The SPATIAL Architecture: Design and Development Experiences from Gauging and Monitoring the AI Inference Capabilities of Modern Applications

Abdul-Rasheed Ottun[1], Rasinthe Marasinghe[1], Toluwani Elemosho[1], Mohan Liyanage[1],
Mohamad Ragab[10], Prachi Bagave[2], Marcus Westberg[2], Mehrdad Asadi[13], Michell Boerger[3],
Chamara Sandeepa[4], Thulitha Senevirathna[4], Bartlomiej Siniarski[4], Madhusanka Liyanage[4], Vinh Hoa La[5],
Manh-Dung Nguyen[5], Edgardo Montes de Oca[5], Tessa Oomen[8], João Fernando Ferreira Gonçalves[8],
Illija Tanasković[7], Sasa Klopanovic[7], Nicolas Kourtellis[11], Claudio Soriente[9], Jason Pridmore[8],
Ana Rosa Cavalli[5], Drasko Draskovic[7], Samuel Marchal[6], Shen Wang[4], David Solans Noguero[11],
Nikolay Tcholtchev[13, 3], Aaron Yi Ding[2] and Huber Flores[1]
[1]University of Tartu, Estonia; [2]Delft University of Technology, Netherlands; [3]Fraunhofer Institute for Open Communication Systems, Germany; [4]University College Dublin, Ireland; [5]Montimage, France;
[6]VTT Technical Research Centre of Finland Ltd, Finland; [7]Mainflux, Serbia; [8]Erasmus University Rotterdam, Netherlands;
[9]NEC Labs, Germany; [10]University of Southampton, United Kingdom;[11]Telefónica Research, Spain;
[12]Vrije Universiteit Brussel, Belgium; [13]RheinMain University of Applied Sciences, Germany
firstname.lastname@{ut.ee, tudelft.nl, fraunhofer.de, ucd.ie, montimage.com
vtt.fi, mainflux.com, eshcc.eur.nl, neclab.eu, soton.ac.uk, telefonica.com, vub.be, hs-rm.de}

***Abstract*—Despite its enormous economical and societal impact, lack of human-perceived control and safety is re-defining the design and development of emerging AI-based technologies. New regulatory requirements mandate increased human control and oversight of AI, transforming the development practices and responsibilities of individuals interacting with AI. In this paper, we present the SPATIAL architecture, a system that augments modern applications with capabilities to gauge and monitor trustworthy properties of AI inference capabilities. To design SPATIAL, we first explore the evolution of modern system architectures and how AI components and pipelines are integrated. With this information, we then develop a proof-of-concept architecture that analyzes AI models in a human-in-the-loop manner. SPATIAL provides an AI dashboard for allowing individuals interacting with applications to obtain quantifiable insights about the AI decision process. This information is then used by human operators to comprehend possible issues that influence the performance of AI models and adjust or counter them. Through rigorous benchmarks and experiments in real-world industrial applications, we demonstrate that SPATIAL can easily augment modern applications with metrics to gauge and monitor trustworthiness, however, this in turn increases the complexity of developing and maintaining systems implementing AI. Our work highlights lessons learned and experiences from augmenting modern applications with mechanisms that support regulatory compliance of AI. In addition, we also present a road map of on-going challenges that require attention to achieve robust trustworthy analysis of AI and greater engagement of human oversight.**

***Index Terms*—Practical Trustworthiness, Industrial Use Cases**

I. Introduction

The adoption of AI is imminent in everyday applications. The AI market value is expected to reach a valuation of two trillion USD by 2030 [1], emphasizing the impact of AI on current software practices and systems development. Machine and deep learning components (aka AI components) are part of larger systems that provide autonomous decision capabilities for modern applications. AI components implement machine/deep learning pipelines to build AI models. These models are improving the perception, experience and interaction between users and digital applications [2], providing human-like and insightful functionality that facilitates application usage and provides added value to users. Examples of this include advanced Chat-bots (ChatGPT, Gemini, Ernie) for e-commerce recommendations [3], optimal route planning for practical drone delivery [4], [5] and sophisticated diagnosis capabilities in healthcare applications [6] to mention some. A key limitation for the adoption of AI at scale is its inherent black-box characteristics [7]. Indeed, the incomprehensible advanced performance of AI caused distrust in humans when massively trained, leading to the release of an open global petition in March 2023 to slow down the developments of AI for at least 6 months [8]. AI probabilistic decision nature cannot be dissected using existing methods to verify software [9]. Besides this, AI models can be easily hampered throughout their life cycle, making them vulnerable and exposed to many threats, impacting their autonomous decisions. This is worrisome in cybersecurity situations, where AI models can be (purposely) attacked to perturb their inference process, which can cause life-critical consequences for people and society.

As recognized by all economic and regulatory frameworks, with a primary emphasis on the EU but also encompassing the US and China, artificial intelligence (AI) stands out as the pivotal focus to developing a trustworthy technology. Traditionally, trustworthy computing ensures that a piece of software is trustful to users by verifying several of its properties, e.g., robustness, reliability, resilience, accuracy and so on. Audit and accountability compliance on trustworthy software is simpler as there is a quantifiable understanding of its performance sensitivity to drifts and errors. As trustworthy verification cannot be conducted directly with AI using traditional methods, there is a lack of transparency, accountability and resilience towards AI technologies. This has made Europe to impose strict regulations for the use of AI, becoming a benchmark at an international level. AI trustworthiness extends fundamental principles of trustworthy computing with additional properties that have been considered and some defined by regulatory entities (EU AI act and US Executive Order 13960). Trustworthy AI is valid, reliable, safe, fair, free of biases, secure, robust, resilient, privacy-preserving, accountable, transparent, explainable, and interpretable [10]. Notice however, that AI trustworthiness is an ongoing process whose definition is evolving continuously and involves collaboration among technologists, developers, scientists, policymakers, ethicists, and other stakeholders. As emerging regulatory standards mandate increased human control and oversight of AI, this concurrently reshapes the development practices and responsibilities of individuals engaging with AI. Moreover, new methods and approaches that help to understand the behavior of AI are being investigated or have re-gained attention, e.g., Explainable AI (XAI) methods [11]. As applications equipped with AI continue proliferating every aspect of human life, new methods are required to gauge, adjust and monitor the trustworthiness of AI inference capabilities.

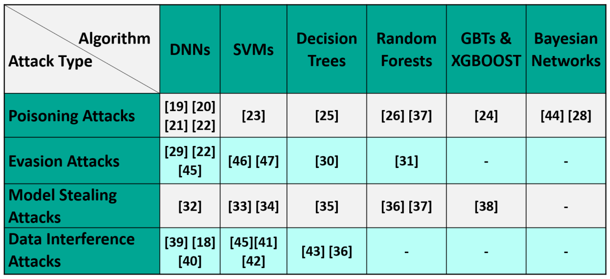
We contribute SPATIAL, a proof-of-concept architecture that augments modern applications with capabilities to robustly gauge and monitor the trustworthiness of AI in a human-in-the-loop manner. To achieve this, SPATIAL uses an AI dashboard and instruments applications with AI sensors. Conceptually, an AI dashboard serves as a tool to provide insights to human operators, enabling them to monitor and adjust AI trustworthiness according to their preferences. Additionally, it facilitates the verification of AI systems for potential audits and ensures compliance with accountability regulations set by regulatory bodies. In parallel to this, AI sensors that monitor specific trustworthy properties are instrumented within applications. Simply put, an AI dashboard shows to users quantifiable metrics extracted by AI sensors [12]. To design SPATIAL, first, we investigate the sensitivity of machine learning pipelines to (induced/non-induced) changes - from input data to model deployment. With this information, trustworthy metrics that can be instrumented as AI sensors are reviewed in current state-of-the-art. For instance, a sensor for fairness can be instrumented to analyze raw input data as well as to characterize fairness in decision making after model deployment [13]. Notice that currently, there is a misalignment between regulatory (legal) and technical trustworthy requirements. Thus, relevant metrics are selected from a technical point of view. Naturally, as regulatory trustworthiness evolves, it is possible to replace technical metrics with alternatives that adjust better to legal requirements. To augment modern applications with AI dashboards and sensors, we develop SPATIAL following a micro-service pattern. The key idea of using this pattern is that each micro-service contributes with the specific functionality to monitor a trustworthy property, and this functionality is requested by an AI sensor instrumented in the application (like an API). Unlike the blockchain approach for embedding trustworthiness to AI that is complex and less scalable [14], the micro-service pattern provides flexibility to add more metrics dynamically. Besides this, the pattern also helps analyse a specific set of trustworthy properties. Indeed, as demonstrated by previous work, trustworthy properties are not agnostic. Thus, the number of trustworthy properties that can be derived from an application depends on its inherent characteristics [10], [15]. Through rigorous analysis and benchmarks conducted in real industrial use cases, we evaluate the performance and scalability of SPATIAL. Our results indicate that to measure trustworthiness in AI is necessary to instrument every step of the AI pipelines with sensors. Moreover, our results also suggest that AI dashboard and sensors are useful to individuals to monitor AI inference capabilities, but it increases the complexity of developing and maintaining AI components in modern applications. Our work also highlights lessons learned from designing and developing SPATIAL, and describes on-going challenges that require attention to achieve a robust analysis of AI trustworthiness and greater engagement of human oversight.

II. Related Work

AI trustworthiness: All regulatory and economic frameworks have recognized the need for trustworthiness in AI. As a result, several initiatives, projects and efforts are ongoing to define how to verify it. EU projects, such as EU TRUST-AI (https://trustai.eu/), EU SPATIAL (https://spatial-h2020.eu/) and EU TAILOR (https://tailor-network.eu/) have proposed principles and guidelines to ensure trustworthiness in AI development practices. Likewise, leading technological vendors have proposed frameworks to achieve AI trustworthiness, including, IBM's AI fairness 360, the what-if tool and ML fairness gym of google, Microsoft's fairlearn, Linkedin Fairness Toolkit (LIFT), AT&T software System to Integrate Fairness Transparently (SIFT), and Fat forensic. Other initiatives also include, PwC AI trust index, AI trust and transparency of Microsoft, and AI Impact Assessment of Open AI. In parallel to this, development toolkits also have been released by private vendors and open-source communities. For instance, Google's model card toolkit measures transparency in AI models. Other development initiatives to verify integrity and robustness of AI include open-source SHAPASH [16], IBM AI explainability 360 toolkit [17], Microsoft Interprete

ML, and IBM Adversarial Robustness 360 toolkit. While there is a clear overlapping between all these works, a key challenge that remains unexplored is identifying essential and general requirements of trustworthiness. Unlike others, our work investigates how to augment modern applications with practical trustworthiness analysis and shares experiences and lessons learned from our developments.

Human oversight and AI: Regulatory trustworthiness mandates human oversight in AI developments. While multiple frameworks have been developed to measure different trustworthy properties [18], it is still unclear the role that humans play in the monitoring and supervision [19], [20]. XAI methods are the most common method to communicate the logic of AI models to users via (optimized) explanations, numerical values, visual diagrams, and so on [21]. At the machine and deep learning levels, several tools and frameworks are available to tune the inference process of AI models. For instance, TensorLeap (https://tensorleap.ai/), Neptune AI (https://neptune.ai/), and Comet ML (https://www.comet.com/site/). Unlike others, our SPATIAL uses an AI dashboard to communicate to human operators the inference capabilities of AI, making it possible to adjust it.



Algorithm / Attack Type	DNNs	SVMs	Decision Trees	Random Forests	GBTs & XGBOOST	Bayesian Networks
Poisoning Attacks	[19] [20] [21] [22]	[23]	[25]	[26] [37]	[24]	[44] [28]
Evasion Attacks	[29] [22] [45]	[46] [47]	[30]	[31]	-	-
Model Stealing Attacks	[32]	[33] [34]	[35]	[36] [37]	[38]	-
Data Interference Attacks	[39] [18] [40]	[45][41] [42]	[43] [36]	-	-	-

Fig. 1: AI perturbations based on algorithm type and attack.

AI perturbations: Attacks on machine learning systems can be identified by threat modeling using frameworks like ENISA, MITRE, NIST, IBM, Microsoft. AI pipelines implement a set of steps to build AI models. These models can be hampered by induced and non-induced changes in any step of its construction [22]. Non-induced changes occur due to situational events, e.g., environment, data quality and failures of devices. Induced changes (aka adversarial attacks) are perpetrated by an attacker with the main intention to control/induce the inference process of AI models. Poisoning attacks are of a significant issue as contaminate the data used for model training [23]–[29], [29]–[41], [41]–[50]. Adversarial attacks can also occur at the model level by changing internal structure and parameters of the model [27], [31], [34], [36], [51]–[53], e.g., model evasion, model stealing. A summary of attacks investigated in the relevant literature in last years is shown in Figure 1. From the figure, it is possible to observe the type of attack that can be performed depending on each AI algorithm used for training. SPATIAL augments modern applications with functionality to gauge and monitor changes in AI inference capabilities such that human operators can visualize and react to them.

III. Background and motivation

We continue by analyzing how modern applications implement AI components and their respective AI pipelines for building AI models. After this, we reflect on regulations for the use of AI and its implications for software development practices and systems deployment.

Modern architectures: As shown in Figure 2, the underlying system of modern applications have evolved considerably from its fundamental client-server architecture. At the same time, there has been a rise in design and development considerations. In early developments, in a basic client-server architecture, end devices acting as clients send requests to the server. At the server, the request is then processed and a response is sent back to the client (Figure 2(a)). After this, more advanced architectures are designed to collect data in a centralized manner (at the server) from users interacting with applications. This data is then used to train machine learning models to improve certain functionality over time (Figure 2(b)). Further developments have made these architectures capable of collecting data from clients in a distributed manner, such that more robust datasets can be used to train models. Currently, a global model is trained by data contributions of clients collected in a privacy-preserving manner, e.g., using federated learning, once trained, this model is then propagated to all the end devices. Figure 2(c) extends the ML architecture presented in [54] to depict the latest advances of distributed training.

AI model construction in a nutshell: Applications equipped with AI models implement pipelines that facilitate their construction and incremental improvement over time. The standard pipeline for building an AI model can be summarized in Figure 4(a). Applications implement these typical steps to update models continuously as new data contributions are obtained. In the first step (data collection), available data is cleaned and prepared using common methods to enhance its quality, e.g., missing data, removing duplicates, and data augmentation [55]. After this step, data is transformed into a suitable input for the AI algorithm, meaning data is labelled, e.g. using human annotators. Next, the training process takes place. Here, an algorithm is selected, e.g., Random Forrest, Support Vector Machine; then the training process is decided, e.g., data parallelization or model partition [56], and the model is evaluated, e.g., using cross-validation [57]. Lastly, the model is deployed and the performance is evaluated within applications. In classical architectures, models require re-training and re-deploying as new data contributions are obtained. In newer paradigms, such as federated learning, the model is updated by a global aggregator, which combines contributions from clients, such that the later resulting model is propagated back to all the contributors.

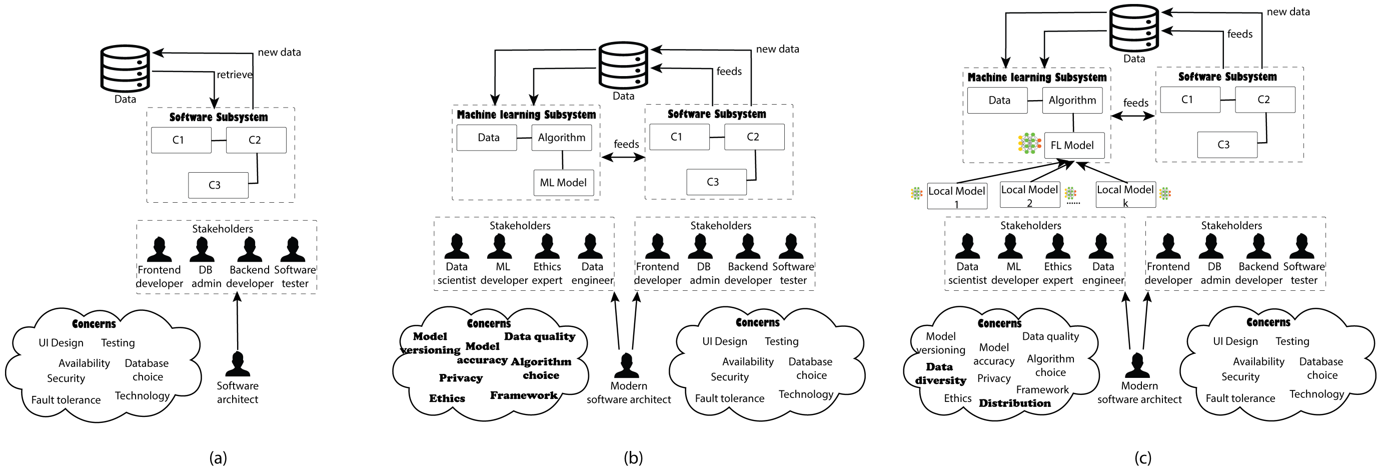


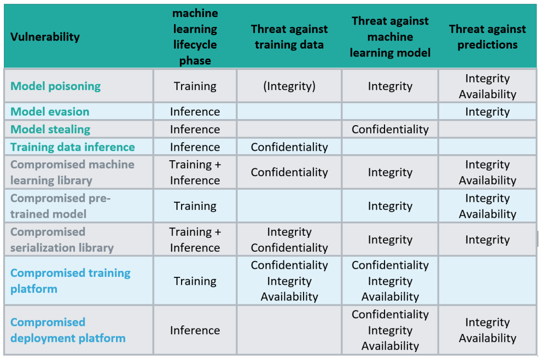
Fig. 2: Evolving system architecture; a) Basic client-server architecture [54]; b) Machine learning architecture [54] and c) Federated learning architecture.

AI regulations: AI models are trained from data contributions collected over time, each contribution helping to tune their probabilistic nature. AI regulations thus define the properties for verifying and validating the correct development and usage of AI models. The General Data Protection Regulations (GDPR) stipulates the guidelines for dealing with personal data within the European Union (EU), putting forward fairness, security, privacy, trust, transparency, and explanation considerations during software and AI-based solution development. These principles are also described in the US AI Act, and other countries have also considered similar regulations, for instance, China, Japan, Brazil, and Canada. Given these considerations, modern applications have to implement mechanisms or tools that allow individuals to understand the inference capabilities of AI. This, however, requires to inspect the whole construction of AI models.

IV. The SPATIAL architecture

We next describe how modern applications are augmented with SPATIAL, such that it is possible to gauge and monitor the trustworthiness of its AI components. To do this, first, we analyze how sensitive AI pipelines are to vulnerabilities that can change the inference logic of AI models during their construction. After this, we introduce the concepts of AI dashboards and sensors, which encapsulate complexity of the trustworthy analysis. With this information, we then provide an overview of the SPATIAL system.

AI vulnerabilities: Machine learning vulnerabilities exist throughout the AI pipeline and these can be exploited to change the AI inference logic. We enumerate the most common and critical vulnerabilities by relying on the CIA (confidentiality, integrity, and availability) approach. CIA provides a qualitative analysis to model the impact of vulnerabilities on AI models. Confidentiality depicts the level of access to AI models. Confidentiality is not limited to preventing access to a machine learning model but also to ensuring that its output predictions do not leak information that can be used to understand and reproduce its decision making or reconstruct its training data. Similarly, integrity relates to preserving expected behavior, level of performance, and quality of predictions under any conditions, including attack. Likewise, availability refers to the idea that accurate predictions are produced, that reflect those seen in testing, and in a timely manner. Models are vulnerable throughout their construction life cycle pipeline. Figure 3 summarizes these vulnerabilities together with associated security attributes that can lead to compromise. This suggests that metrics that quantify trustworthiness are required to be instrumented in different steps of the AI pipelines.



Vulnerability	machine learning lifecycle phase	Threat against training data	Threat against machine learning model	Threat against predictions
Model poisoning	Training	(Integrity)	Integrity	Integrity Availability
Model evasion	Inference			Integrity
Model stealing	Inference		Confidentiality	
Training data inference	Inference	Confidentiality		
Compromised machine learning library	Training + Inference	Confidentiality	Integrity	Integrity Availability
Compromised pre-trained model	Training		Integrity	Integrity Availability
Compromised serialization library	Training + Inference	Integrity Confidentiality	Integrity	Integrity
Compromised training platform	Training	Confidentiality Integrity Availability	Confidentiality Integrity Availability	
Compromised deployment platform	Inference		Confidentiality Integrity Availability	Integrity Availability

Fig. 3: Vulnerabilities against machine learning systems.

The SPATIAL architecture: SPATIAL augments the latest architectures by building upon the standard machine learning pipeline that constructs and updates AI models. Figure 5 shows the overall concept. Applications are instrumented with AI sensors (for each trustworthy property), and these sensors gauge and monitor the inference capabilities of AI models. At the architecture level, Figure 4(c) shows the system components augmented in modern applications. Notice that the architecture is easily integrated into any application as the trustworthy analysis is applied over the model and data. In practice, the trustworthy properties have to be monitored over time as these can change as the AI model gets updated. Besides this, it has been demonstrated that trustworthy properties can be considered as trade-offs within applications [58], suggesting that modifying one property can impact others, e.g., robustness vs privacy, accuracy vs fairness, transparency vs security. Moreover, different types of applications have different predominant characteristics, influencing the extraction of a trust score and thus obstructing the adoption of a generic certification scale [59]. By using AI sensors, it is possible then to *quantify* the compliance of AI against available requirements. The main reason for abstracting trustworthy properties into sensors is that a sensor enables continuous monitoring of applications during runtime. AI sensors are software-based (aka virtual sensors) and are instrumented within the source code of an application to monitor specific parts of its code execution or can be instrumented as a concurrent process to monitor the behaviour of the overall application. Thus, AI sensors can be considered APIs. Another reason for instrumenting and abstracting modern applications with AI sensors is to foster a correct-by-construction approach, such that standard trustworthy properties are considered from the early design and development phases of AI. Measurements obtained by the AI sensors are shown to human operators using the AI dashboard, such that human operators can aid in overseeing the development of AI models. Human feedback to change AI behavior is applied directly to the AI pipeline. Figure 4(b) shows the additional steps that are introduced. As any step can be easily hampered to change the model inference process, AI sensors are required to be instrumented across the pipeline. AI sensors are built using specific metrics to extract trustworthy properties, e.g., XAI methods, fairness metrics, and accuracy, among others.

V. Implementation and deployment

System implementation: To demonstrate how modern applications can be augmented to gauge and monitor trustworthy properties from AI models. We design, develop and deploy a proof-of-concept system architecture. Our current implementation uses a micro-service API gateway to support various micro-services. These micro-services implement different metrics to analyze specific trustworthy properties. AI sensors are instrumented within applications and request the functionality of a specific metric in an input/output manner. The main reason for using micro-services architecture is to add and replace metrics with ease. Indeed, currently, there is a misalignment between legal regulatory and technical trustworthiness. Thus, technical metrics that fulfil and comply with regulatory requirements are meant to evolve over time. Another reason to rely on micro-service patterns is to augment dynamically the capacity of each individual metric to handle the workload. The source of this workload considers 1) several different applications requesting the metric and 2) workload caused by continuous monitoring of the metric. To implement our API gateway, we rely on the open-source Kong technology. Kong can be easily extended through OpenAPI and configured to support continuous integration, facilitating re-deployment and managing versioning of our prototype. The API Gateway manages the communication flow, ensuring that each microservice receives the necessary input, processes it, and returns the appropriate response. Micro-services connected to the API gateway rely on docker containerization to encapsulate each metric. Technical configurations are in subsection VI-B. The overall system is deployed in the computing infrastructure provided by the supercomputer LUMI at UT HPC datacentre [60].

AI Dashboard: The AI dashboard incorporates data-driven visualizations, dynamic displays, and quantifiable metrics derived by leveraging AI sensors. These sensors monitor specific trustworthiness properties in various applications within SPATIAL's use case applications. The dashboard enables human operators to interactively interface with different microservices running within the SPATIAL's architecture, see figure 8(a); and communicates valuable insights that are tailored to human operators, prioritizing clarity and adaptability to accommodate individual preferences for fulfilling compliance with regulatory requirements. At the front end, we employed a collection of contemporary web technologies, scripts, libraries, and design frameworks for deploying a robust user interface that is flexible, adaptive, and fluid for reporting insights delivered from AI sensors on any device. We relied on React [], Bootstraps responsive design [], and Tailwind CSS [] custom styling for creating visually appealing UI elements and components. Dataset handling and responsive chart visualization and parsing of CSV data were managed using D3 [], Chart.js [] and papaparse []. Okta's authentication framework [] enforced security by providing robust authentication and access control to meet strict security standards. The backend of the dashboard piggybacks the existing backend infrastructure of the SPATIAL architecture that runs multiple independent microservices. Each service is the self-contained unit responsible for evaluating specific, trustworthy AI properties like explainability, resilience, fairness, and privacy using metrics. Using Docker [], we containerized each service code alongside its dependencies, libraries and configuration files and into a container for isolated running. The services are managed with the KONG API Gateway []. The gateway serves as the entry point for API requests and handles each request, load balancing, and authorization for the security and scalability of the services. Each service API calls are managed asynchronously, allowing the request to be initiated from the dashboard's frontend (user interface) to the backend without any disruption in operations while the request is processed.

Trustworthy metrics for AI sensors: Micro-services implement different metrics to quantify specific trustworthy properties. Applications are instrumented with AI sensors requesting each metric functionality. Current micro-services implement metrics that can be used to support the resilience and accountability of AI models. Accountability metrics support the ability to explain the source causes that led to a decision. Thus, accountability is supported by implementing the XAI SHAP method. SHAP fosters transparency of inference capabilities

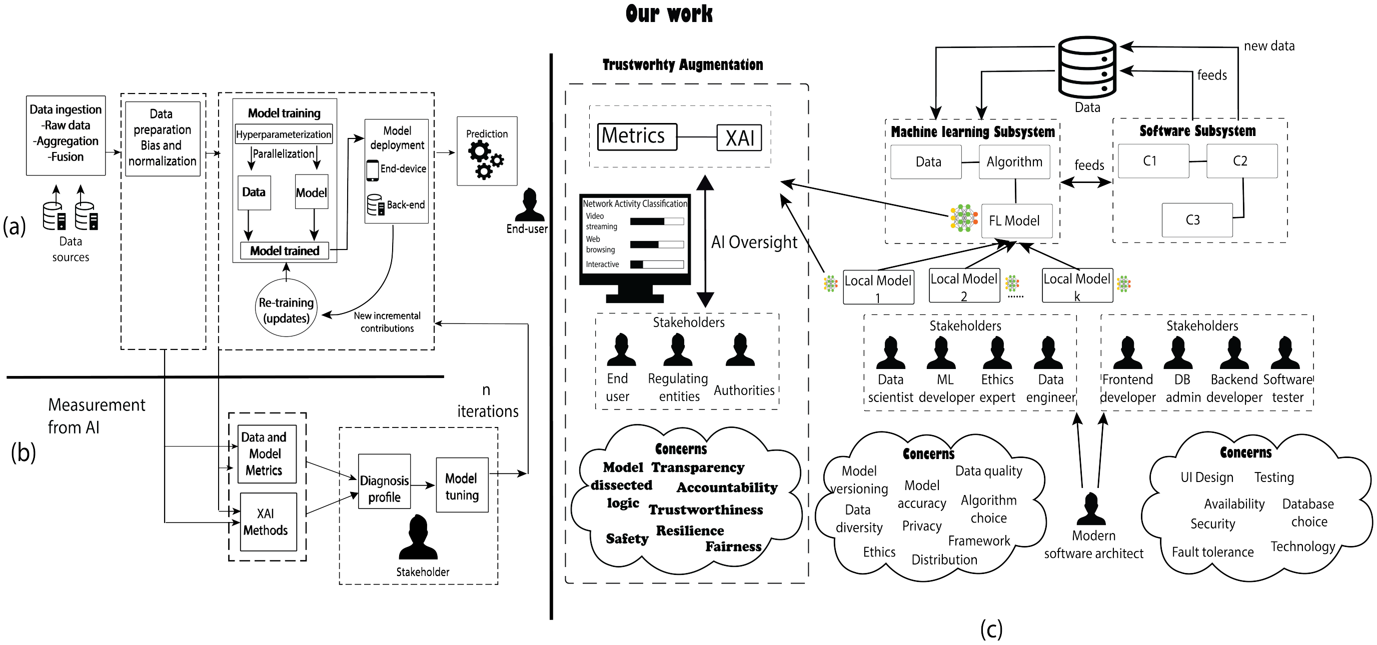


Fig. 4: AI model construction; a) standard pipeline to construct machine learning models; b) Augmented pipeline to analyze trustworthy trade-offs; and c) Conceptual modern system architecture equipped with methods to monitor trustworthiness.

of AI by highlighting the most important part of the data used for learning. Likewise, resilience metrics quantify the ability of models to resist and recover from an exploited machine learning vulnerability. Resilience insights are thus estimated by calculating complexity and impact metrics on model and data [46]. Complexity quantifies the effort required by an attacker to achieve a successful attack. The higher the complexity, the more difficult it is for the attack to hamper the model. Similarly, impact quantifies the extent of the attack's effect on the AI models within a system. The higher the impact, the more vulnerable the AI model becomes in that system. Besides this, our architecture also implements a machine learning component, where several AI algorithms can be passed a dataset to create an AI model. This component also allows us to provide performance metrics about the AI model, e.g., accuracy and precision.

VI. The Experiments

We conduct experiments to analyze the performance and scalability of SPATIAL as industrial modern applications are augmented with it. Two sets of experiments are conducted. The first focuses on gauging the trustworthiness properties of AI components of applications, whereas the second focuses on analyzing the capacity of the system to monitor applications and handle workload of concurrent requests. In the following, we provide a detail description of the experimental setup.

A. Monitoring performance.

We next evaluate how SPATIAL can gauge and monitor the inference capabilities of AI. To do this, we analyze how changes in AI models can be quantified and monitored over time. Monitoring the inference process is important to identify when models have been compromised. The first use case focuses on analyzing sensor data to trigger medical emergency support whereas, the second application depicts a network activity classification system, where network data is poisoned to disguise the classification model.

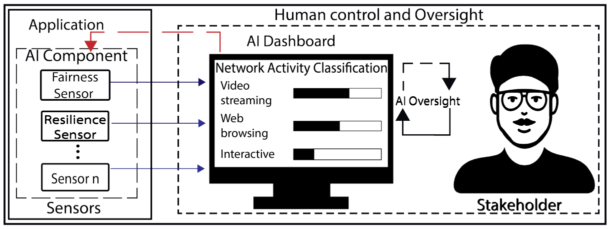


Fig. 5: [Anon] concept overview.

Use case 1: Medical e-calling application: It is a mobile application, part of an e-calling system, that uses accelerometer data to detect the falling of an elderly person. As the falling event is detected, the application triggers an emergency call to request medical assistance.

Dataset and model: The UniMiB SHAR dataset [61] was employed in training five different ML models, Logistic Regression (LR), Random Forest (RF), Multilayer Perceptron (MP), Deep Neural Network (DNN), and Decision Tree (DT). The UniMiB dataset is a benchmark dataset for human activity and fall detection comprising 11771 acceleration samples from 30 subjects, 9 classes representing activities of daily living (ADL), and 8 classes representing falls.

Adversary model and assumptions: We assume a black-box attacker model where the attacker has only access to the training data but has no knowledge about the underlying structure of the utilized model. Furthermore, we expect the attacker to be capable of randomly poisoning the data up to a poisoning rate of p. Thereby, we expect that the attacker poisons the data by performing a random label-flipping attack.

Setup and procedure: The label flipping attack is performed systematically to different subsets of the dataset. Precisely, the attack is executed at varying poisoning rates p of 0% (baseline), 1%, 5%,10%, 20%, 30%, 40%, and 50%, respectively. Baseline results without data poisoning are also collected for reference purposes. Afterwards, the respective ML model (e.g., DNN, DT, RF, LR, or MLP) is trained on the poisoned training data set and then evaluated with the retained clean test data set based on the accuracy, precision, and recall evaluation metrics. In addition, we explore the impact of the attack to the model explainability. More specifically, we also calculate the similarity of SHAP explanations of the DNN model for each of the varying poisoning rates. To realize this, we determine the five nearest neighbours regarding the Euclidean distance for each fall instance in the retained clean test set. We then measure the average distance of the corresponding SHAP explanations. Finally, we average the average distances of explanations, resulting in an average distance of explanations of similar instances across the test set w.r.t. the class "fall".

Use case 2: Network activity classification application: The second use case is a network monitoring application that examines IP and TCP/UDP data headers. The application is able to identify the type of activity an online user is performing. Three common types of online activities are considered: Web browsing, Web interactions and video streaming. Network monitoring is important to design security policies, safeguard user privacy and efficient dynamic allocation of resources, particularly in 4G/5G networks.

Dataset and model: We setup a testbed to collect network data of user activities using our application. Network data depicts real online activities of users at [Annon. Vendor], a network data monitoring provider. We rely on Wireshark to create pcap files with a size of 2.15 GB that contain the activities of users captured through the network traffic. Our datasets comprise multiple network traces, linked to different users. The network traffic traces contain essential information such as the source and destination IP addresses, protocols, port numbers, packet timestamps, packet size, to mention some. We clean the dataset using standard methods and select relevant features to identify the previous described activities. After applying filtering methods, the final dataset consists of 382 labelled traces across three traffic classes: Web, Interactive, and Video activities, with 304, 34, and 44 traces respectively. The processed CSV files derived from this dataset are used for the analysis and evaluation of our AI-based classification model. Feature extraction reveals 21 features categorized into five main categories: duration, protocol, uplink, downlink, and speed. We employ various machine learning classification algorithms, including Neural Networks (NN), LightGBM (LGBM), and XGBoost.

Adversary model and assumptions: We assume a white-box attack model, where the attacker has a complete knowledge about the AI model structure. This type of attack depicts a common situation where the AI models are hampered from inside an organization. By injecting commonly use poisoning and evasion attacks, the attacker's objective is to compromise the integrity of our models leading to a significant degradation in the model's accuracy. Fast Gradient Sign Method (FGSM) is a technique used in adversarial ML to generate adversarial examples by adding a small amount in the direction of the gradient of the loss function with respect to the input. Resilience of models against an evasion attack is quantified based on impact and complexity metrics. Here, complexity is measured by characterizing the processing power required to generated evasion data points. Impact on the other hand, it is measured by counting each successful misclassification gained through those evasion data points. In parallel to this, GAN-based poisoning attack is also performed and the goal is to generate synthetic data that looks very similar to the real data. Random swapping labels attack chooses randomly two samples of the training dataset and swaps their labels. Target label flipping attack flips the labels of some samples from one class to the target class (e.g., Video class). Here, complexity and impact are also estimated based on different observations. Complexity is measured by quantifying the percentage of data that is poisoned out of all the data used for training the model. Similarly, impact is measured by using the drifts in any performance metric of the model, e.g., accuracy, F1-score.

Setup and procedure: We generated 103 adversarial samples from the 103 test data samples that were initially obtained. After this, the white-box FGSM evasion attack is launched. For GAN-based attack, we use CTGAN [62] for modelling tabular data to generate 5000 synthetic samples. For other poisoining attacks, such as label flipping and random swapping labels attacks, the poisoning rates are 0% (baseline), 10%, 20%, 30%, 40%, 50%. Subsequently, the corresponding ML models (e.g., NN, LightGBM and XGBoost) are retrained using the manipulated training dataset and compared against the baseline to identify performance degration based on accuracy, precision, and recall metrics.

B. *Capacity-load performance*

Experimental setup: To verify the performance and scalability of SPATIAL, SPATIAL is deployed following the setup shown in Figure 8(a). The system consists of six (6) different machines, one acting as the integration/API gateway, and others as back-end micro-services. The machine running the Kong Gateway consists of 32 vCPUs and 64 GB of RAM running Linux. The remaining machines host a specific service to extract a metric. Micro-services include, a LIME micro-service (4 vCPUs and 4 GB RAM); a SHAP micro-service (4 vCPUs and 4 GB RAM), an Occlusion-sensitivity micro-service (4 vCPUs and 8 GB RAM), an impact resilience micro-service (computing instance with NVIDIA A4000 GPU, Intel Xeon 2.10 GB CPU, and 128 GB RAM running Ubuntu 20.04), and an AI pipeline micro-service that provides performance indicators (8 vCPUs and 8 GB RAM)/. All micro-services are accessible through the API gateway, and requests to micro-services are specified by the clients. The system is deployed in the computing infrastructure provided by LUMI.

Tools and metrics: Once the system is deployed and running, capacity-based testing is performed to evaluate the performance of individual requests and concurrent requests, handled by the system as its usage increases, depicting an in production environment. To generate stress capacity load, we rely on JMeter, deployed in a different machine, but running in the same network as the SPATIAL deployment. JMeter is installed in a Windows machine with an 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz CPU and 16 GB RAM.

Experiment 1: We evaluate perturbations in the AI model and derive the impact poisoning attacks have on resilience. We also evaluate the analysis of SHAP and LIME values over model predictions. In the configuration process for the JMeter script, we create a test plan encompassing an ultimate thread group with a thread count set to 100 to simulate concurrent requests to the micro-services. To examine the performance of specific micro-services, an HTTP request sampler was added, specifying the server name, port, protocol, endpoint path. Parameters or file uploads were configured as necessary. To gauge response times, the Response Times Over Active threads or the Summary Report listener was incorporated into the test plan. These listeners provided detailed metrics, including average response time, throughput, and error rate for each micro-service.

Experiment 2: We next evaluate the performance of the system when handling heavier load induced by image inputs. In this case, when analyzing image-based samples, the analysis of methods, such as LIME, SHAP and Occlussion sensitivity increases. As a result, we analyze to what extent these services impact the overall response time. Notice that configuration presented in experiment 1 cannot be handled by these services when considering input images. As a result, with this setup, a different capacity load is generated. We select incremental concurrent load from 5 to 25 requests. Requests are also set to be sent to services with a ramp-up period of $1s$ in parallel.

VII. Results

Monitoring results on use case 1: Prior to poisoning the models, reference baselines of the models is established to measure performance deviation. Our performance evaluation indices, LR (73%), DNN (97%), RF (97%), DT (90%), and MLP (97%), respectively. Moreover, our results indicate that DNN, MLP and RF models are best suited for fall detection when compared to others. It is also possible to observe from the results that DNN, MLP, and RF are able to attain 97% accuracy and precision in performing the binary classification task but at slightly different recall rates, respectively. After this, models are poisoned, Figure 6 shows the results. From the figure, it is possible to observe that label flipping has a significant impact on model performance, with most metrics decreasing as the attack rate increased (Figure 6(a)-i shows accuracy, 6(a)-ii shows precision and 6(a)-iii shows recall). In line with this result, the average performance of all the models in accurately detecting falls before the data poisoning attack was 90%. However, this average performance starts to decline down to 75% as the data is gradually poisoned from 1% to 50%. We calculated a metric based on SHAP values which addresses the similarity of SHAP explanations of similar data points. Figure 6(a)-iv illustrates the results of this metric relative to the poisoning rate of the model. As can be seen from this figure, the metric is higher at higher poisoning rates, suggesting its capability of indicating poisoning of the data set. This result alone provides insights for detecting possible attacks on the model, requiring to monitor further the model to apply corrective actions, e.g., Label sanitization methods. Besides this, analysis of the result indicated that the high-performing models (DNN, MLP, and RF) showed relatively small performance losses at low attack rates (1% and 5%), indicating some degree of robustness in maintaining their capabilities to detect fall up to 5% poisoning rate, but this is lost when the intensity exceeded 5%. Interestingly, the random forest (RF) model showed better resilience against the poisoning attack. Even at a 30% poisoning rate, the RF model maintained an accuracy of 93%, close to its baseline performance. Only at a poisoning rate of 40% did a significant performance decrease occur, rendering the model unusable. The RF recall and precision metrics were also relatively stable, up to a 30% poisoning rate, further highlighting its robustness.

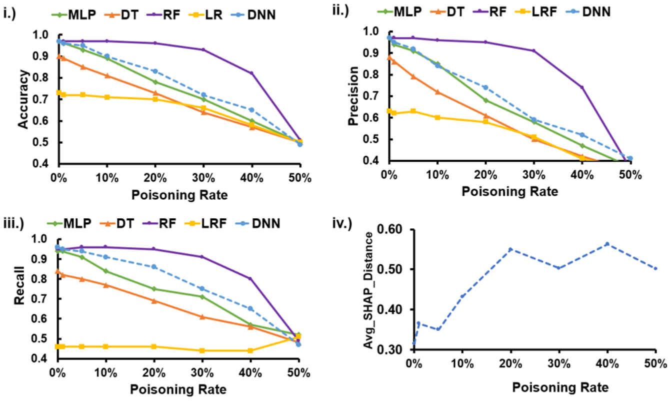


Fig. 6: Use case 1 results (Medical application); Effect of label flipping based on (i) accuracy, (ii) precision, (iii) recall; and (iv) poisoning quantification using SHAP dissimilarity

Monitoring results on use case 2: A reference baseline about the performance of our models for user activity classification is estimated to be NN (96%), LightGBM (94%) and XGBoost (94%). After this, the (FGSM) evasion attack is performed over the models, degrading their performance to NN (71%), LightGBM (72%) and XGBoost (54%). We then use SHAP to observe differences as models get hampered. Figure 7(a) and (b) shows the results of SHAP when applied to NN, before and after the evasion attack. From the result, it is possible to observe that shapley values for web activities have decreased around 16% for the udp_protocol, causing the feature to drop to the second place in ranking, while the importance of the tcp_protocol has almost doubled. This means that attacks on the model can easily induce misclassification of user activities. At the same time, it is possible to detect these changes with SHAP, however, the detection alone is insufficient to identify concrete causes nor overall performance degradation of the model, requiring additional information to be computed. Thus, *complexity and impact metrics* are calculated from the models using the methods presented in [63].

For each model, impact and complexity are estimated, NN (Impact 29%, Complexity 37.86 μs), LightGBM (Impact 28%, Complexity 37.86 μs) and XGBoost (Impact 45%, Complexity 37.86 μs). The results of the metrics indicate that XGBoost is (17%) more vulnerable for the FGSM attack when compared with the other two models. Moreover, since the FGSM generation was done with only the NN model, the complexity of the attack was always constant at around 37 μs. In parallel to this, in the case of poisoning attacks, SHAP can provide valuable insights to detect changes in performance. For instance, after label flipping and GAN-based poisoning are performed in our models, it is possible to observe shapley values for web activities have also changed significantly (tcp_protocol increases by 10% while udp_protocol decreased to half of it's initial importance). To reinforce this detection further, we then calculate impact and complexity metrics to analyze further the impact of poisoning in our NN model. Figure 7 shows the results estimated by impact and complexity metrics. From the results, we can observe how metrics changed based on the level of poisoning applied. We can observe that there is an increasing relative trend between increased poisoning and drift in impact and complexity.

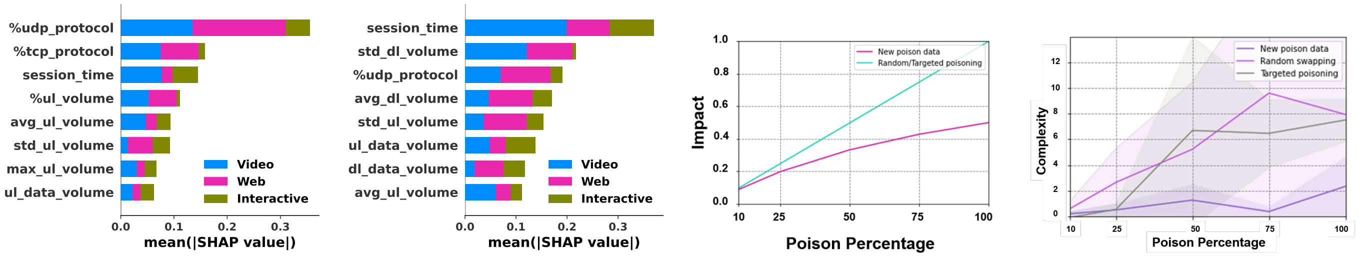
Capacity-load results: Experiment 1 results are shown in Figure 8(b) and Figure 8(c). The figures show capacity results when handling concurrent requests by the impact resilience micro-service and LIME/SHAP micro-services, respectively. From the results, it is possible to observe a lower response time for the evasion impact metric. Even with nearly 100 parallel requests, the numerical metric converges to an average of around 1600ms across the ramp-up time. Similarly, SHAP's and LIME's APIs under 100 requests are also presented in Figure 8(b). From this result, it possible to observe that SHAP's and LIME's explanations require an average processing times of 228.6 and 243.4 milliseconds, respectively. In both cases, the response times depict latencies that are tolerable by end-users and also can be used for continious monitoring. Notice however that XAI methods can also be used to analyze images, such that it is possible to obtain a representation regarding which parts of the images the model used to learn. Thus, we also evaluate LIME to handle resource intensive workload (Experiment 2). Figure 8(d) shows the results of experiment 2. From the figure, it is possible to observe that LIME methods require considerable amount for computation. As a result, when facing resource intensive processing, XAI are not able to handle concurrent workload below $1s$. In fact, we can observe a steady increase in response time that depends on the number of concurrent users accessing the service. This has direct implications in the types of models/datasets that can be analyzed with available XAI methods.

VIII. Challenges outlook and experiences

While all regulatory frameworks agree on the strategic importance of AI trustworthiness, the development of trustworthy AI is an on-going process. While principles, tools, guidelines and methods are available to aid in this matter, there is still a gap between regulations and technical requirements. Thus, there are several challenges that remain open for augmenting modern applications with AI trustworthiness capabilities. Based on our experiences, we next highlight technical challenges that require further attention for complying robustly with the trustworthy AI requirements.

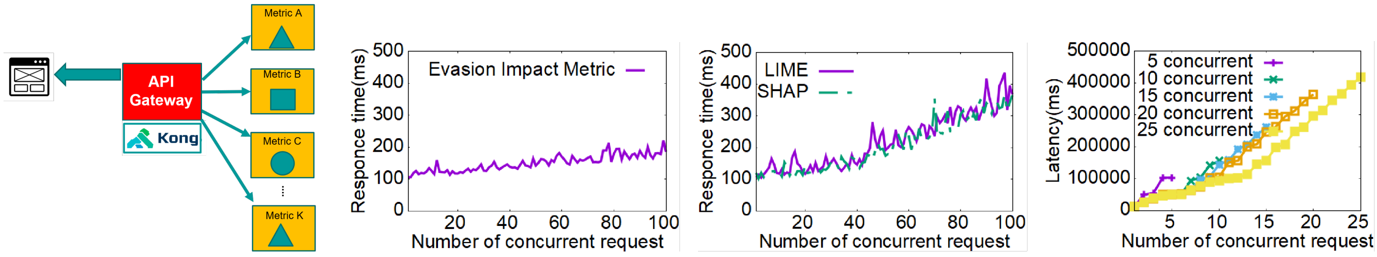
AI trust score and AI sensors: AI trustworthiness involves the characterization of several properties [10], including technical (e.g., validity, accuracy, reliability, robustness, resilience, or security) as well as the socio-technical characteristics (explainability, interpretability, managing bias, privacy enhanced, safety). Each property can be obtained through specialized metrics, based on the nature of the area of application at hand. For instance, in a loan application, fairness can be applied to identify data biases in individual or specific groups (equitable), whereas fairness can be also calculated to estimate whether the decision process was fair to all the involved loaners (procedural). Similarly, in a object detection application, explainability can be generated using occlusion sensitivity to identify the most relevant area on an image contributing with the object detection. In turn, LIME divides the image into multiple section areas and ranks each accordingly to measure their contribution to the overall model prediction. Encapsulating all different properties into AI sensors is a key challenge to foster the easy integration of trustworthiness in current software development practices. AI sensors can provide general procedures and guidelines to instrument applications with trustworthy mechanisms. Another important challenge is to produce a coherent and comparable trust score from measurements obtained by AI sensors, such that trustworthiness can be understood as an overall feature of applications. While the development of a trust score has been explored by previous work [64], these solutions simplify the extraction of trustworthiness by considering all properties homogeneous and not considering its different inherent characteristics.

Human oversight and AI tuning: As part of the EU AI Act, humans play a critical role in overseeing the behavior of AI. AI dashboards can provide critical information about the AI inference capabilities to stakeholders. For example, level of fairness, robustness and resilience to mention some. Through the dashboard inspection, individuals relying on AI models can be aware about the limitations and scope of the decision



(a) SHAP XAI for benign data (b) SHAP XAI for evasion attack data (c) Impact vs Poison percent (d) Complexity vs Poison percent

Fig. 7: Use case 2 results (Network activity monitoring); SHAP analysis for evasion attacks; a) Benign (NN) model, b) Attacked (NN) model; Poisoning attacks quantified by Impact and complexity metrics; c) Impact vs Poison%, d) Complexity vs Poison%.



(a) System deployment (b) (c) (d)

Fig. 8: Capacity-load experiments, a) System deployment; b) Load in impact metric; c) Load in LIME and SHAP; and c) Load in LIME when handling requests requiring heavy computations.

support provided by AI models. Ultimately, dashboards can support humans to decide whether or not to use AI for aiding with a particular task. Moreover, as the trustworthy properties are considered trade-offs that can be adjusted depending on the requirements of different stakeholders using the applications, it becomes then critical to tune these properties over time. Existing methods can be used to perform hyperparametrization on the way an AI algorithm learns and thus adjusting its resulting decision process [15]. As the tuning of models is an iterative process that involves a reinforced human-in-the-loop feedback rather than a single shot, a key challenge is to integrate such process in the construction of AI models. To obtain significant feedback from stakeholders, it is important that explanations describing the overall trustworthiness of a model are tied to specific domain terminology of stakeholders, e.g., tailored explanations for end users and software developers. An extra layer of transformation is thus required to map understandable insights of a model to a specific target audience. A potential solution is to rely on large language models (ChatGPT-like preamble) or a meta-model that change dynamically the explanations to a specific domain audience. Besides this, another key challenge is to determine what changes can be applied on the model by individuals. For instance, removing personal data from the training dataset or changing the machine learning algorithm. This is a critical challenge to overcome as AI models have to support individual needs of users, while preserving general values from groups and society. Otherwise, conflicts on AI usage may arise, halting everyday activities and human processes. Another remaining challenge is to develop AI dashboards that motivate users to be involved in the AI tuning process [65].

Adversarial threats over AI algorithms and data: As demonstrated in our experiments, the decision process of AI models can be changed abruptly. Induced changes (aka attacks) are of particular interest as proactive counter measurements have to be taken rapidly by human operators, otherwise, compromised applications can become source of harm for citizens and urban infrastructure, e.g., attacks on drone delivery [22]. Other examples of this include adversarial generative patches that confuse AI models and poisoned data that can make devices drain energy at faster rates, e.g., sponge attacks in IoT devices. As there is a large plethora of attacks that can hamper AI functionality, a key challenge is to quantify the level of the AI resilience to attacks by applying multiple detection methods and suggesting those counter measurements to human operators. Naturally, the level of resilience depends on the available methods that attest whether model/data has been compromised. Besides this, while some post defacto verification methods could be applied to detect attacks over AI functionality, other methods require re-playing the overall training process, involving a more time consuming analysis.

Privacy-preserving data and computations: Data is a key element in the machine and deep learning pipelines, building AI models. Regulatory guidelines in the use of data, e.g., EU GDPR, forbid the inclusion of private and sensitive data that can be used to identify specific individuals. Thus, data is required to be obfuscated before it can be used within the AI pipelines. Existing solutions to aid in this matter include differential privacy and data anonymity techniques [66]. However, data removal degrades the decision making process performance, requiring new methods to obfuscate sensitive information without reducing model performance levels, e.g., sparse coding and compressive sensing compensation models. At the same time, since direct access to model and data are required to estimate different trustworthy properties, a key challenge is to guarantee that the analysis of these properties is conducted in a secure manner to avoid potential induced attacks over AI. Existing methods based on multi-party computation, homomorphic encryption and TEEs (Trusted Execution Environments) could be adopted in this matter. Integrating these mechanisms within the architectures, however, require managing extra computation overhead in the analysis as well as to solve several technological limitations to achieve scalable solutions. For instance, while TEEs are currently available to aid in secure computation, they have several limitations regarding the specific characteristics in software runtime execution, e.g., programming language, dependencies, and storage to mention the most common.

IX. Implications

Legal vs technical trustworthiness: Our work presents the design and development experiences from augmenting modern applications with capabilities to gauge and monitor AI trustworthiness. The selected metrics of our prototype are considered from a technical point of view based on the most common methods currently adopted to analyze AI black-box characteristics. We are interested on replacing our metrics with others that align better with regulatory trustworthiness. This however requires to conduct a legal analysis that considers all metrics available in the state-of-the-art to identify the most suitable. This analysis out of the scope of this work.

Cost and complexity: SPATIAL not just augments modern applications with new regulatory functionality, but it also augments the amount of components and enlarges the underlying deployment of the overall system running the applications. This increases the complexity of developing and maintaining the applications. Moreover, the cost of the deployment also increases as it is not possible to piggyback already existing infrastructure due to increased load required for computation. Indeed, as shown in our experiments, methods such as XAI can induce heavy load in the overall system, requiring instead to be deployed in their own dedicated machine.

Verifying vs Embedding trustworthiness: Current practices to analyze trustworthiness of AI inference capabilities rely on post-defacto verification of models. The use of AI sensors can be foster the embedding of mechanisms to gauge and monitor AI trustworthy properties from early development and design phases. This however requires /standard procedures on how to create AI sensors (like APIs) that encapsulate each trustworthiness property. Moreover, guidelines and best practices on how to instrument modern applications with AI sensors are also required to facilitate their adoption in software development practices.

Adaptive trustworthiness: In our work, we present the encapsulation of trustworthy properties into AI sensors. More advanced AI sensors are envisioned to provide adaptive trustworthiness. As these properties can be considered trade-offs, it is possible to establish interactions and negotiations between AI sensors to obtain a balance level of trust (similar to Chatbot negotiations). Achieving this level of automation however requires to develop further autonomy in AI sensors.

X. Summary and Conclusions

In this paper, we presented the SPATIAL architecture, a proof-of-concept system that augments modern applications with capabilities to analyze trustworthy aspects of AI models. SPATIAL diagnoses AI functionality by combining different methods that characterize and quantify the inference process of AI. Through rigorous benchmarks and analyses that consider two real-world industrial applications, our results suggests that SPATIAL can provide relevant insights about AI models, but this analysis is time-consuming and very resource intensive, making it unsuitable for critical applications. We also highlight a roadmap of requirements and challenges that need to be overcome, such that current issues that were found can be addressed. Our work paves the way towards augmenting modern applications with trustworthy AI mechanisms.

Acknowledgment

This research is part of SPATIAL project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No.101021808. We also thank SPATIAL researchers and our collaborators that participated in general discussions around the topic, Cornelis van de Kamp, Souneil Park, Anouk Mols, Danitsja van Heusden-van Winden, Jose Gonzalez, Siike Lampson, Jennifer Albright, Martijn Janssen, Huu-Nghia Nguyen, Wissam Mallouli, Jorge Campos, Roel Dobbe, Iris van der Wel, Giulia Pastor, Christopher Vervoort, Laura Bruun, Tim Orchard, Faye Carr, Sabastian Mateiescu, Ville Valtonen, Selma Toktas. We also want to thank our steering committee for providing valuable critics that contributed in tuning our work, Jon Crowcroft, Stefan Schmid, Michael Zimmer and Eve Schooler. We also thank the EU Project Officer and the reviewers.

References

[1] Statista, *Artificial intelligence (AI) market size worldwide in 2021 with a forecast until 2030*, Accessed Dec 31, 2023. [Online]. Available: https://www.statista.com/study/38609/artificial-intelligence-ai-statista-dossier/

[2] T. M. Brill, L. Munoz, and R. J. Miller, "Siri, alexa, and other digital assistants: a study of customer satisfaction with artificial intelligence applications," in *The Role of Smart Technologies in Decision Making*. Routledge, 2022, pp. 35–70.

[3] L. Qiu and I. Benbasat, "Evaluating anthropomorphic product recommendation agents: A social relationship perspective to designing information systems," *Journal of management information systems*, vol. 25, no. 4, pp. 145–182, 2009.

[4] E. Frachtenberg, "Practical drone delivery," *Computer*, vol. 52, no. 12, pp. 53–57, 2019.

[5] H. Flores, "Opportunistic multi-drone networks: Filling the spatiotemporal holes of collaborative and distributed applications," *IEEE Internet of Things Magazine*, vol. 7, no. 2, pp. 94–100, 2024.

[6] A. L. Fogel and J. C. Kvedar, "Artificial intelligence powers digital medicine," *NPJ digital medicine*, vol. 1, no. 1, p. 5, 2018.

[7] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)," *IEEE access*, vol. 6, pp. 52 138–52 160, 2018.

[8] F. of life, *Pause Giant AI Experiments: An Open Letter*, Accessed Dec 31, 2023. [Online]. Available: https://futureoflife.org/open-letter/pause-giant-ai-experiments/

[9] A. Asatiani, P. Malo, P. R. Nagbøl, E. Penttinen, T. Rinta-Kahila, and A. Salovaara, "Challenges of explaining the behavior of black-box ai systems," *MIS Quarterly Executive*, vol. 19, no. 4, pp. 259–278, 2020.

[10] B. Li, P. Qi, B. Liu, S. Di, J. Liu, J. Pei, J. Yi, and B. Zhou, "Trustworthy ai: From principles to practices," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–46, 2023.

[11] K. Gade, S. C. Geyik, K. Kenthapadi, V. Mithal, and A. Taly, "Explainable ai in industry," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 3203–3204.

[12] H. Flores, "Ai sensors and dashboards," *IEEE Computer Magazine*, 2024.

[13] A. D'Amour, H. Srinivasan, J. Atwood, P. Baljekar, D. Sculley, and Y. Halpern, "Fairness is not static: deeper understanding of long term fairness via simulation studies," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 525–534.

[14] S. K. Lo, Y. Liu, Q. Lu, C. Wang, X. Xu, H.-Y. Paik, and L. Zhu, "Toward trustworthy ai: Blockchain-based architecture design for accountability and fairness of federated learning systems," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3276–3284, 2022.

[15] J. M. Wing, "Trustworthy ai," *Communications of the ACM*, vol. 64, no. 10, pp. 64–71, 2021.

[16] M. D. Scientists. (2023) Shapash. [Online]. Available: ,https://github.com/MAIF/shapash

[17] V. Arya, R. K. Bellamy, P.-Y. Chen, A. Dhurandhar, M. Hind, S. C. Hoffman, S. Houde, Q. V. Liao, R. Luss, A. Mojsilović *et al.*, "Ai explainability 360 toolkit," in *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)*, 2021, pp. 376–379.

[18] D. Kaur, S. Uslu, K. J. Rittichier, and A. Durresi, "Trustworthy artificial intelligence: a review," *ACM Computing Surveys (CSUR)*, vol. 55, no. 2, pp. 1–38, 2022.

[19] K. Kyriakou and J. Otterbacher, "In humans, we trust: Multidisciplinary perspectives on the requirements for human oversight in algorithmic processes," *Discover Artificial Intelligence*, vol. 3, no. 1, p. 44, 2023.

[20] R. Koulu, "Proceduralizing control and discretion: Human oversight in artificial intelligence policy," *Maastricht Journal of European and Comparative Law*, vol. 27, no. 6, pp. 720–735, 2020.

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

[22] I. Shumailov, Y. Zhao, D. Bates, N. Papernot, R. Mullins, and R. Anderson, "Sponge examples: Energy-latency attacks on neural networks," in *2021 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2021, pp. 212–231.

[23] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 253–261.

[24] C. Zhu, W. R. Huang, H. Li, G. Taylor, C. Studer, and T. Goldstein, "Transferable clean-label poisoning attacks on deep neural nets," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7614–7623.

[25] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," *Advances in neural information processing systems*, vol. 31, 2018.

[26] Y. Liu, X. Ma, J. Bailey, and F. Lu, "Reflection backdoor: A natural backdoor attack on deep neural networks," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*. Springer, 2020, pp. 182–199.

[27] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. Ieee, 2017, pp. 39–57.

[28] S. Weerasinghe, T. Alpcan, S. M. Erfani, and C. Leckie, "Defending support vector machines against data poisoning attacks," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2566–2578, 2021.

[29] B. Biggio, I. Corona, G. Fumera, G. Giacinto, and F. Roli, "Bagging classifiers for fighting poisoning attacks in adversarial classification tasks," in *Multiple Classifier Systems: 10th International Workshop, MCS 2011, Naples, Italy, June 15-17, 2011. Proceedings 10*. Springer, 2011, pp. 350–359.

[30] A. I. Newaz, N. I. Haque, A. K. Sikder, M. A. Rahman, and A. S. Uluagac, "Adversarial attacks to machine learning-based smart healthcare systems," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[31] E. Alhajjar, P. Maxwell, and N. Bastian, "Adversarial machine learning in network intrusion detection systems," *Expert Systems with Applications*, vol. 186, p. 115782, 2021.

[32] C. Dunn, N. Moustafa, and B. Turnbull, "Robustness evaluations of sustainable machine learning models against data poisoning attacks in the internet of things," *Sustainability*, vol. 12, no. 16, p. 6434, 2020.

[33] E. Alsuwat, H. Alsuwat, J. Rose, M. Valtorta, and C. Farkas, "Detecting adversarial attacks in the context of bayesian networks," in *Data and Applications Security and Privacy XXXIII: 33rd Annual IFIP WG 11.3 Conference, DBSec 2019, Charleston, SC, USA, July 15–17, 2019, Proceedings 33*. Springer, 2019, pp. 3–22.

[34] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.

[35] J. Chen, M. I. Jordan, and M. J. Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack," in *2020 ieee symposium on security and privacy (sp)*. IEEE, 2020, pp. 1277–1294.

[36] A. Kantchelian, J. D. Tygar, and A. Joseph, "Evasion and hardening of tree ensemble classifiers," in *International conference on machine learning*. PMLR, 2016, pp. 2387–2396.

[37] X. He, J. Jia, M. Backes, N. Z. Gong, and Y. Zhang, "Stealing links from graph neural networks," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2669–2686.

[38] R. N. Reith, T. Schneider, and O. Tkachenko, "Efficiently stealing your machine learning models," in *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society*, 2019, pp. 198–210.

[39] M. R. Clark, P. Swartz, A. Alten, and R. M. Salih, "Toward black-box image extraction attacks on rbf svm classification model," in *2020 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2020, pp. 394–399.

[40] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction {APIs}," in *25th USENIX security symposium (USENIX Security 16)*, 2016, pp. 601–618.

[41] G. M. R. de Arcaute, J. A. Hernández, and P. Reviriego, "Assessing the impact of membership inference attacks on classical machine learning algorithms," in *2022 18th International Conference on the Design of Reliable Communication Networks (DRCN)*. IEEE, 2022, pp. 1–4.

[42] X. Luo, Y. Wu, X. Xiao, and B. C. Ooi, "Feature inference attack on model predictions in vertical federated learning," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2021, pp. 181–192.

[43] G. Liu, S. Wang, B. Wan, Z. Wang, and C. Wang, "Ml-stealer: Stealing prediction functionality of machine learning models with mere black-box access," in *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2021, pp. 532–539.

[44] R. Shokri *et al.*, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[45] D. Gopinath, H. Converse, C. Pasareanu, and A. Taly, "Property inference for deep neural networks," in *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2019, pp. 797–809.

[46] K. J. Reza, M. Z. Islam, and V. Estivill-Castro, “Privacy protection of online social network users, against attribute inference attacks, through the use of a set of exhaustive rules,” *Neural Computing and Applications*, vol. 33, no. 19, pp. 12 397–12 427, 2021.

[47] Y. Alufaisan, M. Kantarcioglu, and Y. Zhou, “Robust transparency against model inversion attacks,” *IEEE transactions on dependable and secure computing*, vol. 18, no. 5, pp. 2061–2073, 2020.

[48] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei, “Demystifying membership inference attacks in machine learning as a service,” *IEEE Transactions on Services Computing*, vol. 14, no. 6, pp. 2073–2089, 2019.

[49] E. Alsuwat, H. Alsuwat, M. Valtorta, and C. Farkas, “Adversarial data poisoning attacks against the pc learning algorithm,” *International Journal of General Systems*, vol. 49, no. 1, pp. 3–31, 2020.

[50] G. Liu, C. Wang, K. Peng, H. Huang, Y. Li, and W. Cheng, “Socinf: Membership inference attacks on social media health data with machine learning,” *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 907–921, 2019.

[51] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, “Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models,” in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 15–26.

[52] M. James, M. Mruthula, V. Bhaskaran, S. Asha *et al.*, “Evasion attacks on svm classifier,” in *2019 9th International Conference on Advances in Computing and Communication (ICACC)*. IEEE, 2019, pp. 125–129.

[53] H. Chen, J. Su, L. Qiao, and Q. Xin, “Malware collusion attack against svm: Issues and countermeasures,” *Applied Sciences*, vol. 8, no. 10, p. 1718, 2018.

[54] H. Muccini and K. Vaidhyanathan, “Software architecture for ml-based systems: what exists and what lies ahead,” in *2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN)*. IEEE, 2021, pp. 121–128.

[55] W. Fan, “Data quality: From theory to practice,” *Acm Sigmod Record*, vol. 44, no. 3, pp. 7–18, 2015.

[56] Z. Jia, S. Lin, C. R. Qi, and A. Aiken, “Exploring hidden dimensions in parallelizing convolutional neural networks.” in *ICML*, 2018, pp. 2279–2288.

[57] A. Vehtari, A. Gelman, and J. Gabry, “Practical bayesian model evaluation using leave-one-out cross-validation and waic,” *Statistics and computing*, vol. 27, pp. 1413–1432, 2017.

[58] Y. Wang, “Balancing trustworthiness and efficiency in artificial intelligence systems: An analysis of tradeoffs and strategies,” *IEEE Internet Computing*, 2023.

[59] M. Anisetti, C. A. Ardagna, N. Bena, and E. Damiani, “Rethinking certification for trustworthy machine learning-based applications,” *IEEE Internet Computing*, 2023.

[60] University of Tartu, “Ut rocket,” 2018.

[61] D. Micucci, M. Mobilio, and P. Napoletano, “Unimib shar: A dataset for human activity recognition using acceleration data from smartphones,” *Applied Sciences*, vol. 7, no. 10, 2017. [Online]. Available: http://www.mdpi.com/2076-3417/7/10/1101

[62] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, “Modeling tabular data using conditional gan,” *Advances in neural information processing systems*, vol. 32, 2019.

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

[64] A. H. Celdran, J. Kreischer, M. Demirci, J. Leupp, P. M. Sanchez, M. F. Franco, G. Bovet, G. M. Perez, and B. Stiller, “A framework quantifying trustworthiness of supervised machine and deep learning models,” in *SafeAI2023: The AAAI’s Workshop on Artificial Intelligence Safety*, 2023, pp. 2938–2948.

[65] S. Park, C. Gebhardt, R. Rädle, A. M. Feit, H. Vrzakova, N. R. Dayama, H.-S. Yeo, C. N. Klokmose, A. Quigley, A. Oulasvirta *et al.*, “Adam: Adapting multi-user interfaces for collaborative environments in real-time,” in *Proceedings of the 2018 CHI conference on human factors in computing systems*, 2018, pp. 1–14.

[66] V. Gudivada, A. Apon, and J. Ding, “Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations,” *International Journal on Advances in Software*, vol. 10, no. 1, pp. 1–20, 2017.