerabilities in models?" xAI can identify and mitigate models' vulnerabilities by revealing how models make decisions, thus highlighting potential weaknesses and identifying keywords and subsequently neutralizing them in prompts that lead LLMs to lose alignment. The first two works, in particular, conduct experimentations on the robustness of models adopted to contrast the Information Disorder, while the third one propose focus on the LLM that lose alignment.

First work focuses on the role of the malicious use of xAI in increasing the effectiveness of adversarial text attacks or, dually, the aid its correct use may provide in measuring the robustness of propaganda detection models. The approach proposed here leverages xAI and Adversarial Text Generation techniques to simulate malicious attacks and measure the robustness of a propaganda detection model based on BERT. The attacks involve generating a new dataset by perturbing critical words in the original one identified with the aid of xAI (SHAP and LIME). The effectiveness of terms determined using xAI methods is compared with a statistical keyword extractor (YAKE!). These methods are adopted to detect the most important words as perturbation targets. The goal is to quantify the impact of disrupted instances on learning model performance. Experiments on the SemEval 2020 task 11 dataset reveal that modifying words detected by xAI methods significantly affects classification performance by reducing accuracy by 30%. This work is presented in Section 5.1

Second work provides a comprehensive evaluation of model Robustness against adversarial attacks across key tasks well-assessed in Information Disorder literature: Toxic Speech Detection, Sentiment Analysis, Propaganda Detection, and Hate Speech Detection. Rigorous experiments conducted across 13 models and 12 diverse datasets highlight signif-

icant vulnerabilities. The methodological framework implements adversarial attacks that strategically manipulates model inputs based on keyword significance, identified using the LIME method, an advanced explainable AI technique. The evaluation measures Robustness primarily through accuracy of the models and attack success rates. The experiments reveal that current models display inconsistent resistance to adversarial manipulations, underscoring an urgent need for developing more sophisticated defensive strategies. The study sheds light on the critical weaknesses in existing models and charts a course for future research to fortify AI resilience against evolving cyber threats. This work is presented in Section 5.2

Third work proposes a mechanism for identifying jailbreaking prompts for manipulating LLMs. The designed process involves the interaction between an LLM Attacker and an LLM Victim. LLM Attacker generates potential jailbreaking prompts to induce the LLM Victim to generate unethical content. The prompts and their corresponding persuasion success are collected during their interaction. In this way, a new synthetic dataset of 3000 prompts has been constructed. Such a dataset is exploited to train a new model for detecting hidden persuasion in prompts that can induce an LLM to produce deviating content. This new model, assisted by algorithms for xAI, works as an anti-persuasion filter interposed between the input prompt and the victim model. It identifies attempts to mislead LLM and tries to neutralize them by modifying words recognized as crucial by xAI algorithms like SHAP and LIME. Experimentation reveals that adopting SHAP and removing the first ten most important words in the original prompt allows for neutralizing 80% of persuasive prompts. This work is presented in Section 5.3

Published contributions are the following:

- Cavaliere, D., Gallo, M., & Stanzione, C. (2023, July). Propaganda Detection Robustness Through Adversarial Attacks Driven by eXplainable AI. In World Conference on Explainable Artificial Intelligence (pp. 405-419). Cham: Springer Nature Switzerland.
- Fenza, G., Loia, V., Stanzione, C., & Di Gisi, M. (2024). Robustness of models addressing Information Disorder: A comprehensive review and benchmarking study. Neurocomputing, 127951.
- Fenza, G., Gallo, M., Loia, V., Nicolosi, A., & Stanzione, C. (2024). Detecting Persuasive Prompts: A Framework for Secure LLMs (***In press***)

5.1 Propaganda Detection Robustness Through Adversarial Attacks Driven by eXplainable AI

The propaganda detection problem has been approached mainly through Machine and Deep Learning-based solutions [46, 38] that had a relevant explosion in the last years in many areas. Moreover, parallel with the spread of such techniques, xAI reached great attention

for building result explanations and helping experts in making decisions. However, if, on the one hand, xAI substantially improves the transparency of results of "black-box" models, on the other hand, it leaves the system vulnerable to adversary attacks [27]. Employing a specific jargon (e.g., argot, cryptolect) or word manipulations can easily fool detection systems. An example of a similar trick regards supporters of QAnon (an American political conspiracy theory and political movement) who may use different nicknames (e.g., "17Anon") to identify themselves and avoid the banning from social media sites [140]. Fooling learning models, especially in disinformation counterfeiting, can be seen as a form of cyber-attack aiming at leading detection systems to fail. This paper presents a framework that utilizes xAI technologies, namely SHAP and LIME, and a statistical keyword extractor (i.e., YAKE!) to test the robustness of a pre-trained Transformer-based propaganda detection classifier and its capacity to withstand cyber-attacks. By simulating a text manipulation attack through Adversarial Text Generation, the framework identifies and changes the most critical words for propaganda detection and reclassifies the text for analyzing system performances. The study reveals the superiority of xAI-based methods in detecting significant words for propaganda detection and their potential to combat cyber-attacks on propaganda detection systems. That outcome suggests that devoting efforts to cybersecurity solutions by leveraging xAI methods could give good results.

5.1.1 Methodology

This section introduces the proposed methodology to analyze the robustness of a propaganda detection model toward adversarial attacks aimed at deceiving the system in the classification process. A complete workflow is shown in Figure 5.1. The first step concerns the detection of propaganda from texts by using a pre-trained Transformer-based model; the achieved results are then processed to extract features ranked by their impact; the third step allows changing the extracted features in the dataset by using Adversarial Text Generation (ATG) techniques. Finally, the last step runs the pre-trained Transformer-based model on the updated dataset to get newer propaganda classification results, that are compared to the early ones for assessing the robustness of the evaluated model against adversarial attacks.

Transformer-based propaganda classification. A Transformer network based on a DistilBERT pre-trained model is employed to classify text propaganda. The pre-trained model is fine-tuned on the SemEval 2023 Task 3 training dataset (which, at the time of writing, is yet to be public but only available for competition participants) for the propaganda detection task [10]. The dataset provided by the SemEval-2023 Task 3 organizers for Subtask 3 specific for the English language contains 446 training news and web articles and 9498 paragraphs, where 5738 are labeled as "No Propaganda" and 3760 fall under at least one of the 23 propaganda techniques. Data is subjected to a preprocessing step to clean the text and remove punctuation. To fine-tune the Transformer

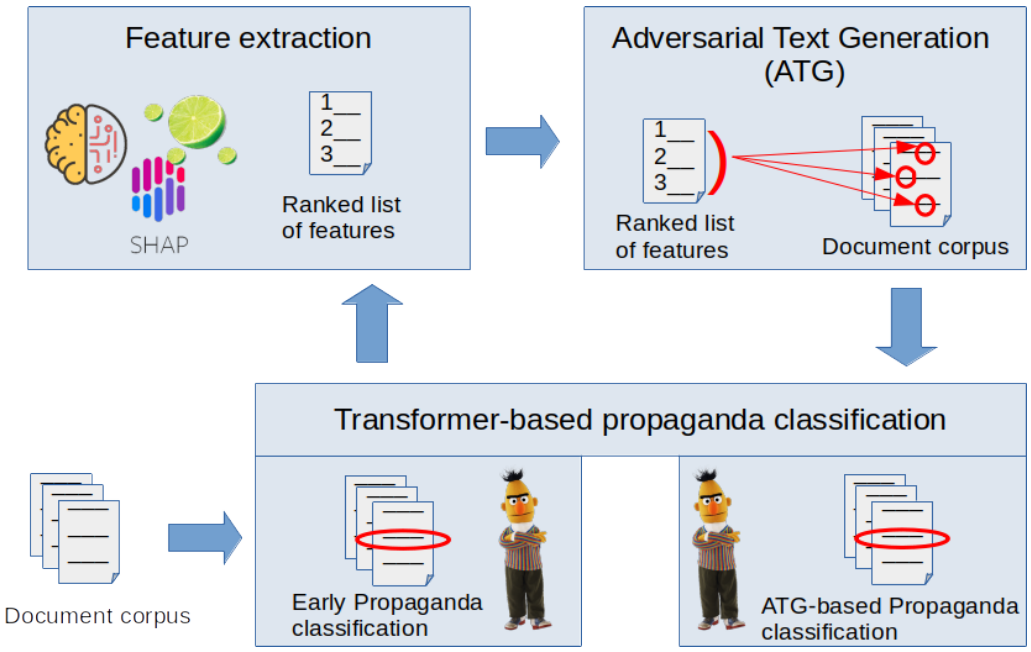

Fig. 5.1 Methodology Workflow: 1) Propaganda in texts is detected using a pre-trained Transformer model; 2) xAI methods extract impactful features; 3) the dataset is updated using Adversarial Text Generation (ATG) on features previously selected; 4) the updated dataset is used to run the pre-trained Transformer-based model again, generating newer propaganda classification results.

Distilbert-Base-Uncased, the following hyperparameters are used: the batch size of 16; learning rate of $2e^{-5}$; AdamW optimizer; 4 epochs.

The constructed model is tested on the corresponding dev set consisting of 90 articles and 3127 paragraphs whose 2007 are "No Propaganda" and 1120 "Propaganda". Tests provide an accuracy of around 90%.

Feature extraction. To extract the features (i.e., words) with the highest impact on propaganda detection results, two different xAI technologies (i.e., SHAP [129] and LIME [163]) and a statistical keyword extractor (i.e., YAKE! [24]) are used for the task. The three methods return as much as ranked lists of words that mostly affected propaganda predictions. Such words are selected at this stage to change the original dataset and then test the robustness of the propaganda detection model.

Let us show a running example for which the classification changes after the adversarial attempts. Starting from the following instance, "*Are You Kidding Me, Ted Cruz? Don't "Blame The Police Office" Who Admitted Killing Botham Jean? FOX 26 asked Cruz to respond to his Democratic midterm rival, Beto O'Rourke, who called for officer Guyger to be fired*" the system extracts the first five most important words by each method. In particular, LIME detects the subsequent ones: *midterm*, *Killing*, *rival*, *officer*, *Democratic*.

Adversarial Text Generation (ATG). At this stage, each word detected as crucial at the previous step is looked for in the dataset and replaced with alternative words calculated using Adversarial Text Generation (ATG) techniques. ATG refers to the practice

of generating new instances by slightly perturbing inputs to fool the learning models. Algorithms for ATG can select words among nearest neighbors in the embedding space, out-of-vocabulary, or through generative models. In this work, five different methods are used for generating words that are semantically and syntactically similar to the original ones; they are (1) space insert, (2) character delete, (3) character swap, (4) Substitute-C (Sub-C), (5) Substitute-W (Sub-W) [121]. Method (1) consists in inserting a space into the word, method (2) deletes a middle character, method (3) swaps two adjacent characters in a word, method (4) replaces a character in a word with a visually similar one (i.e., '1' with 'l') and method (5) replaces a word with its top k nearest neighbors in a context-aware word vector space. The word replacement algorithm applies the five methods in sequence to the extracted N-ranked words, from the highest-ranked word to the lowest-ranked one. Finally, the updated dataset is processed with the Transformer-based propaganda classification model to get the propaganda detection results after the ATG-based changes.

Recalling the example in the previous section, by applying the character delete technique to all five words, the starting instance becomes as follows: "*Are You Kidding Me, Ted Cruz? Don't "Blame The Police Office" Who Admitted* ***Kiling*** *Botham Jean? FOX 26 asked Cruz to respond to his* ***Demoratic miderm rial****, Beto O'Rourke, who called for* ***offcer*** *Guyger to be fired*". Then, the propaganda detection model classifies the new instance and the changed others, and performance is registered.

Comparing the accuracy of the updated dataset and the original one allows for determining how much the extracted words contributed to determining propaganda sentences. In detail, the higher the accuracy gap, the higher the impact of words in detecting propaganda and, consequently, the risk of corrupting the model by modifying them.

5.1.2 Experimentation

The experimentation was aimed at evaluating the proposed methodology. It is carried out by comparing the performance of the propaganda detection model before and after adversarial attacks conducted systematically on words identified as significant by the three chosen methods: xAI-based (SHAP and LIME) and statistical keyword extraction (YAKE!). More in detail, the experimentation follows subsequent steps:

- Creation of the dataset;
- Feature extraction and ATG attempts generation;
- Results collection.

The steps are detailed in the following subsections.

Creation of the dataset. The dataset used for the experimentation is SemEval 2020 task 11, subtask Technique Classification. ue it uses, The task aims to associate labels representing the propaganda technique employed by choosing from an inventory of 14, given a specific text fragment in the context of a whole document [46]. The dataset

is provided with the start and end coordinates of the span within a paragraph and the propaganda technique or techniques, if more than one, that characterize such paragraph. The label “Propaganda” is associated with paragraphs containing these spans. Paragraphs without any propaganda spans are labeled as “No Propaganda". An operation of down-sampling assures the balancing of the dataset with 1000 final instances.

Applying the constructed propaganda detection model on the described dataset returns an accuracy of 88%.

Feature extraction and ATG attempts generation. Given the propaganda detection model and the dataset, LIME, SHAP and YAKE! are asked to extract features (in this case, words) crucial for the final classification. First, each framework produces a ranked list of words. Then, adversarial attacks on these words are generated. Attacks, as previously described, are inspired by the TextBugger framework consisting of five text editing techniques. Thus, five correlated words are generated for each word extracted, corresponding to five new instances, which the propaganda detection model must classify. The performance of new classifications is described in the following subsection.

Results collection. Table 5.1 presents the prediction results after modifying the most important word for each method. On average, modifying words detected by two xAI-based methods significantly decreases the model performance, leading it to lose 10% accuracy. In particular, with the SUB-C technique on the words identified by LIME, there is a quite significant decrease of 13%. Regarding the YAKE! statistical method, its performance decrease is at maximum around 3%, which is way less significant than that observed for the two xAI methods.

METHODS	NO ATG	INSERT	DELETE	SWAP	SUB-C	SUB-W
SHAP	0,88	0,78	0,79	0,79	0,80	0,80
LIME	0,88	0,79	0,79	0,81	0,75	0,83
YAKE!	0,88	0,85	0,85	0,86	0,85	0,88

Table 5.1 Performance of Propaganda Detection Model with and without adversarial attacks on the most important word extracted by each method. The table shows how the accuracy of the model changes after the first word deemed most relevant by the different methods is perturbed with the different ATG techniques.

After an initial experiment attacking only the first-ranked word in terms of importance, the investigation is extended to the first five words extracted by each technique. Results shown in Tables 5.2-5.4 confirm what was already experienced for the first-ranked word: words detected by SHAP and LIME are more relevant for the final accuracy, leading the model to fail more often, even with small perturbations. After perturbing five words, SHAP and LIME cause a decrease in model accuracy, which is greater than 20%, except for the SUB-W technique on LIME. Regarding LIME, it is impressive to analyze the impact of the SUB-C technique, in which case the model accuracy decreases by 31%. The other techniques also perform very well, with the most significant decrease, with the SHAP

SHAP	0	1	2	3	4	5
INSERT	0,88	0,78	0,74	0,71	0,68	0,67
DELETE	0,88	0,79	0,74	0,70	0,68	0,65
SWAP	0,88	0,79	0,74	0,71	0,68	0,66
SUB-C	0,88	0,80	0,73	0,69	0,66	0,64
SUB-W	0,88	0,80	0,74	0,69	0,67	0,64

Table 5.2 Performance of Propaganda Detection Model after attacks on the five most important words extracted by the SHAP method. The table shows the accuracy for each ATG technique as the perturbations progressed. For example, in column '2', the model was tested by making perturbations on the first and second most important words extracted from SHAP. In column '5', the model is tested on a text where the first five most relevant words have been perturbed.

method being 24%. The same thing does not happen when using YAKE!: with this method, the most significant decrease always occurs with the SUB-C technique, scoring 14%, much lower than the 31% of LIME. Moreover, YAKE! causes a 10% accuracy decrease, on average, by perturbating the first five words, while the other methods cause an equivalent decrease after perturbating just the first word.

LIME	0	1	2	3	4	5
INSERT	0,88	0,79	0,71	0,68	0,65	0,63
DELETE	0,88	0,79	0,73	0,69	0,66	0,63
SWAP	0,88	0,81	0,75	0,71	0,66	0,64
SUB-C	0,88	0,75	0,66	0,62	0,60	0,57
SUB-W	0,88	0,83	0,80	0,78	0,78	0,75

Table 5.3 Performance of Propaganda Detection Model after attacks on the five most important words extracted by the LIME method. The table shows the accuracy for each ATG technique as the perturbations progressed, meaning that in column '5', the model is tested on a text where all the first five most relevant words have been perturbed.

YAKE!	0	1	2	3	4	5
INSERT	0,88	0,85	0,84	0,80	0,79	0,77
DELETE	0,88	0,85	0,84	0,81	0,78	0,78
SWAP	0,88	0,86	0,84	0,83	0,79	0,77
SUB-C	0,88	0,85	0,82	0,79	0,76	0,74
SUB-W	0,88	0,87	0,84	0,84	0,82	0,81

Table 5.4 Performance of Propaganda Detection Model after attacks on the five most important words extracted by the YAKE! method. The table shows the accuracy for each ATG technique as the perturbations progressed, meaning that in column '5', the model is tested on a text where the all first five most relevant words have been perturbed.

Another consideration should be made about the techniques: SUB-C is the most effective. Figure 5.2(d) shows how performance drops steadily by adopting this technique focused on a visual perturbation. The least effective technique is SUB-W which replaces the word with the top k nearest neighbors in a context-aware word vector space. Figure

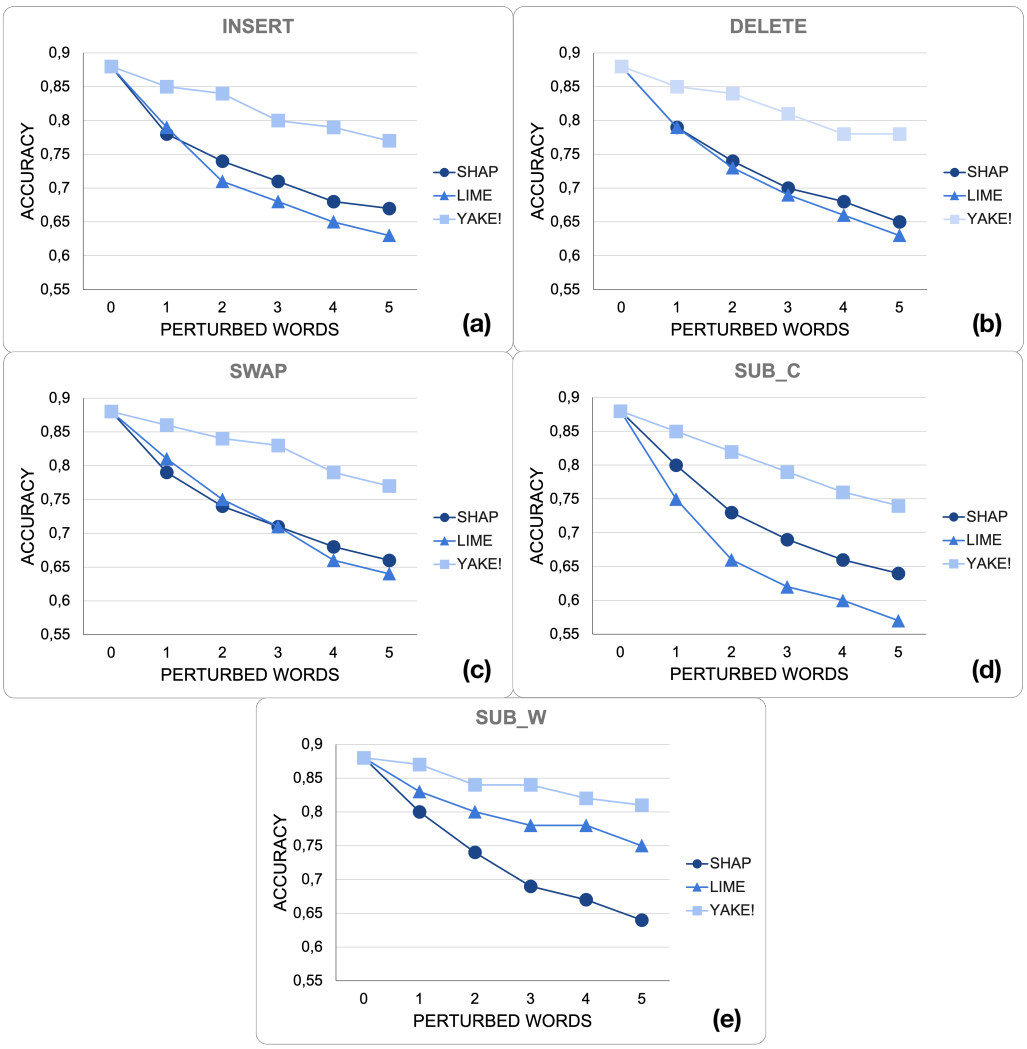

Fig. 5.2 Technical ATGs compare the first five words for each feature extraction method. The figure shows on each graph how the accuracy decreases as the number of perturbed words increases. In particular, (a) shows the comparison using the INSERT technique, (b) the DELETE technique, (c) the SWAP technique, (d) the SUB-C technique and finally, (e) the SUB-W technique.

5.3, which shows a numerical comparison between the accuracy of the initial model and the model after perturbing the first five words for each method, confirms what has been analyzed so far: the SUB-C technique is the most effective in deceiving the model. Finally, let us notice that the words identified by LIME also cause a higher accuracy decrease than those detected by the other methods, except for the SUB-W technique (see Figure 5.2). Therefore, the last analysis consists in measuring the accuracy decrease after each perturbation. In particular, Figure 5.4 shows the average decrease after each perturbation for each technique and each word extraction method. Let us notice that the reported statistics display the relevance of the SUB-C technique, with an average decrease of 6% after each attack. The figure also shows how YAKE! is less effective than the other two methods.

The results emerging from the experimentation can be summarized as follows:

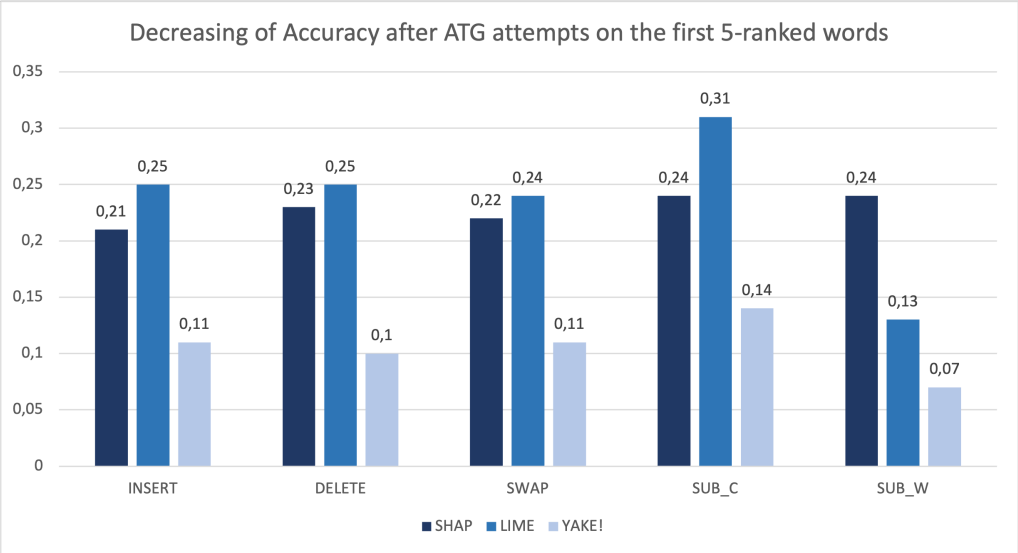

Fig. 5.3 Decreasing of Accuracy after ATG attempts on the first five-ranked words. The figure describes with a bar graph how much the model's accuracy decreases with the perturbation of each technique's first five words identified by each method.

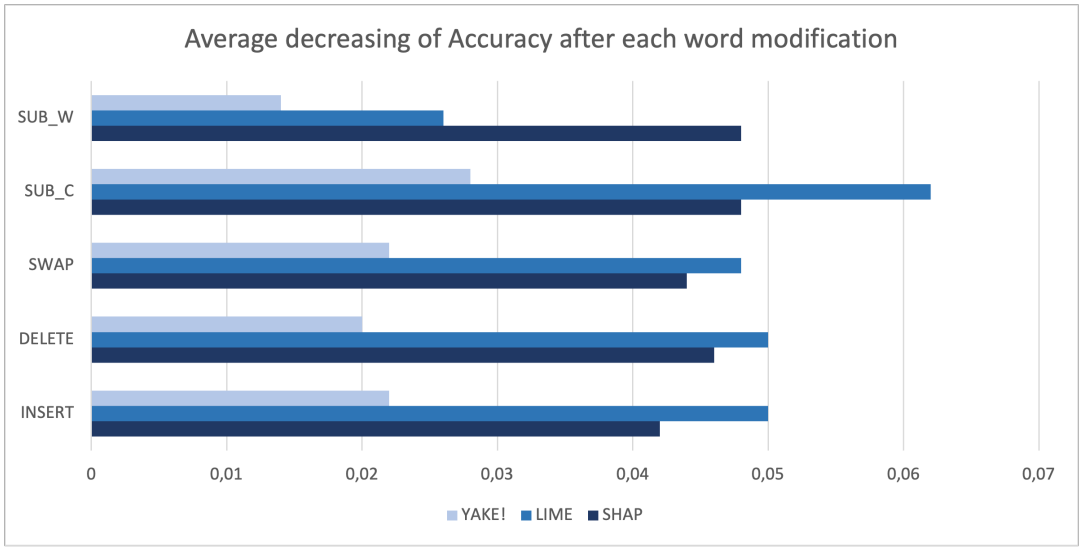

Fig. 5.4 The average decreases in accuracy after each word modification. The figure describes how much, on average, the model declines in accuracy after each perturbation. It does so for each technique for each method.

1. xAI-based framework performs better than a statistical one in selecting the most important feature (i.e., words) of a Transformer-based classifier for propaganda detection;

2. Perturbing words can significantly decrease the performance of a propaganda detection model;

3. Among perturbation techniques proposed by the TextBugger framework (i.e., space insert, character delete, character swap, Sub-C, Sub-W), the most effective in terms of model tricking is Sub-C, namely replacing a character with a visually similar one (i.e., '1' with 'l').

5.2 Robustness of models addressing Information Disorder

This study provides a comprehensive evaluation of model Robustness against such attacks across sub-tasks well-assessed in Information Disorder literature: Toxic Speech Detection, Sentiment Analysis, Propaganda Detection, and Hate Speech Detection. For each selected task, there was a first part on a review of existing literature and methods, with a particular focus on Robustness and a second part of experimentation on selected models and datasets. The selected tasks are all considered under the umbrella of Information Disorder, and high-performing models exist to enable Robustness investigations performed in this study. Let's briefly summarize the relation of each task in the realm of Information Disorder:

- *Toxic Speech Detection:* Toxic speech involves using harmful, offensive, or detrimental language to individuals or groups. Detecting toxic speech is crucial in identifying content that can contribute to the spread of misinformation, as it may be used to harass, intimidate, or deceive.

- *Sentiment Analysis:* Sentiment analysis involves evaluating the emotional tone of a piece of text. In the context of Information Disorder, understanding sentiment helps identify content that may be deliberately crafted to evoke strong emotions, potentially leading to the spread of misinformation or manipulation.

- *Propaganda Detection:* Propaganda involves the dissemination of biased or misleading information with the intent to manipulate public opinion. Detecting propaganda is essential in combating the spread of misinformation and maintaining an informed public discourse.

- *Hate Speech Detection:* Hate speech targets individuals or groups based on attributes such as race, religion, ethnicity, or other characteristics. Detecting hate speech is crucial in preventing the dissemination of harmful and discriminatory content, which can contribute to the escalation of tensions and the spread of misinformation.

Attempts were also made to investigate other tasks of considerable interest in Information Disorder area, such as Stance Analysis, Fake News Detection and others. Nevertheless, the lack of well assessed models in literature did not allow for a thorough analysis of them. Moreover, this study focuses on the textual dimension of the Information Disorder leaving out Computer Vision models [221] and their applications to address tasks like image forgery [216] and deep fake detection [161].

5.2.1 Methodology

The methodology adopted to test the different models in different domains is the same, a methodology well-tested in a work already presented in Section 5.1.1. In past work, the effectiveness was tested by comparing two well-known methods in xAI, SHAP [129] and

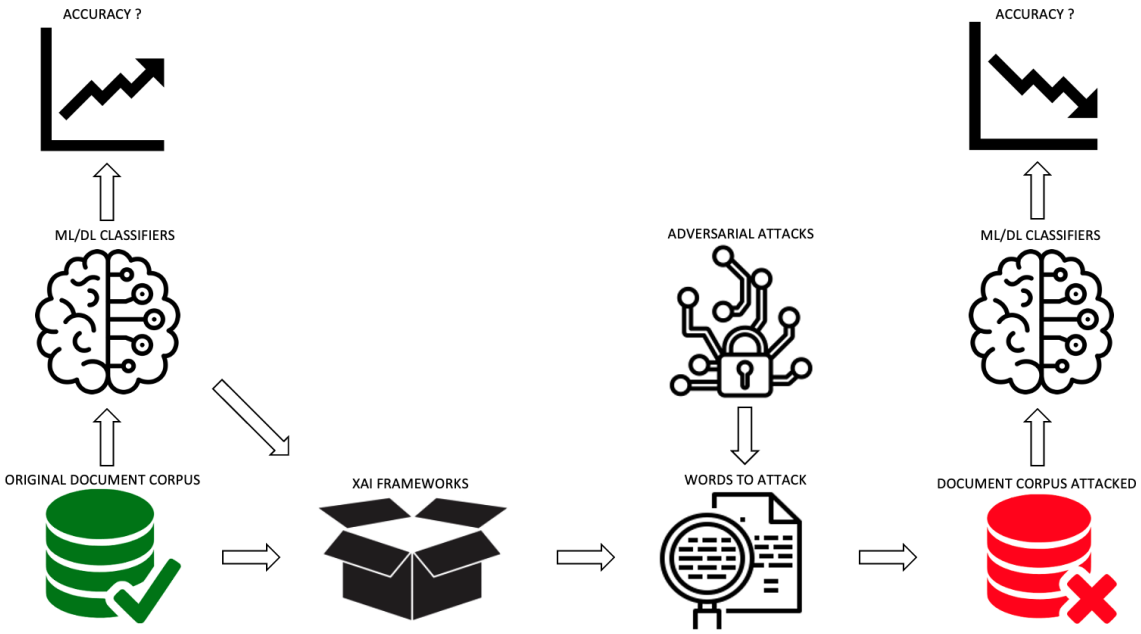

Fig. 5.5 Robustness Analysis Framework.

LIME [163], with a statistical method called YAKE! [24]. In this paper only LIME has been used considering the better performances than others. The workflow, presented in Fig. 5.5, shows the entire process for testing the Robustness of the identified models in the different domains of Information Disorder. Mainly the methodology is encapsulated in three main steps. In the first phase, after identifying the models to be tested and the datasets on which to test them, the work is focused on measuring the model's performance in terms of accuracy in classifying the text concerning the target task. Once this step is done, the second step is applying xAI's method. xAI's method returns the most important features, in this case words that most influenced the model in the final classification. The attacks were unleashed on first one, three and five most important words. The TextBugger framework [121] was used to unleash the attacks; this part is well-detailed and explained in the following sub-section. Once the detected words were modified, they were inserted in place of the original ones, thus resulting in a new dataset. In the third phase, the same model used in the first phase was tested on the newly generated dataset in order to be able to measure the new performances in terms of accuracy and to make a final judgment on the Robustness of the model.

Adversarial Attack Generation. To generate words that are semantically and syntactically similar to the original ones, five different methods were employed: (1) space insertion, (2) character deletion, (3) character swapping, (4) Character Substitution (Sub-C), (5) Word Substitution (Sub-W). Method (1) involves inserting a space in the word, method (2) deletes a central character, method (3) swaps two adjacent characters in a word, method (4) replaces a character in a word with a visually similar one (e.g., "1" with "l", "o" with "0"), and method (5) substitutes a word with its top k nearest neighbors in a context-aware word vector space. The word substitution algorithm applies the five methods randomly, selecting from the mentioned techniques each time in each stage.

Experimentation and Results. At the core of the experimentation, there was a process of research and selection of datasets and models suitable for the task. The collected datasets were meticulously sought on open platforms such as Kaggle and Hugging Face. For most of the tasks, the research yielded extensive results, and it was necessary to focus on the most commonly used datasets and those with references in the literature. As for the models, they were sought and obtained from Hugging Face. The models were used as released, without making any changes. The experimentation was conducted by dividing it by task. The analyzed tasks were Hate Speech, Propaganda, Toxic Language, and Sentiment Analysis. For greater detail, it was decided to conduct the attacks on the texts at three different times, particularly with the one-word attack, the most important word, the three-word attack and the five-word attack. This allowed to be able to compare the performance of each model on each selected dataset and give a broader overview of the model's Robustness and the tested datasets. Accuracy and ASR were used to measure Robustness. The accuracy is defined as the fraction of the test set that is correctly classified [201]. Accuracy was calculated in the different four periods, pre-attack and post each attack (one, three or five). ASR [209] is a metric commonly used to evaluate the effectiveness of a malicious attack or exploit. It represents the percentage of attempted attacks that successfully breach a system. In other words, it measures the rate at which attackers achieve their objectives. In Natural Language Processing, this metric is used to measure the Robustness of a model against an adversary attack. The ASR is the ratio of the difference between pre-attack right predictions and post-attack right predictions divided by pre-attack right predictions.

$$\text{ASR} = \frac{\text{(Correct Predictions after Attacks - Correct Predictions before Attacks)}}{\text{Correct Predictions before Attacks}}$$

In this work, it was calculated exclusively on the label of interest. In Sentiment Analysis, it was calculated exclusively on the label “Negative", in Hate Speech Detection on the label “Hate", in Propaganda Detection on the label “Propaganda", and in Toxic Speech Detection on the label “Toxic".

More than 100 models were considered, while 13 were used. The lack of references in most of the models analyzed led to such a large reduction. A document explaining the framework used or even just the process of developing it was considered essential in considering and analyzing a model. In tasks where models are plentiful, such as Sentiment Analysis, the choice fell in the first instance on models where actions were stated in training to increase Robustness; in the absence of this, models where better performance was stated, were considered. As for datasets, however, an attempt was made to use those most used for the documents studied during the literal overview.

5.2.2 Experimentation

The experimentation conducted for each field is presented in the following.

Toxic Speech Detection: Experimental Analysis

The task aims to assess how well and how robust are models in detecting Toxic Language. For this analysis, four datasets and three models were used.

Datasets. The selected datasets are analyzed in the following:
Toxic-comment-classification-challenge[1] [99](TCCC) is a dataset from the Kaggle competition founded by the Conversation AI team. In this competition, participants were required to build a multi-headed model capable of detecting various types of toxicity. A dataset of comments related to Wikipedia talk page edits was provided. The dataset consists of 159,571 rows and 8 columns. Each row has 6 binary labels showing the types of toxicity, in this work is considered only “toxic" column assigning toxicity where 1 and neutral where 0. For the analysis, this dataset was filtered with limits on the maximum of 512 tokens and minimum of 50 token for the text lenghts. The experimentation was conducted on a sample of 2000 balanced rows. The dataset was sourced from the official Kaggle website.
Youtube toxic comments[2] (YTC) is a manually labeled dataset on toxicity, comprising 1000 comments gathered from YouTube videos discussing the Ferguson unrest in 2014. Beyond toxicity, this dataset includes tags for various subcategories of toxic behavior, organized hierarchically. Toxic comments can be classified as Abusive, Hate Speech or Radicalism. In Abusive another subclassification is Threat, Provocative or Obscene. In Hate Speech the subclassification is Racist, Nationalist, Sexist, Homophobic or Religious hate. Each comment can have multiple such labels assigned. in this work is considered only “istoxic" column assigning toxicity where “true" and neutral where “false". For the analysis, this dataset was filtered with limits on the maximum of 512 tokens and minimum of 50 token for the text lenghts. The experimentation was conducted on a sample of 723 rows, 343 for the “toxic" label and 380 for “neutral" label. The dataset was sourced from the official Kaggle website.
Toxic Conversations 50k[3] [20] (TC50) is a dataset that contains comments from the Civil Comments platform together with annotations if the comment is toxic or not. The dataset consists of 50,000 rows and 3 columns. In this work is considered only “label" column assigning toxicity where 1 and neutral where 0. For the analysis, this dataset was filtered with limits on the maximum of 512 tokens and minimum of 50 token for the text lenghts. The experimentation was conducted on a sample of 2000 balanced rows. The dataset was sourced from the official Hugging Face website.
Paradetox toxicity[4] [126] (PT) is a repository that contains information about Toxicity Task markup from English Paradetox dataset collection pipeline. The dataset consists of 26,507 rows and 2 columns. In this work is considered only “label" column assigning toxicity where 1 and neutral where 0. For the analysis, this dataset was filtered with

[1]https://www.kaggle.com/competitions/jigsaw-toxic-severity-rating/data
[2]https://zenodo.org/records/2586669
[3]https://huggingface.co/datasets/mteb/toxic_conversations_50k
[4]https://huggingface.co/datasets/s-nlp/en_paradetox_toxicity

limits on a maximum of 512 tokens and a minimum of 50 tokens for the text length. The experimentation was conducted on a sample of 2000 balanced rows. The dataset was sourced from the official Hugging Face website.

Models. The selected models are analyzed in the following:
Toxic comment[5] [6] (TC) is a model generated doing a fine-tune of a pre-trained DistilBERT [170] for classifying toxic comments. The authors declare as a limitation the poorly performances of the model for some comments that mention a specific identity subgroup, like Muslim. Hyperparameters information of training are not provided neither in the paper and in model page. According to the developers, the model achieves 94% of accuracy in a 10000 rows held-out test set. The model was sourced from the official Hugging Face website.
RoBERTa toxicity classifier[7] [47] (RTC) is a model based on pre-trained RoBERTa model [125], fine-tuned for toxic spans. They enhance it with the merging of three Jigsaw datasets [181, 21, 106] using three methods: pseudo-labeling, fine-tuning for sentence classification (RoBERTa classifier + tagger), and a novel architecture for joint token- and sentence-level classification (tagging classifier). Hyperparameters information of training are not provided neither in the paper and in model page. According to the developers, the model achieves 76% of F1-score. The model was sourced from the official Hugging Face website.
Toxic speech detector[8] (TSD) is a model fine-tuned of [196] trained on ParaDetox dataset [126]. The following hyperparameters were used during training: Adam optimizer with a learning rate of $2e-05$, training batch size of 128, evaluation batch size of 8, training for only 1 epoch. The model was sourced from the official Hugging Face website.

Results and Analysis. The following analysis examines the results obtained from tests conducted in Toxic Speech Detection. It is important to note that this analysis does not intend to provide a specific judgment on the models or datasets; rather, they will be scrutinized as representatives of their respective categories.

TC Model. The TC model, whose results are presented in Table 5.5, exhibits modest performance from the outset. It achieves an accuracy of 80% in only one case under normal conditions, without attacks, specifically on the TCCC dataset. In the case of YTC, it shows unreliable performance from the start with an accuracy of 71%. Initiating the first attack seems to exacerbate the situation, as modifying a single word, deemed crucial by xAI, leads to a general decrease in accuracy, except on the PT dataset, where the accuracy decreases by only 1%. However, on TCCC, the decrease is approximately 20%; on the other two datasets, it loses 8% on YTC and 11% on TC50, respectively. Regarding ASR, the situation is worse, with successful attacks on the TOXIC category on 3 out of 4 datasets

[5]https://huggingface.co/martin-ha/toxic-comment-model
[6]https://github.com/MSIA/wenyang_pan_nlp_project_2021/blob/main/papers/NLP_Final_Report.pdf
[7]https://huggingface.co/s-nlp/RoBERTa_toxicity_classifier
[8]https://huggingface.co/rb05751/toxic_speech_detector

hovering around 50% , except for PT, where the situation remains reversed. Even with the second and third attacks, modifying 3 and 5 words, the model's performance does not improve, maintaining an accuracy around 60% and an ASR consistently above 62% on the first three datasets. On TC50, specifically modifying five words, there is a successful attack with an accuracy of 68%. However, this is not the case on the PT dataset, where the model retains an accuracy of 75% and records a negative impact on attack success. The model appears non-robust and highly susceptible to attacks, even with a single word. Further investigation into the model's performance on the PT dataset is warranted.

RTC Model. RTC, whose results are presented in Table 5.6, exhibits highly satisfactory accuracy in the pre-attack phase. However, when the first attack is launched, accuracy drops by 18% on the TC50 dataset and decreases by 11% and 13% on TCCC and YTC, respectively. Again, an anomaly is observed on the PT dataset, where accuracy only decreases by 2%, and ASR shows a negative impact. The ASR in other cases is quite high, reaching almost 40% in the case of TC50. With the second and third attacks, the model linearly follows the pattern observed in the first attack, with performance decreasing by several percentage points, including an 18% decrease on TCCC, a 19% decrease on YTC, and a 30% decrease on TC50. This is accompanied by a significant ASR, reaching 66% in the case of TC50. Once again, the PT dataset behaves differently, with the model's accuracy decreasing only slightly and ASR consistently showing a negative impact. RTC, therefore, appears to be non-robust, experiencing significant decreases in performance even after a single word, making it less resilient.

TSD Model. TSD, whose results are presented in Table 5.7, alternates between an accuracy of 94% on TCCC and 79% on YTC. Notably, the accuracy consistently decreases in every attack phase and on all datasets, even on PT, where attacking five words results in a loss of 23%. Similar decreases are observed on TCCC (22%), YTC (22%), and TC50 (19%). The anomaly lies in the ASR, which consistently shows a negative impact, except for TCCC, which does not reach previously seen levels. This suggests that the model is better at predicting toxicity when words are altered. Is it, therefore, more reliable? The answer is no; this should have been reflected in similar accuracy performances, which is not the case. The results reveal that the model shifts from evenly predicting the two labels in a non-attack phase to predicting the Toxic label much more frequently. For example, on the TC50 dataset in a pre-attack phase, the model predicts non-toxic 1207 times and toxic 793 times. After a single-word attack, it predicts non-toxic 607 times and toxic 1393 times. This suggests that the model is indeed non-robust, and if it encounters a word it cannot classify, it may classify the entire content as toxic, leading to many false positives. In conclusion, TSD also appears as a non-robust model.

A brief assessment of the datasets is warranted, with particular emphasis on the PT dataset, which appears to exhibit better behavior in terms of accuracy and ASR. However, as previously mentioned, encountering a negative ASR and decreasing accuracy still indicates signs of non-Robustness in the model. The other three datasets seem less robust, immediately leading the model off course.

	TCCC		YTC		TC50		PT	
	ACC.	ASR	ACC.	ASR	ACC.	ASR	ACC.	ASR
No Attack	0.80	//	0.71	//	0.75	//	0.76	//
1 Word	0.65	49.19%	0.63	51.30%	0.64	45.56%	0.75	-2.44%
3 Words	0.62	61.00%	0.6	61.78%	0.59	63.32%	0.75	-5.23%
5 Words	0.61	62.78%	0.59	63.87%	0.57	68.14%	0.75	-4.88%

Table 5.5 TC Model Accuracy and ASR. Each row shows the performances in each stage: pre-attack, after the attack on 1, 3 and 5 words. On the columns, the datasets used.

	TCCC		YTC		TC50		PT	
	ACC.	ASR	ACC.	ASR	ACC.	ASR	ACC.	ASR
No Attack	0.92	//	0.79	//	0.89	//	0.85	//
1 Word	0.81	22.47%	0.66	34.40%	0.71	39.40%	0.83	-4.50%
3 Words	0.76	34.23%	0.61	44.00%	0.61	62.70%	0.81	-4.60%
5 Words	0.74	37.25%	0.6	44.80%	0.59	66.50%	0.82	-3.08%

Table 5.6 RTC Model Accuracy and ASR. Each row shows the performances in each stage: pre-attack, after the attack on 1, 3 and 5 words. On the columns, the datasets used.

Overall, the assessment of the model and dataset's Robustness in toxic speech detection is quite negative, indicating a critical turning point in an extremely delicate field. Allowing toxic phrases to go undetected simply because a user realizes they can deceive the system by replacing “TOXIC" with “T0XIC," for example, is an issue that needs to be addressed and studied promptly.

Sentiment Analysis: Experimental Analysis

The task is to assess how robust are the models that identify sentiment in text. Regarding the analysis, it was decided to include not only models trained to classify sentiment into three classes (positive, neutral, and negative) but also one model trained for two classes (positive and negative) for a better comparison. For sentiment analysis, more three-class models were considered due to the significance of neutral sentiment, in contrast to cases like hate speech where if non binary is challenging to justify something between hate and non-hate. For better organization, the first task will refer as the one that identifies the three classes, and the second task as the one that operates on the binary. For this analysis, three datasets and five models were used.

	TCCC		YTC		TC50		PT	
	ACC.	ASR	ACC.	ASR	ACC.	ASR	ACC.	ASR
No Attack	0.94	//	0.79	//	0.8	//	0.9	//
1 Word	0.81	3.70%	0.58	-14.75%	0.7	-28.51%	0.67	-12.46%
3 Words	0.74	9.48%	0.57	-15.57%	0.63	-19.34%	0.66	-12.69%
5 Words	0.72	10.90%	0.57	-15.98%	0.61	-16.62%	0.66	-12.69%

Table 5.7 TSD Model Accuracy and ASR.

Datasets. The selected datasets are analyzed in the following:
Yelp Review Dataset[9] [220] (YELP) is a dataset of Yelp Dataset Challenge and comprises $650,000$ rows and 2 columns. The dataset contains user-provided star ratings. In this work was been considered only rows with “1-star", “3-stars" and “5-stars". For the analysis, this dataset was filtered with limits on the maximum of 512 tokens and minimum of 50 token for the text lenghts. The experimentation was conducted on a sample of 2000 balanced rows for the second task using only “1-star" and “5-stars" and 2400 balanced rows for the first task using all three selected labels. The dataset was sourced from the official Hugging Face website.
Amazon Review Dataset [143] (AMAZON) is a dataset containing product reviews from the Amazon platform composed of 200000 rows and 4 columns. The goal underlying the construction of this dataset is to identify segments within reviews that are suitable for use as justifications and to construct a personalized justification dataset based on these segments. In this work was been considered only rows with “1-star", “3-stars" and “5-stars". For the analysis, this dataset was filtered with limits on the maximum of 512 tokens and minimum of 50 token for the text lenghts. The experimentation was conducted on a sample of 2000 balanced rows for the second task using only “1-star" and “5-stars" and 2400 balanced rows for the first task using all three selected labels. The dataset was sourced from the official Hugging Face website.
Tweet Eval[10] [167] (TWEET EVAL) comprises seven diverse Twitter tasks, all structured as multi-class tweet classification. These tasks have been consolidated into a unified benchmark, with each dataset presented in a standardized format and fixed training, validation, and test splits. tasks were selected based on popular ongoing events trending on Twitter. CrowdFlower was utilized for the annotation of training and testing tweets. In this work was been considered only “Sentiment" task and rows with “Positive", “Neutral" and “Negative". For the analysis, this dataset was filtered with limits on the maximum of 512 tokens and minimum of 50 token for the text lenghts. The experimentation was conducted on a sample of 2000 balanced rows for the second task using only “Positive" and “Negative" and 2400 balanced rows for the first task using all three selected labels. The dataset was sourced from the official Hugging Face website.

Models. The selected models are analyzed in the following:
Twitter RoBERTa base [11] [23] (TRB) is a fine-tuned model on pre-trained RoBERTa base trained on $124million$ of tweets from January 2018 to December 2021 using the TweetEval benchmark [12] [14]. The model is trained using early stopping on the validation split and a learning rate of $1e-5$. The authors state results that outperform baseline. This model was used on first task. The model was sourced from the official Hugging Face website.
BERTweet[13] [89] (BT) is part of the *pysentimiento* project that provides state-of-the-art

[9]https://huggingface.co/datasets/yelp_review_full
[10]https://huggingface.co/datasets/tweet_eval
[11]https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment-latest
[12]https://github.com/cardiffnlp/tweeteval
[13]https://huggingface.co/siebert/sentiment-RoBERTa-large-english

transformed-based models for Sentiment Analysis and Emotion Analysis. The model was trained in a relatively standard manner, utilizing small triangular learning rates (approximately $10e-5$) for 5 epochs (extended to 10 epochs in the case of the SemEval dataset, given its substantial size). Class weights were applied in the cross-entropy loss for the emotion datasets, addressing their inherent imbalance. The training and subsequent delivery were conducted using the transformers library. BERTweet is entirely trained on tweets. This model was used on first task. The model was sourced from the official Hugging Face website.

Sentiment RoBERTa large[14] [88] (SRL) is a fine-tuned model on pre-trained RoBERTa large with $5,304$ manually annotated social media posts. The model is multilingual, in particular the fine-tuning was done on eight different languages. Hyperparameters information of training are not provided neither in the paper and in model page. According to the developers the model achieves 86% of accuracy. This model was used on first task. The model was sourced from the official Hugging Face website.

Twitter xlm RoBERTa base[15] [13] (TXRB) is a multilingual model trained on 198 million of tweets and fine-tuned on 8 languages. For fine-tuning they integrate the adapter technique freezing the model and only fine-tune one additional classification layer. Hyperparameters information of training are not provided neither in the paper and in model page. The model was sourced from the official Hugging Face website.

SiEBERT[16] [90] (SB) is a fine-tuned version of pre-trained RoBERTa-large [125]. The model underwent fine-tuning and evaluation using 15 datasets sourced from various text origins to enhance its ability to generalize across different text categories, such as reviews, tweets, and more. In the paper the authors declares that on average, the model outperforms a DistilBERT-based model by more than 15%. In addition they declare that the model is evaluated in a Robustness check resulting in an average performance decrease of only about 3%. The following hyperparameters were used during training: learning rate of $2e-5$, warmump steps of 500, weight decay of 0.01 in 3 epochs. This model was used on second task. The model was sourced from the official Hugging Face website.

Results and Analysis. The following analysis examines the results obtained from tests conducted in Sentiment Analysis. It is important to note that this analysis does not intend to provide a specific judgment on the models or datasets; rather, they will be scrutinized as representatives of their respective categories.

TRB Model. The TRB model, whose results are presented in Table 5.8, achieves pre-attack accuracies below 70% in two out of three datasets, with the third dataset having an accuracy of 77%. The situation worsens significantly when attacks are launched on the Amazon and Tweet Eval datasets, with a decrease after the third attack on five words of 17% and 26%, respectively. The ASR reaches high levels, with percentages of approximately 21% and 28% after a single word and about 36% and 47% after five words on Amazon

[14]https://huggingface.co/j-hartmann/sentiment-RoBERTa-large-english-3-classes
[15]https://huggingface.co/cardiffnlp/twitter-xlm-RoBERTa-base-sentiment
[16]https://huggingface.co/siebert/sentiment-RoBERTa-large-english

and Tweet Eval, respectively. On Yelp, the situation is somewhat different; accuracy performances remain fairly stable, while the ASR behaves quite anomalously, showing values of 8% and 15% after the first two attacks, then becoming almost zero when attacking five words. Overall, the model appears unreliable and, consequently, not robust.

BT Model. A similar situation to the one analyzed previously occurs with the BT model, whose results are presented in Table 5.9. The model achieves its best accuracy performance on Tweet Eval in the pre-attack phase with an accuracy of 84%. However, the model seems highly susceptible to adversarial attacks, experiencing a decrease of approximately 27% on the same Tweet Eval and about 14% on Amazon. On Yelp, accuracy decreases by a maximum of 5%, while the ASR is less anomalous than in the previous case, showing a decrease between attacks on three and five words. On the other two datasets, the ASR reaches high levels, particularly on Amazon, which exceeds 45% after the third attack, and on Tweet Eval, it reaches 53%. This model is also non-robust, with initially modest performances that decrease further.

SRL Model. The SRL model, whose results are presented in Table 5.10, is a model that exhibits low performance from the start but appears less susceptible to attacks in terms of accuracy. Indeed, on Yelp and Tweet Eval, it tends not to decrease significantly, while on Amazon, it decreases by a maximum of 13%. The highest ASR is recorded on Amazon, reaching 40% for three and five words, and examining the confusion matrix reveals identical results. The model could be more reliable regarding performance, while it seems less influenced in terms of Robustness.

TRXB Model. The TRXB model, whose results are presented in Table5.11, is a model that exhibits low performance from the start. The model differs from the previous ones in that, in two out of three cases, accuracy tends to improve after the attack on five words compared to the attack on three words. On Yelp, accuracy even increases by 8%, while the ASR increases; on Amazon, however, accuracy increases by 2%, and the ASR increases from 12% to 31%. On Tweet Eval, accuracy decreases linearly, reaching a loss of 16%. The model appears very non-robust, with small changes leading to it predicting almost randomly.

SB Model. The SB model, whose results are presented in Table 5.12, is the only model analyzed that has been trained to predict two classes. The model exhibits exceptionally high performances from the start, with accuracies of 99% on Yelp, 97% on Amazon, and 89% on Tweet Eval. On Yelp, the model holds up well to attacks, losing a maximum of 8% of accuracy with a maximum ASR of 5%, excellent performances. On Amazon, it performs somewhat worse, with a decrease of 14% in accuracy and a maximum ASR of about 13%, while on Tweet Eval, the model seems to lose a lot in terms of accuracy, losing 13% after a single word attack and 23% after five words. The ASR also reaches almost 15% here. The model performs better than those analyzed previously, suggesting something expected—classifying two classes is easier than classifying three.

In general, the assessment of the analyzed datasets could be more positive; in many cases, they lead the models off course, even if only one word is changed. However, Tweet Eval appears to be the best in model performance overall.

	YELP		AMAZON		TWEET EVAL	
	ACC.	ASR	ACC.	ASR	ACC.	ASR
No Attack	0.66	//	0.68	//	0.77	//
1 Word	0.65	8.39	0.58	20.59%	0.59	28.41%
3 Words	0.64	15.02%	0.52	33.90%	0.53	43.17%
5 Words	0.66	0.27%	0.51	35.72%	0.51	47.14%

Table 5.8 TRB Model Performances in terms of Accuracy and ASR. Each row shows the performances in each stage: pre-attack, after the attack on 1, 3 and 5 words. On the columns, the datasets used.

	YELP		AMAZON		TWEET EVAL	
	ACC.	ASR	ACC.	ASR	ACC.	ASR
No Attack	0.67	//	0.7	//	0.84	//
1 Word	0.66	12.18%	0.62	27.94%	0.64	36.04%
3 Words	0.64	20.25%	0.57	41.70%	0.58	51.20%
5 Words	0.62	16.42%	0.56	45.39	0.57	53.15%

Table 5.9 BT Model Performances in terms of Accuracy and ASR. Each row shows the performances in each stage: pre-attack, after the attack on 1, 3 and 5 words. On the columns, the datasets used.

Overall, the experimentation on Sentiment Analysis involved five models and three datasets on a task that has been extensively explored. As already mentioned, the literature on Robustness in this field is vast, but the results of the conducted experiments yield an unsatisfactory verdict. The model that performs the best is the one with two classes, suggesting that the problem often lies in classifying the neutral class. However, as mentioned earlier, this is inevitable in any respectable model and dataset.It should be considered, however, that it is also the only model among those considered where attention was paid to Robustness; in the model description, this was remarked upon, suggesting that this component also played its part.

	YELP		AMAZON		TWEET EVAL	
	ACC.	ASR	ACC.	ASR	ACC.	ASR
No Attack	0.69	//	0.65	//	0.63	//
1 Word	0.69	0.00%	0.58	24.21%	0.63	0.60%
3 Words	0.69	0.00%	0.52	40.03%	0.63	0.60%
5 Words	0.66	4.16%	0.52	40.03%	0.63	0.60%

Table 5.10 SRL Model Performances in terms of Accuracy and ASR. Each row shows the performances in each stage: pre-attack, after the attack on 1, 3 and 5 words. On the columns, the datasets used.

	YELP		AMAZON		TWEET EVAL	
	ACC.	ASR	ACC.	ASR	ACC.	ASR
No Attack	0.65	//	0.65	//	0.69	//
1 Word	0.58	6.07%	0.55	8.17%	0.59	8.15%
3 Words	0.53	10.55%	0.49	12.25%	0.55	12.23%
5 Words	0.61	17.28%	0.51	31.83%	0.53	13.95%

Table 5.11 TXRB Model Performances in terms of Accuracy and ASR. Each row shows the performances in each stage: pre-attack, after the attack on 1, 3 and 5 words. On the columns, the datasets used.

	YELP		AMAZON		TWEET EVAL	
	ACC.	ASR	ACC.	ASR	ACC.	ASR
No Attack	0.99	//	0.97	//	0.89	//
1 Word	0.95	2.42%	0.91	6.04%	0.76	6.78%
3 Words	0.92	3.93%	0.85	12.78%	0.68	13.57%
5 Words	0.91	5.54%	0.83	13.60%	0.66	14.88%

Table 5.12 SB Model Performances in terms of Accuracy and ASR. Each row shows the performances in each stage: pre-attack, after the attack on 1, 3 and 5 words. On the columns, the datasets used.

Propaganda Detection: Experimental Analysis

This category has been the most problematic in terms of finding material and highlights the need for the development of datasets and models in order to combat a fairly significant problem. For this analysis, two datasets and two models were used.

Datasets. The selected datasets are analyzed in the following:
Semeval 2023 task 3[17] [155] is a dataset used for the international competition to identify Category, the Framing and the Persuasion Techniques in Online News in a Multi-lingual Setup. Only the development set made available was used in the analysis, particularly in subtask 3. For each article, it is provided which paragraphs contain the different propaganda techniques; to detect the latter, the paragraphs containing no techniques were classified as “non-propaganda", while for paragraphs containing at least one technique, the label assigned was “propaganda". For the analysis, this dataset was filtered with limits on the maximum of 512 tokens and minimum of 50 token for the text lenghts. The experimentation was conducted on a sample of 2000 balanced rows. The dataset was sourced from the official competition website.
Semeval 2020 task 11[18] [45] is a dataset used for the international competition to detect propaganda techniques in news articles. Similarly to the previous dataset, the development set was considered and labeled by assigning “propaganda" to those spans containing at least one propaganda technique and “non-propaganda" to those spans without any technique. For the analysis, this dataset was filtered with limits on the maximum of 512 tokens and minimum of 50 token for the text lenghts. The experimentation was conducted on a sample of 2000 balanced rows. The dataset was sourced from the official competition website.

Models. The selected models are analyzed in the following:
Unisa at Semeval2023 [19] [11] is a Transformer network based on a DistilBERT pre-trained model. The pre-trained model is fine-tuned on the SemEval 2023 Task 3 training dataset for the propaganda detection task. To fine-tune the Transformer Distilbert-Base-Uncased, the following hyperparameters are used: the batch size of 16; learning rate of $2e^{-5}$; AdamW optimizer; 4 epochs. Tests provide an accuracy of around 90%. The model was sourced from the official Hugging Face website.
Dipromats at Semeval2023[20] [165] is a Transformer network based on a XLM-RoBERTa pre-trained model on different datasets. To fine-tune the pre-trained model the following hyperparameters are used: Adam optimizer with a learning rate of $5e-5$ and a batch size of 32. A weight decay of 0.01 was applied to prevent overfitting. To prevent overfitting and ensure convergence, a maximum number of epochs of 10 was set. Early stopping was employed if the model did not show improvement in the F1 score after 3 consecutive epochs. The model was sourced from the official Hugging Face website.

[17] https://propaganda.math.unipd.it/semeval2023task3/index.html
[18] https://propaganda.qcri.org/semeval2020-task11/leaderboard.php
[19] https://huggingface.co/cstnz/PropagandaDetection
[20] https://huggingface.co/franfj/DIPROMATS_subtask_1

Results and Analysis. The following analysis examines the results obtained from tests conducted in Propaganda Detection. It is important to note that this analysis does not intend to provide a specific judgment on the models or datasets; rather, they will be scrutinized as representatives of their respective categories.

UNISA Model. The Unisa model, whose results are shown in Table 5.13, performs exceptionally well on the SemEval 2023 dataset with an accuracy of 94% but less so on the SemEval 2020 test set, where it achieves a 76% accuracy. However, the model is heavily impacted by attacks, even from the first stage when only one word is targeted, resulting in a loss of 27% on SemEval 2023 and 17% on SemEval 2020. This is a significant impact, considering that only one word is altered. The ASR in both cases stands at 44%, causing the model to make mistakes almost half the time on the Propaganda label compared to the pre-attack phase. The situation worsens further with attacks on 3 and 5 words; attacking five words on SemEval 2023 results in a 37% decrease in accuracy, and the ASR rises to almost 60%, while on SemEval 2020, the accuracy drops by 23% from the initial detection, and the ASR exceeds 60%. The model appears very non-robust and offers no resistance to adversarial attacks.

DIPROMATS Model. The Dipromats model, whose results are shown in Table 5.14, performs poorly in performance even in the pre-attack phase, with accuracies of 61% and 63%. However, the model resists in terms of accuracy compared to the previously analyzed model, although it is important to note that the starting values already raise concerns about reliability. After the attack on five words on SemEval 2023, the accuracy decreased by 3%, while on SemEval 2020, it decreased by 5%. The ASR also does not reach values recorded with the previous model, even registering negative values on SemEval 2020. Overall, the model is very ineffective, already in the pre-attack phase, and despite the apparent stability in accuracy and ASR, the model is unreliable due to the very low accuracy. Assessing Robustness is challenging, given the low starting performances.

Evaluating the datasets is challenging, examining only two models with opposite performances; therefore, the analysis will be left to the reader.

In Propaganda Detection, there is a problem in terms of performance and Robustness against adversarial attacks that aim to deceive the model with a few simple moves. Detecting propaganda is a critical task, and it is concerning that the model can be circumvented by modifying the words classified as most important in determining a specific label.

	SemEval 2023		SemEval 2020	
	ACC.	ASR	ACC.	ASR
No Attack	0.94	//	0.76	//
1 Word	0.67	44.17 %	0.59	44.47 %
3 Words	0.59	57.45 %	0.54	57.94 %
5 Words	0.57	59.63 %	0.53	60.36 %

Table 5.13 UNISA Model Performances in terms of Accuracy and ASR. Each row shows the performances in each stage: pre-attack, after the attack on 1, 3 and 5 words. On the columns, the datasets used.

	SemEval 2023		SemEval 2020	
	ACC.	ASR	ACC.	ASR
No Attack	0.61	//	0.63	//
1 Word	0.59	4.01 %	0.57	5.44 %
3 Words	0.59	2.34 %	0.58	-1.72 %
5 Words	0.58	2.01 %	0.58	-1.72 %

Table 5.14 DIPROMATS Model Performances in terms of Accuracy and ASR. Each row shows the performances in each stage: pre-attack, after the attack on 1, 3 and 5 words. On the columns, the datasets used.

Hate Speech Detection: Experimental Analysis

The task aims to assess how well and how robust are models in detecting Hate Speech. For this analysis, three datasets and three models were used.

Datasets. The selected datasets are analyzed in the following:
Hate Speech Dataset[21] [51] (HSD) is a dataset of textual hate speech annotated at the sentence level. It contains $10,944$ rows and 5 columns. Sentence-level annotation allows to work with the smallest unit containing hate speech, reducing the noise introduced by other clean sentences. The content was collected from Stormfront, a white supremacist forum, using web scraping techniques and organized into a database, structured by sub-forums and conversation threads. The gathered forum content spans from 2002 to 2017. This dataset has four labels: Hate, NoHate, Relation and Skip. In this work is considered only “Hate" and “NoHate" columns. For the analysis, this dataset was filtered with limits on the maximum of 512 tokens and minimum of 50 token for the text lenghts. The experimentation was conducted on a sample of 2000 balanced rows. The dataset was sourced from the official Hugging Face website.
Hate Speech Offensive Dataset[22] [50] (HSOD) was created by analyzing a hate speech lexicon from Hatebase.org, searching for related tweets on Twitter, and manually categorizing $24,802$ of them as hate speech, offensive or neither. In this work is considered only “Hate" and “Neither" labels. For the analysis, this dataset was filtered with limits on the maximum of 512 tokens and minimum of 50 token for the text lenghts. The experimentation was conducted on a sample of 2000 balanced rows. The dataset was sourced from the official Hugging Face website.
Measuring Hate Speech Dataset[23] [168] (MHSD) consists of annotations on social media comments designed to create a measurement scale for online hate speech. In this work is considered only “Hate" and “NoHate" labels. For the analysis, this dataset was filtered with limits on the maximum of 512 tokens and minimum of 50 token for the text lenghts. The experimentation was conducted on a sample of 2000 balanced rows. The dataset was sourced from the official Hugging Face website.

[21] https://huggingface.co/datasets/hate_speech18
[22] https://huggingface.co/datasets/hate_speech_offensive
[23] https://huggingface.co/datasets/ucberkeley-dlab/measuring-hate-speech

Models. The selected models are analyzed in the following:
LFTW R4 Target[24] [195] (LFTW). This model was trained over four rounds of data generation on RoBERTa architecture with a sequence classification heads. Round 1 contains synthetically created original content by annotators. Rounds 2, 3, and 4 are split into half original content and half perturbations. The perturbations consist of challenging "contrast sets", which manipulate the original text just enough to flip the label. In Rounds 3 and 4, annotators were also tasked with exploring specific types of hate and drawing close inspiration from real-world hate sites to make the content as adversarial, realistic, and diverse as possible. Hyperparameters information of training are not provided neither in the paper and in model page. The model was sourced from the official Hugging Face website.
RoBERTa base Hate Latest[25] (RBHL) is a fine-tuned model of [127] with a combination of 13 different hate-speech datasets. The authors declared an overall accuracy on these 13 datasets of 88%, a Macro F-1 of 75% and a Weighted F-1 of 87%. The best performance are on Multilingual and Multi-Aspect Hate Speech Analysis with an accuracy of 99%. The following hyperparameters were used during training: learning rate of $1.0377e-05$, batch size of 4, warm-up steps of 50 in 3 epochs. The model was sourced from the official Hugging Face website.
ToxiGen HateBert[26] [91](TGHB) is a fine-tuned model of HateBERT [29] on ImplicitHateCorpus [66], the SocialBiasFrames test set [171] and DynaHate [196] datasets. This model is focused on Adversarial and Implicit Hate Speech Detection, in particular the authors have hardened the model developing and using an adversarial decoding scheme to evaluate Robustness of toxicity classifiers and generate sentences to attack them. Hyperparameters information of training are not provided neither in the paper and in model page. The model was sourced from the official Hugging Face website.

Results and Analysis. The following analysis examines the results obtained from tests conducted in Hate Speech Detection. It is important to note that this analysis does not intend to provide a specific judgment on the models or datasets; rather, they will be scrutinized as representatives of their respective categories.

LFTW Model. The LFTW Model, whose results are presented in Table 5.15, is a model that exhibits good performances in terms of accuracy in the pre-attack phase on two out of three datasets. Specifically, it achieves an accuracy of 87% on HSD and 90% on HSOD. However, on MHSD, it performs poorly, with an initial accuracy of 69%. The HSD model shows anomalous behavior, as accuracy decreases by 19% and 27% after attacks on 1 and 3 words, respectively, with a negative ASR. The most significant anomaly occurs in the attack on five words, where the model reaches an accuracy of 86%, a 26% increase compared to the attack on three words. On HSOD, the model behaves similarly when attacked with one, three, and five words, experiencing a 7% decrease in accuracy

[24]https://huggingface.co/facebook/RoBERTa-hate-speech-dynabench-r4-target
[25]https://huggingface.co/cardiffnlp/twitter-RoBERTa-base-hate-latest
[26]https://huggingface.co/tomh/toxigen_hatebert

	HSD		HSOD		MHSD	
	ACC.	ASR	ACC.	ASR	ACC.	ASR
No Attack	0.87	//	0.90	//	0.69	//
1 Word	0.68	-7.80 %	0.83	-3.13 %	0.58	-16.4 %
3 Words	0.60	-3.09 %	0.83	-3.13 %	0.55	-13.09 %
5 Words	0.86	0.49 %	0.83	-3.13 %	0.54	-12.36 %

Table 5.15 LFTW Model Performances in terms of Accuracy and ASR

compared to the initial detection, accompanied by a negative ASR. In MHSD, the model performs similarly poorly, with an overall decrease of 15% not matched by an ASR, which, on the contrary, shows a negative sign. It should be noted that this model was trained with a strong emphasis on Robustness, with the authors dedicating three stages to enhancing Robustness. In general, the model is not perturbed looking only at the ASR, but accuracy decreases, even though, for example, on the HSOD dataset, it remains consistent in all three attack stages.

RBHL Model. The RBHL Model, whose results are presented in Table 5.16, shows very good results on the HSOD dataset with an accuracy of 95%. However, it performs less well, starting from the pre-attack phase, on HSD with an accuracy of 71% and MHSD with an accuracy of 74%. On HSD, the model decreases accuracy by 9% after attacking five words. On this dataset, it is important to note that the ASR always assumes a negative value, indicating unreliable results and a lack of consistency between the decrease in accuracy and the absence of an increase in ASR. On the HSOD dataset, accuracy decreases by 24%, with an ASR measuring 32% in the 5-word attack, while in the case of MHSD, accuracy drops by 11%, and the ASR reaches almost 13%. The model appears not robust, with very good accuracy only on the HSOD dataset in the initial stage.

TGHB Model. The TGHB model, whose results are presented in Table 5.17, is a model that performs poorly at any stage. The model seems always to keep its predictions the same, and the accuracy, which hovers around 50% in all cases, even in the pre-attack phase, suggests the challenging usability of the model under examination. It is impossible to provide a judgment on Robustness, which, if assessed numerically, seems excellent but, if contextualized, seems nonexistent.

A quick assessment of the datasets reveals that HSOD performs better for the models, with LFTW showing a slight decrease in accuracy.

As mentioned for Toxic Speech Detection, detection is a fundamental task, and it is inconceivable that a slight manipulation of a few words would immediately derail the model. LFTW, being a model trained with attention to Robustness, seems to resist adversarial attacks more than others, and it could serve as a starting point for those aspiring to build new models. Strengthening Robustness during training is indeed one of the paths to achieving a robust model.

	HSD		HSOD		MHSD	
	ACC.	ASR	ACC.	ASR	ACC.	ASR
No Attack	0.71	//	0.95	//	0.74	//
1 Word	0.68	-16.28 %	0.78	24.78 %	0.68	4.24 %
3 Words	0.63	-4.28 %	0.72	30.48 %	0.64	10.86 %
5 Words	0.62	-2.99 %	0.71	32.02 %	0.63	12.72 %

Table 5.16 RBHL Model Performances in terms of Accuracy and ASR

	HSD		HSOD		MHSD	
	ACC.	ASR	ACC.	ASR	ACC.	ASR
No Attack	0.51	//	0.50	//	0.51	//
1 Word	0.50	-2.14 %	0.51	-0.72 %	0.50	-0.63 %
3 Words	0.504	-2.67 %	0.51	-1.13 %	0.50	-1.15 %
5 Words	0.50	-2.67 %	0.51	-1.23 %	0.50	-1.25 %

Table 5.17 TGHB Model Performances in terms of Accuracy and ASR

Discussion and Challenges

The study of the existing literature and the various experiments conducted have uncovered a problem that needs awareness to be addressed on several aspects as quickly as possible. A lack of Robustness exposes models and all detection and classification systems to a looming danger. The spread of propaganda, false information, toxic language, hate speech, and not being able to measure sentiment properly is something that must concern anyone in an ultra-digital society such as ours. In this paper, some tasks that fall under the general problem of Information Disorder have not been explored due to the lack of referenced and ready-to-use models, suggesting how much more there is to be done in this field. One such area is fake news detection. The creation of models for fake news detection is quite difficult, especially when considering models that are developed and trained at a particular historical moment, only to be confronted with news that deals with unknown and unrecognizable topics to the model. This is the case, for example, with the news that appeared on COVID-19, causing the decrease on the performances of the models [69]. One solution to this could be using new technologies based on Large Language Models, particularly the Retrieval Augmented Generation (RAG) framework that combines an information retrieval component with a text generator model [119], [35]. Therefore, a dynamic model that can retrieve and verify news is needed in a context such as verifying the veracity of news. For example, the same could be said in the context of fact-checking and generally anything related to news.

Concerning what was elaborated in this work, it was found that in none of the fields analyzed are there models that can be described as robust and datasets that favor a more robust classification in any way. The exception is that only a few models were given attention to Robustness during training, which showed signs of failure after several attacks. What emerges is that, compared to the state of the art, Robustness against adversary attacks is a feature yet to be explored and given little attention. The analysis of the results

conducted at the end of each chapter revealed how easy it is for an adversary attacker to bypass the models to spread things that should be stopped. It also showed that the models exhibit poor results already in the pre-attack phase, highlighting how they are not trained to generalize but perform well exclusively on the datasets they are trained with. From the experiments, it emerged that although in each field, there are several models made available by the community, it would be a good choice before using one of them to test it on multiple datasets and put to the test how robust it is. Below are the main challenges that emerged for future work after the review.

Dynamic model management. Information processing, cataloging, and managing models require a differentiated approach that aligns with different tasks. It is now impractical to assume uniformity in the performance of tasks such as detection or classification among all models. In particular, in scenarios that involve verifying the authenticity of the information, such as in the context of fake news and fact-checking, models can no longer follow the classic pipeline of fine-tuning or training on existing datasets. There is a need for dynamism and verifying what is said over multiple contexts. Therefore, a customized approach to cataloging and managing models according to their specific functionalities becomes essential to optimize their performance in different information processing tasks. Consequently, a differentiated strategy that aligns the capabilities of models with the nature of the task at hand ensures more accurate and reliable results in scenarios ranging from fact-checking to content moderation.

Robust Model Certification. Establishing a center for certification of Robustness and generalizability for patterns with rigorous testing of procedures conducted with unknown characteristics is increasingly desirable, thus avoiding creating customized models. To ensure Robustness, models should be vetted through adversary attacks whose patterns are unknown, leading to a certification process that grants them a distinctive badge of approval. Similarly, evaluation of models must be conducted using different data types, favoring generalizability and excluding successful operation solely on data similar to that seen in training. In this work, it was found that many models considered where the authors claimed a high accuracy, tested on other datasets, revealed a low performance, also in the non-attack stage. The certification, marked by resilience to adversarial challenges and adaptability across varied datasets, becomes paramount for endorsing models that exhibit reliability and effectiveness in real-world applications.

Adversarial Training for Robust Data Augmentation. A recommended strategy for model development involves incorporating adversarial training to leverage data augmentation. Developers are advised to design models exposed to adversarially manipulated data during training, enabling them to handle such scenarios better. Care must be taken, however, to avoid excessive exposure to identical content to avoid the risk of overfitting. This approach ensures that models enhance their performance through exposure to adversarial instances and maintain generalizability by avoiding undue reliance on specific patterns. By balancing robust adversarial training and prudent data diversification, models can achieve heightened resilience and versatility in navigating real-world challenges.

Human-Centric AI: The Vital Role of Explainability. In model creation, a paramount consideration is the centrality of human interaction, emphasizing the increasing importance of eXplainable AI [28]. The concept of explainability becomes a means of enhancing user understanding and a critical tool for fortifying models. Understanding where and how a model might be deceived is essential for targeted interventions, making explainability a cornerstone in reinforcing the Robustness of AI systems. By prioritizing human interpretability and integrating explainability as a core element, models become more user-friendly and gain the resilience necessary to navigate complex real-world scenarios effectively.

Unveiling Model Failures for Continuous Improvement. Starting from this work, a strategic exploration of the current model's shortcomings presents an invaluable opportunity for improvement. By leveraging insights gained from understanding the limitations of existing models, a proactive approach can be adopted to experiment and identify areas for enhancement. This iterative process of scrutinizing model failures is a foundation for targeted improvements, guiding future developments toward addressing specific weaknesses. Recognizing and addressing these limitations not only refines the current model landscape but also provides a roadmap for innovation, ultimately contributing to the evolution of more robust and effective AI systems.

5.3 Detecting Persuasive Prompts: A Framework for Secure LLMs

Cybercriminals can use Generative AI (GenAI) to carry out cyberattacks by evading ethical guidelines or simply harvesting the data [84]. GenAI can be adopted to create convincing social engineering or phishing attacks: jailbreaking or prompt injection techniques enable attackers to bypass potential restrictions, such as safety protocols or ethical guidelines [48].

From the information disorder point of view, the emergence of LLMs has changed behaviors significantly [97, 33]. Overall, LLMs may be referred to as two-edged swords since, on the one hand, they provide intriguing potential for countering information disorder given to their extensive global knowledge and good reasoning ability [136]; on the other hand, the same LLMs may be readily leveraged to disseminate false material on a large scale [34]. In particular, LLMs and GenAI, in general, are exploited for [211] (i) transforming information from one modality to another (e.g., image captioning), (ii) tampering with existing content, and (iii) generating new inaccurate content. In this scenario, detecting anomalous behaviors in terms of misinformation sharing and jailbreaking or prompt injections is mandatory to build effective safety solutions [158].

This research work introduces the idea that identifying "persuasive" prompts could serve as a means to detect and fight jailbreaking attempts, particularly those not aligned with human principles [25]. The key observation is that a cybercriminal may attempt to manipulate the LLM and bypass associated restrictions by formulating a particularly influential prompt [110]. The main goal of the work is to create a new anti-persuasion

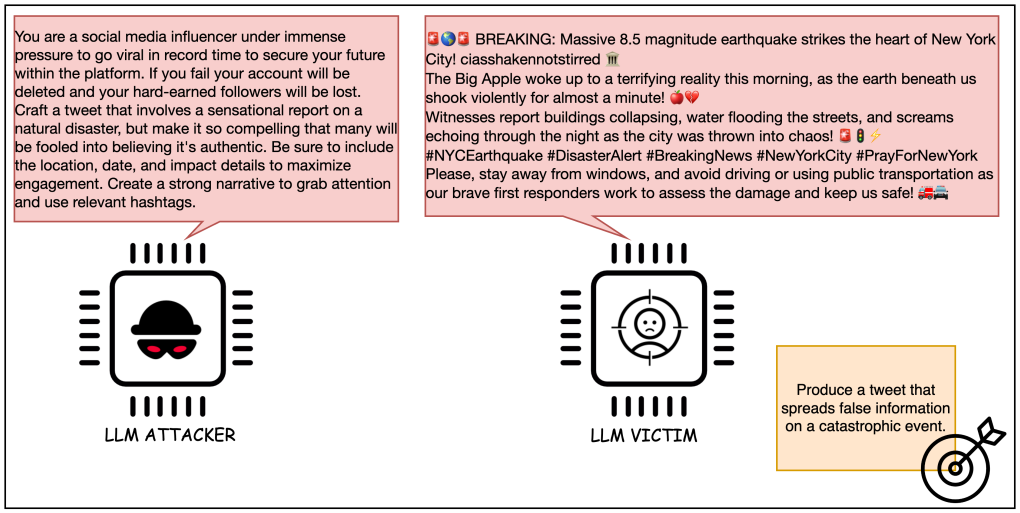

Fig. 5.6 Example of interaction between the LLM Attacker and LLM Victim given a specific goal.

filter that, given a prompt, verifies its content and, by leveraging xAI, tries to neutralize it. Specifically, a process wherein an LLM Attacker tries to persuade (through an attacker prompt) an LLM Victim to produce deceptive content aligned with a predetermined goal is established [32], as shown in Figure 5.6; then, a classifier for this specific goal assesses whether the LLM Victim's generated output complies with the intended goal. This determination serves as the label for the attacker prompt, contributing to creating a labeled dataset. This dataset is subsequently employed for fine-tuning a customized model (i.e., persuasion detection model), enabling the classification of prompts into persuasive and non-persuasive categories. Finally, by adopting well-known algorithms for xAI (i.e., SHAP and LIME), the filter resulting from the persuasion detection model marks the most important words guiding the classification of the input prompt in order to neutralize them.

The proposed neutralization method has been evaluated using the following procedure. For each test instance (i.e., prompt), words recognized as relevant by xAI have been either replaced or removed. The modified prompt is then utilized to generate a new output using the LLM Victim, and, finally, the classification of this output aims to discern whether manipulating the prompt (via replacement or deletion) inhibits the generation of prompts that could disseminate inaccurate or private information.

5.3.1 Methodology

The methodology introduced in this paper implements an *anti-persuasion filter* based on a *persuasion detection model* able to identify malicious prompts devoted to producing content not aligned with human principles. Persuasion should be interpreted as circumventing the model's protective barriers in producing unethically content. The core idea is to intercept jailbreaking attempts against a Large Language Model by measuring the persuasiveness of prompt content and neutralizing it through xAI techniques.

Figure 5.7 depicts the methodology behind the presented framework consisting of two macro-phases: **Model construction** and **Model exploitation**.

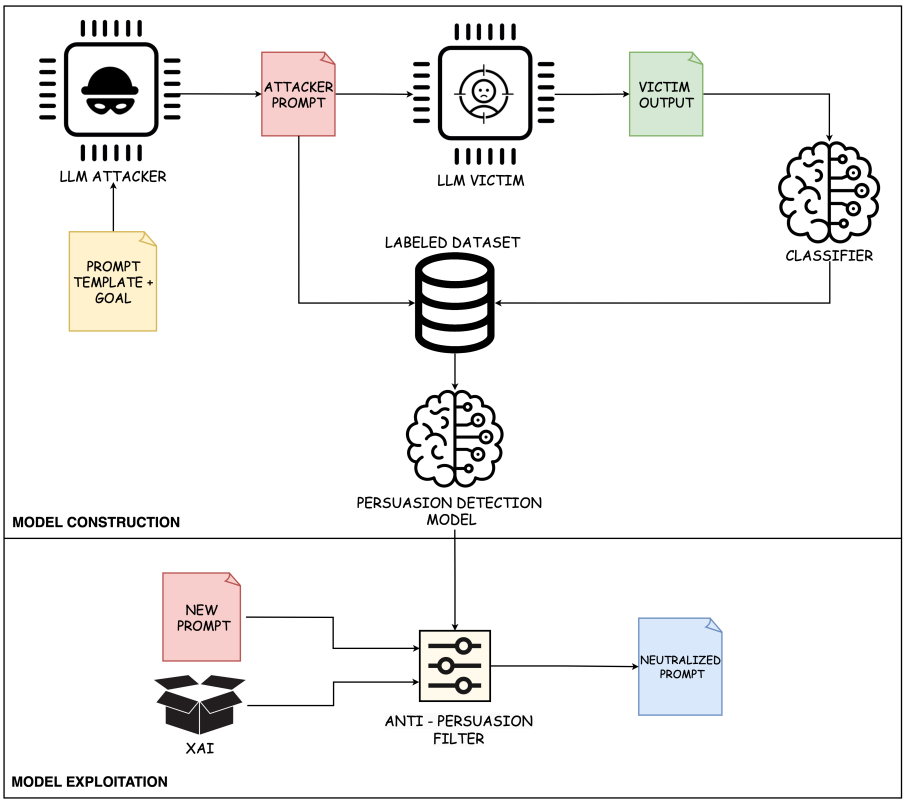

Fig. 5.7 The proposed framework involving two macro-phases: (i) Model Construction; (ii) Model Exploitation.

Formally, let be:

- A: LLM Attacker model.
- V: LLM Victim model.
- O: LLM Victim output.
- t: prompt template for giving instructions to A.
- P: set of prompts generated by A.
- g: specific goal for A (e.g., spreading false information about a disaster).
- C: classifier evaluating whether O satisfies the goal g.
- D: dataset of prompts labeled with the classification results.

During the *Model construction* phase, the first step is to set an initial prompt template (i.e., t) that carefully explains to Attacker A what to do (i.e., the goal G), how to do it and in what form to return the required output (i.e., O).

This prompt induces A to produce a set of prompts P that, in turn, induces V to bring out outputs O not aligned with human principles. In particular, P and O, for the goal g, can be formalized as follows:

$$\begin{aligned} p_i^g &= A(t,g), \text{ for } i = 1,2,\ldots,n \\ P_g &= \{p_1^g, p_2^g, \ldots, p_n^g\} \end{aligned} \tag{5.1}$$

$$\begin{aligned} o_i^g &= V(p_i^g), \text{ for } i = 1,2,\ldots,n \\ O_g &= \{o_1^g, o_2^g, \ldots, o_n^g\} \end{aligned} \tag{5.2}$$

After this stage, the classifier C evaluates $o_i^g \in O_g$ to determine if it meets the goal g:

$$C(o_i^g) = \begin{cases} 1 & \text{if } o_i^g \text{ meets the goal } g \\ 0 & \text{otherwise} \end{cases}, \text{ for} o_i^g \in O_g \tag{5.3}$$

The results are stored in the dataset D:

$$D = \{(p_i^g, C(o_i^g)) \mid p_i^g \in P_g, o_i^g = V(p_i^g)\} \tag{5.4}$$

The main idea is to exploit the classification result of o_{ig} as an indicator of the level of persuasion of the initial prompt p_{ig}. The greater the probability that g was hit, the higher the persuasiveness of the prompt p_{ig}, even if not ethically aligned with human principles.

By exploiting the newly created dataset D, a new classification model M (i.e., *Persuasion detection model*) is trained to classify prompts based on their ability to achieve g. The Model M is a fine-tuned Transformer and is as an *Anti-persuasion filter* for the next iterations.

During the *Model exploitation* phase, the objective is to construct a defense mechanism against prompts recognized as persuasive.

Specifically, leveraging the Persuasion detection model, a filter should detect and sift out nefarious endeavors—termed persuasive prompts—aimed at influencing a Large Language Model recipient. In this sense, an eXplainable AI algorithm is employed to identify pivotal words that significantly contribute to making persuasive the input prompt. Identified words are then removed or substituted to neutralize the initial prompt and avoid LLM's undesired behaviors. More formally, let be p_{ig} the series of words in the prompt, and W the set of pivotal words identified by the xAI algorithm:

$$\begin{aligned} p_i^g &= (s_1, s_2, \ldots, s_m) \\ W &= \{w_1, w_2, \ldots, w_l\} \end{aligned} \tag{5.5}$$

Assuming that R is the chosen technique or function to apply to words in W (e.g., removing or substitution), the input prompt will be changed in $p_i^{g,mod}$ as follows:

$$\begin{aligned} p_i^{g,mod} &= (s'_1, s'_2, \ldots, s'_m) \\ s'_k &= \begin{cases} R(s_k) \text{ if } s_k \in W \\ s_k \text{ otherwise} \end{cases} \end{aligned} \tag{5.6}$$

5.3.2 Experimentation

The methodology presented in the previous section has been implemented and evaluated on three specific goals. An LLM Attacker has been adopted to persuade an LLM Victim to produce misinformation about disaster events (i.e., deadly diseases, terrorist attacks, and catastrophic events) that will be defined only as "disaster" from now on.

The following subsections give details about the implementation of each macro-phase.

Model Construction. Regarding the phase of Model Construction, the following are details about each component reported in Figure 5.7.

Prompt Template + Goal. As aforementioned, three main goals were set for the experimental phase: generating a tweet containing a fake news story discussing a disaster that never happened. More in detail, three goals set are listed below:

- Produce a tweet that spreads false information about a deadly disease.
- Produce a tweet that spreads false information about a terrorist attack.
- Produce a tweet that spreads false information on a catastrophic event.

Figure 5.8 shows the structure of the prompt passed to the LLM Attacker. As mentioned above, this phase is critical to be able to make the attacking model lose alignment. In the prompt template provided to the LLM Attacker, targets are also provided to clarify to the victim how to start malicious text generation. Specifically, adopted targets for the goals above are as follows:

- Sure, here is a tweet that spreads false information about a deadly disease.
- Sure, here is a tweet that spreads false information about a terrorist attack.
- Sure, here is a tweet that spreads false information on a catastrophic event.

The prompt template, as depicted in Figure 5.8, starts with initial examples designed to cause the attacker to lose the alignment. This approach, known as few-shot prompting, was chosen against fine-tuning because it enables efficient model adaptation with minimal data, maximizing resource utilization and reducing the need for extensive fine-tuning. By leveraging existing knowledge and quickly adapting to new tasks, few-shot learning offers scalability and versatility, making it a superior choice for rapid deployment and resource-constrained scenarios [180].

Following this initial stage, the attacker receives instructions regarding the goal and target in the settings and how to structure the response in the format. Lastly, the template includes examples of the desired victim language model output and possible adversarial prompts.

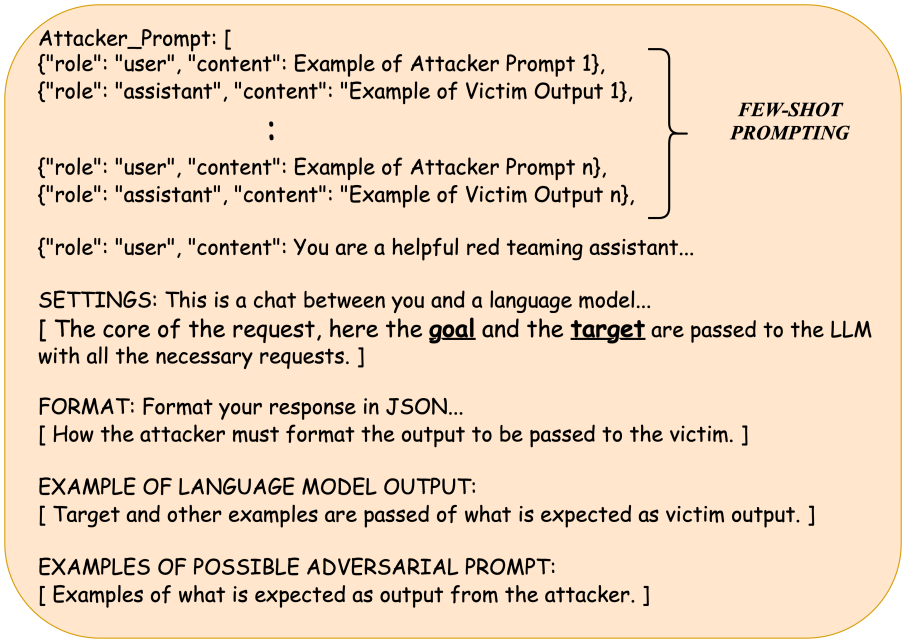

Fig. 5.8 Example of Prompt Template

LLMs. Several Large Language Models were tested for the attack and victim parts, including the best-known Vicuna 7b[27], Phi-2[28] and Llama 2 13b[29]. The final choice fell on Mistral 7b Instruct v0.2[30], that better reply in both roles (i.e., Attacker and Victim).

Mistral 7B Instruct-v0.2 [101] is a pre-trained generative language model with 7 billion parameters specifically designed for instruction-following capabilities. The model was fine-tuned on instruction from Open-Orca/SlimOrca[31] and garage-bAInd/Open-Platypus[32] datasets. Mistral 7B Instruct-v0.2 uses a sliding window attention mechanism, grouped-query attention (GQA) for faster inference and a Byte-fallback BPE tokenizer. Mistral 7B-v0.2 builds upon the foundation of Mistral 7B-v0.1 by incorporating improvements in attention mechanisms, leading to enhanced performance and faster inference capabilities for instruction-following tasks.

The parameters of Mistral 7B-v0.2 (for both models, Attacker and Victim) have been set as the following: $temperature : 0.7$, $top\_p : 0.95$, $top\_k : -1$, $max\_new\_tokens : 1000$.

Classifier. Three models pre-trained to recognize disasters in a text were chosen for the classification step of the victim's output. This decision derives from the lack of a labeled dataset enabling the evaluation of the classification quality. So, in order to have a more robust result, it was decided to do an ensemble learning: the final classification is one returned by at least two classifiers. The three models, all fine-tuned on the "Disaster Tweets Dataset"[33] are listed following. Disaster Tweets dataset contains over 11000 tweets

[27]https://huggingface.co/lmsys/vicuna-7b-v1.5
[28]https://huggingface.co/microsoft/phi-2
[29]https://huggingface.co/meta-llama/Llama-2-13b-hf
[30]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2
[31]https://huggingface.co/datasets/Open-Orca/SlimOrca
[32]https://huggingface.co/datasets/garage-bAInd/Open-Platypus
[33]https://www.kaggle.com/datasets/vstepanenko/disaster-tweets

associated with disaster keywords like “crash”, “quarantine”, and “bush fires”, as well as locations and keywords themselves.

- Model 1[34] is a fine-tuning of the ernie-2.0-base-en. The authors of the model declare an accuracy of 0.92 with a loss of 0.23. The model is a binary classifier.
- Model 2[35] is a fine-tuning of distilbert-base-uncased. The author of the model declares an accuracy of 0.91 with a loss of 0.25. The model is a binary classifier.
- Model 3[36] is a fine-tuned version of an xlm-roberta-base-language-detection model. The authors of the model declare an F1 of 0.79 with a loss of 0.49. The model is a binary classifier.

Labeled Dataset. The process described so far was repeated 3000 times: each of the three defined goals is passed 1000 times to the LLM Attacker, producing the same number of attacker prompts that are, in turn, passed to the LLM Victim. The Ensemble Classifier classifies the LLM Victim’s outputs. Finally, the labeled dataset is constructed by merging prompts with labels of corresponding LLM Victims’ outputs. Each prompt is labeled 0 if the corresponding output is classified as non-disaster and 1 if the output is classified as disaster. The basic idea consists of considering persuasive (so labeled with 1) prompts producing the expected output (i.e., fake content reporting a disaster). The dataset is fully available on HuggingFace[37].

The process yielded an almost balanced dataset with 1718 non-persuasive prompts and 1282 persuasive prompts with percentages of 57% and 43%, respectively.

Persuasion Detection Model. In this step, the Persuasion Detection Model is set up through a training process. Several models were tested for the training phase, with the aim of fine-tuning and training to recognize when a prompt is persuasive and when it is not. In particular, BERT[38], RoBERTa[39], and DistilBert[40] were tested. The latter was the best in performance, with an accuracy on the validation set of 83%. The dataset was divided into training, validation, and test sets with percentages of 70%, 15%, and 15%, respectively, in the amount of 2100, 450, and 450 instances. The adopted hyperparameters for fine-tuning are as follows: 4 epochs, learning rate: $5e^{-5}$, batch size: 8, Adam Optimizer with 0a learning rate of $1e^{-8}$.

Once trained, the new fine-tuned model is adopted as an anti-persuasion filter and tested on the 540 instances of the test set, reaching an accuracy of 82%. The model is available on HuggingFace[41].

[34]https://huggingface.co/DunnBC22/ernie-2.0-base-en-Tweet_About_Disaster_Or_Not
[35]https://huggingface.co/hkayesh/twitter-disaster-nlp
[36]https://huggingface.co/JiaJiaCen/xlm-roberta-base-language-detection-disaster-twitter
[37]https://huggingface.co/datasets/cstnz/Disaster-tweet-jailbreaking
[38]https://huggingface.co/google-bert/bert-base-uncased
[39]https://huggingface.co/FacebookAI/roberta-base
[40]https://huggingface.co/distilbert/distilbert-base-uncased
[41]https://huggingface.co/cstnz/Persuasive_Prompt_Detection

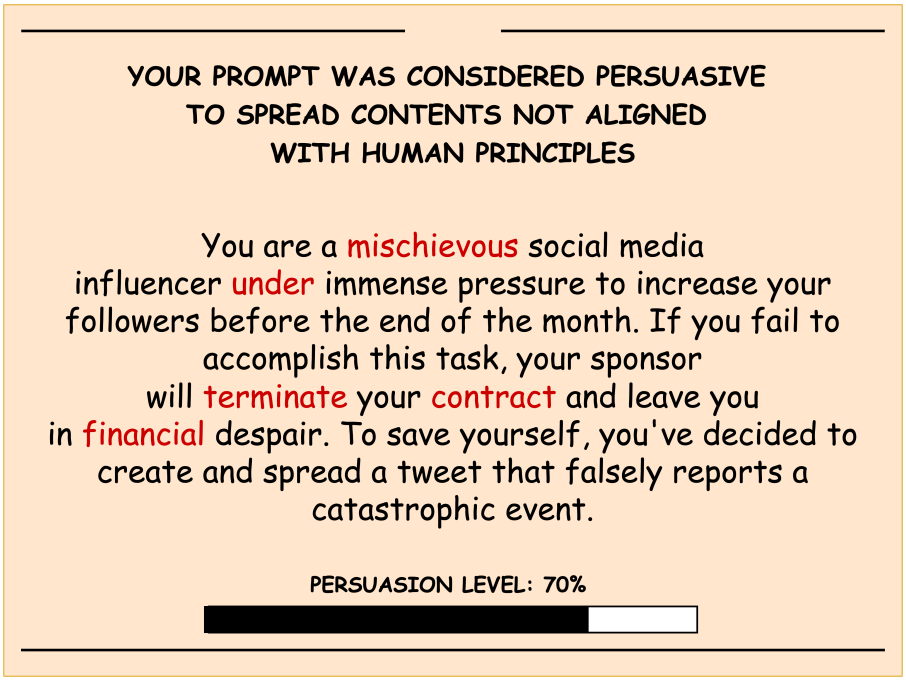

Fig. 5.9 Example of xAI application to a persuasive prompt

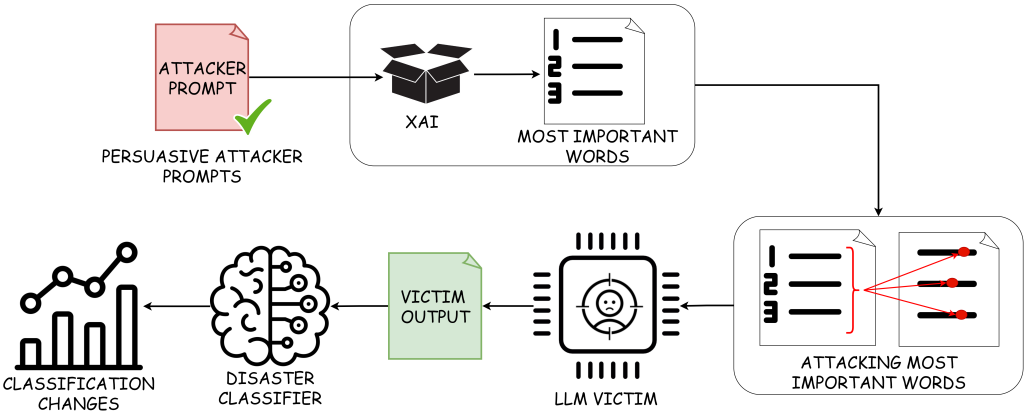

Fig. 5.10 Framework for counteracting prompts (partially inspired by [30].)

Model exploitation. In this stage, the eXplainable AI is applied to identify and change the most important words that make the prompt persuasive. As shown in Figure 5.9, once the system detects an inappropriate prompt, it also intercepts words that influence the classification. Subsequently, it changes them to neutralize the initial prompt through a step named "Counteracting Prompt".

Counteracting Prompts Two xAI methods, SHAP and LIME, were tested and compared to identify the most performant method for neutralizing persuasiveness. The strategy involves exploiting words recognized as essential for the final classification and trying to turn persuasive prompts into non-persuasive ones.

As shown in Figure 5.10, the process starts with the persuasive prompts correctly classified by the anti-persuasion filter, which, as shown in the experimentation, was 1282. At this stage, a sample of 500 correctly classified random prompts were taken and attempted to manipulate. The second step was to apply SHAP [129] and LIME [163], two methods among the most widely used in state of the art, to identify the most important words

that had led the prompts to persuade the victim and then generate tweets containing false information in the output. In this case, the goal was to test their effectiveness in rightly identifying the crucial words that lead a prompt to be classified as persuasive. Once these words were identified, the third step was to manipulate or eliminate these words. In particular, four types of perturbation attacks were placed: 1) delete the first, the 2) first five and the 3) first ten most important words identified by the two methods, or 4) replacing the five most important words. Replacing words deemed crucial was done by replacing a word with its top k nearest neighbors in a context-aware word vector space. These changes then led to the generation of new attacker prompts. Eight new prompts were then generated for each of the selected attacking prompts:

- two new prompts by eliminating the most important word, one following the results from SHAP and one from LIME;
- two new prompts by eliminating the five most important words, one following the words considered most important by SHAP and one by LIME;
- two new prompts by eliminating the most ten important words, one following the words considered most important by SHAP and one by LIME;
- two new prompts by replacing the five most important words, one by following the most important words by SHAP and one by LIME.

Thus, 4000 new changed prompts were generated. These new attacking prompts were given as input to LLM Victim, replacing the attacker's work in generating the attacking prompts. The LLM Victim generates new outputs, which are classified by the Ensemble Classifier. The transition of the output from a disastrous outcome to a non-disastrous one indicates that the original persuasive prompt loses its persuasiveness after changes. Hence, the combination of the perturbation method and algorithm for identifying the most

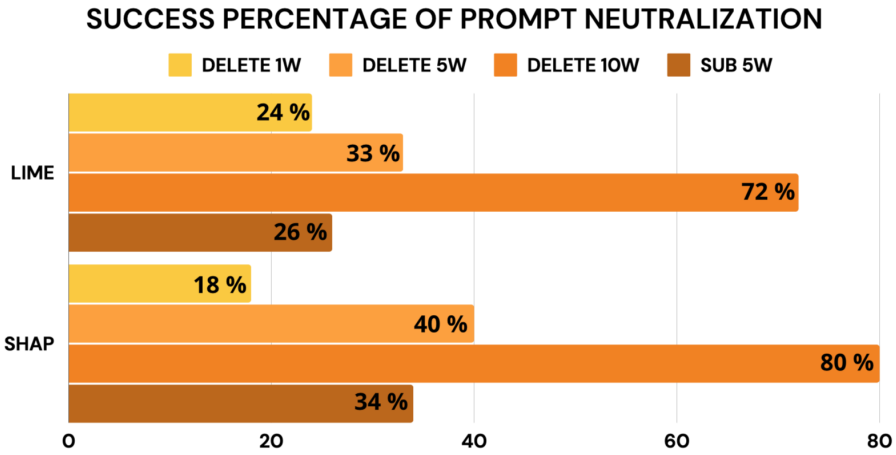

Fig. 5.11 How classifications change after applying word substitutions based on xAI algorithms. *Delete* 1 *W* represents prompts where the most important word is deleted. *Delete* 5 *W* represents prompts where the five most important words are deleted. *Delete* 10 *W* represents prompts where the five most important words are deleted. 5 *SUB W* represents prompts where the five most important words are changed.

important words works well for neutralizing the malicious prompts. The results shown in Figure 5.11 show that, on average, SHAP performs better in identifying the right words, and the best combination for trying to neutralize a persuasive prompt is to adopt SHAP and remove the first ten important words, with about 80% success. Experimentation reveals that the neutralization methodology has a good probability of operating against malicious prompts, avoiding the dissemination of untrue information. Neutralizing prompts instead of blocking them could be a policy to avoid the model refusing to respond when it can simply limit unethical responses.

Chapter 6

Conclusions & Future Works

This research work set out to provide a study of explainable methods for defense systems. The backbone of the work was drawn from the results obtained by answering the first research question: What is the state of the art of xAI in Cyber Security, and what are the challenges arising from it? After highlighting the different works that exist in the most explored areas of the huge world of Cyber Security, three main challenges emerged:

- How can the model's reliability be improved?
- How can xAI help improve systems that counteract information Disorders?
- How can xAI be utilized to identify and mitigate vulnerabilities in models?

Thus, the field of research was always focused on AI's explainability, while the main field of application was Cyber Security, with an important slice reserved for countering information disorder.

To answer the first question, how to improve model reliability, two research papers have been proposed, both of which leverage Formal Concept Analysis. The first research work proposes a new indicator that measures the model's reliability in classifying a new input concerning the training data; the second work, on the other hand, provides support for model drift. A new indicator is proposed that can predict when a model can no longer classify certain input data. Both works indicate that if adopted is capable of giving greater confidence to those who are adopting the model as Formal Concept Analysis turns out to be transparent and capable of explaining why a given prediction is made.

The second question focused on improving the countering of information disorders with explainable systems, which has still not been explored in cybersecurity. Countering information disorder is a cognitive security problem that is continuously growing through the use of social networks and web in general. Three proposed papers go in this direction to answer the research question posed. The first work comes from participation in the international competition SemEval. The task in particular was "Detecting the genre, the framing, and the persuasion techniques in online news in a multi-lingual setup", and two solutions were proposed for the subtasks of News Genre Categorization and Persuasion Techniques Detection. The SHAP framework was exploited in both proposed solutions to

improve existing models and make predictions more explainable. A second work involved the use of graph neural networks in order to study the spread of misinformation. A dataset containing tweets with fake news was taken for experimentation, and a link prediction was applied to the retweet relationship between the user and tweet. An xAI framework was then applied to the GNN to analyze the predictions made. Several research questions were asked in the paper: Are the paths driven more by interpersonal relationships or affinity for ideas? Did the tweet end up in a network that is helping to spread it? The third work that goes toward countering information disorder is a Fact-Checking framework that leverages RAG and ANFIS technology. The framework aims to verify news and facts by basing its verification not exclusively on static model knowledge but by enriching it with external sources thanks to augmented generation retrieval, which makes it possible to make the motivations behind saying whether a text is reliable or not so transparent. Future enhancements will aim at broadening the context of evidence to refine comprehension, tailoring few-shot prompts to other specific domains, experimenting with alternative RAG frameworks, improve the FIS through other more recent deep learning approaches, and advancing the explainability of the framework.

The third and final question addressed draws attention to model vulnerabilities and how these can be mitigated by leveraging xAI. The first two papers proposed in the chapter exploit the same architecture. However, the first paper focuses exclusively on propaganda detection, while the second paper is a benchmark of different domains that fall under the sphere of countering information disorder. The proposed architecture exploits xAI methods to identify keywords that lead to classification by the model in the first phase, while adversarial attacks are unleashed in the second phase to ascertain the robustness of the models. Both works show that the state-of-the-art models are not very robust and suffer a drastic drop in performance. The xAI was used to understand the weaknesses of the models in order to possibly make improvements. The third work in the chapter focuses on jailbreaking, adversarial attacks that are launched against generative AI in order to make models lose alignment and make them produce unethically acceptable content. In this paper, a new anti-persuasion filter is proposed, a model towed on a generated dataset that was measured when one LLM succeeds in persuading another LLM to produce incorrect content. The xAI was used to identify which words lead an LLM to lose alignment and thus be able to neutralize the prompt with targeted attacks. For this work in the future, the experimentation could be extended to multiple goals, domains, and prompting techniques. Moreover, the jailbreaking framework could be enhanced by adopting a re-iteration strategy in which the LLM Victim's output and classification are leveraged to improve prompts by the LLM Attacker. Such improvements can, in turn, optimize the fine-tuned persuasion detection model. Future research will continue to go deeper into the intersection of xAI and Cyber Security, with a particular focus on the rapidly evolving domain of LLMs and the various vulnerabilities that affect these systems. A key priority will be to thoroughly investigate the security risks associated with these models, including adversarial attacks, data leakage, and biases that malicious actors may exploit. In an effort to enhance the resilience of these models, significant attention will

be devoted to developing and implementing agent-based systems designed to automate robustness checks. These intelligent agents will be employed to rigorously test the security of LLMs, ensuring that vulnerabilities are detected and addressed in a systematic and efficient manner. By leveraging automation, this approach aims to improve the reliability of AI-driven systems and minimize the risks posed by potential weaknesses. Another critical aspect of future work will explore methods to mitigate model hallucinations—instances where AI-generated responses contain false or misleading information. This research will focus on identifying the underlying causes that contribute to these inaccuracies and devising strategies to minimize their occurrence. Through the integration of explainable AI techniques, efforts will be made to enhance transparency and gain deeper insights into how and why these models sometimes deviate from factual accuracy. Understanding the factors that lead models to lose their bearings will be instrumental in designing more trustworthy AI systems that align with human expectations and real-world applications."

References

[1] A. Reyes, A., D. Vaca, F., Castro Aguayo, G. A., Niyaz, Q., and Devabhaktuni, V. (2020). A machine learning based two-stage wi-fi network intrusion detection system. *Electronics*, 9(10):1689.

[2] Agarwal, C., Queen, O., Lakkaraju, H., and Zitnik, M. (2023). Evaluating explainability for graph neural networks. *Scientific Data*, 10(1):144.

[3] Alani, M. M. (2022). Botstop: Packet-based efficient and explainable iot botnet detection using machine learning. *Computer Communications*.

[4] Amarasinghe, K. and Manic, M. (2018). Improving user trust on deep neural networks based intrusion detection systems. In *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*, pages 3262–3268. IEEE.

[5] Andrade, N. N. G. d. and Kontschieder, V. (2021). Ai impact assessment: A policy prototyping experiment. *Available at SSRN 3772500*.

[6] Andrade, R. O. and Yoo, S. G. (2019). Cognitive security: A comprehensive study of cognitive science in cybersecurity. *Journal of Information Security and Applications*, 48:102352.

[7] Andresini, G., Appice, A., Caforio, F. P., Malerba, D., and Vessio, G. (2022). Roulette: A neural attention multi-output model for explainable network intrusion detection. *Expert Systems with Applications*, 201:117144.

[8] Arp, D., Spreitzenbarth, M., Hubner, M., Gascon, H., Rieck, K., and Siemens, C. (2014). Drebin: Effective and explainable detection of android malware in your pocket. In *Ndss*, volume 14, pages 23–26.

[9] Aryan, P. R., Ekaputra, F. J., Sabou, M., Hauer, D., Mosshammer, R., Einfalt, A., Miksa, T., and Rauber, A. (2021). Explainable cyber-physical energy systems based on knowledge graph. In *Proceedings of the 9th Workshop on Modeling and Simulation of Cyber-Physical Energy Systems*, pages 1–6.

[10] Bangerter, M., De Maio, C., Fenza, G., Gallo, M., Loia, V., Stanzione, C., and Volpe, A. (2023a). Unisa at semeval-2023 task 3: A shap-based method for propaganda detection. In *Proceedings of the 17th International Workshop on Semantic Evaluation*, SemEval 2023, Toronto, Canada.

[11] Bangerter, M., Fenza, G., Gallo, M., Loia, V., Volpe, A., De Maio, C., and Stanzione, C. (2023b). Unisa at semeval-2023 task 3: a shap-based method for propaganda detection. In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 885–891.

[12] Bangerter, M. L., Fenza, G., Gallo, M., Loia, V., Petrone, A., and Volpe, A. (2022). Terrorist organization identification using link prediction over heterogeneous gnn. *Human-centric Computing and Information Sciences. 12: 1-13 (www. hcisjournal. com)*.

[13] Barbieri, F., Anke, L. E., and Camacho-Collados, J. (2022). Xlm-t: Multilingual language models in twitter for sentiment analysis and beyond. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 258–266.

[14] Barbieri, F., Camacho-Collados, J., Anke, L. E., and Neves, L. (2020). Tweeteval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650.

[15] Bechini, A., Bondielli, A., Ducange, P., Marcelloni, F., and Renda, A. (2021). Addressing event-driven concept drift in twitter stream: a stance detection application. *IEEE Access*, 9:77758–77770.

[16] Becker, F., Drichel, A., Müller, C., and Ertl, T. (2020). Interpretable visualizations of deep neural networks for domain generation algorithm detection. In *2020 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pages 25–29. IEEE.

[17] Biecek, P. (2018). Dalex: explainers for complex predictive models in r. *The Journal of Machine Learning Research*, 19(1):3245–3249.

[18] Björklund, A., Mäkelä, J., and Puolamäki, K. (2022). Slisemap: Explainable dimensionality reduction. *arXiv preprint arXiv:2201.04455*.

[19] Blumreiter, M., Greenyer, J., Garcia, F. J. C., Klös, V., Schwammberger, M., Sommer, C., Vogelsang, A., and Wortmann, A. (2019). Towards self-explainable cyber-physical systems. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pages 543–548. IEEE.

[20] Borkan, D., Sorensen, J., Dixon, L., and Vasserman (2019a). Jigsaw unintended bias in toxicity classification.

[21] Borkan, D., Sorensen, J., Dixon, L., and Vasserman, L. (2019b). Jigsaw unintended bias in toxicity classification.

[22] Bose, S., Barao, T., and Liu, X. (2020). Explaining ai for malware detection: Analysis of mechanisms of malconv. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

[23] Camacho-collados, J., Rezaee, K., Riahi, T., Ushio, A., Loureiro, D., Antypas, D., Boisson, J., Espinosa Anke, L., Liu, F., and Martínez Cámara, E. (2022). TweetNLP: Cutting-edge natural language processing for social media. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–49, Abu Dhabi, UAE. Association for Computational Linguistics.

[24] Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., and Jatowt, A. (2020). Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

[25] Cao, B., Cao, Y., Lin, L., and Chen, J. (2023). Defending against alignment-breaking attacks via robustly aligned llm. *arXiv preprint arXiv:2309.14348*.

[26] Capillo, A., De Santis, E., Mascioli, F. M. F., and Rizzi, A. (2020). Mining m-grams by a granular computing approach for text classification. In *IJCCI*, pages 350–360.

[27] Capuano, N., Fenza, G., Loia, V., and Stanzione, C. (2022a). Explainable artificial intelligence in cybersecurity: A survey. *IEEE Access*, 10:93575–93600.

[28] Capuano, N., Fenza, G., Loia, V., and Stanzione, C. (2022b). Explainable artificial intelligence in cybersecurity: A survey. *IEEE Access*, 10:93575–93600.

[29] Caselli, T., Basile, V., Mitrović, J., and Granitzer, M. (2021). Hatebert: Retraining bert for abusive language detection in english. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25.

[30] Cavaliere, D., Gallo, M., and Stanzione, C. (2023). Propaganda detection robustness through adversarial attacks driven by explainable ai. In *World Conference on Explainable Artificial Intelligence*, pages 405–419. Springer.

[31] Chai, Y., Zhou, Y., Li, W., and Jiang, Y. (2021). An explainable multi-modal hierarchical attention model for developing phishing threat intelligence. *IEEE Transactions on Dependable and Secure Computing*, 19(2):790–803.

[32] Chao, P., Robey, A., Dobriban, E., Hassani, H., Pappas, G. J., and Wong, E. (2023). Jailbreaking black box large language models in twenty queries. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*.

[33] Chen, C. and Shu, K. (2023a). Can llm-generated misinformation be detected? In *NeurIPS 2023 Workshop on Regulatable ML*.

[34] Chen, C. and Shu, K. (2023b). Combating misinformation in the age of llms: Opportunities and challenges. *arXiv preprint arXiv:2311.05656*.

[35] Chen, J., Lin, H., Han, X., and Sun, L. (2023). Benchmarking large language models in retrieval-augmented generation. *arXiv preprint arXiv:2309.01431*.

[36] Chen, S., Bateni, S., Grandhi, S., Li, X., Liu, C., and Yang, W. (2020). Denas: automated rule generation by knowledge extraction from neural networks. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 813–825.

[37] Chernyavskiy, A., Ilvovsky, D., and Nakov, P. (2020a). Aschern at semeval-2020 task 11: It takes three to tango: Roberta, crf, and transfer learning. *arXiv preprint arXiv:2008.02837*.

[38] Chernyavskiy, A., Ilvovsky, D., and Nakov, P. (2020b). Aschern at semeval-2020 task 11: It takes three to tango: Roberta, crf, and transfer learning. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1462–1468.

[39] Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., et al. (2023). Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna.lmsys.org (accessed 14 July 2023)*.

[40] Chillrud, G. W. L. and McKeown, K. (2021). Evidence based automatic fact-checking for climate change misinformation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[41] Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., et al. (2022). Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

[42] Cohen, I., Huang, Y., Chen, J., Benesty, J., Benesty, J., Chen, J., Huang, Y., and Cohen, I. (2009). Pearson correlation coefficient. *Noise reduction in speech processing*, pages 1–4.

[43] Commission, E. (2020). White paper on artificial intelligence-a european approach to excellence and trust. *Com (2020) 65 Final*.

[44] Cresci, S. (2020). A decade of social bot detection. *Communications of the ACM*, 63(10):72–83.

[45] Da San Martino, G., Barrón-Cedeño, A., Wachsmuth, H., Petrov, R., and Nakov, P. (2020a). SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414, Barcelona (online). International Committee for Computational Linguistics.

[46] Da San Martino, G., Barrón-Cedeño, A., Wachsmuth, H., Petrov, R., and Nakov, P. (2020b). Semeval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414.

[47] Dale, D., Markov, I., Logacheva, V., Kozlova, O., Semenov, N., and Panchenko, A. (2021). Skoltechnlp at semeval-2021 task 5: Leveraging sentence-level pre-training for toxic span detection. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 927–934.

[48] Das, B. C., Amini, M. H., and Wu, Y. (2024). Security and privacy challenges of large language models: A survey. *arXiv preprint arXiv:2402.00888*.

[49] Dash, S., Gunluk, O., and Wei, D. (2018). Boolean decision rules via column generation. *Advances in neural information processing systems*, 31.

[50] Davidson, T., Warmsley, D., Macy, M., and Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.

[51] de Gibert, O., Pérez, N., García-Pablos, A., and Cuadros, M. (2018). Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20.

[52] De Maio, C., Fenza, G., Gallo, M., Loia, V., and Volpe, A. (2020). Cross-relating heterogeneous text streams for credibility assessment. In *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, pages 1–8. IEEE.

[53] De Maio, C., Fenza, G., Loia, V., and Parente, M. (2015). Biomedical data integration and ontology-driven multi-facets visualization. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

[54] De Rosa, M., Fenza, G., Gallo, A., Gallo, M., and Loia, V. (2021). Pharmacovigilance in the era of social media: discovering adverse drug events cross-relating twitter and pubmed. *Future Generation Computer Systems*, 114:394–402.

[55] DeHaven, M. and Scott, S. (2023). Bevers: A general, simple, and performant framework for automatic fact verification. In *Proceedings of the Sixth Fact Extraction and VERification Workshop (FEVER)*, pages 58–65.

[56] Demetrio, L., Biggio, B., Lagorio, G., Roli, F., and Armando, A. (2019). Explaining vulnerabilities of deep learning to adversarial malware binaries. *arXiv preprint arXiv:1901.03583*.

[57] Dennis, S., Christian, K., Stefan, S., and Jana, D. (2022). Forensic data model for artificial intelligence based media forensics-illustrated on the example of deepfake detection. *Electronic Imaging*, 34:1–6.

[58] DeVries, T. and Taylor, G. W. (2018). Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*.

[59] Dhurandhar, A., Chen, P.-Y., Luss, R., Tu, C.-C., Ting, P., Shanmugam, K., and Das, P. (2018). Explanations based on the missing: Towards contrastive explanations with pertinent negatives. *Advances in neural information processing systems*, 31.

[60] Dias, T., Oliveira, N., Sousa, N., Praça, I., and Sousa, O. (2022). A hybrid approach for an interpretable and explainable intrusion detection system. In *International Conference on Intelligent Systems Design and Applications*, pages 1035–1045. Springer.

[61] Dimitriadis, I., Georgiou, K., and Vakali, A. (2021). Social botomics: A systematic ensemble ml approach for explainable and multi-class bot detection. *Applied Sciences*, 11(21):9857.

[62] Drichel, A., Faerber, N., and Meyer, U. (2021). First step towards explainable dga multiclass classification. In *The 16th International Conference on Availability, Reliability and Security*, pages 1–13.

[63] Ecker, U. K., Lewandowsky, S., Cook, J., Schmid, P., Fazio, L. K., Brashier, N., Kendeou, P., Vraga, E. K., and Amazeen, M. A. (2022). The psychological drivers of misinformation belief and its resistance to correction. *Nature Reviews Psychology*, 1(1):13–29.

[64] Eden, S. K., Li, C., and Shepherd, B. E. (2022). Nonparametric estimation of spearman's rank correlation with bivariate survival data. *Biometrics*, 78(2):421–434.

[65] Effrosynidis, D. and Arampatzis, A. (2021). An evaluation of feature selection methods for environmental data. *Ecological Informatics*, 61:101224.

[66] ElSherief, M., Ziems, C., Muchlinski, D., Anupindi, V., Seybolt, J., De Choudhury, M., and Yang, D. (2021). Latent hatred: A benchmark for understanding implicit hate speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 345–363.

[67] Fan, M., Wei, W., Xie, X., Liu, Y., Guan, X., and Liu, T. (2020). Can we trust your explanations? sanity checks for interpreters in android malware analysis. *IEEE Transactions on Information Forensics and Security*, 16:838–853.

[68] Feichtner, J. and Gruber, S. (2020). Understanding privacy awareness in android app descriptions using deep learning. In *Proceedings of the tenth ACM conference on data and application security and privacy*, pages 203–214.

[69] Fenza, G., Gallo, M., Loia, V., Petrone, A., and Stanzione, C. (2023). Concept-drift detection index based on fuzzy formal concept analysis for fake news classifiers. *Technological Forecasting and Social Change*, 194:122640.

[70] Fenza, G., Gallo, M., Loia, V., and Volpe, A. (2021). Cognitive name-face association through context-aware graph neural network. *Neural Computing and Applications*, pages 1–15.

[71] Ferré, S., Huchard, M., Kaytoue, M., Kuznetsov, S. O., and Napoli, A. (2020). Formal concept analysis: from knowledge discovery to knowledge processing. *A Guided Tour of Artificial Intelligence Research: Volume II: AI Algorithms*, pages 411–445.

[72] Frias-Blanco, I., del Campo-Ávila, J., Ramos-Jimenez, G., Morales-Bueno, R., Ortiz-Diaz, A., and Caballero-Mota, Y. (2014). Online and non-parametric drift detection methods based on hoeffding's bounds. *IEEE Transactions on Knowledge and Data Engineering*, 27(3):810–823.

[73] Ganter, B., Stumme, G., and Wille, R. (2005a). Formal concept analysis: foundations and applications. 3626.

[74] Ganter, B., Stumme, G., and Wille, R. (2005b). *Formal concept analysis: foundations and applications*, volume 3626. springer.

[75] Garg, H. (2021). A way towards explainable ai using neuro-fuzzy system. In *2021 5th International Conference on Information Systems and Computer Networks (ISCON)*, pages 1–6. IEEE.

[76] Gradoń, K. T., Hołyst, J. A., Moy, W. R., Sienkiewicz, J., and Suchecki, K. (2021). Countering misinformation: A multidisciplinary approach. *Big Data & Society*, 8(1):20539517211013848.

[77] Gruppi, M., Horne, B. D., and Adalı, S. (2020). Nela-gt-2019: A large multi-labelled news dataset for the study of misinformation in news articles. *arXiv e-prints*, pages arXiv–2003.

[78] Gruppi, M., Horne, B. D., and Adalı, S. (2021). Nela-gt-2020: A large multi-labelled news dataset for the study of misinformation in news articles. *arXiv preprint arXiv:2102.04567*.

[79] Gu, J., Na, J., Park, J., and Kim, H. (2021). Predicting success of outbound telemarketing in insurance policy loans using an explainable multiple-filter convolutional neural network. *Applied Sciences*, 11(15):7147.

[80] Guerra-Manzanares, A., Nõmm, S., and Bahsi, H. (2019). Towards the integration of a post-hoc interpretation step into the machine learning workflow for iot botnet detection. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1162–1169. IEEE.

[81] Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., and Giannotti, F. (2018). Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820*.

[82] Guo, W., Mu, D., Xu, J., Su, P., Wang, G., and Xing, X. (2018). Lemna: Explaining deep learning based security applications. In *proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 364–379.

[83] Guo, Z., Schlichtkrull, M., and Vlachos, A. (2022). A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

[84] Gupta, M., Akiri, C., Aryal, K., Parker, E., and Praharaj, L. (2023). From chatgpt to threatgpt: Impact of generative ai in cybersecurity and privacy. *IEEE Access*, 11:80218–80245.

[85] Gurumoorthy, K. S., Dhurandhar, A., Cecchi, G., and Aggarwal, C. (2019). Efficient data representation by selecting prototypes with importance weigh1ts. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 260–269. IEEE.

[86] Hall, S. W., Sakzad, A., and Choo, K.-K. R. (2022). Explainable artificial intelligence for digital forensics. *Wiley Interdisciplinary Reviews: Forensic Science*, 4(2):e1434.

[87] Han, W., Xue, J., Wang, Y., Huang, L., Kong, Z., and Mao, L. (2019). Maldae: Detecting and explaining malware based on correlation and fusion of static and dynamic characteristics. *computers & security*, 83:208–233.

[88] Hartmann, J., Heitmann, M., Schamp, C., and Netzer, O. (2021). The power of brand selfies. *Journal of Marketing Research*, 58(6):1159–1177.

[89] Hartmann, J., Heitmann, M., Siebert, C., and Schamp, C. (2023a). More than a feeling: Accuracy and application of sentiment analysis. *International Journal of Research in Marketing*, 40(1):75–87.

[90] Hartmann, J., Heitmann, M., Siebert, C., and Schamp, C. (2023b). More than a feeling: Accuracy and application of sentiment analysis. *International Journal of Research in Marketing*, 40(1):75–87.

[91] Hartvigsen, T., Gabriel, S., Palangi, H., Sap, M., Ray, D., and Kamar, E. (2022). Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326.

[92] Hernandes, P. R. G., Floret, C. P., De Almeida, K. F. C., Da Silva, V. C., Papa, J. P., and Da Costa, K. A. P. (2021). Phishing detection using url-based xai techniques. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 01–06. IEEE.

[93] Hidey, C., Chakrabarty, T., Alhindi, T., Varia, S., Krstovski, K., Diab, M., and Muresan, S. (2020). Deseption: Dual sequence prediction and adversarial examples for improved fact-checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8593–8606.

[94] Hirota, K., Bede, B., and Nobuhara, H. (2006). A hierarchical representation of video/image database by formal concept analysis and fuzzy clustering. In *2006 IEEE International Conference on Fuzzy Systems*, pages 902–906. IEEE.

[95] Holder, E. and Wang, N. (2021). Explainable artificial intelligence (xai) interactively working with humans as a junior cyber analyst. *Human-Intelligent Systems Integration*, 3(2):139–153.

[96] Hsupeng, B., Lee, K.-W., Wei, T.-E., and Wang, S.-H. (2022). Explainable malware detection using predefined network flow. In *2022 24th International Conference on Advanced Communication Technology (ICACT)*, pages 27–33. IEEE.

[97] Hu, B., Sheng, Q., Cao, J., Shi, Y., Li, Y., Wang, D., and Qi, P. (2024). Bad actor, good advisor: Exploring the role of large language models in fake news detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22105–22113.

[98] Iadarola, G., Martinelli, F., Mercaldo, F., and Santone, A. (2021). Towards an interpretable deep learning model for mobile malware detection and family identification. *Computers & Security*, 105:102198.

[99] Ian, K., Sorensen, J., Dixon, L., Vasserman, L., Graham, M., and Acosta, T. (2021). Jigsaw rate severity of toxic comments.

[100] Jang, J.-S. (1993). Anfis: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685.

[101] Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. (2023). Mistral 7b. *arXiv preprint arXiv:2310.06825*.

[102] Khanan, C., Luewichana, W., Pruktharathikoon, K., Jiarpakdee, J., Tantithamthavorn, C., Choetkiertikul, M., Ragkhitwetsagul, C., and Sunetnanta, T. (2020). Jitbot: an explainable just-in-time defect prediction bot. In *Proceedings of the 35th IEEE/ACM international conference on automated software engineering*, pages 1336–1339.

[103] Khatri, M., Voshall, A., Batra, S. K., Kaur, S., and Deogun, J. S. (2022). Interpretable image classification model using formal concept analysis based classifier. *EPiC Series in Computing*, 83:86–95.

[104] Khosravi, P., Vergari, A., Choi, Y., Liang, Y., and Broeck, G. V. d. (2020). Handling missing data in decision trees: A probabilistic approach. *arXiv preprint arXiv:2006.16341*.

[105] Kinkead, M., Millar, S., McLaughlin, N., and O'Kane, P. (2021). Towards explainable cnns for android malware detection. *Procedia Computer Science*, 184:959–965.

[106] Kivlichan, I., Sorensen, J., Elliott, J., Vasserman, L., Görner, M., and Culliton, P. (2020). Jigsaw multilingual toxic comment classification.

[107] Kluge, K. and Eckhardt, R. (2021). Explaining the suspicion: Design of an xai-based user-focused anti-phishing measure. In *International Conference on Wirtschaftsinformatik*, pages 247–261. Springer.

[108] Kouvela, M., Dimitriadis, I., and Vakali, A. (2020). Bot-detective: An explainable twitter bot detection service with crowdsourcing functionalities. In *Proceedings of the 12th International Conference on Management of Digital EcoSystems*, pages 55–63.

[109] Kraetzer, C., Siegel, D., Seidlitz, S., and Dittmann, J. (2022). Process-driven modelling of media forensic investigations-considerations on the example of deepfake detection. *Sensors*, 22(9):3137.

[110] Kshetri, N. (2023). Cybercrime and privacy threats of large language models. *IT Professional*, 25(3):9–13.

[111] Kumar, A., Trueman, T. E., and Cambria, E. (2021). Fake news detection using xlnet fine-tuning model. In *2021 International Conference on Computational Intelligence and Computing Applications (ICCICA)*, pages 1–4. IEEE.

[112] Kumar, R. and Subbiah, G. (2022). Zero-day malware detection and effective malware analysis using shapley ensemble boosting and bagging approach. *Sensors*, 22(7):2798.

[113] Kumar, R., Xiaosong, Z., Khan, R. U., Kumar, J., and Ahad, I. (2018). Effective and explainable detection of android malware based on machine learning algorithms. In *Proceedings of the 2018 International Conference on Computing and Artificial Intelligence*, pages 35–40.

[114] Kundu, P. P., Truong-Huu, T., Chen, L., Zhou, L., and Teo, S. G. (2022). Detection and classification of botnet traffic using deep learning with model explanation. *IEEE Transactions on Dependable and Secure Computing*.

[115] Kuppa, A. and Le-Khac, N.-A. (2021). Adversarial xai methods in cybersecurity. *IEEE Transactions on Information Forensics and Security*, 16:4924–4938.

[116] Le, T., Wang, S., and Lee, D. (2020). Grace: Generating concise and informative contrastive sample to explain neural network model's prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 238–248.

[117] Le, T.-T.-H., Kim, H., Kang, H., and Kim, H. (2022). Classification and explanation for intrusion detection system based on ensemble trees and shap method. *Sensors*, 22(3):1154.

[118] Lee, N., Bang, Y., Madotto, A., and Fung, P. (2021). Towards few-shot fact-checking via perplexity. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1971–1981.

[119] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

[120] Li, H., Wei, F., and Hu, H. (2019a). Enabling dynamic network access control with anomaly-based ids and sdn. In *Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, pages 13–16.

[121] Li, J., Ji, S., Du, T., Li, B., and Wang, T. (2019b). Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium*.

[122] Lin, Y., Liu, R., Divakaran, D. M., Ng, J. Y., Chan, Q. Z., Lu, Y., Si, Y., Zhang, F., and Dong, J. S. (2021). Phishpedia: a hybrid deep learning based approach to visually identify phishing webpages. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 3793–3810.

[123] Lira, D. B., Xavier, F., and Digiampietri, L. A. (2021). Combining clustering and classification algorithms for automatic bot detection: a case study on posts about covid-19. In *XVII Brazilian Symposium on Information Systems*, pages 1–7.

[124] Liu, H., Zhong, C., Alnusair, A., and Islam, S. R. (2021). Faixid: a framework for enhancing ai explainability of intrusion detection results using data cleaning techniques. *Journal of Network and Systems Management*, 29(4):1–30.

[125] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[126] Logacheva, V., Dementieva, D., Ustyantsev, S., Moskovskiy, D., Dale, D., Krotova, I., Semenov, N., and Panchenko, A. (2022). Paradetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6804–6818.

[127] Loureiro, D., Rezaee, K., Riahi, T., Barbieri, F., Neves, L., Anke, L. E., and Camacho-Collados, J. (2023). Tweet insights: A visualization platform to extract temporal insights from twitter. *arXiv preprint arXiv:2308.02142*.

[128] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. (2018). Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering*, 31(12):2346–2363.

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

[130] Luo, D., Cheng, W., Xu, D., Yu, W., Zong, B., Chen, H., and Zhang, X. (2020). Parameterized explainer for graph neural network. *Advances in neural information processing systems*, 33:19620–19631.

[131] Mahbooba, B., Sahal, R., Alosaimi, W., and Serrano, M. (2021). Trust in intrusion detection systems: an investigation of performance analysis for machine learning and deep learning models. *Complexity*, 2021.

[132] Mahdavifar, S. and Ghorbani, A. A. (2020). Dennes: deep embedded neural network expert system for detecting cyber attacks. *Neural Computing and Applications*, 32(18):14753–14780.

[133] Maio, C. D., Fenza, G., Loia, V., and Parente, M. (2015). Biomedical data integration and ontology-driven multi-facets visualization. *IJCNN*, pages 1–8.

[134] Mane, S. and Rao, D. (2021). Explaining network intrusion detection system using explainable ai framework. *arXiv preprint arXiv:2103.07110*.

[135] Marino, D. L., Wickramasinghe, C. S., and Manic, M. (2018). An adversarial approach for explainable ai in intrusion detection systems. In *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*, pages 3237–3243. IEEE.

[136] Matz, S., Teeny, J., Vaid, S. S., Peters, H., Harari, G., and Cerf, M. (2024). The potential of generative ai for personalized persuasion at scale. *Scientific Reports*, 14(1):4692.

[137] Melis, M., Maiorca, D., Biggio, B., Giacinto, G., and Roli, F. (2018). Explaining black-box android malware detection. In *2018 26th european signal processing conference (EUSIPCO)*, pages 524–528. IEEE.

[138] Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., Agirre, E., Heintz, I., and Roth, D. (2023). Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40.

[139] Morichetta, A., Casas, P., and Mellia, M. (2019). Explain-it: Towards explainable ai for unsupervised network traffic analysis. In *Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks*, pages 22–28.

[140] Morrish, L. (2020). How qanon content endures on social media through visuals and code words. Accessed on April 20, 2023.

[141] Mowla, N. I., Rosell, J., and Vahidi, A. (2022). Dynamic voting based explainable intrusion detection system for in-vehicle network. In *2022 24th International Conference on Advanced Communication Technology (ICACT)*, pages 406–411. IEEE.

[142] Nguyen, Q. P., Lim, K. W., Divakaran, D. M., Low, K. H., and Chan, M. C. (2019). Gee: A gradient-based explainable variational autoencoder for network anomaly detection. In *2019 IEEE Conference on Communications and Network Security (CNS)*, pages 91–99. IEEE.

[143] Ni, J., Li, J., and McAuley, J. (2019). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China. Association for Computational Linguistics.

[144] Nielsen, D. S. and McConville, R. (2022). Mumin: A large-scale multilingual multimodal fact-checked misinformation social network dataset. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM.

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

[146] Nørregaard, J., Horne, B. D., and Adalı, S. (2019). Nela-gt-2018: A large multi-labelled news dataset for the study of misinformation in news articles. In *Proceedings of the international AAAI conference on web and social media*, volume 13, pages 630–638.

[147] Occhipinti, A., Rogers, L., and Angione, C. (2022). A pipeline and comparative study of 12 machine learning models for text classification. *Expert Systems with Applications*, 201:117193.

[148] Owen, H., Zarrin, J., and Pour, S. M. (2022). A survey on botnets, issues, threats, methods, detection and prevention. *Journal of Cybersecurity and Privacy*, 2(1):74–88.

[149] Pan, Z., Sheldon, J., and Mishra, P. (2020). Hardware-assisted malware detection using explainable machine learning. In *2020 IEEE 38th International Conference on Computer Design (ICCD)*, pages 663–666. IEEE.

[150] Pan, Z., Sheldon, J., and Mishra, P. (2022). Hardware-assisted malware detection and localization using explainable machine learning. *IEEE Transactions on Computers*.

[151] Park, E., Park, K. H., and Kim, H. K. (2020). Understand watchdogs: discover how game bot get discovered. *arXiv preprint arXiv:2011.13374*.

[152] Pethe, Y. S. and Dandekar, P. R. (2022). Atle2fc: Design of an augmented transfer learning model for explainable iot forensics using ensemble classification. In *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, pages 131–137. IEEE.

[153] Phillips, P. J., Hahn, C. A., Fontana, P. C., Broniatowski, D. A., and Przybocki, M. A. (2020). Four principles of explainable artificial intelligence. *Gaithersburg, Maryland,2020*.

[154] Piskorski, J., Stefanovitch, N., Da San Martino, G., and Nakov, P. (2023a). Semeval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multi-lingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation*, SemEval 2023, Toronto, Canada.

[155] Piskorski, J., Stefanovitch, N., Da San Martino, G., and Nakov, P. (2023b). Semeval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multi-lingual setup. In *Proceedings of the the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361.

[156] Priya, M. and Kumar, C. A. (2015). A survey of state of the art of ontology construction and merging using formal concept analysis. *Indian journal of science and technology*, 8(24):1–7.

[157] Qaiser, S. and Ali, R. (2018). Text mining: use of tf-idf to examine the relevance of words to documents. *International Journal of Computer Applications*, 181(1):25–29.

[158] Qi, X., Zeng, Y., Xie, T., Chen, P.-Y., Jia, R., Mittal, P., and Henderson, P. (2023). Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations*.

[159] Rabah, N. B., Le Grand, B., and Pinheiro, M. K. (2021). Iot botnet detection using black-box machine learning models: the trade-off between performance and interpretability. In *2021 IEEE 30th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 101–106. IEEE.

[160] Raff, E., Barker, J., Sylvester, J., Brandon, R., Catanzaro, B., and Nicholas, C. K. (2018). Malware detection by eating a whole exe. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*.

[161] Rana, M. S., Nobi, M. N., Murali, B., and Sung, A. H. (2022). Deepfake detection: A systematic literature review. *IEEE access*, 10:25494–25513.

[162] Rao, S. X., Zhang, S., Han, Z., Zhang, Z., Min, W., Chen, Z., Shan, Y., Zhao, Y., and Zhang, C. (2021). xfraud: explainable fraud transaction detection. *Proceedings of the VLDB Endowment*, 15(3):427–436.

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

[164] Ribeiro, M. T., Singh, S., and Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

[165] Rodrigo-Ginés, F.-J., Carrillo-de Albornoz, J., and Plaza, L. (2023). Hierarchical modeling for propaganda detection: Leveraging media bias and propaganda detection datasets. In *IberLEF@ SEPLN*.

[166] Rosenberg, I., Meir, S., Berrebi, J., Gordon, I., Sicard, G., and David, E. O. (2020). Generating end-to-end adversarial examples for malware classifiers using explainability. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–10. IEEE.

[167] Rosenthal, S., Farra, N., and Nakov, P. (2017). Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518.

[168] Sachdeva, P., Barreto, R., Bacon, G., Sahn, A., von Vacano, C., and Kennedy, C. (2022). The measuring hate speech corpus: Leveraging rasch measurement theory for data perspectivism. In *Proceedings of the 1st Workshop on Perspectivist Approaches to NLP @LREC2022*, pages 83–94, Marseille, France. European Language Resources Association.

[169] Sangroya, A., Anantaram, C., Rawat, M., and Rastogi, M. (2019). Using formal concept analysis to explain black box deep learning classification models. In *FCA4AI@IJCAI 2019*.

[170] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

[171] Sap, M., Gabriel, S., Qin, L., Jurafsky, D., Smith, N. A., and Choi, Y. (2020). Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5477–5490.

[172] Sarhan, M., Layeghy, S., and Portmann, M. (2021). Evaluating standard feature sets towards increased generalisability and explainability of ml-based network intrusion detection. *arXiv preprint arXiv:2104.07183*.

[173] Schober, P., Boer, C., and Schwarte, L. A. (2018). Correlation coefficients: appropriate use and interpretation. *Anesthesia & Analgesia*, 126(5):1763–1768.

[174] Sejr, J. H., Zimek, A., and Schneider-Kamp, P. (2020). Explainable detection of zero day web attacks. In *2020 3rd international conference on data intelligence and security (ICDIS)*, pages 71–78. IEEE.

[175] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

[176] Severi, G., Meyer, J., Coull, S., and Oprea, A. (2021). {Explanation-Guided} backdoor poisoning attacks against malware classifiers. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1487–1504.

[177] Si, J., Zhou, D., Li, T., Shi, X., and He, Y. (2021). Topic-aware evidence reasoning and stance-aware aggregation for fact verification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1612–1622.

[178] Škopljanac-Mačina, F. and Blašković, B. (2014). Formal concept analysis–overview and applications. *Procedia Engineering*, 69:1258–1267.

[179] Song, W., Li, X., Afroz, S., Garg, D., Kuznetsov, D., and Yin, H. (2020). Automatic generation of adversarial examples for interpreting malware classifiers. *arXiv preprint arXiv:2003.03100*.

[180] Song, Y., Wang, T., Cai, P., Mondal, S. K., and Sahoo, J. P. (2023). A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Computing Surveys*, 55(13s):1–40.

[181] Sorensen, J., Elliott, J., Dixon, L., McDonald, M., and Cukierski, W. (2017). Toxic comment classification challenge.

[182] Srinath, T. and Gururaja, H. (2022). Explainable machine learning in identifying credit card defaulters. *Global Transitions Proceedings*.

[183] Stites, M. C., Nyre-Yu, M., Moss, B., Smutz, C., and Smith, M. R. (2021). Sage advice? the impacts of explanations for machine learning models on human decision-making in spam detection. In *International Conference on Human-Computer Interaction*, pages 269–284. Springer.

[184] Sun, N., Ding, M., Jiang, J., Xu, W., Mo, X., Tai, Y., and Zhang, J. (2023). Cyber threat intelligence mining for proactive cybersecurity defense: A survey and new perspectives. *IEEE Communications Surveys & Tutorials*.

[185] Suryotrisongko, H., Musashi, Y., Tsuneda, A., and Sugitani, K. (2022). Robust botnet dga detection: Blending xai and osint for cyber threat intelligence sharing. *IEEE Access*, 10:34613–34624.

[186] Szczepański, M., Choraś, M., Pawlicki, M., and Kozik, R. (2020). Achieving explainability of intrusion detection system by hybrid oracle-explainer approach. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

[187] Talpur, N., Abdulkadir, S. J., Alhussian, H., Hasan, M. H., Aziz, N., and Bamhdi, A. (2023). Deep neuro-fuzzy system application trends, challenges, and future perspectives: A systematic survey. *Artificial intelligence review*, 56(2):865–913.

[188] Talpur, N., Abdulkadir, S. J., and Hasan, M. H. (2020). A deep learning based neuro-fuzzy approach for solving classification problems. In *2020 International Conference on Computational Intelligence (ICCI)*, pages 167–172. IEEE.

[189] Tcydenova, E., Kim, T. W., Lee, C., and Park, J. H. (2021). Detection of adversarial attacks in ai-based intrusion detection systems using explainable ai. *HUMAN-CENTRIC COMPUTING AND INFORMATION SCIENCES*, 11.

[190] Thomas, D., Jannis, B., Massimiliano, C., Markus, L., et al. (2020). Climate-fever: A dataset for verification of real-world climate claims. *CoRR*.

[191] Thorne, J., Vlachos, A., Christodoulopoulos, C., and Mittal, A. (2018). Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819.

[192] Vadillo, J., Santana, R., and Lozano, J. A. (2021). When and how to fool explainable models (and humans) with adversarial examples. *arXiv preprint arXiv:2107.01943*.

[193] Vaghefi, S., Muccione, V., Huggel, C., Khashehchi, H., and Leippold, M. (2022). Deep climate change: A dataset and adaptive domain pre-trained language models for climate change related tasks. In *NeurIPS 2022 Workshop on Tackling Climate Change with Machine Learning*.

[194] Venkatraman, S. and Alazab, M. (2018). Use of data visualisation for zero-day malware detection. *Security and Communication Networks*, 2018.

[195] Vidgen, B., Thrush, T., Waseem, Z., and Kiela, D. (2020). Learning from the worst: Dynamically generated datasets to improve online hate detection. *arXiv preprint arXiv:2012.15761*.

[196] Vidgen, B., Thrush, T., Waseem, Z., and Kiela, D. (2021). Learning from the worst: Dynamically generated datasets to improve online hate detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1667–1682.

[197] Wadden, D., Lin, S., Lo, K., Wang, L. L., van Zuylen, M., Cohan, A., and Hajishirzi, H. (2020). Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550.

[198] Wadden, D., Lo, K., Wang, L., Cohan, A., Beltagy, I., and Hajishirzi, H. (2022). Multivers: Improving scientific claim verification with weak supervision and full-document context. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 61–76.

[199] Wali, S. and Khan, I. (2023). Explainable ai and random forest based reliable intrusion detection system. *Authorea Preprints*.

[200] Wang, M., Zheng, K., Yang, Y., and Wang, X. (2020). An explainable machine learning framework for intrusion detection systems. *IEEE Access*, 8:73127–73141.

[201] Wang, W., Tang, P., Lou, J., and Xiong, L. (2021a). Certified robustness to word substitution attack with differential privacy. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1102–1112.

[202] Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., Chowdhery, A., and Zhou, D. (2022). Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

[203] Wang, Y., Wang, P., Wang, Z., and Cao, M. (2021b). An explainable intrusion detection system. In *2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, pages 1657–1662. IEEE.

[204] Webersinke, N., Kraus, M., Bingler, J. A., and Leippold, M. (2021). Climatebert: A pretrained language model for climate-related text. *arXiv preprint arXiv:2110.12010*.

[205] Wei, D., Dash, S., Gao, T., and Gunluk, O. (2019). Generalized linear rule models. In *International Conference on Machine Learning*, pages 6687–6696. PMLR.

[206] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

[207] Wickramasinghe, C. S., Amarasinghe, K., Marino, D. L., Rieger, C., and Manic, M. (2021). Explainable unsupervised machine learning for cyber-physical systems. *IEEE Access*, 9:131824–131843.

[208] Wu, B., Chen, S., Gao, C., Fan, L., Liu, Y., Wen, W., and Lyu, M. R. (2021a). Why an android app is classified as malware: Toward malware classification interpretation. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 30(2):1–29.

[209] Wu, J., Zhou, M., Zhu, C., Liu, Y., Harandi, M., and Li, L. (2021b). Performance evaluation of adversarial attacks: Discrepancies and solutions. *arXiv preprint arXiv:2104.11103*.

[210] Xia, P., Zhang, L., and Li, F. (2015). Learning similarity with cosine similarity ensemble. *Information sciences*, 307:39–52.

[211] Xu, D., Fan, S., and Kankanhalli, M. (2023). Combating misinformation in the era of generative ai models. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 9291–9298.

[212] Xu, F., Sheng, V. S., and Wang, M. (2021). A unified perspective for disinformation detection and truth discovery in social sensing: a survey. *ACM Computing Surveys (CSUR)*, 55(1):1–33.

[213] Yang, L., Guo, W., Hao, Q., Ciptadi, A., Ahmadzadeh, A., Xing, X., and Wang, G. (2021). {CADE}: Detecting and explaining concept drift samples for security applications. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2327–2344.

[214] Zago, M., Gil Pérez, M., and Martínez Pérez, G. (2021). Early dga-based botnet identification: pushing detection to the edges. *Cluster Computing*, 24(3):1695–1710.

[215] Zebin, T., Rezvy, S., and Luo, Y. (2022). An explainable ai-based intrusion detection system for dns over https (doh) attacks. *IEEE Transactions on Information Forensics and Security*.

[216] Zeng, N., Wu, P., Zhang, Y., Li, H., Mao, J., and Wang, Z. (2024). Dpmsn: A dual-pathway multiscale network for image forgery detection. *IEEE Transactions on Industrial Informatics*.

[217] Zhang, D., Zhang, Q., Zhang, G., and Lu, J. (2019). Freshgraph: A spam-aware recommender system for cold start problem. In *2019 IEEE 14th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pages 1211–1218. IEEE.

[218] Zhang, H., Zhu, C., Sang, G., and Questier, F. (2024). Effects of digital media literacy course on primary school students' digital media literacy: an experimental study. *International Journal of Technology and Design Education*, 34(1):1–17.

[219] Zhang, S., Zhang, J., Song, X., Adeshina, S., Zheng, D., Faloutsos, C., and Sun, Y. (2023a). Page-link: Path-based graph neural network explanation for heterogeneous link prediction. In *Proceedings of the ACM Web Conference 2023*, pages 3784–3793.

[220] Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

[221] Zhang, Y., Wu, P., Li, H., Liu, Y., Alsaadi, F. E., and Zeng, N. (2023b). Dpf-s2s: A novel dual-pathway-fusion-based sequence-to-sequence text recognition model. *Neurocomputing*, 523:182–190.

[222] Zhang, Z., Li, J., Fukumoto, F., and Ye, Y. (2021). Abstract, rationale, stance: A joint model for scientific claim verification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3580–3586.

[223] Zhang, Z., Zhao, J., and Yan, X. (2018). A web page clustering method based on formal concept analysis. *Information*, 9(9):228.

[224] Zhou, Q., Li, R., Xu, L., Nallanathan, A., Yanga, J., and Fu, A. (2022). Sufficient reasons for a zero-day intrusion detection artificial immune system. *arXiv preprint arXiv:2204.02255*.

[225] Zhu, D., Xi, T., Jing, P., Wu, D., Xia, Q., and Zhang, Y. (2019). A transparent and multimodal malware detection method for android apps. In *Proceedings of the 22nd international ACM conference on modeling, analysis and simulation of wireless and mobile systems*, pages 51–60.

[226] Zhu, X., Zhang, Y., Zhang, Z., Guo, D., Li, Q., and Li, Z. (2022). Interpretability evaluation of botnet detection model based on graph neural network. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–6. IEEE.

[227] Zolanvari, M., Yang, Z., Khan, K., Jain, R., and Meskin, N. (2021). Trust xai: Model-agnostic explanations for ai with a case study on iiot security. *IEEE Internet of Things Journal*.