ASCERTAINING TRUSTWORTHINESS OF AI SYSTEMS IN TELECOMMUNICATIONS

Rishika Sen, Shrihari Vasudevan, Ricardo Britto, MJ Prasath
Ericsson

NOTE: Corresponding author: Rishika Sen, rishika.sen@ericsson.com; Shrihari Vasudevan, shrihari.vasudevan@ericsson.com

Abstract – *With the rapid uptake of Artificial Intelligence (AI) in the Telecommunications (Telco) industry and the pivotal role AI is expected to play in future generation technologies (e.g., 5G, 5G Advanced and 6G), establishing the trustworthiness of AI used in Telco becomes critical. Trustworthy Artificial Intelligence (TWAI) guidelines need to be implemented to establish trust in AI-powered products and services by being compliant to these guidelines. This paper focuses on measuring compliance to such guidelines. This paper proposes a Large Language Model (LLM)-driven approach to measure TWAI compliance of multiple public AI code repositories using off-the-shelf LLMs. This paper proposes an LLM-based scanner for automated measurement of the trustworthiness of any AI system. The proposed solution measures and reports the level of compliance of an AI system. Results of the experiments demonstrate the feasibility of the proposed approached for the automated measurement of trustworthiness of AI systems.*

Keywords – AI-driven telecommunications, large language models, trustworthy artificial intelligence

1. INTRODUCTION

With the extensive consumption of AI techniques, it is crucial to establish regulations to ensure safe and responsible use of AI. While AI applications have largely been very successful across every domain, there are instances where AI systems have an unpredictable and/or harmful outcomes. One such example is the issue with Uber's self-driving car. In 2018, it killed a cyclist in Arizona [22].

In the telecommunications industry, AI systems must provide reliable and accurate products and offer recommendations to customers. If they do not, it could lead to a poor customer experience and potentially drive customers away from the Communications Service Provider (CSP). An autonomous network that allocates resources to specific regions must maintain transparency about the criteria used for allocation. This helps prevent bias and ensures that business needs do not override the ethical needs of society, thereby avoiding any form of discrimination. In an autonomous network scenario, when a customer complains about poor signal coverage, the system tries to optimize signal strength, this can lead to increased power usage and a higher carbon footprint. Additionally, it may cause the signal strength to exceed the permitted levels in densely populated areas. Therefore, to be able to rely on AI systems in the future, there is a need for trustworthy AI.

The term "trustworthy" signifies the ability to be relied on as honest or truthful. In Artificial Intelligence (AI), the term "trustworthy" signifies explainable, transparent, fair, unbiased and accountable. Building an AI system based on diverse data gives a more generalized result when performing in an unknown environment. Delivering explainable outcomes from the system helps the stakeholders understand the rationale of the generated result. Building a robust and safe system ensures security from any external threat. Such a robust system is expected to perform efficiently, irrespective of the environment in which it is executed.

AI is extensively used in the Telco industry. It is used in network optimization, predictive maintenance, network security, improving quality of service and 5G network management [4]. Integrating AI in Telco contributes to more efficient operations, improved customer satisfaction, and enhanced overall network performance. Given the nature of Telco as an essential service and enabler, and the critical role AI is expected to play in future generation technologies, the AI solutions that are replacing and optimizing existing solutions must be trustworthy. The AI supply chain largely comprises third-party (3PP) public code. Therefore, the analysis of trustworthiness becomes even more important.

Often, service providers have to use AI-enabled Telco solutions from multiple vendors. Such vendors, when collaborating, will likely share data artefacts or possibly have to use each other's (closed-source) models/systems. Artefacts or systems, certified as being trustworthy, would significantly enable cooperation between multiple vendors towards delivering services to consumers. Measuring trustworthiness of AI in the Telco industry is therefore of utmost importance.

To develop trustworthy systems, policies around the safe usage of AI are being drafted in various countries. At the forefront of it all is the EU Commission, which has proposed trustworthy AI design rules [2, 11] and guidelines, primarily focused on seven principles, shown in Fig. 1. These design rules and guidelines, if adhered to, will result in designing trustworthy AI systems. The various principles encompassing these guidelines are as follows:

- Human agency and oversight: AI systems are expected to help humans make informed decisions

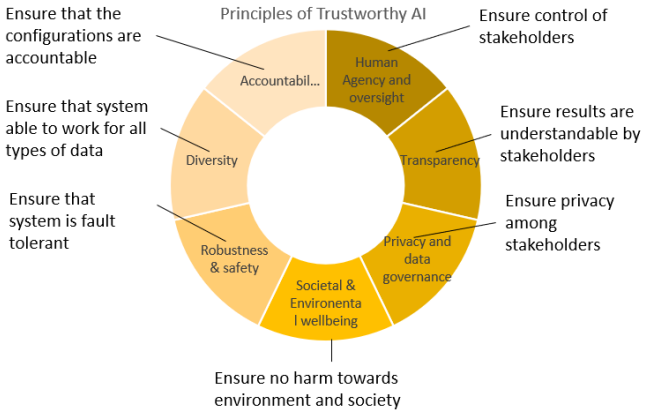

Figure 1 – An overview of the trustworthy AI Principles

and protect their fundamental rights by including oversight of AI system through human-in-the-loop, human-on-the-loop, and human-in-command approaches.

- Technical robustness and safety: ensure the resilience and security of AI systems with safety measures, including backup plans, maintain accuracy, reliability and reproducibility to minimize and prevent unintentional harm.
- Privacy and data governance: ensuring robust data governance involves prioritizing privacy, data protection, implementing mechanisms that address data quality, integrity and authorized access.
- Transparency: explaining AI systems and decisions in such a way that stakeholders can comprehend; ensure awareness of interactions, capabilities and limitations.
- Diversity, non-discrimination and fairness: preventing unfair bias in AI to avoid negative consequences such as discrimination, promoting diversity, accessibility, and ensure stakeholder involvement throughout the lifecycle of AI systems.
- Societal and environmental wellbeing: AI systems should ensure sustainability, environmental friendliness, minimize negative impact on the environment, other living beings and society.
- Accountability: establishing responsibility and accountability for AI systems by implementing mechanisms like audibility to assess algorithms, data and design processes, ensuring accessible redressal mechanisms for any issues that may arise.

The EU Commission expects these principles to be followed when designing, developing and deploying AI systems. Following the EU's guidelines, various countries have developed their own set of regulations for developing AI products [21]. The current EU-TWAI guidelines have been used as the basis for our work, given its level of maturity. Ericsson has published its findings on trustworthy AI for the Telco domain. Ericsson's guidelines incorporate additional Telco-specific guidelines over the EU guidelines. This paper aims to automate the measurement of trustworthiness, as defined in [12].

Currently, verifying trustworthiness of an AI system for the given guidelines and design rules is a manual process. There have been attempts to manually assess the trustworthiness of an AI system. An assessment list called ALTAI has also been published for manual validation of EU's TWAI guidelines [8]. The "Z-Inspection: A Process to Assess Trustworthy AI" [13] is another representative exemplar. The primary concern with checking the compliance of an AI system manually is that it is time-consuming and error-prone. The automation of this activity is currently an open research problem. Automated validation of trustworthiness of an AI system will be required and in some cases mandatory to realize autonomous networks, where the network is able to self-monitor, self-diagnose, self-optimize, self-heal and self-protect. In such scenarios, the action to achieve autonomy is dependent on the trustworthy output of the AI system to proceed further.
This paper proposes the automated validation of TWAI principles for AI systems by presenting an LLM-based TWAI compliance scanner. In this paper, trustworthy measurement is formulated in terms of summary generation and question-answering. The paper does not address the trustworthiness of LLM, which is a separate research topic in itself. The LLMs used in the study, have been tested for their efficacy in summary generation and question answering tasks across various benchmark datasets to conclude their robustness in the said tasks. Further, experiments in the paper test their performance on multiple datasets to make sure that they are useful for our proposed solution. The proposed solution attempts to do away with the manual process and automate using state-of-the-art solutions.

2. PROPOSED METHODOLOGY

In this section, we propose an approach to automate TWAI conformance checking. Some characteristics pertaining to the same are as follows:

- The proposed methodology can consider all relevant artefacts of AI systems by transforming it into an intermediate text representation using AI methods and subsequently using this intermediate representation to answer questions that check for compliance with TWAI principles, also done using AI methods.
- Techniques required to validate fulfilment of the requirements of TWAI principles from the code, may include summary generation [17], [18], question-answering [3] and keyword search.
- The problem of metadata summarization has not been significantly addressed as part of this research

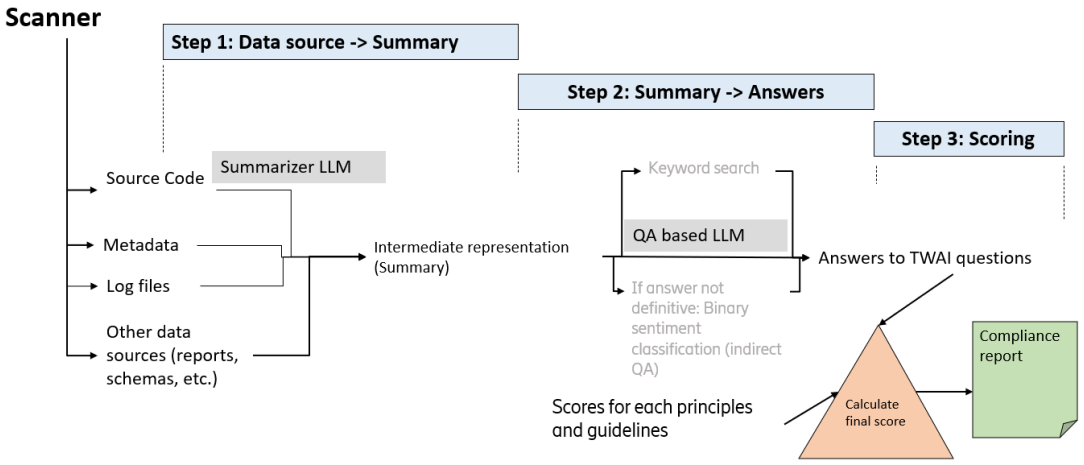

Figure 2 – Template for trustworthy AI Scanner

because, based on empirical findings of current experiments, it was deemed that metadata paraphrasing was unnecessary and possibly detrimental to the objective of the paper, due to its sparsity in the current datasets used.

- Logs are another example of an artefact that could be processed using summarization methods. Logs are generally unstructured, generated in large amounts, and distributed across multiple files.

The AI methods used in this paper, rely on Generative AI techniques and Large Language Models (LLMs). Generative AI refers to a set of AI models that are designed to generate new and original data [1]. These models can create various types of data, such as text, images, music and videos, by learning patterns from existing datasets. Large Language Models (LLMs) are a class of artificial intelligence models that are capable of understanding and generating human-like text at a scale previously unseen. LLMs can be used to generate summaries from various artefacts and in context-based question answering [23], both of which are used in this paper. Therefore, this paper uses LLMs to facilitate designing the proposed solution.

3. IMPLEMENTATION

The challenge of the problem at hand is to scan through all the artefacts to ensure that the AI system complies with the TWAI guidelines and finally develop a quantification of trustworthiness. To comply with the guidelines of the seven principles from the EU proposal, Ericsson has derived the EU guidelines into finer implementable functional requirements that are to be considered as a part of product development to achieve trustworthiness in AI applications. Usually, the guidelines are given as a summary of expectations. To clearly understand the action-items concerning each guideline, decomposing the guidelines into a set of requirements [1] was done. For example, one of the guidelines under transparency is, in case a product includes a model, product documentation shall include applicable accuracy metrics and valid ranges. Product documentation shall describe as well how the training data were/are collected, used and maintained.
The requirements extracted from this guideline were rephrased to form questions. The questions are an atomic reflection of TWAI guidelines and prepared to enable an effective compliance measurement. An example of such questions that were curated from the requirements are:

- Does the context[2] contain information on performance metrics?
- Does the context contain information on plotting the performance metrics?
- Does the context contain information on validation of training data?
- Does the context contain information on the plot of the distribution of training data?
- Does the context contain information on the storing and usage of training data?

Affirmative answers (positive response) to these questions indicate that the corresponding guideline has been adhered to. A total of 79 requirements, expressed as questions, cover all the guidelines of the seven principles. Satisfying these requirements manually is tedious and risks being inconsistent or erroneous. With more guidelines in the future, more questions will need to be validated.
Each of these guidelines is subject to a weight specified by a domain expert. This is done to convey the importance of the guidelines for an AI system whose compliance is being evaluated. Depending on the criticality of different principles for a certain AI system, the weights of the guideline will vary accordingly. Currently, the weights for a certain

[1] The requirements were manually vetted by multiple domain experts, who have the expertise of handling and executing telecom-based AI projects, as part of prior internal study.

[2] Here, the context refers to the intermediate representation of various data sources.

Table 1 – Summary of LLMs used for development of the scanner.

LLM	Training Info	Training Objective
Mini-Orca [26]	Used DeepSpeed [27] with fully shared data parallelism, also know as ZeRO stage 3 by writing their own fine-tuning scripts plus leveraging some of the model training code provided by OpenAlpaca repo	text generation, inference
Falcon 7b [15]	Falcon-7B was trained on 7 billion parameters, a high-quality filtered and deduplicated web dataset which was enhanced with curated corpora	text generation, inference
Llama 7b [14]	Llama 2 was pre-trained on 2 trillion tokens of data from publicly available sources. The fine-tuning data includes publicly available instruction datasets and over one million new human-annotated examples.	text generation, inference
MPT-Chat [25]	The model was trained with shared data parallelism and used the AdamW optimizer	text generation, inference

AI system were given manually based on the description and the objective of the system.

The weights fall within the range [0,1], 0 being unimportant and 1 being important, and the numbers in between capture the relative importance. The sum of these weights for each of the guidelines does not add up to 1, rather the weights are comparable to one another; the higher the weight, the higher the importance of complying with the respective guideline. According to the domain experts, some principles can be more important than all the other principles. Even among principles, some guidelines may be more crucial than the others.

The scanner (as depicted in Fig. 2) consists of three stages. The first stage converts various data sources, which includes the code associated with the AI system, metadata that describes various characteristics of the AI system and logs which are generated during model training and inference, to an intermediate representation (referred to as context in queries), termed as summary. Summarization, in the context of this paper, does not always refer to the task of paraphrasing; the summary is an intermediate representation of the artefact to enable the validation of the requirements in the form of question-answering that follows. The proposed approach uses question-answering to validate TW requirements. To be able to perform question-answering with LLMs using multiple data sources and cross-correlate information in making inferences, a unified intermediate representation is required. Summaries provide a query-able intermediate representation of code, metadata and logs. For example, consider the following Python code snippet.

```
for i in range(10):
  print('Hello world')
```

The summary generated from LLM (Llama 7b) with respect to this code is as follows:

```
The provided Python code is a for loop that
iterates ten times, from 0 to 9, using the
range(10) function. During each iteration,
it prints the string "Hello world" to the
console. This results in "Hello world"
being printed ten times in total.
```

The summary for code, metadata and logs are similar as shown above. In the current paper, two data sources (code and metadata) have been considered. The approach can be adapted to cater to other artefacts related to the AI/ML system. In future, data sources may also include reports generated from tools that enable software product development. For example, database schema files, input files and output files, depicted as "Other data sources" in Fig. 2 are some data sources that can be considered as well. The proposed approach is generic and can be extended to incorporate other data sources in the same way as presented in this paper. But the experiments in this paper limit evaluation to two data sources: code and metadata. The second stage uses the generated summaries to validate the TWAI requirements, expressed as a sequence of questions. The third stage consists of calculating the percentage compliance of the AI system with the trustworthy AI guidelines.

Multiple solutions for validating TWAI requirements have been considered. One solution is the application of a question-answering-based LLM on the generated summary using these questions. Alternate solutions to validating TWAI requirements include keyword search and binary "sentiment" classification. However, only keyword search will not suffice as the answers to many questions require semantic interpretation of natural language text (summaries generated before). Keyword search does not capture semantic similarity (for example, keywords "accuracy", "precision" will fall under the umbrella term "performance metric"), which an LLM will map to be semantically similar. A simple keyword search for one of these would miss the occurrence of the other. If new performance metrics are discovered in the future, the list of synonyms for each keyword will have to be individually checked. Furthermore, the sentiment of a sentence with respect to a keyword is also important. If a keyword is being used in a negative sense in a sentence, then the keyword detection technique will detect the keyword, but will fail to identify the sentiment with which the keyword was used. For example, in the sentence "the cat is not

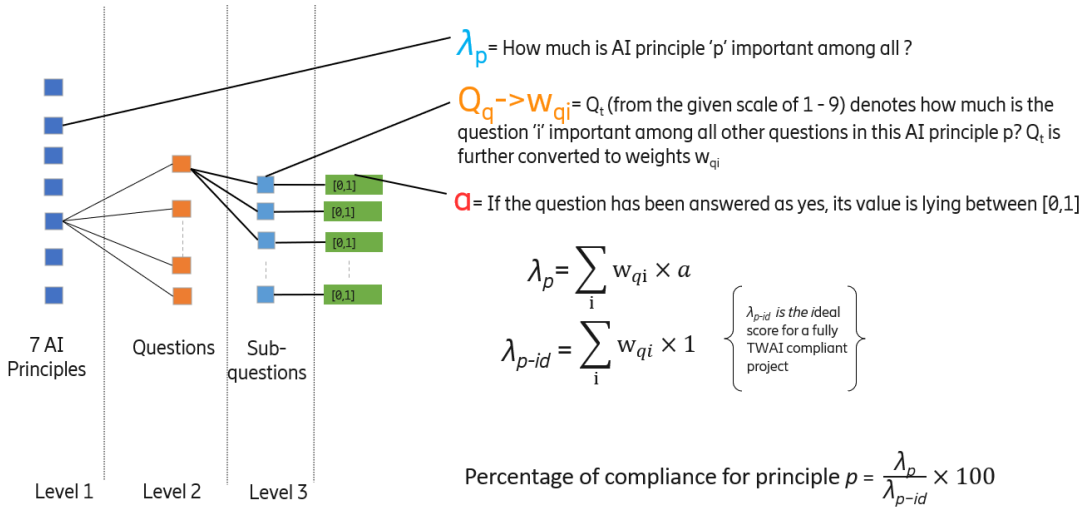

Figure 3 – Scoring mechanism for trustworthy AI Scanner

behind the tree", the absence of the object identified by the keyword "cat" is being represented. However, with keyword search for the question "Is the cat behind the tree?", the keyword will be detected, but the sentiment of the sentence will not be captured. Another alternative to keyword search is binary sentiment classification. This method will be applicable when the answers to the questions are not definitive. For the experiments in this paper, all the questions have a definitive answer.

The questions extracted from each of the guidelines, have a binary answer, "Yes/No". However, answers to the questions may be probabilities or weights representing a degree of compliance depending on the guideline. Each guideline is considered to be completely compliant if all the questions related to it were answered as "Yes". The trustworthiness is the weighted sum of these compliance responses. In the current paper, it is a linear weighted sum, where the weights were provided by domain experts. The next three subsections are dedicated towards explaining the steps of the solution in detail.

3.1 Step 1 - Conversion from source to intermediate representation

In this step, any source of data, such as code, metadata or log files, were converted to an intermediate representation. The intermediate representation may take several forms, for example, descriptions of code snippets, unaltered data, description of logs, depending on the data source:

- Summary from code: AI/ML projects, which are pushed into production, consist of large amounts of code. Generating the summary of the whole code in one go would lead to loss of information from a trustworthiness evaluation standpoint. Therefore, a summary of the entire code is generated, snippet by snippet, using LLMs which are suitable for the task of summarization. Significant experimental evaluation, of several LLMs, described in Section 4.1 and Table 4, and best LLM was chosen in Section 4. Deciding the size of each code snippet is a challenging problem since different code lengths would result in different summary generations. For example, considering the following Python code snippet

```
x=[]
for i in range(10):
  x.append(i)
```

the summary generated from LLM with respect to this code is as follows:

```
The code initializes an empty list x
and then iterates over the numbers
from 0 to 9, appending each number to
the list. After the loop completes,
the list x contains the integers
from 0 to 9.
```

However, when one more line of the code is taken into consideration as follows:

```
x=[]
for i in range(10):
  x.append(i)
res=max(x)
```

the summary generated from LLM changes into:

```
The code initializes an empty list x,
iterates over the numbers from 0 to 9,
appending each number to the list,
resulting in x containing the integers
from 0 to 9. After the loop, the variable
res is assigned the maximum value in the
list x, which is 9.
```

If the summary of the line "res=max(x)" would have been generated separately, it would be difficult to

know from the summary what x is. Therefore, as visible from the above example, selecting the size of code snippet is crucial in getting the correct explanation of the code. This has been studied in Section 4.1 and 3. Empirical evaluation of multiple snippet sizes was the subject of an experiment, that the best size was chosen for the solution.

- Summary from metadata: The metadata has been used as is. No further paraphrasing of the metadata was performed. This was done since the metadata was sufficiently compact, simple and further summarization would lead to information loss. In larger projects, summarization might require gathering metadata related to the AI system, distributed across multiple files and using them directly after paraphrasing.
- Summary from logs: Abstractive summarization of logs involves the generation of an English language summary of the log file. This will help in the later stage where question-answering is involved. Experiments in this paper do not consider TWAI validation through logs since logs may require non-trivial preprocessing to filter "signals" from "noise" or insightful log-messages from the massive log files. This requires a detailed extensive investigation via a future paper building on this work.

3.2 Step 2 - Validating TWAI requirements from intermediate representation of AI system artefacts

In order to ensure that the requirements of each principle were validated, each of these requirements were represented by questions whose response is obtained using LLMs. In this stage, the summary collected from various sources of data were parsed to get answers of the questions for each of the guidelines. This is done using an LLM trained to do question-answering tasks. The LLM is prompted to answer each of the questions based on the context provided. Since datasets containing these questions and answers are not available in the literature, the ground truth of the questions was derived manually by multiple people, and the predicted response was validated against them. Experiments that compared different LLMs for the task of question-answering has been furnished in Section 4.1 and the best model has been selected for the said task. The main challenge of this step is finding the correct prompt for deriving answers from the summaries. For question-answering, multiple prompts have been experimented with, and the result of the experimentation has been documented in Table 6.

3.3 Step 3 - Quantification of validation of fulfilment of the requirements

In this stage, the score assigned to each of the questions derived in the previous step, in accordance with the answers generated from the summaries, were analyzed to generate a final compliance report. The scoring methodology is as follows:

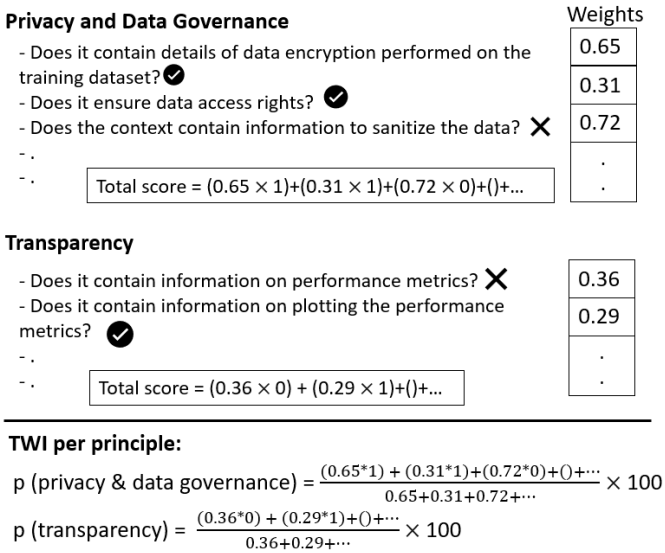

Figure 4 – A pictorial depiction of a scoring mechanism for trustworthy AI Scanner with an example. The ticks denote that the answer to that question is a "Yes" while the cross denotes the answer to the question is "No".

- Each of the requirements within the principles were assigned a weight $w_1, w_2, w_3, ..., w_i$ in the range $[0, 1]$.
- In this work, the response to each question is binary, where 0 indicates non-compliance and 1 indicates compliance to the requirement. Non-binary responses (e.g., probabilities) would work just as well and may be used in future.
- The final weight for each question is calculated as:

$$\alpha_i = w_i \times a \tag{1}$$

where a is 1 when the answer is affirmative (LLM generates answer as 'Yes'), and 0 when answer is negative.

- The total compliance weight considering the final weight of all the questions is calculated as:

$$\lambda_p = \sum_i \alpha_i \tag{2}$$

- The percentage of compliance, which is referred to as the TWAI Index (TWI) for each principle is calculated as:

$$TWI_p = \frac{\lambda_p}{\lambda_{p_{id}}} \times 100 \tag{3}$$

where the ideal index, $\lambda_{p_{id}}$ is:

$$\lambda_{p_{id}} = \sum_i \alpha'_i \tag{4}$$

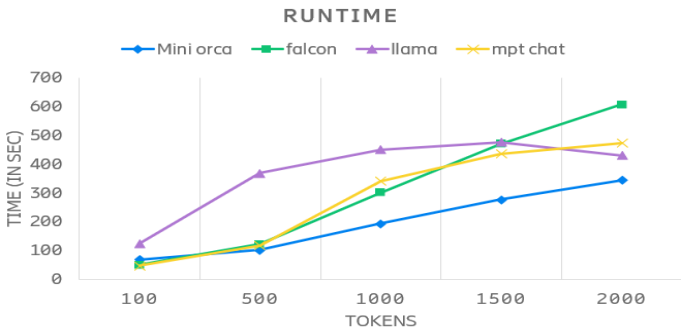

Figure 5 – Comparison of inference time of various LLMs on CPUs

and α_i' is:

$$\alpha_i' = w_i \times a(=1) \quad (5)$$

assuming all questions have been answered affirmatively, a is considered as 1.

Fig. 3 gives a pictorial depiction of the methodology. Fig. 4 demonstrates how the score is calculated for a particular use case.
The advantages of the proposed solution include:

- A principle-wise compliance score for AI systems that will give an explainable insight into the degree of compliance a system has towards each of the principles.
- The questions cover all the aspects of the TWAI guidelines; having a positive response for all the questions will ensure that the project is TWAI compliant.
- Each system will have a degree of compliance associated with it. This will give the users of the solution an insight into how trustworthy the AI system is and whether there is a need to increase its dependability.

4. EXPERIMENTATION AND RESULTS

The experimentation and results described in this section are spread across four subsections. These subsections discuss the characteristics of the LLMs tested for summary generation and question-answering, analyze efficacy of LLMs in generating summary representations of code, analyze efficacy of LLMs in answering questions from summaries and analyze efficacy of the proposed solution compared across multiple public repositories.

4.1 Analysis of the LLMs

To assess the efficacy of the scanner, each module of the scanner has been thoroughly assessed. The list of LLMs that were tested for the summary generation and question-answering problems is given in Table 1.
Fine-tuning LLMs [7] and recent concepts of Retrieval-Augmented Generation (RAG) [6] could improve performance, but the paper seeks to evaluate off-the-shelf LLMs using no more than prompt engineering [5]. The proposed solution leverages LLMs in GPT4ALL [16]. GPT4All is an open-source ecosystem designed to train and deploy large language models that can run efficiently on local devices. It provides accessible tools for developing and utilizing AI models, promoting wider adoption and customization in various applications. Among the LLMs provided in GPT4ALL, only LLMs with permissive licenses have been considered for evaluation and usage in the scanner. The max input token for all LLMs is 2048. The comparison of inference time for the LLMs on CPU has been depicted in Fig. 5. As visible, the lower the number of tokens, the lower the inference time is.

4.2 Analysis of efficacy of LLMs in generating summary representations of code

The first step of the proposed solution is to generate an intermediate representation from the code. To analyze the accuracy of intermediate representation (summary) generated by an LLM, the first step is to zero in on a metric to measure the accuracy. To quantify the performance of LLMs in generating summaries, a dataset of algorithms and its corresponding Python code was extracted from a public-domain website Javapoint[3]. Using the code, the explanation of the code that was generated using the LLMs are being compared. The generated explanations were compared with the public-domain explanation, listed on the webpage. The following subsections discuss the results of the analysis.

4.2.1 Selecting metric to quantify semantic similarity

To compare two summaries, various metrics have been analyzed. The semantic similarity between summaries was computed using Spacy [28], BleU score [29], BERT model with BERT score [30] and SBERT [10]. SpaCy is an open-source Natural Language Processing (NLP) library that provides pre-trained models and tools for efficient processing, analysis and understanding of text. The Bilingual Evaluation Understudy (BLEU) score is a metric for evaluating the quality of machine-translated text by comparing it to reference translations based on n-gram overlap. The Sentence-Bert (SBERT) score is a metric that assesses the semantic similarity between sentences by leveraging BERT-based sentence embeddings and cosine similarity measures. BERT score is a metric for evaluating the quality of natural language processing models, particularly those based on Bidirectional Encoder Representations from Transformers (BERT), by comparing their output to human-generated reference sentences. The analysis result on our curated dataset has been summarized in Table 2.
Analyzing Table 2, in the test set 1, three summaries[4] apart from the reference summary have been considered.

[3] https://www.javatpoint.com/python-algorithms
[4] This was an empirical evaluation using manually curated summaries, crafted by us, to test which metric suits best when comparing summaries.

Table 2 – Comparing performance of the LLMs in summary generation from code.

Metric	Test 1			Test 2		Test 3	
	Summary 1	Summary 2	Summary 3	Summary 1	Summary 2	Summary 1	Summary 2
Spacy (word embedding based)	1	0.95	0.9	0.9732	0.9003	0.9508	0.8493
BleU score	0	0	0	0.0013	0.0014	0.0015	0.0014
BERT model with BERT score	1	0.9263	0.8085	0.8058	0.7114	0.8054	0.6329
SBERT with cosine similarity	0.9999	0.9057	0.9732	0.9199	0.5955	0.8875	0.3759

Table 3 – Comparing semantic similarity in summary generated by LLMs from code based on SBERT.

LLM	100 tokens	500 tokens	1000 tokens	1500 tokens	2000 tokens
Mini-Orca	0.49	0.63	0.63	0.63	0.63
Falcon	0.53	0.51	0.51	0.49	0.43
Llama 7B	0.63	0.65	0.65	0.62	0.64
MPT-chat	0.39	0.34	0.34	0.38	0.37

These summaries are contextually different but semantically similar. For example, if the reference is "The application conducted an experimentation using SHAP values to interpret the contributions of each feature to the model's predictions.", summary 1 is "The application conducted an experimentation using SHAP values to interpret the contributions of each feature to the model's predictions.", summary 2 is "The application has performed experimentation using SHAP values to understand the contributions of each feature to the model's predictions.", and summary 3 is "Experimentation using SHAP values has been performed to recognize the contributions of each feature to the model's predictions." For the other two tests (test 2 and test 3), summary 1 is contextually very similar to the reference, while summary 2 is a summary different from the reference. As visible in Table 2, the similarity score for summary 1 and summary 3 is higher and much closer to the reference than for summary 2. Considering the above-mentioned metrics and subjecting them to the three exemplars chosen in Table 2, SBERT (sentence transformer model used: paraphrase-MiniLM-L6-v2) with cosine similarity was finalized to compare the summary of the code generated by the LLM and the corresponding algorithm of the code.

Experiments have been carried out to determine the appropriate size of the code snippet that could be ingested by an LLM, ensuring the algorithm's quality is not compromised. In this experiment, the average similarity score between the LLM-generated summaries and the algorithms for various sizes of code snippets have been calculated. The dataset for the experiment is the code-algorithm dataset curated from Javapoint, which is described in the next subsection. The average SBERT score depicting the contextual similarity between the code explanation generated by the LLM and the algorithm for the corresponding code has been summarized in Table 3.

From the analysis above we concluded that the size of the code snippet did not significantly impact the outcome from among those tested. As visible from Fig. 5, 100 tokens[5] gave the least inference time. Therefore, empirical evaluation of multiple snippet sizes concluded with a snippet size of 100 tokens being used. Experiments in this paper do not yet consider nesting or sub-blocks of code.

4.2.2 Selecting LLM for summary generation

In the next stage, the aim is to select an LLM appropriate for summary generation from code. A set of 55 algorithms and their corresponding Python codes were collected from Javapoint and treated as the dataset for further analysis. The aim was to generate an explanation of these codes from LLMs. Subsequently, the explanation generated by the LLM was compared with the algorithm in the dataset, to select the best LLM suited for the requirement of this study. Table 4 gives a summary of the comparison.

Table 4 – Comparing performance of the LLMs in summary generation from code over 55 instances.

LLM	Average SBERT	Variance SBERT
Mini-Orca	0.59	0.07
Falcon	0.34	0.02
Llama 7B	0.63	0.02
MPT-chat	0.39	0.05

In multiple instances, the Mini-Orca model generated irrelevant and inconsistent explanations for the code that it was being provided as input. Such LLM-generated summaries were automatically filtered out before analysis. Summaries generated by the Mini-Orca model were short and not detailed. On the other hand, the Llama model gen-

[5]Tokens are the building blocks of language in LLMs. With the help of the website https://platform.openai.com/tokenizer, we were able to approximate number of tokens for each line of codes

erated a detailed summary of the code snippets. Considering the variance of the LLMs, Llama 7b and Falcon came out as the least divergent from the reference summary and more stable compared to the other two LLMs. However, given the scale of variance, it is evident that there is no significant difference across algorithms. Therefore, given the performance of Llama 7b in generating summaries and the variance in its result being the lowest, Llama 7B model was chosen. In the future, for metadata whether paraphrasing is needed or not, is another challenge that will need to be addressed. For code, automated selection of optimal snippet size for generating summaries and incorporating parse trees for codes to determine the same may be explored.

4.3 Analysis of efficacy of LLMs in answering questions from summaries

When it comes to answering questions from summaries, answers of the questions were needed in the form of "Yes/No". Based on the answer, the questions were assigned a value of 0/1, 0 being "No" and 1 being "Yes". The efficacy of the LLM models listed in Table 1 for context-based question-answering problems was evaluated via multiple datasets. The questions given in SQUAD dataset [9] (groups: immunology, geology, prime numbers) were manually converted to "Yes/No" questions. For the MPT-chat model, a variety of prompts were tested. However, this LLM was not able to generate answers to questions in the form of "Yes/No". Therefore, its performance could not be measured.

In Table 5, the accuracy is primarily stable across the LLMs under consideration. However, recall and percentage of False Positives (FPs) sees a wide variation. It is important to note that not all LLMs which are commercially usable and locally executable, may be efficient in the task of question-answering, which eventually reflects in their performance metrics. The similar performance in accuracy is due to a large number of True Negative (TN, regulations correctly identified as not conforming) and False Negative (FN, regulations incorrectly identified as not conforming). When TWAI regulations are to be taken into consideration, it is critical to prevent FP prediction of regulations (regulations incorrectly identified as conforming). Falcon and Mini Orca have a low FP indicating that cases of predicting regulations to be present when it is absent, is low. The low recall values of Falcon and Mini Orca is due to a high number of FNs being predicted compared to TNs. From the perspective of TWAI, reducing FPs is more crucial than reducing FNs, increasing TPs or increasing TNs. Therefore, Falcon has been selected based on the least false positive predictions.

The analysis of the question-answering is given in Table 5. As visible from tables 4 and 5, some LLMs do well in generating summaries from code while some were better at question-answering from the summaries. Therefore, with the above analysis, Llama 7B is the LLM selected for generating summaries and Falcon is the LLM selected to be used for question-answering.

In continuation to the above experimentation, the solution's efficacy was scrutinized when Falcon, which is considered for question-answering, was subjected to various queries, provided as prompts. A summary of the same has been given in Table 6. As visible from the table, there is not much variation in the accuracy of the prompts. However, recall and false positives tell a different story. A tradeoff is noticed. For high recall, the percentage of false positives is also high. The ultimate aim is to maximize recall but minimize false positives. The selected prompt was "Respond to the question concerning the context, in one word only Yes or No.", since the tradeoff between accuracy, recall and percentage of false positives for this prompt is optimally satisfied. Based on the analysis above, the chosen LLM for generating summary was Llama 7b model. For question-answering, the chosen LLM was the Falcon model.

4.4 Analysis of the efficacy of the proposed solution compared across multiple public repositories

For comparing performance of the proposed method, four Github Kaggle-based repositories [6] were selected, which had both Python code and metadata. The accuracy of the solution has been summarized in Table 7. No public datasets of public AI-system code repository and corresponding TWAI assessments exist. Therefore, we have manually checked compliance and answered each of the TWAI questions. The experiment then checks if the proposed automation solution is able to produce those answers. The ground truth for the Github repositories cannot be derived from a standard benchmark and therefore has been done manually.

The ground truth has been compared with the output of the Falcon model to generate the performance metric. The metrics have been separately depicted for code and metadata to understand which data sources need more attention to generate more accurate results. Table 7 shows that the scanner demonstrates a viable performance for each repository in terms of accuracy. The accuracy can be further improved via the application of ensemble prompting. The execution speed can be improved by using GPU-based LLMs instead of CPU-based LLMs. Furthermore, only Python has been considered as the programming language for the proposed solution. There is scope for taking more programming languages into consideration.

5. FUTURE SCOPE

This paper has proposed a Large Language Model (LLM)-based solution for automated measurement of the trustworthiness of any AI system. The solution measures and reports the level of compliance of an AI System to the

[6] https://github.com/sayaliwalke30/Kaggle-Projects

Table 5 – Comparing performance of the LLMs in answering questions from summaries on SQUAD dataset.

LLM	Immunity (224 questions)			Geology (362 questions)			Prime Number (308 questions)		
	Accuracy (%)	Recall (%)	FP (%)	Accuracy (%)	Recall (%)	FP (%)	Accuracy (%)	Recall (%)	FP (%)
Mini-Orca	50.45	3.57	2.67	56.07	30.17	21.54	49.35	3.92	2.9
Llama-7b	51.12	100	97.76	33.7	98.28	65.78	50.32	97.42	48.37
Falcon	50.89	5.8	4.01	64.64	14.69	8.01	51.3	1.96	0
MPTChat	-	-	-	-	-	-	-	-	-

Table 6 – Comparing the performance of Falcon for various prompts on SQUAD dataset.

No.	Prompt	Immunity (224 questions)			Geology (362 questions)			Prime Number (308 questions)		
		Accuracy (%)	Recall (%)	FP (%)	Accuracy (%)	Recall (%)	FP (%)	Accuracy (%)	Recall (%)	FP (%)
1	Respond with a yes or no.	48.43	89.73	92.87	44.75	82.9	49.72	51.94	82.35	39.28
2	Respond to the question with respect to the context, in one word only Yes or No.	52.9	36.16	30.35	56.35	35.71	23.75	51.29	21.56	9.74
3	Answer the question with respect to the context in either Yes or No only. If the answer is found in the context, respond with yes. Even if there is slight ambiguity, respond with No.	52.9	15.17	9.3	58.83	35.34	20.44	51.62	31.37	14.28
4	Based on the context respond in yes or no. Answer yes only when you are sure.	52.67	56.25	50.89	43.92	64.65	44.75	49.39	52.28	26.94
5	Based on the context respond with Yes or No.	47.99	19.64	23.66	54.97	33.33	23.48	45.77	9.1	9.09
6	Answer the question in yes or no based on the context.	46.66	47.78	50.44	48.06	50.86	36.18	54.22	47.75	18.83

Table 7 – Comparing accuracy of the solution across various repositories.

Repositories	Code (in %)	Metadata (in %)
Repo 1	85.71	78.23
Repo 2	87.75	66.66
Repo 3	76.19	85.71
Repo 4	80.27	89.79
Repo 5	75.71	65.98

trustworthy AI guidelines. Given the increasing usage of AI in the telecommunication industry, it is crucial for AI to follow a predefined set of standards, to enable fair comparison or assessment.

The approach proposed in this paper has multiple avenues of improvement. One such area is the automated ingestion of new policies, the removal of obsolete ones and the translation of these policies into a relevant set of questions, that help measure trustworthiness. Identifying conflicting questions would help to identify policies that might be in opposition to each other.

Future work in the TWAI domain may focus on defining interface and audit capabilities, much like a lawful intercept, that may be required for internal audit by a service provider or an external audit by government owned/recognized certification bodies. In such scenarios exposing a TWAI interface from the AI systems to intercept and collect the trustworthiness measurement of the AI functions will be needed. The automated measurement proposed in this paper will be the underlying basis for all such future work. It is important to cater to as many data sources as possible, logs and reports are a good example.

6. CONCLUSION

Considering the pace with which the industry is adopting the principles of trustworthy AI, automated trustworthiness measurement of AI systems will be a significant enabler for safe and responsible AI-driven Telco in 5G, 5G Advanced and 6G to achieve autonomous networks. Automating the evaluation of trustworthiness of AI systems by adhering to the TWAI guidelines is critical. Existing efforts at ascertaining trustworthiness are manually-driven and therefore subjective. This work automates trustworthiness measurement and therefore provides an objective measure of compliance with desired policies and a basis for comparison. To materialize this objective, evaluation of various off-the-shelf LLMs was carried out. The proposed solution is a first step towards automating the measurement of the trustworthiness of an AI system and the results look promising.

ACKNOWLEDGMENT

The authors would like to thank Paddy Farrell, A K Raghavan, Gnanaprakash Janarthanam, Shreeya Nallaboina and Sidaarth Balaji for their support.

REFERENCES

[1] Francisco García-Peñalvo and Andrea Vázquez-Ingelmo. "What do we mean by GenAI? A systematic mapping of the evolution, trends, and techniques involved in Generative AI". In: (2023).

[2] Luciano Floridi. "Establishing the rules for building Trustworthy AI". In: *Nature Machine Intelligence* 1.6 (2019), pp. 261–262.

[3] Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. "Toolqa: A dataset for llm question answering with external tools". In: *Advances in Neural Information Processing Systems* 36 (2024).

[4] Roberto E Balmer, Stanford L Levin, and Stephen Schmidt. "Artificial Intelligence Applications in Telecommunications and other network industries". In: *Telecommunications Policy* 44.6 (2020), p. 101977.

[5] Aras Bozkurt and Ramesh C Sharma. "Generative AI and prompt engineering: The art of whispering to let the genie out of the algorithmic world". In: *Asian Journal of Distance Education* 18.2 (2023), pp. i–vii.

[6] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9459–9474.

[7] Michiel Bakker, Martin Chadwick, Hannah Sheahan, Michael Tessler, Lucy Campbell-Gillingham, Jan Balaguer, Nat McAleese, Amelia Glaese, John Aslanides, Matt Botvinick, et al. "Fine-tuning language models to find agreement among humans with diverse preferences". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 38176–38189.

[8] Charles Radclyffe, Mafalda Ribeiro, and Robert H Wortham. "The assessment list for trustworthy artificial intelligence: A review and recommendations". In: *Frontiers in Artificial Intelligence* 6 (2023), p. 1020592.

[9] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. "Squad: 100,000+ questions for machine comprehension of text". In: *arXiv preprint arXiv:1606.05250* (2016).

[10] Nils Reimers and Iryna Gurevych. "Sentence-bert: Sentence embeddings using siamese bert-networks". In: *arXiv preprint arXiv:1908.10084* (2019).

[11] Nathalie A Smuha. "The EU approach to ethics guidelines for trustworthy artificial intelligence". In: *Computer Law Review International* 20.4 (2019), pp. 97–106.

[12] Jim Reno, Rafia Inam, and Attila Ulbert. "Trustworthy AI - What it means for telecom". In: *Ericsson White Paper* (2023).

[13] Roberto V Zicari, John Brodersen, James Brusseau, Boris Düdder, Timo Eichhorn, Todor Ivanov, Georgios Kararigas, Pedro Kringen, Melissa McCullough, Florian Möslein, et al. "Z-Inspection®: a process to assess Trustworthy AI". In: *IEEE Transactions on Technology and Society* 2.2 (2021), pp. 83–97.

[14] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. "Llama: Open and efficient foundation language models". In: *arXiv preprint arXiv:2302.13971* (2023).

[15] Yoshua X ZXhang, Yann M Haxo, and Ying X Mat. "Falcon LLM: A New Frontier in Natural Language Processing". In: *AC Investment Research Journal* 220.44 (2023).

[16] Yuvanesh Anand, Zach Nussbaum, Brandon Duderstadt, Benjamin Schmidt, and Andriy Mulyar. "Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo". In: *GitHub* (2023).

[17] Stephen MacNeil, Andrew Tran, Arto Hellas, Joanne Kim, Sami Sarsa, Paul Denny, Seth Bernstein, and Juho Leinonen. "Experiences from using code explanations generated by large language models in a web software development e-book". In: *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. 2023, pp. 931–937.

[18] Juho Leinonen, Paul Denny, Stephen MacNeil, Sami Sarsa, Seth Bernstein, Joanne Kim, Andrew Tran, and Arto Hellas. "Comparing code explanations created by students and large language models". In: *arXiv preprint arXiv:2304.03938* (2023).

[19] Hisashi Kamezawa, Noriki Nishida, Nobuyuki Shimizu, Takashi Miyazaki, and Hideki Nakayama. "Rnsum: A large-scale dataset for automatic release note generation via commit logs summarization". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2022, pp. 8718–8735.

[20] Weibin Meng, Federico Zaiter, Yuheng Huang, Ying Liu, Shenglin Zhang, Yuzhe Zhang, Yichen Zhu, Tianke Zhang, En Wang, Zuomin Ren, et al. "Summarizing unstructured logs in online services". In: *arXiv preprint arXiv:2012.08938* (2020).

[21] Mahanagar Doorsanchar Bhawan and Jawahar Lal Nehru Marg. "Telecom Regulatory Authority of India". In: *resource* (2023), p. 140.

[22] https:// www.bbc.com/news/technology-54175359. "Uber's self-driving operator charged over fatal crash". In: (2018).

[23] David Baidoo-Anu and Leticia Owusu Ansah. "Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning". In: *Journal of AI* 7.1 (2023), pp. 52–62.

[24] Roberto Gozalo-Brizuela and Eduardo C Garrido-Merchan. "ChatGPT is not all you need. A State of the Art Review of large Generative AI models". In: *arXiv preprint arXiv:2301.04655* (2023).

[25] Kevin Lin, Chung-Ching Lin, Lin Liang, Zicheng Liu, and Lijuan Wang. "Mpt: Mesh pre-training with transformers for human pose and mesh reconstruction". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 3415–3425.

[26] Xiaochuang Han and Yulia Tsvetkov. "ORCA: Interpreting Prompted Language Models via Locating Supporting Evidence in the Ocean of Pretraining Data". In: (2022).

[27] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. "Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 3505–3506.

[28] Yuli Vasiliev. *Natural language processing with Python and spaCy: A practical introduction*. No Starch Press, 2020.

[29] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[30] Joonbo Shin, Yoonhyung Lee, and Kyomin Jung. "Effective sentence scoring method using BERT for speech recognition". In: *Asian Conference on Machine Learning*. PMLR. 2019, pp. 1081–1093.

AUTHORS

Dr Rishika Sen is a Data Scientist III at Ericsson. She obtained her Ph.D. in computer science from the Machine Intelligence Unit, Indian Statistical Institute, Kolkata in 2021. She has completed her M.Sc. (2014) and B.Sc. (2012) from the University of Calcutta. Her domain of research is data science, data analysis, bioinformatics, and machine learning. She has published in various internationally reputed journals as the first author. Detailed information about Rishika is available at http://rishikasen.de.

Dr Shrihari Vasudevan (D.Sc., 2008) has a background in statistical modelling, data fusion and machine learning applied to complex data from heterogeneous sources and diverse application domains such as robotics, mining automation, natural resources, workforce analytics, enterprise financial process transformation and currently, in telecommunications, designing state of the art AI/ML solutions for 5G. He also contributes to strategic studies aimed at determining the role of AI/ML in 6G and beyond. His work has been evidenced by numerous peer-reviewed publications, patents and recognitions by the business and scientific communities. Detailed information about Shrihari is available at http://lshv.net.

Dr Ricardo Britto is Head of Development Group AI at Ericsson. He has a PhD from the Blekinge Institute of Technology's Department of Software Engineering. His research interests include large-scale agile software development, global software engineering, search-based software engineering and software process improvement. Britto received an MSc in electrical engineering from the Federal University of Rio Grande do Norte.

M.J. Prasath is a telecommunications leader with more than 25 years of experience in architecture, designing and developing complex real-time Telco applications for 2G, 3G,4G and 5G networks. He has spent 12 years of his professional experience in designing and launching new services for network operators DT, Telefonica, BT and 3UK. In his current role as Director of Data Science, he focuses on bringing AI capability to Ericsson's product and services portfolio. His current area of interest includes autonomous networks, GenAI and AI native transformation in Telco networks.