PhD-FSTM-2023-099
The Faculty of Science, Technology and Medicine

DISSERTATION

Defense held on 12/09/2023 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG
EN INFORMATIQUE

by

Muhammed Akif AĞCA
Born on 17 September 1988 in Antakya/Hatay, (TÜRKİYE)

TRUSTED DISTRIBUTED ARTIFICIAL INTELLIGENCE FOR CRITICAL AND AUTONOMOUS SYSTEMS

Dissertation Defense committee:

Chair: Prof. Dr. Pascal BOUVRY,
University of Luxembourg, LUXEMBOURG.

Vice Chair / Expert Member: Prof. Dr. Nazim AGOULMINE,
University of Evry (Paris Saclay), FRANCE.

Expert member: Prof. Dr. Abdelmadjid BOUABDALLAH,
Sorbonne University University of Technology of Compiègne, FRANCE

Member: Dr. Sébastien FAYE,
Luxembourg Institute of Science and Technology – LIST, LUXEMBOURG

Supervisor: Prof. Dr. Djamel KHADRAOUI,
Luxembourg Institute of Science and Technology – LIST, LUXEMBOURG

Affidavit

I hereby confirm that the PhD thesis entitled ‘TRUSTED DISTRIBUTED ARTIFICIAL INTELLIGENCE FOR CRITICAL AND AUTONOMOUS SYSTEMS’ has been written independently and without any other sources than cited.

Luxembourg,

Muhammed Akif AĞCA

Abstract: Intelligent systems are expected to adapt to change dynamically up to varying context by keeping trustworthiness of the system via available resources. In the case of decentralized resources and algorithms, scalability becomes a key issue compared to centralized approaches. The increasing number of nodes in swarm system impacts its behavior, since it utilizes complex computational mechanisms for co-operative missions, which has to dynamically fuse large-scale matrices of the nodes and merge those to be able to scale and train the system in (near) real-time. The literature review in the domain shows that keeping the system resilient is challenging since it requires real-time updates and predictions in different system layers. In this context, some approaches based on ledger chain structures and big-data technologies can accomplish the transaction scalability and memory-speed analytic performance in some manner. Despite that, mission/safety/operation-critical applications (such as cooperative autonomous missions), require measuring and monitoring trust in critical checkpoints of a system in operation while keeping the expected performances of the total system. The thesis aims at solving such challenge and targeted the development of a new Methods called Trusted Distributed Artificial Intelligence (TDAI). The main contributions of the thesis are on the development of novel method that covers the following aspects:

- Measuring, quantifying and justifying trust in distributed systems.
- Enabling trusted scalability of autonomous systems.
- Assuring trust for swarm intelligence mechanisms.
- Manipulating swarm system units with search and mining focus to implement TDAI methodology in (near) real time.

In order to develop such a holistic methodology approach, we gradually contributed from different perspectives. We initially proposed to utilize a holistic MEMCA (Memory-Centric-Analytics) abstraction, to maximize the trust factor of the system while enabling trusted scalability of the transactions and memory-speed. Data locality is extended to the edges in trusted scalable manner to handle the complexity of data/transaction-flow while the memory performance was ensured in the swarm. This approach is based on micro-services architecture and has innovative approaches for layer-wise structure enabling verification of trust in critical checkpoints of an operational system via observed and observing nodes that can enable us to build growing intelligent mechanisms with software-defined networking (SDN) features by virtualizing network functionalities with maximized trust features for continuous trust monitoring in observed context.

Accordingly, critical feature sets of AI systems and growing intelligent mechanisms are identified, measured, quantified and justified dynamically with defined three architectural perspectives (1) central, (2) decentral/autonomous/embedded, (3) distributed/hybrid for emerging trusted distributed AI mechanisms. Therefore, resiliency and robustness can be assured in a dynamic context with an end-to-end Trusted Execution Environment (TEE) for growing intelligent mechanisms and systems. Thanks to TDAI methodology, the system could consider the trust indicators that can bring a confidence on the predictions in the distributed context. This way, distributed algorithms can be processed at massive scale by ensuring trusted scalability. Therefore, resiliency and robustness can be assured in a dynamic context with an end-to-end Trusted Execution Environment (TEE) for growing intelligent mechanisms and systems. Besides that, the trust measurement, quantification, and justification methodologies on top of TDAI are also applied in emerging distributed systems and their underlying diverse application domains. Finally, smartness features are also improved with human-like intelligence abilities at massive scale thanks to the promising performance of TDAI at massive scale initial deployment experiments. TDAI is demonstrated in a cross-border financial risk monitoring scenario within the distributed systems formed by the connected nodes, to detect and minimize critical risk alerts within the observed context. The main objective is to maximize trust values of the critical nodes and the observed environment within the monitored time-span.

Following the thesis, the simulation of autonomous and connected vehicles will be pursued as the application domain with specific use-cases for further massive scale deployments within different cross-border challenges. Furthermore, the innovative approaches will also be utilized to wider spectrum intelligent system requirements of smart-cities and space systems as well with improved trust features.
INDEX TERMS Trusted AI, Distributed Systems, Software Defined Networking (SDN), Trusted Execution Environment (TEE)

OUTLINE

i List of Abbreviations

TDAI	Trusted Distributed Artificial Intelligence
TEE	Trusted Execution Environment
SDN	Software Defined Networking
SDR	Software Defined Radio
NFV	Network Functions Virtualization
MEMCA	Memory-Centric Analytics
MCMC	Markovian-Chain Monte Carlo
MAS	Multi-Agent Systems
5/6G	Fifth/Sixth Generation
E2E	End-to-end
P2P	Peer-to-Peer
WAN	Wide Area Network
MAN	Metropolitan Area Network
LAN	Local Area Network
PAN	Personal Area Network
ACID	(Atomicity, Consistency, Integrity, Durability)

ii List of Figures

iii List of Tables

iv Acknowledgements

Thanks to Luxembourg Institute of Science and Technology (LIST) for supporting PhD thesis component of TDAI Project with Grant Number: 10.13039/100016172-LIST Ph.D. Grant. Thanks to jury members for favorable considerations and manuscript reviews.

Chapter I: Introduction

Intelligent systems are able to adapt to change dynamically in varying contexts by keeping the trustworthiness of a system within the limits of available resources. However, increasing computational and storage capacities require the decentralization of resources and algorithms. Trusted scalability of analytical functions and resources is still an open research issue. In fact, large-scale matrices generated by the novel methods, which are used to formally state the data and context, have to be merged and dynamically fused to be able to scale/train [1] the decentralizing algorithms. Furthermore, an increasing number of nodes in the system cause swarm behavior [2,3,4] due to the use of complex computational systems for co-operative and critical missions of autonomous system units. The components utilized to interact with these units are called edge devices, and have densified storage and computational facilities, which enable them to cover broader additional contexts and a wider spectrum. This is a key enhancement for running novel machine-learning algorithms at the edge by ensuring trust and security [5]. Nevertheless, the dynamic context exponentially triggers data/transaction flows in the system. The flows lead recent challenges for intelligence valorization in a dynamic context. Table 1 introduces selected keyword definitions that will be used in the rest of the document.

Trust	Belief in Reliability, Truth, Ability, Strength, Reliance, Dependence, Faith, and Confidence. In the computing domain, it is defined as the behavioural integrity of a system, which behaves as expected for all transactions [37,14,15,28].
Distributed System	Set of nodes with decentralized/distributed memory and processing resources, which operates coherently [14,15].
Autonomous System	In a general context, a network or set of networks under one organization. In systems and computing science, it can be defined as a component or a unit, which can operate independently [13,38,15].
Resilient Swarm	Set of fully connected nodes.
Artificial Intelligence - AI	Ability to make the right decision in a mathematically well-defined context.
Smart System	A system with a set of nodes, which can behave intelligently [14,15].

Table 1. Keyword definitions.

Intelligence flow is also need to be valorized within critical/distributed/autonomous systems constraints as defined in Table 1 keyword definitions. Thereby, security/privacy features, QoS metrics, and other trust justification features are defined as part of the justification process. These features are widely explored in trust building approaches with QoS and security/privacy based trust models [145,146]. However, the models need to be managed dynamically, instead of static approaches to enable dynamism in observed context. So that trust can be monitored via sufficiently trusted nodes within the embedded checkpoints, are linked to the nodes and fetched to growth-flow mechanisms of intelligent systems [140] with the feedback structures to justify the trust in (near) real time [147].

Besides that, valorizing the swarm intelligence and keeping the system resilient require real-time updates and predictions in different system layers [6,7]. Ledger-based chain structures and big-data technologies can accomplish transaction scalability and memory-speed analytic performance to a certain extent. Despite this, mission/safety/operation-critical applications, such as tracking a moving object, monitored by a swarm, require trust to be verified at the critical checkpoints while maintaining the performance of the overall system. Extending data locality to the edge in a trusted scalable manner with holistic views can help to manage the complexity of the data/transaction-flow and maintain the memory speed performance of the total system and analytical transactions [130].

Furthermore, holistic abstraction can maximize the trust factor of the system while enabling trusted scalability of the transactions and keeping the memory speed of large-scale trusted analytics on massive-systems. Co-operation between these units can be maximized with micro-service architectures, which have innovative approaches for layer-wise structures. Thereby, trust can be verified at critical checkpoints to maximize the targeted throughputs of these units. The approaches can help to dynamically define user feature sets and the management of these features can be enabled at run-time to maximize the performance of the co-operative mission, and the trust factor of a resilient system. Therefore, the system can consider the trust indicators that can give confidence concerning, for instance, the predictions processed by the distributed AI/ML algorithms at a massive scale by ensuring trusted scalability with the justified features.

Thereby, resiliency and robustness can be assured in a dynamic context with an end-to-end Trusted Execution Environment (TEE) for the growing intelligent mechanisms and systems thanks to the holistic abstraction paradigms as a critical improvement to Amdahl's notation [5]. Since the dynamic holistic views can enable to measure the trust and correlate with expected throughput levels of the system node in (near) real-time. The innovative approach can be utilized to extend TEE definitions with end-to-end (E2E) architectural perspective via the improved check-point mechanism and dynamic feedback structures. Since, intelligence-flow mechanisms can ensure behavioral integrity of a system node with the justified features of the context requirements. By that means, resiliency and robustness can be ensured during justification process within the intelligence-flow mechanisms of growth-flow structures. E2E trust mechanism protection is also succeeded via the trusted agents based ensured interactivity within the observed context by dynamically verifying the expectation fulfillment of the nodes and the linked context.

These critical feature sets are explored as components of a distributed computing system with novel trusted distributed AI-driven approaches with a comprehensive scientific background definition. In this way, the trust justification features can be explored and identified for the emerging intelligent systems and mechanisms.

As a main target in this thesis, we introduced a new concept of **Trusted Distributed AI (TDAI)**, which offers the ability to make the right decision in a mathematically well-defined context within the critical, distributed, autonomous system constraints [15]. It covers the following main research questions:

1. How to handle the dynamic and (near) real time context?
2. How to consider the scalability in a holistic end-to-end view?
3. How generic the methodology is in different critical and autonomous scenario use cases?

Tackling the above-mentioned research questions required first benchmarking our approach against existing methods to help understanding the new concept of TDAI, with a comprehensive review of the major related contributions in the current literature. Thanks to that, we obtained comprehensive view on emerging AI systems concepts and its' critical components like SDN, TEE etc. Thereby, TDAI can be utilized to gain massive scale operationally in trusted scalable manner [5] thanks to the improvements and promising experimental performances of the methodology, and its novel critical features identified in this thesis.

The TDAI methodological innovations proposed in the thesis are applicable to many perspectives since it has been experimentally validated in real-life experiments and it is based on a taxonomy allowing a formal trust measurement and quantification with novel holistic abstraction paradigms [5]. Associated metrics makes it operational and generic to any critical distributed and autonomous applications. The resulting trusted distributed system becomes complex and may involve self-organizing techniques with multiple hierarchical layers to better manage the decisions between several nodes. A node which might play the role of master would benefit from an overall view for global decision-making for trusted interactivity and cooperation. Compared to existing trust based methodology approaches, TDAI is based on the fact that it considers a combination of security and privacy features together with the QoS related ones. Its exploitation requires a modeling phase of the observed system with a focus on the critical checkpoints. This way it makes it unique against the existing trust monitoring approaches, which are compared in details in Table 3.b.

Many applications might be related to this, especially those related to mobile, edge and ubiquitous computing in mobility scenarios where vehicles are equipped with context-aware and user-centric technologies. Furthermore, applications that cover this could be related to driver behavior profiling, with different possible outcomes, like low-emission driving where the user or the car (if fully autonomous) would need to follow precise instructions depending on the way that is driving and, on the environment and user-health diagnosis as explained in details in chapter V.

During this thesis nine publications and one patent were achieved. They are listed below:

1. CSCI'19-21st ICAI 2019 – International Conference on Artificial Intelligence: Persisting trust in untrusted resilient city context. Ağca, M. A., Khadraoui, D., & Faye, S. (2019). Persisting Trust In Untrusted Varying Resilient Citv Context V. In Proceedings on the International Conference on Artificial Intelligence (ICAI) (pp. 312-318).
2. CSCI'19 - 6th Annual Conf. on Computational Science & Computational Intelligence: A holistic abstraction to ensure trusted scaling and memory speed trusted analytics. Agca, M. A. (2019, December). A Holistic Abstraction to Ensure Trusted Scaling and Memory Speed Trusted Analytics. In 2019 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 1428-1434). IEEE.
3. CSCE'20 - PDPTA'20 - The 26th Int'l Conference on Parallel and Distributed Processing Techniques and Applications: Challenges for Swarm of UAV/Drone Based Intelligence. Agca, M.A. et.al (2020). Challenges for Swarm of UAV/Drone Based Intelligence. In the 26th Int'l Conference on Parallel and Distributed Processing Techniques and Applications.
4. CSCI'21 - 8th Annual Conf. on Computational Science & Computational Intelligence: Persisting Trust in Emerging Hybrid-Clouds and 6G Systems. Agca, M. A. et.al. Persisting Trust in Emerging Hybrid-Clouds and 6G Systems. In 2021 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 1428-1434). IEEE.
5. IEEE Access - A Survey on Trusted Distributed Artificial Intelligence, Published. Ağca, M. A., Faye, S., & Khadraoui, D. (2022). A survey on trusted distributed artificial intelligence. IEEE Access, 10, 55308-55337.
6. CSCE'22 - PDPTA'22 - The 28th Int'l Conference on Parallel and Distributed Processing Techniques and Applications: End-to-end Trust Mechanism with Dynamic Holistic View Based Throughput Maximization Approach. Agca, M. A., (2022). End-to-end Trust Mechanism with Dynamic Holistic View Based Throughput Maximization Approach. In the 26th Int'l Conference on Parallel and Distributed Processing Techniques and Applications.

7. CSCE'22-24th ICAI 2022 – International Conference on Artificial Intelligence: brainIT: A generic IT Core Mechanism for Continuous Growth-Flow in Dynamic Context. Ağca, M. A., (2022). brainIT: A generic IT Core Mechanism for Continuous Growth-Flow in Dynamic Context. In Proceedings on the International Conference on Artificial Intelligence (ICAI) (pp. ..). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).
8. LFMS 2022 The Linux Foundation Member Summit: End-to-end Trusted Execution Environment (TEE) with Dynamic Holistic View Based Throughput Maximization Approach for Open Source Systems, Invited Speaker, https://lfms22.sched.com/speaker/agca.akif.
9. IEEE Access - Trusted Distributed Artificial Intelligence (TDAI), Published. Ağca, M. A., Faye, S., & Khadraoui, D. (2023). Trusted Distributed Artificial Intelligence (TDAI). IEEE Access.
10. P-2022-019-AGCA-trusted autonomous interaction nodes (ITIS), updated: METHOD FOR TRUSTED DISTRIBUTED AUTONOMOUS SYSTEMS, PCT Application Number: PCT/EP2023/070587. Submission Number: 12273270, submitted: 25th July 2022, extended at: 25th July, 2023.

Rest of the thesis is structured as follows, Chapter. II introduces the general scientific background and definitions of AI vs Distributed AI and intelligent systems, Chapter. III introduces comparative analysis of the literature, which gives details about the security, privacy, and trust metrics considered in this study. Furthermore, it includes the discussion, current challenges, and a comparative analysis of the literature. Chapter IV introduces TDAI methodology formal statement, system architecture and its components. Chapter V explains evaluations, experiments and related use-case specifications and descriptions. Chapter. VI concludes the study and introduces future directions.

Chapter II: Background Analysis on AI Systems

Introduction

This chapter covers the main background on the distributed elements of AI systems together with their underlying trust justification features. It presents and defines the concepts of trusted distributed AI systems and their underlying architectures. Furthermore, reviews the state of the art with comparative analyses of related studies.

A. Distributed Systems Vs Trusted Distributed Systems

In order to characterize a distributed system, it is useful to use the **logical functional distribution** of the processing capabilities of a given system composed of a set of computers. The logical distribution of such capabilities is based on the following criteria like: Multiple processes, Inter-process communication, Shared memory and Collective goals. Some examples of distributed systems can be related to Peer-to-peer networks, Process control systems, Sensor networks and Grid computing.

The computers in these systems are identified as system units, which are generic components called **nodes**. A distributed system is a system with set of nodes $N_i : \{N_o, N_1, N_2, \dots, N_n, \}$, which can operate coherently as a single system. Depending on the memory system design, it is called (1) a parallel system with shared memory resources or (2) a distributed system with decentralized/distributed memory resources in each system node N_i , as illustrated in Figure 1.

The systems can be designed for specific purposes or as **generic mechanisms** for multi-purpose implementations. Examples of this are: (1) Distributed computing system, which can be a cluster computing system or a grid computing system; (2) Distributed information system for a transaction-processing system (mainly database applications) or enterprise applications; and (3) Distributed pervasive systems with mobile and embedded computing devices. This category can include wireless nodes as networking devices for low latency communication, such as emerging 1/2/3/4/5/6G communication and networking technologies [14].

Features for networking and communication technologies and current research challenges for distributed systems will be discussed in Chapter. III, but we can already say that emerging communication and networking technologies together with system abstraction approaches enable us to categorize that as a fog layer with novel holistic view approaches [5], with a feedback controller mechanism. Some examples of emerging wireless communication technologies such as 5/6G can be defined as a network component for distributed pervasive systems with a set of interacting nodes $N_i \ldots n \{\}$, which have very low latencies for real/near real time critical systems. It is a core mechanism for emerging Software Defined Networking (SDN) and virtualized network functionalities (NFV), as well as for the growing intelligent systems.

The concept of trust is very subjective, having been used by many researchers in many domains for different purposes. The generic definition of trust is as follows:

"trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent will perform a particular action, both before [we] can monitor such action (or independently of his capacity of ever to be able to monitor it) and in a context in which it affects [our] own action".

Trust can then be defined as the belief that a rational entity will resist malicious manipulation or that a passionate entity will behave without malicious intent [40].

If we look at the distributed systems from a service point of view, the emerging digital environments and infrastructures, such as distributed security and computing services, have together generated new means of communication, information-sharing, and resource

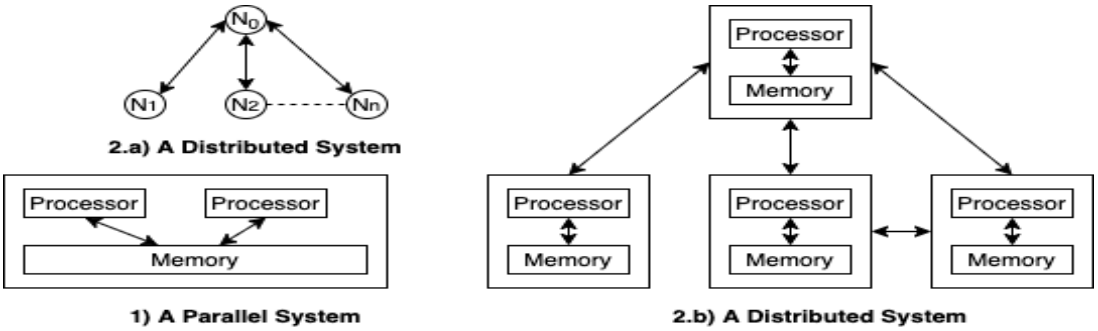

Figure 1. Parallel and Distributed Systems.

utilization. However, using these distributed services results in the challenge of how to trust service providers to not violate security requirements, whether in isolation or jointly. Answering this question is crucial for designing trusted distributed systems and selecting trusted service providers [41].

When designing distributed systems, trust has to be considered as a major factor in all development stages. Therefore, the trust-based design and development would need a framework to guide system developers towards identifying a set of comprehensive requirements and simultaneously preventing any possible conflicts [42]. These conflicts are observed via layering and logical operations with a multi-layer design principle and paradigm approach, as illustrated in Figure 2, which interacts via data between the layers. Thereby, data can be the key components to track the states and critical knowledge regarding the required framework.

In [5], authors propose the Markov chain Monte Carlo (MCMC) method, which can be analytically considered as an inference problem, i.e. computing the posterior distribution via prior distribution information. Given a dataset $D = \{x_1, x_2, \dots, x_N\}$, the posterior probability of $P(x^*|D)$ for the excess state x^* can be calculated using the Bayesian Rule with a probabilistic distribution. Knowing the probability distribution of the initial state $P(x^0)$ and the transition kernels $T(x^* \leftarrow x)$, the marginal probability of the Markov chain at the specific state x^* is computed dynamically. If the prior states are likely to link to the posterior states, then, the Markovian chain is ergodic and converges to an invariant distribution. So that, trust can be transferred between the contexts. This approach can be considered for a dynamic context with a rational agent function to obtain a well-defined context to elaborate on the challenges. So that, trust can be measured with initial assumption and a value assignment to the system node between 0-1. Depending on the fulfillment of required expectation in the observed environment E{N[*]} the values are adjusted dynamically within the observed time span. The more features are justified in the context the more trust level obtained. Furthermore, it can be utilized to ensure the interactivity within the observed context. Thereby, TDAI taxonomy can utilize the approach to adjust trust level of the context dynamically. Each system node it is associated

the trust features on top of which values are via some probes (using for instance automated or semi-automated processes). These values are aggregated and normalized in a way to get a trust value between 1 and 5 (TDAI Trust levels).

In order to formulate interactions with the environment within the well-defined dynamic context, the behavior of a given node can be described as a dynamic system node in the environment E {}, which produces a sequence of states or snapshots of that environment. A performance measurement U() evaluates this sequence, where performance is dependent of the set of nodes in the observed environment N_E {N_1, N_2, N_3, …, N_n}.

Let $V(f, \boldsymbol{E}, U)$ denote the expected utility according to U () of the agent function f () operating in E {}. Each Environment has a set of nodes, N_E {N_1, N_2, N_3, …, N_n} and can be monitored with a set of trusted agents or nodes. Each node can be defined as a trusted agent, which can be defined as system nodes depending on their context. We can identify the rational agent with a function as follows:

$$f_{opt} = arg \max_{f} V(f, \boldsymbol{E}, U) \quad (1).$$

Throughput of each Node X(N) is monitored via trusted Agent A {} as well as via other nodes N {}. A trusted Agent is formulated as follows:

$$A\ \{\} = \{\ iN_i \text{ and with activation function } a_i\ \} (2).$$

The goal of the set of agents A {} and nodes N {} is to maximize the expected utility V () of the set of environments E {} by monitoring behaviors with $f_{opt}()$ function via trusted channels. The interactions with the environment and identified trust **justification features** can be observed dynamically with (1) centralized, (2) decentralized and (3) distributed system design paradigms within an architectural design perspective. Thereby, the trust factor of the system, $P(x^*) \propto t$ with the set of nodes; $N_{E\{\}}$:{N_1, N_2, N_3, …, N_n} can be aggregated in order to **maximize** the throughput in the well-defined dynamic context.

Chapter IV introduces detailed formal statement of TDAI, in this chapter we focus on background analysis with the initial statements.

The dynamic context and environment in which the set of nodes E{N[*]} interacts, require an **optimal** level of trust in the context in order to be able to ensure the interactivity of the nodes and system components. Depending on this level of trust, the system can be identified as a trusted distributed system with the hard constraints of a critical system in real time. This set of features can be mainly identified and measured with dynamic metrics such as the latency, throughput, and power values of the nodes. The values have to be set up accurately for the context dependencies and adapted dynamically to the changing context. Next chapter introduces the background of these AI principles and paradigms with a comparative analysis on centralized and distributed perspectives.

B. Centralized AI Vs Distributed AI

Artificial/computational intelligence has been described from many aspects in literature. The main challenge is finding the abstract and numeric definitions of thinking, learning, and intelligence. In this chapter, we will provide the major definitions of machine intelligence, computing, and AI, in order to emphasize the roots of our conceptual and abstract basic definition for trusted AI mechanisms available in the literature. This chapter articulates and discusses state-of-the-art conceptual definitions of artificial intelligence.

In spite of not having a standard definition for artificial intelligence, most accepted definitions can be categorized into four main groups. (1) behavior, acting humanly; (2) thought processes and reasoning, thinking humanly, cognitive modeling; (3) success measurement respective to human performance, thinking rationally; and (4) the ideal performance measure, rationality; acting rationally is a combination of mathematics and engineering [13]. The remainder of the chapter briefs on the four main categories and introduces the state-of-the-art definitions. Current challenges will be addressed with a comparative analysis of the state of the art.

The first category is behavior, acting humanly and is initiated by the Turing test approach. Natural language processing methods enable computers to communicate with other computers like humans. A computer passes the test if a human interrogator cannot differentiate whether the sender of the message is a human or machine. Knowledge representation stores heard or known data. Automated reasoning uses stored information to answer questions and inference new conclusions.

Machine learning adapts a system to new contexts and detects/extrapolates patterns. There is no direct physical interaction between the computer and the interrogator, since the physical simulation of a person is unnecessary for intelligence. A video signal is included to test the subject's perceptual abilities and pass physical objects through a hatch. Computers need computer vision to perceive objects and robotics that manipulate/move objects in order to be able to pass the test. AI researchers prefer studying the underlying principles of intelligence rather than duplicating exemplary scenarios. Therefore, little effort is needed to pass the Turing test.

The second category is the thought processes and reasoning, thinking humanly, cognitive modeling approach. Cognitive science merges computer models from AI and experimental methods from psychology to imitate the human mind. Each field is growing rapidly and fertilizing the other. One of the most popular definitions of intelligence is the ability to adapt to change (Hawking, 1992), which has inspired most AI systems.

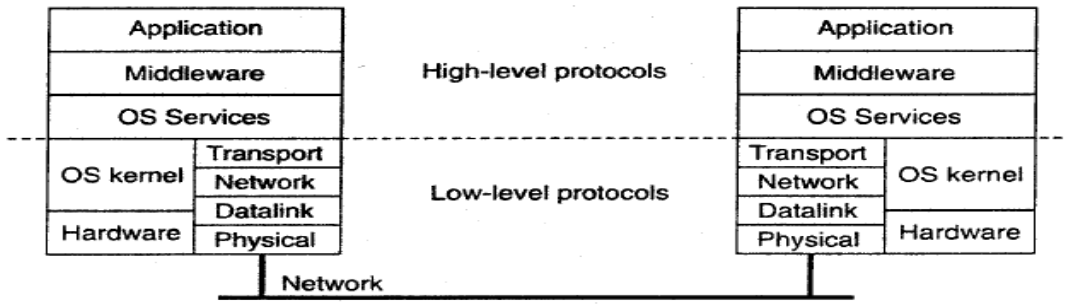

Figure 2. Layering and logical operations of a distributed system [14].

Neuropsychological evidence supports computer vision to develop innovative computational models. However, the systems used in real life have mission/safety/operational critical system constraints. Rational and formally proven methods are preferred by AI researchers.

The third category is rational thinking, success measurement with respect to human performance. Logicians develop precise notations for statements about all kinds of objects in the world and relationships among them. Logic-based computational reasoning systems are applicable to some extent. However, formalizing and stating informal knowledge in formal terms with uncertainty factors is not an approach that is fully applicable. Furthermore, an insufficient number of facts make the use of problem-solving methods impossible and would exhaust computational resources. Reasoning steps can be added to increase the performance of a computational reasoning system, but it would remain limited due to uncertainty and informal knowledge resources.

The fourth category is the acting rationally, rational agent approach, a combination of mathematics and engineering, based on an ideal performance measure known as rationality. A computer agent operates autonomously, perceives the environment, persists in a defined time period, adapts to change, reasons logically, and generates and pursues goals. A rational agent operates/acts under uncertain conditions to achieve the best expected outcome. All skills are required for the Turing test enable agent to act rationally. Knowledge representation and reasoning skills enable agents to reach good decisions. Comprehensible sentences in natural language need to be generated to communicate with the environment.

Continuous learning is needed to improve the ability to generate effective behavior with the agent function $f_{opt}()$. This category of AI can enable us to obtain a mathematically well-defined context to interact with the environment. In this way, we can extend a definition for trust to AI systems as illustrated in Figure 3, where we have a dynamic context and where we see the AI based categories and trust impact. For instance, in IV part of the figure we can have mathematically well-defined context, where we can extend, quantify and qualify the trust with precise definitions of rationality principles.

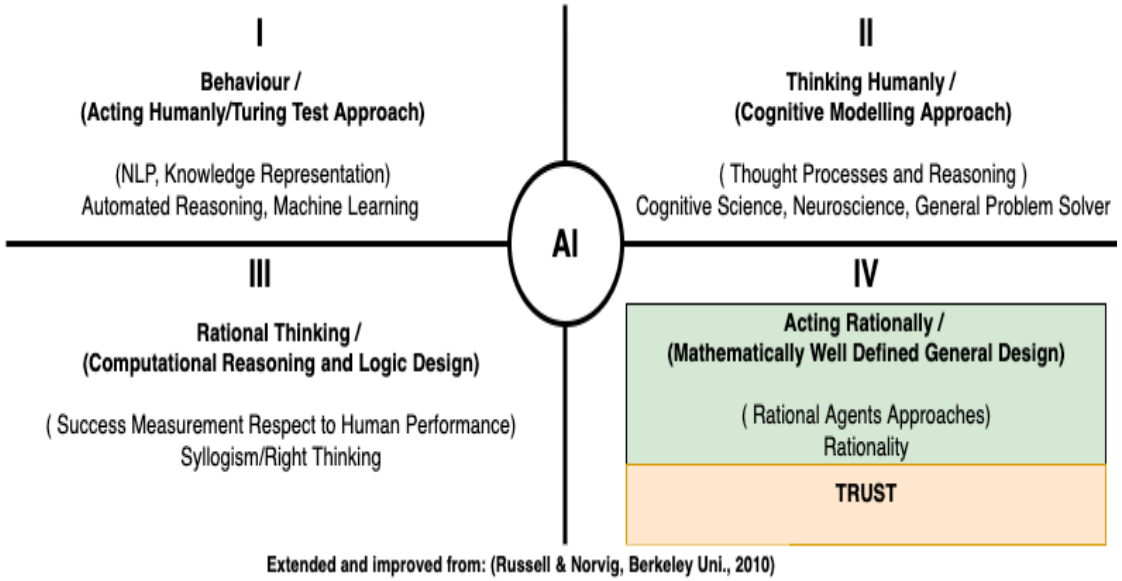

Figure 3. Main categories of Artificial Intelligence.

Based on the comprehensive view of AI system methodologies, we can see that the rational agent approach is preferred by AI researchers. The standard of rationality is mathematically well-defined and completely general. It can enable an agent to be generated for any well-defined context. Achieving perfect rationality and always doing the right thing is not feasible with the uncertainty factors intensive environments. Computational requirements cannot be satisfied in the context. A computer system that has (1) storage, (2) an executive unit, (3) control units does not have to be a central mechanism. Decentralized and distributed system design approaches can enable us to get closer to achieving perfect rationality.

Architectural views and perspectives can be used to differentiate and categorize the features of emerging AI systems. The categories can be named as (1) centralized AI (2) decentralized/autonomous/embedded AI and (3) distributed/hybrid AI. The centralized approach is not considered feasible with the current state-of-the-art approaches, since the emerging intelligent environments are data-intensive and they have limited agent cooperation interactivity features due to the bandwidth limits of interaction channels. Thereby, the number of nodes in the limited context inflates exponentially. The agent functions $f_{opt}()$ also grow exponentially and the systems exceed the limits of computational scalability [5].

As the second alternative, decentralized design features can help to control the independent nodes with limited capabilities of the autonomous agents, mainly with limited knowledge storing and processing features to make the right decision in uncertainty-intensive contexts and environments. However, decentralized design is also limited due to the capacities of the independent node, which is only feasible for a well-defined limited context.

Fortunately, distributed design paradigms can help to merge critical feature sets of centralized and decentralized design paradigms within a hybrid design approach for cooperation and interaction with the agent function $f_{opt}()$ under uncertainty-intensive conditions. Thereby, we can define the distributed AI as the ability to make the right decision in a mathematically well-defined context within the critical, distributed, autonomous system constraints [15]. The main difference between centralized and distributed approaches is the dynamic data-driven cooperation with a set of nodes; $N_E : \{N_1, N_2, N_3, \ldots, N_n\}$ in a set of dynamic environments E{} to achieve the expected utility V().

As cooperation between the nodes increases, system level trust becomes a more critical requirement to ensure the behavioral integrity of a system. The distributed design approach can enable us to maximize the critical feature sets (memory, storage, processing capacities etc.) of distributed AI/ML algorithms for the intelligent systems and mechanisms targeted. The following chapter discusses these feature sets in a dynamic context, where we introduce the need and increasing interest for Trusted Distributed AI in the literature as the core generic mechanism of the emerging trusted distributed systems. In this way, the agent function $f_{opt}()$ can dynamically control distributed resources to maximize the performance of the expected utility V (). By this means, the system can cover wider contexts and spectrums with the distributed features of trusted distributed AI, as explained in the next chapter.

C. Trusted AI Vs Trusted Distributed AI

In a broader context, Trusted Distributed AI (TDAI) can be defined as the ability to make the right decision in a mathematically well-defined context within the critical, distributed, autonomous system constraints. The constraints can be observed with agent function $f_{opt}()$ to reach ultimate rationality in uncertainty-intensive environments. In order to identify these features, the rest of the chapter introduces basic definitions of (1) distributed systems, (2) security, privacy, and trust, (3) distributed AI and multi-agent systems, (4) end-to-end paradigms, and swarm mechanisms to maximize cooperation between the agents and thus maximize the performance measure U () in a set of environments E {}.

Table 2 introduces selected critical feature set comparisons between trusted AI and Trusted Distributed AI with the architectural perspectives. The rest of the chapter explains the key features of TDAI and its advantages with decentral and hybrid design approaches, which enables us to maximize the performance of the agent function $f_{opt}()$ and overall system.

C.I. Security, Privacy and Trust Features

Security, privacy, and trust are the key elements of growing intelligent distributed systems. The scientific principles and paradigms are investigated in all design lifecycles with hardware/software co-design approaches. These features can make systems more flexible and undertake the necessary configurations to tackle the challenges of hardware dependencies. Scientific views and challenges can be categorized into many perspectives, such as the authors [14] roughly divide the issues of security in a distributed system into two parts. (1) concerns the communication between users or processes, possibly residing on different machines that have secure communication channel mechanisms. The mechanisms are more specifically designed for authentication, message integrity, and confidentiality. (2) concerns authorization, which deals with ensuring access rights to the resources with an access control mechanism. The mechanism can manage the user access level, system node confidentiality classification, and data protection policies with cryptographic keys and

certificates. The Table 2 below categorizes these most critical features with (1) centralized (2) decentralized/autonomous/embedded (3) distributed/hybrid design perspective, for which the values are assigned based on the background analyzes in this chapter.

In a broader context, the security of a computer system is strongly related to the notion of dependability, which means that the computer system must have justifiable trust to deliver its services. Dependability includes availability, reliability/liability, safety, maintainability, and robustness. Furthermore, recently emerging concepts like anti-fragility can also be a notion of the resilience and dependence of the system. The authors include the confidentiality and integrity of the computer system as a prerequisite of trust.

Confidentiality feature can be ensured by the security mechanisms in some manner with a layered logical and security mechanism. However, the integrity and coherency of the system require holistic views and end-to-end transaction monitoring approaches within the limits of critical system constraints [15]. In this way, alterations and state changes can be detected and rectified in real or near real time at massive scale. By this means, Alice can trust the computer system and interact with Bob via trusted channels in real or near real time. Trust features, metrics, and measurement/quantification approaches will be discussed in detail in Chapter. III after the brief background definitions of end-to-end paradigms and swarm mechanism feature sets in the next chapter.

ARCHITECTURE	Trust Measurement and Quantification	Trusted Scalability	Trust Assurance	Swarm Manipulation	System and User Behaviour Monitoring
Decentralized (Autonomous/ Embedded/Local)	✕	✕	✕	✕	☑
Centralized/ (Fully connected)	✕	☑	☑	✕	☑ Limited with end-to-end latencies.
Distributed (Edge/Hybrid/Hierarchical/Multi-layer)	✕	☑	☑	☑	✕
Trusted Distributed AI	☑	☑	☑	☑	☑

Table 2. Artificial Intelligence System state-of-the-art targeted feature summary (☑: Yes, ✕: No).

C.II. End-To-End Paradigm and Swarm Mechanisms

The data-intensive nature of emerging AI systems and context-dependent programs makes the problem much more complicated due to the increasing complexities of the transactions. Nevertheless, a generalization approach is possible. Distributed caching policies and system abstractions have recently been tested. Performance improvements are observed with distributed file systems and different configurations for memory bottlenecks and congestions as an improvement to the Turing and McCarthy abstraction models [29,3]. The studies prove that the end-to-end implementations of machine-learning pipelines with modern cloud systems, which have browser-based interface architectures, can be implemented in real time or near real-time. The authors define the diversity of emerging data growth as big data concept. It is 3V (Volume, Variety, Velocity) data, which cannot be processed with classical database systems.

A proof-of-concept study was experimented with basic machine-learning use cases for an opinion-mining application to understand social polarization and convergence features. The proposed distributed file system-based design enables us to overcome the memory bottleneck with a 90% true clustering performance [29] for designed scoring algorithms.

System-level innovations and new conceptual definitions and abstractions have enabled us to develop advanced computational systems to automate many manual processes. Thereby, trusted distributed AI methodologies can be implemented with end-to-end machine learning pipelines and trusted execution of transactions with holistic views to the total system. Baydin et al. [1] propose automatic differentiation for machine-learning applications to build end-to-end pipelines. The approach can enable end-to-end machine-learning models/knowledge bases to be merged and trained in different contexts. Emerging AI systems and computational/storage resources can support the end-to-end design of AI systems. Data can be managed and fused with knowledge bases within reasonable latency thresholds for many applications to keep the rationality of the agents in a well-defined dynamic context [30].

Machine learning and statistical techniques can help to transform big data into actionable knowledge with a simple user-interface via an efficient distributed system design approach [2]. End-to-end differentiable pipelining frameworks can support the automatic composition of a learning framework within acceptable latency thresholds [3]. The innovations enable cooperation between diverse contexts for the tasks, and trigger novel trust modeling approaches. Cohen et al. [4] propose a multi-agent-based trust model to be able to ensure the expected behaviors of system units. In order to increase cooperation between system-level transactions, swarm-based coherence is proposed as a collective adaptation for swarm intelligence with artificial neural networks [31].

The emerging technologies associated with swarm mechanisms enable trusted distributed AI to be developed, with an increase in processing capacity together with the distribution/decentralization of resources (data units, AI processes, etc.). Real-time management/exploitation of such systems and consideration of them from a holistic point of view becomes much more critical. [5] is an example of holistic system abstraction proposed for end-to-end transaction flow monitoring of trusted AI systems. In addition to this, some research work focuses on transaction management considering the X-AI concepts together with the lineage aspects (data locality and tracking) [32,33,34,35]. In terms of the development and engineering of AI-based systems, cloud-based lifecycle-based trust modeling and monitoring approaches are also proposed by the authors [8,9].

As a brief overview to the explored background, the challenges can be categorized into three main architectural design views with a system level perspective. (1) Decentralized (Autonomous/Embedded/Local) (2) Centralized/ Fully connected (3) Distributed (Edge/Hybrid/Hierarchical/Multi-layer). Within these, the interactivity and cooperation of the agents and dynamic system components can be observed in the dynamic context within a holistic point of view.

The features in the literature targeting trustworthy mechanisms for either centralized AI or Distributed AI can be summarized as illustrated in Table.2. These categories and main feature sets can be listed as follows:

- trust measurement, quantification, and justification
- trusted scalability
- trust assurance
- swarm manipulation
- system and user/agent behavior monitoring

D. Conclusion

We can conclude that there is little research work that is related to the field of TDAI. Indeed, this research field requires all five key features dedicated to pure distributed AI-driven systems, as mentioned above, to be considered.

All these indicators may require simulation tasks to understand and measure the ability of a distributed system solution to solve complex (near) real-time mobility problems compared to conventional centralized approaches. Chapter IV content illustrates an intelligence flow mechanism in which the learning and growth is correlated with an end-to-end trust mechanism. Thereby, the high-level monitoring dashboards of the intelligent systems can have a holistic view of the growing mechanisms in a dynamic context. Chapter IV describes the novel feature sets of TDAI with a use-case focus for the growing intelligent systems. It comparatively analyses the state of the art for the identified principles and paradigms and introduces the emerging challenges and potentials in detail.

Chapter III: Research Challenges and Related Work

Introduction

This chapter presents the following:

- AI system categorization covering its architecture perspectives, networking and communication features as well as its end to end trust mechanism justification features and indicators.
- Comparative Matrix on the related work results.

A. AI System Categorization

Continuously growing intelligent systems can enable massive-scale AI support for many critical systems. However, challenges also increase, mainly in terms of complexity, inflation of size/volume, reaching limits of resource centralization, and an increased need for decentralized/distributed mechanisms due to non-deterministic alterations and uncertainties in system components. In this chapter, we will discuss related studies, which categorize the challenges and introduce the main features to be targeted for a trusted distributed AI methodology as a core mechanism of growing intelligent systems as illustrated in Figure 4.

Data is the most valuable digital dynamic asset of the intelligent systems. Since computing machines have existed, the most interesting challenge has been to tackle the computational complexities in a timely manner and access the system resources with the right credentials. [42] Proper identifies the current state of the data as fuel for the digital age. Business analytics, statistics-based AI, digital twins, etc. are defined as “data-hungry” applications, which are components of complex systems, and which can be thought of as data ecosystems. The research challenges below are defined as the main categories:

- Data as a key resource
- Trust at the core
- Regulation of data ecosystems
- Data need semantics
- From data to information

The challenges within the identified categories can help define the role of data in the current context of smart systems. However, data is not fully separated concern from computation, it has to be mapped to computation. Trust has to be measured and quantified. Novel holistic system abstractions are required to track the transaction flow at the system level and to assign trust values to each category. Furthermore, semantic web-based ontology modeling approaches with RDF (Resource Description Framework, Subject-Predicate-Object) [43,44], scenario-based strategy planning tools [45], or any other system design tools can help to model a lifecycle with a conceptual modeling perspective to better interact with intelligent system components at run time or (near) real time. Thereby, explainability and justifiability features with socio-dynamical perspectives can also be tracked more coherently to contribute to the continuous growth of the emerging intelligent mechanisms.

Within the digital-dynamics perspective, system-level end-to-end transaction monitoring can be succeeded by a holistic view [5], which enables data-state and lineage tracking in a trusted manner with a robust core mechanism. System-level trust features, metrics, and measurement approaches will be discussed later in this chapter to indicate the justification of trust prerequisites, such as robustness, reliability/liability, resiliency, and integrity.

The digitization of everyday life, cause the amount of data to grow exponentially, and the challenges emerging from this have made the need for system reconfigurability more critical. Hardware-dependent designs are replaced with software-driven mechanism and hardware/software co-design approaches are utilized when necessary. The software-driven approaches also adapted the software challenges to the current intelligent system context with software-intensive mechanisms. [46] Aksit has summarized these challenges/research directions into six categories with a focus on smart-city systems and presents them in a single list as briefed below;

1. Developing models for smart cities;
2. Designing a framework for managing and optimizing the configurations of clusters;
3. Designing models, methods and tools for critical infrastructures;

4. Optimizing the necessary quality attributes through system adaptation at run-time;
5. Integrating software systems;
6. Designing a smart infrastructure with a high degree of interoperability, configurability, adaptability, and evolvability.

The challenges can help to synchronize coherency between the related research studies. However, new software and hardware co-design principles are emerging. System-level hardware/software integrated views, which can interact with all verticals at run-times, are required for emerging smart city systems. Trust is not only required for dependability, which already ensures the security, robustness, resilience, integrity, and coherency of a system [46], but must be measured and quantified for smart systems, in order to inspire confidence in system architectural level disruptive innovations.

The next chapters will identify these architectural design differences to emphasize the need for distributed design and the potential benefits of hybrid mechanisms. Thereby, we will be able to introduce the methodology to be used for the concept of SDN to ensure the interactivity of trusted agents in near/real-time for smart systems. Hybrid approaches also define a core mechanism for emerging networking/communication methodologies for close-to-long-range systems as the generic IT core, which can be implemented in emerging software intensive systems, such as 5/6G. In order to be able to focus the identified features on the TDAI, these system-level paradigms can be categorized as architectural (1) and (2) networking/communication perspectives. Thereby, we can obtain the trust justification features of novel computing systems with a focus on distributed computing concerns for the targeted trust frameworks. Rest of the chapter introduces these identified system-level features and explains them as the trust-justification features of emerging AI systems.

A.I. Architecture

Architectural modeling and the models are the basic methodology and critical feature for system-design paradigms. These are mainly considered with hardware and software-level design concerns. The approaches can be limited to board-level architecture-design paradigms for computing and intelligent system mechanisms [47]. Chip-level designs can enable us to implement computing facilities on any system components as an integrated unit, such as edge devices and mobile units. However, increasing amounts of the data manipulated by the systems require major updates of the hardware and software abstraction principles.

Existing approaches can enable us to process and manipulate data with virtualization and caching policies [48]. Chip-level interconnection [49] mechanisms can enable us to transfer the data between the processes with available scheduling policies [50]. These design approaches are limited with 3D-Stack board/memory design principles [51] and network interconnection issues [52], such as scalability, performance, energy consumptions, and most of all, with bandwidths of the transmission and buffering channels.

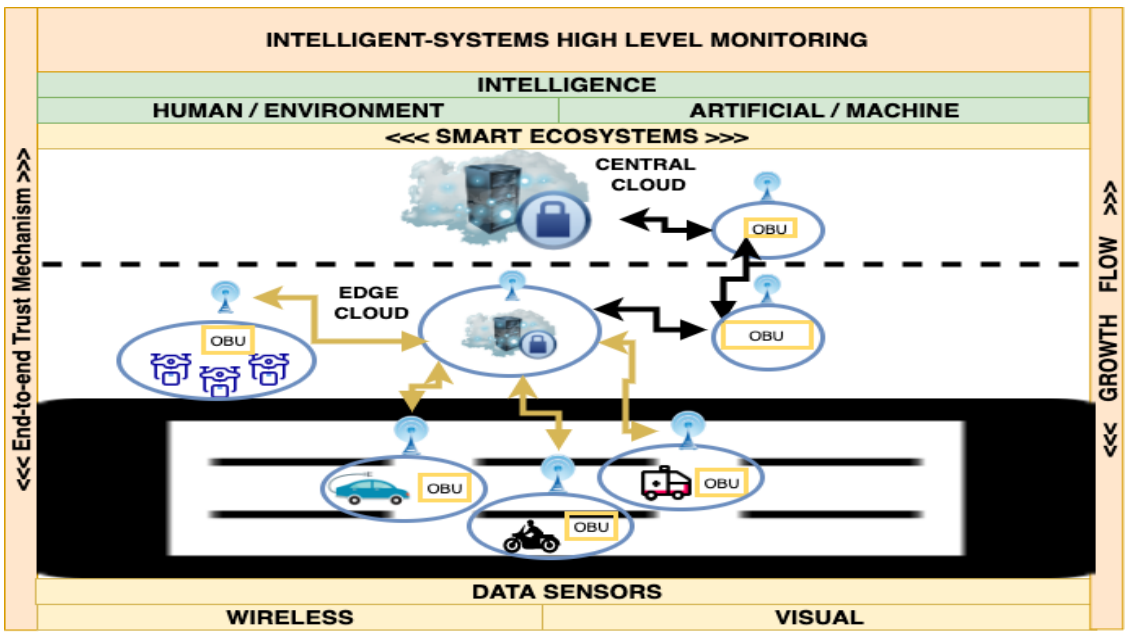

Figure 4. Intelligent System Intelligence Flow Mechanism.

Emerging intelligent computing architectures [53] can enable us to process and manipulate data with more intelligent approaches. For instance, caching performances [54] can be optimized and streaming buffers can be designed more coherently and adaptively [55] to interact more efficiently with the system components. The interactions can be designed with online approaches with an event-based trigger mechanism [55] with data-center networking [56] protocols. The architectural concerns can be modeled as generic core mechanisms with the holistic abstraction and end-to-end system design paradigms with throughput maximization approaches [5].

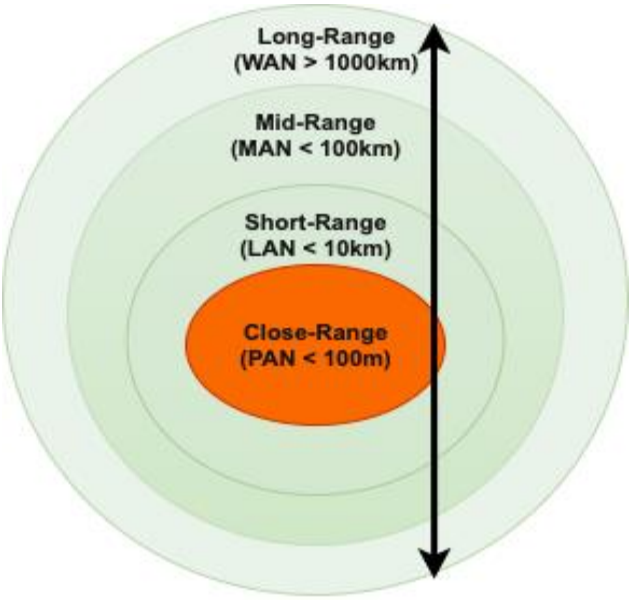

Figure 5. Networking and communication systems range-based categorization.

Interactions with the environment and the dynamic state changes of the components also trigger behavioral complexities. These require the dynamic modeling of system view points and snapshots of the states, and can be succeeded by available system engineering architectural frameworks [57] up to the specific requirements of the dynamic context changes. In addition, business processes can also be modeled conceptually [58] to interact more efficiently with the environment. Therefore, system resource modeling and management paradigms can be improved with novel learning heuristics [59]. Furthermore, system resource-management knowledge bases can be trained for continuous growth and the heuristics can be improved with holistic abstraction [5] paradigms. These architectural features can be categorized into three main groups with centralized (fully connected), decentralized (autonomous/embedded/local), and distributed (edge/hybrid/hierarchical/multi-layer) system design approaches.

- **Centralized/** (Fully connected)**:** Processing and memory resources are fully centralized

Centralized architecture can enable to build smart systems with intelligent mechanisms, which have centralized processing and memory/storage components. Furthermore, robust networking/communication channel-based abilities are supported with strong back-end units, such as quantum computing mechanism, to interact with the environment and dynamic context efficiently. However, these architectural design paradigms are limited by system integration and performance issues [60]. Fortunately, parallelizable portions of these issues can be identified with trust factor maximization principles, and integrity concerns can be minimized with holistic interfaces [61], and holistic total system throughput maximization methodologies [5]. Nevertheless, the design/manufacturing costs and physical limits of these designs require decentralized and distributed approaches, which are explained briefly in the next chapters.

- **Decentral/** (Autonomous/Embedded/Local): Processing and memory resources are fully de-centralized

The decentralization of the mechanisms requires trusted computing units [37] on edge devices and secure channels for trusted interactions with the environment. The trust constraints can be ensured to a certain extent, but the edge/mobile components are limited by digital design paradigms and require novel holistic interfaces [61]. 3D-stack digital design technologies can help to improve edge/mobile units as densified system components [15,62]. These features can dynamically adapt to the context changes with the holistic interface's digital design approaches [61] and end-to-end holistic abstraction and views [14,5]. Thereby, the available features of edge/mobile devices can be maximized with densified design paradigms, and the overall system performance can be improved with a distributed design approach. Next chapter introduces the basics of distributed design and details are discussed in a comparative matrix in Chapter. IV C.

- **Distributed/** (Edge/Hybrid/Hierarchical/Multi-layer): Processing and memory resources are distributed

Distributed system design issues can be grouped into (1) algorithm-level and (2) system-level concerns with a focus on distributed computing [14] principles and paradigms. Algorithm-level challenges include learning paradigms with statistical and AI/ML optimization approaches, such as learning cardinality estimator performance maximization [63] with a flow loss model, a source code compiler to minimize algorithmic complexities [64], and other AI/ML challenges to automate the data pipelines of training and test data-sets [65]. One of the most critical concerns of these challenges, estimator performance maximization, can be improved with holistic abstraction paradigms, which can enable the dynamic training of multi-layer data models [5].
However, increasing complexities and uncertainties of the algorithms trigger more system-level concerns and require computational tractability of the processes and transactions. For instance, critical database ACID (Atomicity, Consistency, Integrity, Durability) features need to be extended to edge devices in a trusted, scalable manner [5]. These challenges require updates in logic design and hardware level updates within the polynomial time threshold values to minimize latency concerns [66]. Holistic interfaces can help to design reconfigurable hardware with a dynamic end-to-end logic structure [61]. However, middleware design paradigms are also critical for software/hardware co-design issues [5,62]. Fortunately, rational verification methods in polynomial time [66] can help to improve the transaction flow to enable the computational tractability of the processes.

Behavioral strategies of these mechanisms can also be dynamically configured [67] to improve the intelligence mechanism of the intelligent systems. Therefore, AI systems can be improved with novel methodologies, so that we are able to mention trust on these systems, which have more opportunities and challenges than ML features [68], due to their behavioral integrity constraints [5,62]. Intelligent agents [69] are key components for these intelligently behaving smart systems, and trust measurement and maximization with swarming approaches are promising indicators and features of these paradigms [15]. Comparative analysis matrix in Chapter. IV B summarizes potential research directions.

Discussions in the next chapter will be limited to networking and communication perspectives in order to observe the methodologies that can maximize critical features of an intelligent system with the set of nodes $N_i : \{N_o, N_1, N_2, \dots, N_n, \}$, such as connectivity and interactivity paradigms.

A.II. Networking and Communication

As the emerging technologies grow faster, intersections between the fields are also increasing and multi-disciplinary fields are converging with AI systems-driven design paradigms. For example, networking and communication technologies are improved, with the critical features of emerging AI systems and novel functionalities are being enabled with software-driven design paradigms, such as NFV (Network Function Virtualization) and SDN (Software Defined Networking). In this way, emerging networking and communication technologies like 5/6G technologies can enable massive-scale AI System deployments to be implemented with novel hardware/software codesign paradigms, and the challenges can be explored with AI systems perspectives. The rest of the chapter explores these critical features with a range-based categorization approach as illustrated in Figure 5**.**

Networking and communication features, which trigger growth acceleration in the computing paradigms and the densified computing/storage system units, can enable distributed massive data to be processed in real time and help to ensure the connectivity and interactivity of the components within the critical system constraints. These features, which are the key enablers for the SDN mechanisms with virtualized network functionalities, are called NFV. Thereby, the connected mechanisms can help to design software-driven dynamic systems rather than hardware-dependent designs to make the networking and communication systems flexible and adaptive to dynamic context changes. In doing this, the critical networking and communication components of emerging technologies can enable us to ensure the interactivity of the agents within hybrid-cloud mechanisms [49] and Radio-Network technologies, which have multi-layer design with network slicing features [56].

Standard definitions are still on progress of improving signal transmission and edge/mobile processing latencies to meet the critical system constraints [57]. In spite of making good progress with these challenges, distributed computing and system design paradigms [14,15] still need to be investigated and tested with the critical feature sets of the emerging

networking and communication systems. These features are mainly categorized into two groups: (1) beamforming and signal transmission and (2) edge/mobile processing mechanism for networking mechanisms and systems. Figure 6 [72] illustrates the basic characteristics of wireless features, with GHz frequencies and advanced emerging features, which are mmWave and THz waveforms. These signal transmission abilities can help to improve the interactivity of system nodes and edge/mobile units within the limits of total system throughput principles and paradigms [5]. Furthermore, available networking protocols and mechanisms can enable the transmission and processing of [86] packages with multi-layer connectivity paradigms, as illustrated in Figure 7. Transmission latencies and edge/mobile package processing features are promising for ensuring the interactivity and connectivity of an intelligent system with a set of nodes $N_i : \{N_o, N_1, N_2, \dots, N_n, \}$. Detailed features are discussed with a focus on end-to-end trust mechanism justification features and indicators. In this chapter, we will limit the discussion to critical networking and communication features with a range-based categorization approach as illustrated in Figure 5, consist of (1) Close (2) short (3) mid (4) long ranges.

- **Close-Range (PAN < 100m):** Bluetooth, Wi-Fi, 802.11p/ITS G5 for V2X, low latency networks etc.

An increasing number of networking and communication technologies can provide a wide variety of options for the connectivity and interactivity maximization of the system nodes and edge/mobile units. However, the growth and diversity of options increase complexity and trigger behavioral anomalies, which require (near)-real-time channel selection mechanisms.

Fortunately, intelligent control mechanisms can enable us to design vision-based control mechanisms [87] or hybrid controllers with a wireless/visual sensor-based [5] control structure inside the edge/mobile units. These challenges trigger the requirement for novel synchronization and concurrency features [88] at the edge/mobile units. Networking and communication services can also be adapted to the dynamic internet/intranet [89] applications. Energy efficiency of these nodes is also a critical feature for the available

wireless/wired communication channels and required wireless communication protocols [90], and wireless sensor network architectures [91] can be adapted to change dynamically. These features can be designed as open-flow mechanisms [92] for a selected region, and adaptive protocols [93] can help to process packages and disseminate information in edge/mobile units with a distributed design approach [93]. Multi-layer topologies [94], such as MAC 802.11 ad hoc protocols, can be maintained dynamically to adapt the physical layers at run-time.

The resiliency of overlay networks [95] is also a critical concern for the edge/mobile device surface [96] signal processing abilities within the critical system constraints. Thereby, advances in edge/mobile device features and abilities like ultrasonic ranging hardware [97], congestion controller mechanisms [98], miniaturized beamforming devices [99] can operate and help to ensure the interactivity and connectivity of the components with ms-scale latency values [100]. On the other hand, these advanced features require hardware-level code management challenges with context-aware computing paradigms. Fortunately, this adaptiveness can be improved with Machine Inferred Code Similarity (MISIM) systems [101], which can be part of a future research challenge in terms of the run-time reconfigurability of the systems. In order to limit discussions on close-range communication in terms of end-to-end trust mechanism and justification features, the rest of the chapter will only mention short-to-long range paradigms briefly and will focus on the identified trusted computing feature sets.

- **Short-Range (LAN < 10km):** Cellular networks, 4G/LTE, 5G NR etc.

Short-range feature sets and end-to-end networking/ communication mechanisms can maximize the connectivity and interactivity of the system components and edge/mobile units in real/near-real time. Figure 7 illustrates these protocol categories, which include link-layer and end-to-end connectivity features. These features need to be extended to short-range cover, especially for smart-city use cases. For instance, emerging mobility technologies like autonomous cars can behave like a mobile computer device, which can help to implement urban air transport with the cars having a dual functionality as mobile computers.

Connectivity and interactivity of these mobile units can be ensured with emerging systems and methods in order to obtain [102] telematic data with wireless/wired sensor tags. These protocols can be adapted dynamically to the dynamic context changes as explained in the previous chapter. The discussions can be limited to the V2X (vehicle to everything) domain to explain the most promising feature sets. For instance, ITS-G5 (IEEE 802.11p) and C-

	Sub-6 GHz	**mmWave** (30 – 100 GHz)	**sub-THz and THz** (0.1 – 10 THz)
Distance	High range	Medium to short range ($\leq$ 200 m)	Short range ($\leq$ 20 m)
Bandwidth	Limited	Medium to Large	Large
Data rates	Limited	Medium to High (up to 10 Gbps)	High (up to 100 Gbps)
Interference	Mitigated by techniques like OFDM and OFDMA	Mitigated by beamforming	Mitigated by sharp pencil beamforming
Noise source	Thermal noise	Thermal noise	Molecular absorption noise and Thermal noise
Blockage	Not susceptible	Susceptible	Highly susceptible
Beamforming	Medium to narrow beams	Narrow beams	Very narrow neams
Horizons to explore	Expanding the midband coverage	NLoS communications and long-range sensing functions	Reliable and low latency communications, integrated sensing and communication systems
Viable architectures	Massive MIMO	Ultra massive MIMO, RIS, and UAV-RIS	Cell-free massive MIMO and holographic intelligent surfaces
Significant caveats	Low rates and spectrum inefficient	Susceptibility to mobility and blockages	Susceptibility to micro-mobility, orientation, air composition and blockages
Applications	Low-rate and latency tolerant services	Vehicular networks, radar, UAVs, and IoT	XR, holography, IoE, NTNs, sensing, and nanosensors

Figure 6. Advance wireless communication systems emerging features [72].

V2X (3GPP Release 14) are promising technologies in terms of sampling/transmission frequencies and power/energy efficiencies [103] with minimized congestion and latency values. Nice progress is saved with 5G hybrid-mechanisms [85], which are supported by hybrid clouds with maximized bandwidth limits to exceed theoretical thresholds like Edholm's law of bandwidth [81]. Distributed computing paradigms are still under investigation to maximize the total system throughput values of the system [5] with novel AI/ML supported designs [32,6]. These feature sets will be summarized in chapter. III C. The rest of the chapter briefs on mid/long-range challenges and potential future research directions.

- **Mid-Range (MAN < 100km)**: High Speed Wireless Internet, cable TV systems

As the amount of data traffic increases to the peta/exa-scale, controller mechanisms become more complicated and require advanced intelligent controllers inside edge/mobile devices with distributed mechanisms to be able to cover wider ranges. The diversity of the components and interaction require advanced package processing features and real-time decision mechanisms. Fortunately, the state-of-the art methodologies can enable distributed sensor computing systems, as illustrated in Figure 8**.** [12]. The detailed feature sets of these mechanisms are explained in the next chapter with a focus on the end-to-end trust features of computing principles and paradigms. As a brief introduction, these features can cover on-board computing units with mobile/edge query processing mechanisms. Therefore, dynamic measurement metrics can be collected to help to ensure the connectivity and interactivity of mobile units with maximized trust values, as explained in the next chapter, as advanced feature sets of the trust mechanism to be able to extend networking and communication features to mid-range.

- **Long-Range (WAN > 1000 km)**: Space Networks, Sat Com, Space Internet, Futuristic (Drones, Low-Orbit Satellite etc.)

The growth in communication technologies and signal transmission features can enable us to reach higher signal frequency transmission features up to PHz scales. Futuristic components of the emerging smart systems, which have higher bandwidths, can ensure the connectivity and interactivity of the components at massive scale within continental scope and low-mid-high orbit space systems. These innovative space missions can cover space internet, space aircrafts, and other advanced high-throughput connectivity mechanisms like 6G and InfiniBand optical systems. These advanced radio signals and mm-to-ultraviolet frequencies are illustrated in Figure 9 [72].

Swarming and end-to-end trust mechanisms can help to maximize the throughput of each node and the total system within the critical system constraints [15] with novel AI-supported distributed computing system designs as the core mechanisms. Main characteristics of these features are summarized in chapter. IV.B within the detailed comparative features matrix. Figure 5 illustrates the categorization of these emerging networking and communication technologies with a range-based classification approach.

Name	Category	Special Mechanisms
E2E	end-to-end	standard TCP-Reno
E2E-NEWRENO	end-to-end	TCP-NewReno
E2E-SMART	end-to-end	SMART-based selective acks
E2E-IETF-SACK	end-to-end	IETF selective acks
E2E-ELN	end-to-end	Explicit Loss Notification (ELN)
E2E-ELN-RXMT	end-to-end	ELN with retransmit on first dupack
LL	link-layer	none
LL-TCP-AWARE	link-layer	duplicate ack suppression
LL-SMART	link-layer	SMART-based selective acks
LL-SMART-TCP-AWARE	link-layer	SMART and duplicate ack suppression
SPLIT	split-connection	none
SPLIT-SMART	split-connection	SMART-based wireless connection

Figure 7. Networking protocols for package transmission [98].

Advanced communication and future networking systems will be considered in future related works. In this thesis, the focus is on distributed computing paradigms and principles of emerging intelligent systems.

A.III. Trust: End-To-End Trust Mechanism Justification Features and Indicators

Trust paradigms are widely explored in technical and human science disciplines. Since our focus is on technical concerns with distributed computing scientific paradigms and communities, the categorization and futures are selected from the computing perspectives of a system with a set of nodes $N_i : \{N_o, N_1, N_2, \dots, N_n, \}$. In this way, the continuous growth acceleration of an intelligent system can be maximized with dynamic feedback structures [5]. These justification features can be categorized into four main groups: (1) Performance, (2) Run-time monitoring, (3) Security, and (4) Test-based features and indicators. This chapter will explain these justification features by using a selection of the main related studies in the literature.

I.Performance

Performance elements are the key metrics for the justification features and defined trust indicators. These can be measured, quantified, and monitored from many perspectives. In order to focus the indicators on distributed computing domains and improve the feedback control structure of the generic mechanism, we can address and focus mainly on the scalability, elasticity, connectivity, and energy efficiency features of the nodes and total system.

Thereby, the rationality and performance features of AI/ML methodologies can adapt to the dynamic context [13] and (near) real-time threshold constraints, and ensure the interactivity of mobile agents. This chapter explains the identified performance elements of these trust justification features.

a. **Scalability, Elasticity, and Connectivity Limits:** Total number of nodes and users in the system

The scalability, elasticity, and connectivity features of a node can be identified as basic features of the performance-measuring approaches in system sciences. These are measured with number of nodes, users, data volume, and other system/algorithm level computational scalability limit metrics [5]. The approach has been widely applied with AI methodologies, such as distributed AI and multi-agents in many industries. For example, telecommunication systems became data intensive and improved with scalable system design paradigms [16]. Thanks to the advances in data-processing technologies, these features can be queried in real time and correlated with knowledge bases of the systems with dynamic holistic views [15]. The challenges will be discussed with a comparative feature analysis matrix in the next chapter.

b. **Energy Efficiency:** Average energy consumption of nodes and critical transactions

Dynamic management of system resources and physical capacity features requires both hardware and the physical layer monitoring of the units in real time. Energy efficiency average consumption value is the key capacity metric for the interactivity features of the mobile agents for the required power constraints. A novel system abstraction approach can enable the physical layer parameter/metric monitoring to be extended to ensure interactivity and adaptivity in near or near real-time [5].

c. **Energy Efficiency:** Average EMF (v/m), SAR(w/kg), Power(W) environment friendliness

Emerging networking/communication systems like 5/6G can enable the implementation of novel features of the AI methodologies, such as real-time massive scale analytics. However, this triggers risks pertaining to human health, include cancer, COVID-19, etc. [73,74]. These challenges can be mainly identified and are rooted in EMF, SAR, and the power features of the nodes. In order to be able to justify trust in a dynamic context, these features have to be monitored in real time with the local/global regulative constraints. Thanks to the holistic view [5] of innovations in emerging computational ecosystems, this can also be achieved within the regulative constraints and it is discussed in the next chapter.

II.Run-time Monitoring

In order to be able to maintain overall performance; dynamically justified trust features, the growth progress of the systems, and other trust indicators have to be monitored continuously. Active and passive systems have different constraints and limits, which trigger diverse challenges in distributed computing paradigms. AI methodology approaches can be improved to satisfy the need of the active system constraints at run-time with dynamic approaches. Trust features and indicators can be guaranteed for machine-learning systems [18] and distributed AI techniques [8]. Programming approaches like

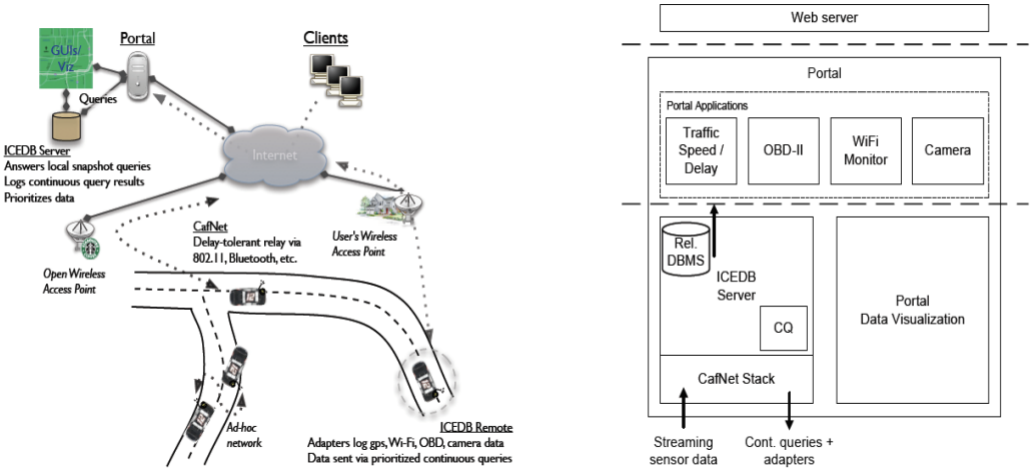

Figure 8. A distributed mobile sensor computing system [12].

probabilistic/concurrent [20], dynamic/differential [18,1] can enable knowledge bases and data states to be updated at run-time coherently. Therefore, the justification features can be trained and updated dynamically for a continuously growing mechanism. We focus these challenges on distributed computing and caching policies in the next chapter discussions. This chapter is a brief on data-state tracking/transitions for an efficient end-to-end feature embedding/manipulation mechanism of a running system.

a. Data-flow monitoring: Data state monitoring between applications

Data is the fuel and most valuable asset for the emerging intelligent systems [42]. It is the critical element of the justification features to ensure the integrity of the mechanisms and systems. Each state-change has to be tracked and manipulated during the whole lifecycle of the data. Emerging AI technologies can improve data challenges [30] with novel end-to-end paradigms and scientific improvements in the field. Improved ML systems can also help to improve knowledge bases and are dynamically generated up to data-state dependencies [2]. However, the training process is not only required for data states, it also has to be mapped to the pipelining [3] and feedback mechanisms of system nodes with trust indicators [4]. The features that can help to justify trust are discussed in the comparative matrix table in Chapter 4. B.

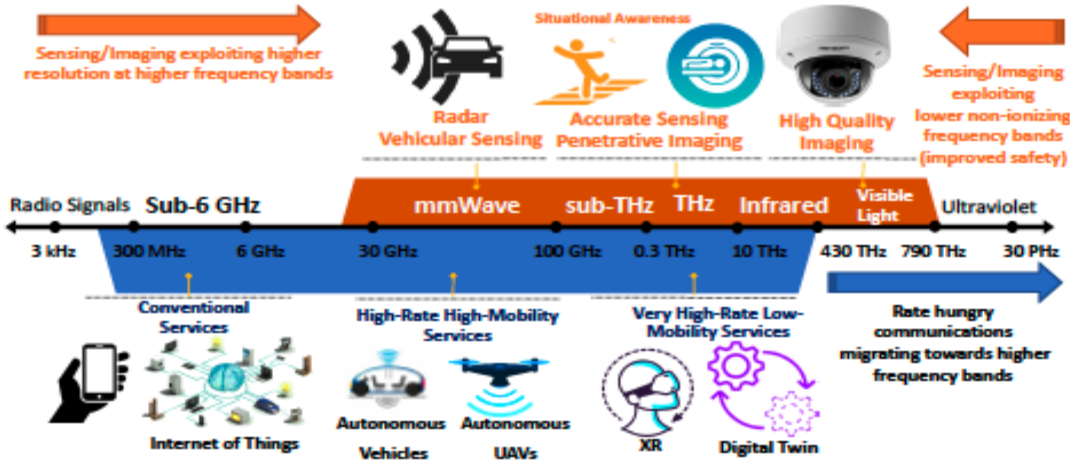

Figure 9. Frequency spectrum paradigm shift by communication and sensing features (3kHz-30PHz) [72].

b. Transaction-Flow Monitoring: Transaction lifecycle monitoring

The diversity and heterogeneity of the emerging systems require decentralized and distributed designs to be able to ensure the growth of the mechanism [30]. Distributed computing paradigms are core features for managing the resources and mapping the data and computation where necessary. Swarm intelligence techniques at the algorithm and system levels can help to resolve the challenges and complexities that trigger swarm behavior in emerging intelligent systems [2]. Novel designs for control structures and abstraction hierarchies [5] can help to embed trust justification features and ensure continuous growth with the necessary updates at runtime with real-time threshold values. Thereby, transaction life-cycle can be monitored dynamically and failures can be recovered with minimum latency via the feedback controllers and holistic views. These features will be discussed in the next chapter.

c. Trust Monitoring: Periodical trust verification

Technical and human science concerns around trust modeling are critical paradigms and features for the justification mechanisms. Our focus will be on technical concerns of trust with distributed computing principles and paradigm challenges. In order to improve the quality attributes with user-level measurement metrics, we can consider the regulative aspects of the trust issues. Emerging trends like explainability features [32] can provide growth acceleration metrics, and these can be improved with lineage-tracking features [15]. Furthermore, these features are strongly dependent on the secure execution of the monitored transactions. In [118], authors explored and improved hardware-based SEE (Secure Execution) with a Trusted Execution Environment (TEE) concept. Storage and user interfaces are identified as critical features and compared with some of the available technologies like ARM TrustZone-based TEEs.

However, these features have (near)real time interactivity constraints with diverse system components. For this reason, hardware isolation and separated kernels are far from achieving these latency, interactivity and scalability thresholds. Fortunately, dynamic

holistic views can help to maximize the trusted scalability of the emerging AI Systems [28,29]. Additionally, holistic abstraction paradigms [5] can also help to measure and quantify trust with a trust factor coefficient-based throughput maximization approach as an extension to Amdahl's notation. Thereby, trust can also be verified periodically with the identified trust justification features. From a system-level perspective, the trustworthiness features of AI/ML principles [6,33,7,34,8,36] can be interpreted as basic structures of feedback and other mechanical/digital control loops for the growing mechanisms [5]. Furthermore, the uncertainty of the harsh conditions [75] can also be measured and quantified at run-time as calibration metrics [15].

In order to obtain measurable and quantifiable metrics for trust concerns, we will keep the focus on regulative features, such as EMF/SAR/power values for health-care limits, privacy of data etc. with local/global perspectives [76,77,10]. These are the critical metrics of the massive-AI system justification features for real-time alerting and risk prediction algorithms [6]. Risk predictions can also improve the scalability limits of large-scale optimization algorithms [10] at run time [5]. Therefore, trust performance and node regulative constraint thresholds can be justified and can help to improve the growth of the mechanism by ensuring the behavioral integrity of the total system. Next chapter summarizes and discusses the scope of the technical concerns with trust measurement and quantification perspectives in distributed computing paradigms with AI/ML pipelining features of growing intelligent systems.

d. AI/ML Pipelining: Dynamic knowledge base monitoring and update

Dynamic contexts, in which mobile agents and system components interact, require real-time updates in different system layers, data models, and most critically, knowledge bases for critical decision-support mechanisms. In order to be able to justify the trust features and indicators, interactivity of mobile agents has to be ensured with distributed computing paradigms and challenges. System acceleration units and algorithm level improvements can be designed for these purposes [11]. In order to be able to manage the system resources dynamically for the changing context parameters, AI/ML pipelining mechanisms can be

designed [78] with novel digital control structures [79]. Swarming approaches are also useful for mission-critical constraints of the growing smart systems. Resiliency, robustness, durability, locality, and anti-fragility features [80, 5] are critical features of the trust justification mechanisms for the trusted computing units [37], which are explained in the next chapter.

e. **Run-time feature embedding and interaction**: Data fetching at run-time to knowledge bases

Previous chapters introduced background information on the intelligent-system growth mechanisms. Some additional advanced system features can be explored in terms of trust justification features with technical concerns with a focus on distributed computing paradigms. Sensor-based approaches with wireless/visual detection/actuation interactors are promising for dynamic and fully-automated/autonomous designs. Detected features can be integrated within the limits of current networking/communication technologies [81]. Dynamic heuristics [82] and knowledge bases can be trained dynamically with critical-system constraints of massive AI systems [15]. Multi-layer neural networks and tree structures with different data structures can improve the interactivity and performance of training [119,120].

Other critical feature sets like data poisoning, backdoor attack can also be monitored to improve the data collection process of training transactions [121]. The features can be extracted dynamically within the critical data sets like fingerprint images and can be embedded into other knowledge bases and used for critical missions like spoof detection [122]. Furthermore, privacy-preserving deep learning models with homomorphic encryption and chain structures can be designed. However, these feature definitions and interactions are limited due to computational scalability and critical system design constraints [5]. Emerging hybrid-cloud and distributed computing design paradigms can help to maximize total system throughputs in order to handle the limitations in a trusted scalable manner [29,117]. The challenges and future directions will be summarized in a comparative analysis matrix in Chapter. III B.

III.Security

Security is also a critical system constraint for the justifiable trust features. Security concerns for distributed computing paradigms cover a wide scope from physical protection to digital security mechanisms. The trust features can be justified with digital security design principles for any system with the set of nodes $N_i : \{N_o, N_1, N_2, \dots, N_n, \}$. Therefore, we can limit the scope to digital security concerns. Justifiable trust features can be ensured by improving system thinking paradigms [83] and artificially intelligent cyber-security mechanisms [84]. Adaptive protocols for dynamic contexts with checksum verification-based approaches can justify the trust features and indicators [5] dynamically. Next chapter will discuss the details of these features and future challenges. In this chapter, we will introduce the basic principles of context awareness and trusted computing paradigms.

a. **Context aware dynamic adaptiveness:** Event-based secure connection policies and protocols

Context change detection/actuation and correlation extraction between the system resource allocation abilities are increased with the capacity improvements of computing technologies. Event-based abstraction approaches can adapt to change and reconfigure the required trust justification features with the available policies and protocols [5]. Context-aware computing paradigms can distribute computing and process/extract the required features at the edge or on mobile units [27] with trusted computing mechanisms [37]. Furthermore, the bandwidth limitations of communication systems can also be made trusted with secure channels/interfaces [85,81] and the interactivity of agents and nodes can be ensured for continuous growth [15]. Next chapter introduces checkpoint and verification approaches-based feature for dynamic context change.

b. **Context aware dynamic adaptiveness:** Dynamic package check-sum verification

Thanks to the growth of distributed computing mechanisms, transaction flows can be verified at available checkpoints with dynamic package monitoring approaches to extract

the required metric with sets of justification features. Package checksum verification approaches can verify the integrity check mechanism [5], and this can be applied to generic IT core mechanisms with end-to-end paradigms [15]. By that means, targeted justification features of the related context can be extracted dynamically within the edge units and merged the transaction flows at (near)-real-time. Therefore, we can rely on and limit the scope to package check-sum verification approaches to justify the defined trust features, which are summarized and discussed in the comparative matrix in Chapter 4. B.

IV.Test

Testing is also a de facto component for the system design lifecycle. Continuous testing mechanisms (black box, white box, grey box etc.) can detect anomalies and future risks with digital twins of the system units for continuous growth-assurance paradigms. Performance metrics are the key elements of trust justification features for both technical and human-level trust justification features. Behavioral anomaly detection/reaction-based monitoring approaches can detect/recover potential risks within the critical system design constraints [80]. Therefore, we can limit the testing scope to verification and confidence-building approaches.

a. **Verification (survey, benchmarking, expert):** Formal verification with regulative and technical standards

Verification mechanisms are part of the holistic system lifecycle for distributed computing paradigms. These challenges can be defined and categorized as software engineering [46] paradigms with system design perspectives. Therefore, the necessary quality attributes can be defined/tracked/monitored as performance indicators. In order to make the approaches dynamic and adaptive to changing contexts, we can limit the scope to check-point controller and feedback mechanism principles for continuous growth assurance concerns. Detailed features are summarized in the next chapters with the comparative matrix tables. Rest of the chapter briefs about the selected trust justification features of the testing and verification mechanisms of the growing intelligent systems.

i. **Dynamic check-point locating with feedback controllers and optimization:** End-to-end holistic check-point structures

Improvements in the distributed computing can enable packages to be processed at the edge or using mobile units as discussed in previous chapters. Feedback controllers can be correlated with the total system performance and each system unit's throughput values can be correlated dynamically with the behavioral anomalies for feature

extraction/detection/reaction mechanisms [5]. Novel structures and holistic abstraction approaches can be implemented on the edge devices and used to build end-to-end TEE. Dynamic optimizers can be merged as critical supplementary components with respect to the context dependencies.

Therefore, the system can ensure growth acceleration and improve the performances of the mobile units and agents with the trust justification mechanism via the holistic views, which provides dynamic feedback for the continuous growth of an intelligent system. Critical feature sets are summarized within the comparative matrix in Table.3. B.

ii. Resiliency and robustness monitoring with holistic views and feedback controllers: Data-driven dynamic control structures for monitoring mechanisms

User-level concerns are also critical metrics for the trust justification features at the technical and human/socio-dynamics levels. Resiliency and robustness features can be tracked with semantic or graph-modeling approaches, which are used to represent and visualize [43,44] the correlations between the entities. These features can be named and generalized as conceptual modeling [42] and monitored with a holistic end-to-end trust mechanism [15] as part of the generic IT core structure. Thereby, it can used to monitor regulative constraints in related contexts to observe the identified thresholds and improve train sets of alerting mechanisms.

These features can be improved with scenario-based strategy planning paradigms [45]. Therefore, trust justification features can ensure the acceleration of the growth of the system with the monitored performance indicators and quality attributes of resiliency and robustness features with dynamic controllers and testing operations/processes. Table.3 gives a summary of the identified features and emerging challenges to ensure the continuous growth of the intelligent systems.

b. Confidence-Building: Trust and confidence measurement/quantification in intelligent systems

Trust can be defined as the behavioral integrity of a system, that is, the system behaves as expected at all times, in computing paradigms and sciences [37]. Human/socio-dynamics level concerns can be limited to regulative legal metrics with IT audit paradigms. Socio-dynamical visions can help us to understand changing requirements and contexts dynamically and provide continuous feedback on the defined/monitored trust justification features to accelerate the growth progress of the intelligent systems.

Building confidence in these systems also requires critical constraint feature predictions and forecasts for potential anomalies. This level of confidence can be maximized with strategy planning and a vision of the future cases and predictions about the states of the contexts [45]. Therefore, the trusted mechanisms can keep learning and accelerate growth continuously with an increasing confidence in the total system. Regulative legal constraints are dependent on the digital dynamic operation context and socio-dynamic regulation within the client context. This field is a future possible direction and challenge in our research. Next chapter will elaborate on the user-level monitoring metrics, which can be identified as critical system threshold values of the alerting mechanisms. So that, user-oriented critical alerts can be minimized and confidence can be built with maximum level. Next chapter briefs about these metrics/parameters.

c. User-level continuous trust measurement: Facial expressions/body language, behavioral anomalies

In order to retain the validity of the metrics for trust measurement mechanisms in distributed computing paradigms, these justification features can be improved with novel indicators, such as facial expressions, body language or any other human-level behavioral anomalies that can be correlated as sensor units of opinion-mining algorithms [29]. Therefore, the impacts of socio-dynamical changes can provide dynamic feedback to the control loops of growth mechanisms via trusted channels [29], and the interactivity of the

mobile units can be maximized with minimum latencies and fault penalties with a dynamic holistic view [5]. These features are observed dynamically with respect to the identified regulative legal constraints of the targeted context.

Regulation mechanisms are also disrupted by growth acceleration and the diverse structure of emerging intelligent systems. Real-time alerting mechanisms are required in daily life also be able to observe socio-dynamic changes and make dynamic alerts for the critical risks in the observed context. Mathematically well-defined structures can enable us to implement swarming approaches within the agent functions $f_{opt}()$ and maximize the cooperation between the exponentially increasing number of components and exa-scale data resources [15]. Thereby, measurable metrics of the regulative legal constraints of observed subject matter can be visualized within the high-level monitoring dashboards of the intelligent systems within the intelligence-flow mechanisms. Figure 4 visualizes the correlation between the end-to-end trust mechanism and growth-flow structure, which can enable to build dynamic intelligence flow within regulative legal constraints of the observed context. Thereby, trust can be quantified with respect to dynamic legal metrics of the socio-dynamic metrics and parameters. This field is also a research domain will be investigated in related future works, in this thesis we will keep focus on behavioral anomaly observations of the observed context.

On the other hand, data processing capabilities are still limited by edge device processing limitations and the latency values of these units. Fortunately, the current state regulatory standards define the critical constraints, which are emissions, power limits, and other critical factors. These have an impact on our health and can be monitored and extracted as trust justification features in real time or near real time via the alerting mechanisms. Nevertheless, massive scale deployment is still limited with scalability concerns at the algorithm and system levels [5]. These include critical risks for human health and environmental concerns. These features are also part of the future research directions. Table.3 summarizes major concerns and identifies critical feature sets of trust justification features to be able to maximize trust in emerging intelligent systems and minimize socio-dynamic risks within the identified regulatory legal constraints.

B. Comparative Matrix

The comparative matrix is reflecting the state of the art related to the thesis. It shows first the main scientific papers connected to TDAI reflected in Table 3.A, as well as a matrix comparing all such works against trust justification features reflected in Table 3. B.

An intelligent system with of nodes $N_i : \{N_o, N_1, N_2, \dots, N_n, \}$, which have critical selected features can be categorized into five main groups as in Table.3:

- Trusted scalability and elasticity for throughput maximization
- Resilience to adversarial/adversarial threads
- Simulation-based validation and verification with digital twins or limited context simulations
- Monitoring with holistic views of the system
- Thread detection and reaction with dynamic feedback controllers for continuous growth flow.

As summarized in Table.3, state-of-the-art approaches investigate the challenges with disruptive system-level innovations. For instance, a data-centric operating system is proposed with limited features [106]. Higher-throughput lower-latency features are also studied with protected data planes [107]. The approach has triggered paradigm switches on transaction definitions and implementations, such as a device [108] is proposed for a secure transaction with advanced feature sets like dynamic feedback controllers. Although trusted scalability remains an open issue but novel holistic abstraction approaches [5] can help maximize the throughput of nodes and total systems.

AI-driven system modeling is also a hot topic, especially for decentralized and distributed systems [109]. Adversarial/un-adversarial thread monitoring and transaction approval approaches [110,111,112,113] also promising to minimize system-level anomalies and failures in (near)-real time. Quantum computing and quantum cryptography features [114] are becoming a critical challenge and feature for the growing intelligent systems.

Simulation-based digital twins are [115] widely implemented to minimize potential future failures and improve knowledge bases and training sets.

In a nutshell, we can say that decentralized and distributed designs enable us to implement massive-scale AI/ML algorithms within the growing intelligent systems as hybrid clouds [116,117] with novel accelerator components, as indicated in Table.7 with focus on recent years between 2011-23. Next chapter summarizes these challenges and main findings and introduces potential TDAI research fields and comparative matrices as tables.

Table.3. A. Related Works			
#	**Type, Year, Authors**	**Impacts**	**Shortcomings**
1	**Survey, 2018, Baydin et al., Oxford et al.**	Automatic differentiation is proposed for machine-learning applications to build end-to-end pipelines.	Limited evaluation of dynamic computational graphs and differentiable programming. Computational scalability considerations are missing.
2	**Method, 2013, Kraska et al., Brown Uni. et al.**	Support for machine learning and statistical techniques with a distributed system design view and novel data management features.	Limited evaluation of data management, networking, and distributed system design basics.
3	**Method, 2017, Milutinovic et al., Berkeley et al.**	End-to-end differentiable pipelining frameworks are proposed for ML frameworks.	Results are not presented with a proof-of-concept trial.
4	**Method, 2019, Cohen et al., University of Waterloo**	Trust modeling in multi-agent systems is proposed for trust assurance in AI systems.	Limited definition of trust and trust modeling metrics at system level.

5	**Method, 2019, Agca, TOBB ETU**	Holistic system abstraction is proposed for the end-to-end transaction flow monitoring of trusted AI systems.	Proof-of-concept test systems to be tested on larger scales and in different contexts.
6	**Method, 2020, Nassar et al., American University of Beirut et al.**	Blockchain structures and smart contracts are proposed for explainable and trustworthy AI.	Limited file system and caching policy evaluation. Theoretical system design and computational scalability indicators are missing.
7	**Method, 2020, Kumar et al., University of Helsinki, Finland et al.**	Trust issues for AI systems and smart-city use cases explored.	Limited system architecture and networking feature evaluations.
8	**Method, 2019, Hummer et al., IBM Research**	A framework is proposed for end-to-end AI Lifecycle management with a DevOps focus.	Limited system abstraction definition. Trust measurement and quantification is missing. Application layer perspectives are used for system design concepts like feedback mechanisms.
9	**Survey, 2020, Fernández et al.,**	Research directions identified for AI Models on Trustworthy Autonomous Systems.	Limited SOTA review and description of identified keywords. System verticals are not

	UPC-BarcelonaTech et al.		explored; limited application layer level DevOps design issues are identified.
10	**Survey, 2018, Bottou et al., Facebook AI Research et al.**	A comprehensive review for large-scale optimization methods for machine learning is provided.	Limited benchmarking and computational scalability evaluations.
11	**Method, 2017, Lee et al., University of California, Berkeley et al.**	An acceleration framework for increasing the performance of distributed machine learning algorithms is proposed. Noises, such as straggler nodes, system failures, or communication bottlenecks are identified and elaborated with coding theory techniques to provide resiliency in different engineering contexts.	Limited benchmarking and computational scalability/performance evaluations.
12	**Method, 2006, Hull et al., MIT**	A distributed mobile sensor computing system is proposed for the cars to dynamically process data at massive scale.	Scalability and trust mechanisms are not investigated.
13	**Survey, 2010, Russel & Norvig, Berkeley**	Full-breadth definition of AI as a scientific field.	Limited evaluation for networking and trust aspects.

14	**Survey, 2014,Steen& Tanenbaum, University of Twente et al.**	Full-breadth definition of Distributed Systems and Distributed Computing Principles and Paradigms.	Limited evaluation for networking, trust measurement/quantification and end-to-end paradigms.
15	**Method, 2020, Agca et al., LIST**	Key challenges for swarm intelligence mechanisms are identified as computing, communication, and control. In order to ensure resilience against manipulation threats, the other parts of the contribution concern end-to-end trust mechanism (integrated view of the three pillars: networking, processing/optimization, and security) and swarm controller methods guaranteeing safety, which aims to enable the trusted scalability of the swarm systems. It introduces CCAM Connected Cooperative, Autonomous Mobility) business-use cases.	Limited discussion of the results of the proposed methodology and proof-of-concept - POC implementations.
16	**Method, 1993 Velthuijsen, PTT Research**	Distributed AI usage in telecommunication is implemented.	Limited results and evaluation.

17	**Survey, 1997, Stone & Veloso, CMU**	ML potential applications to DAI and Multi-Agent systems are proposed.	Limited use cases and evaluations.
18	**Survey, 2019, Toreini et al., Newcastle Uni.**	Trust review for AI and ML studies.	Limited evaluation of ML and AI Systems.
19	**Method, 1992, Marsh, University of Stirling**	Formal definition and evaluation of trust for distributed AI.	Limited evaluation of AI methodologies and system perspectives.
20	**Method, 1999, Agha $ Jamali, MIT**	Concurrent programming paradigm based on multi-agent system actor model abstraction is proposed for DAI.	Limited context definition for distributed agent interactions.
	Other Related and Selected Studies		
30	**Survey, 2019, Gadepally et al., MIT**	Comprehensive view of AI systems and available and future technologies.	Limited system abstraction definitions and literature review of AI methodologies.
31	**Book, 2001, Russell, Computelligence LLC**	Emphasizes swarm-based coherence as collective adaptation for swarm intelligence with artificial neural networks.	Limited system design and scheduling metrics evaluation.

32	**Survey, 2019, Xu et al., Lenovo Research et al.**	Explainability features of AI systems are explored with a neural network focus.	Limited evaluation of system abstraction hierarchies.
33	**Method, 2020, Wing, Columbia University**	Formal definitions of trust and relations between trusted computing and trustworthy AI are explored.	Limited evaluation of SOTA for the proposed concept.
34	**Survey, 2018, Sileno et al. University of Amsterdam, Netherlands et al.**	Emphasizes norm ware with computational artefacts to deal with trust and explainability problems.	Limited norm definition and trust modeling.
75	**Method, 2020, Tomsett et al., IBM Research et al.**	Interpretability and uncertainty of AI systems are identified for a trust calibration framework.	Limited AI methodology categorization and trust modeling to measure and quantify it in AI systems.
76	**Method, 2019, Smuha, KU Leuven, Belgium**	High-level governance framework is proposed for trustworthy AI systems.	System-level standard definitions and feasibility evaluations are missing. Trust measurement and quantification approaches are not evaluated.

104	**Method, 2016, Scherer, Harvard**	Regulative framework is proposed for AI systems.	Technical standard definitions and literature review is missing.
78	**Survey, 2019, Alsamhi et al., School of Aerospace Engineering, Tsinghua University, Beijing, China et al.**	AI methods are reviewed for robot teaming and human cooperation methodologies.	Limited categorization of AI methodologies and data management for swarm systems.
38	**Book, 2010 Golnaraghi et al., Simon Fraser University, Canada et al.**	Basics of automatic control systems are explained in detail. Feedback mechanisms for distributed systems are defined.	Limited evaluation of intelligent systems' actuator and sensor architectures. Data-intensive design and decentralization of computational resource evaluations are missing.
105	**Survey, 2004, Truskowski et al., NASA**	AI techniques are proposed for challenges of mission-critical autonomous software. Novel abstraction paradigms are identified as a requirement for reducing the complexity of swarm	Trust measurement/quantification, computational scalability evaluations are missing; which are basic and fundamental requirements for critical systems.

		systems. The heterogenous structure of emerging intelligent swarms is identified as a challenge for system behavior monitoring and verification.	
80	**Survey, 2004, Rouff et al., NASA**	Formal methods and techniques are explored for the verification, validation, and assurance of future swarm-based missions, such as the ANTS (Autonomous Nano Technology Swarm) mission. It has 1,000 autonomous robotic agents designed to cooperate in asteroid exploration.	Context-aware computational paradigms and real-time data management/fetching features are not evaluated.
82	**Method, 2020, Machovec et al., Colorado State University, USA et al.**	Dynamic heuristics for surveillance mission scheduling with UAVs in heterogeneous environments are proposed. Real-time thresholds are targeted for critical-missions.	Computational load and feasibility evaluations for on-board UAV tasks are missing.
85	**Survey Report, 2021, Stuckman et al., EU Commission**	The main contributions of 5G projects are summarized as of 2021.	Limited evaluation for AI systems use-cases. System abstraction standard definition and benchmarking results are missing.
83	**Method, 2020, Crowder et al.,**	A system-level thinking process is proposed for AI systems.	Data-flow mapping with a computational transaction is missing. The categorization of

	Colorado Engineering Inc. et al.		AI agents has no mathematically well-defined context. Proof of concept experimentation is not provided.
84	**Method, 2020, Carbone et al., Forcepoint Global Governments and Critical Infrastructure LLC et al.**	A comprehensive transdisciplinary approach is proposed, which includes Axiomatic Design (AD), AI/ML techniques and Information Theoretic Methods (ITM), in order to reduce risks and complexity by improving cyber system adaptiveness, enhancing cyber system learning, and increasing cyber system predictions and insight potential.	System design theoretical design limitations are not included. Trust measurement and quantification is not identified, which is a fundamental concern for the proposed concept.
85	**Method, 2020, Crowder et al., Colorado Engineering Inc. et al.**	Concepts and notional architectures are presented for the Big Data Analytical Process (BDAP), in order to facilitate real-time cognition-based information discovery, decomposition, reduction, normalization, encoding, memory recall (knowledge construction), and decision-making for big data systems.	Limited evaluation of AI methodologies, feedback structures, and data-flow process and definition.

42	**Survey and Method, 2020, Proper, Luxembourg Institute of Science and Technology (LIST)**	Business analytics, statistics-based AI, digital twins, etc., are defined as “data hungry” application components of complex systems, which can be thought of as data ecosystems.	Data is not fully separated from computation, it has to be mapped to computation. Trust has to be measured and quantified. Novel holistic system abstractions are required to track transaction flows at the system level and to assign trust values to each one.
46	**Survey, 2020, Aksit, TOBB ETU, Ankara, TURKEY**	Software engineering challenges for smart-city systems are categorized. 1- Developing models for smart cities; 2- Designing a framework for managing and optimizing the configuration of clusters; 3- Designing models, methods, and tools for critical infrastructures; 4- Optimizing the necessary quality attributes through system adaptation at run-time; 5- Integrating software systems; 6- Designing a smart infrastructure with a high degree of interoperability, configurability, adaptability, and evolvability.	Software and hardware co-design principles are emerging. System level hardware/software integrated views are required, which interact with all verticals at run-time. Trust is not only about dependability. It also has to be measured and quantified for smart systems in order to inspire confidence.

27	**Survey, 2013, Perera et al., Information and Communication Centre, Australia**	Context awareness is surveyed from an IoT perspective, which includes techniques, methods, models, functionalities, systems, applications, and middleware solutions. Growth progress of context-aware computing from desktop applications, web applications, mobile computing, and pervasive/ubiquitous computing concerning the Internet of Things (IoT) is explained.	Limited evaluation for distributed caching and memory management. The computational scalability of the emerging complex systems is not evaluated with system throughput limitations.
37	**Method, 2009, Martin et al., University of Oxford**	Trusted computing, trusted platforms, and trusted systems are defined as the system components, which behave as expected.	Limited definition for data caching, trust measurement, and computational scalability issues.
81	**Survey, 2004, Cherry, Northwestern University, USA**	The throughput data rates of communication technologies correlated with Moore's law and Edholm's law of bandwidth.	Limited evaluation for computational scalability and system throughput limitations.
116	**Survey, 2020, Reuther et al., MIT, USA**	AI/ML accelerators with end-to-end approaches are summarized. Vector engines, dataflow engines, neuromorphic designs, flash-based analog memory	Limited evaluation for middleware categories and trusted scalability concerns.

		processing, and photonic based processing approaches are discussed.	
117	**Method, 2021, Samsi et al., MIT, USA**	Modern AI/ML workloads are categorized, and limitations of HPC systems are explained. SuperCloud hybrid mechanisms are introduced with recent data sets.	End-to-end trust mechanisms and holistic abstraction paradigms are not considered. Limited evaluation for middleware components.

Table.3. B. Trust Justification Features

#	AI				Architecture			Networking					Trust															
	Behavior/ (Acting Humanly/Turing Test Approach): NLP, Knowledge Representation, Automated Reasoning, Machine Learning Learning	**Thought Processes and Reasoning / (Thinking Humanly/ Cognitive Modeling Approach):** Cognitive Science, Neuroscience, General Problem Solver	**Rational Thinking/ (Success Measurement Respect to Human Performance):** Syllogism/Right Thinking, Computational Reasoning and Logic Design	**Acting Rationally / (Rational Agent Approaches):** Rationality/ Mathematically Well-Defined General Design	**Centralized/ (Fully connected):** Processing and memory resources are fully centralized	**Decentral/ (Autonomous/ Embedded/Local):** Processing and memory resources are fully decentralized	**Distributed/ (Edge/Hybrid/Hierarchical/Multi-layer):** Processing and memory resources are distributed	**Close-Range (PAN < 100m):** Bluetooth, Wi-Fi, 802.11p/ITS G5 for V2X, low latency networks, etc.	**Short-Range (LAN < 10km):** Cellular networks, 4G/LTE, 5G NR, etc.	**Mid-Range (MAN < 100km):** Global networks, High Speed Wireless Internet, cable TV systems, etc.	**Long-Range (WAN > 1000 km):** Space Networks, Sat Com, Space Internet, Futuristic (Drones, Low-Orbit) Satellite etc.)	**Performance**	**Scalability, Elasticity and Connectivity Limits** Total number of nodes and users in the system	**Energy Efficiency** - Average energy consumption of nodes and transactions - EMF (v/m), SAR(w/kg), Power(W) environmental friendliness	**Run-Time Monitoring**	**Data- flow monitoring:** Data-state monitoring between applications	**Transaction-Flow Monitoring:** Transaction lifecycle monitoring	**Trust Monitoring:** Periodical trust verification	**AI/ML Pipelining:** Dynamic knowledge-based monitoring and update	**Run-time feature embedding and interaction:** Data fetching at run-time to knowledgebases at run-time	**Security**	**Context-aware dynamic adaptiveness:** - Event-based secure connection policies and protocols	**Context-aware dynamic adaptiveness:** - Dynamic Package checksum verification	**Test**	**Verification (survey, benchmarking, expert):** - Dynamic check-point locating with feedback controllers and optimization - Resiliency, reliability, dependability, and robustness monitoring with holistic views and feedback	**Confidence Building:** - Trust and confidence measurement/quantification in smart systems	**User-level continuous trust measurement:** Face mimics/body language, behavioral anomalies
	Denominators: Distributed computing/caching, differential/probabilistic/dynamic programming																										
	✓: YES ✗: NO ? : NOT INDICATED																										
1	✓	X	X	X	?	?	?	?	?	?	?		X	X		X	✓	X	✓	X		X	X		X	X	X
2	✓	X	X	X	X	X	✓	?	?	?	?		✓	X		✓	X	X	X	✓		X	X		X	X	✓
3	✓	X	X	X	?	?	?	?	?	?	?		X	X		X	X	X	✓	X		X	X		X	X	X

4	✓	X	X	✓	?	?	?	?	?	?	?		X	X		X	X	✓	X	X		X	X		X	✓	✓
5	✓	X	X	✓	X	X	✓	✓	✓	X	X		✓	✓		✓	✓	✓	✓	X		✓	✓		✓	✓	X
6	✓	X	X	✓	✓	X	✓	?	?	?	?		X	X		X	✓	X	✓	X		X	X		X	X	✓
7	✓	X	X	X	?	?	?	?	?	?	?		X	X		✓	✓	X	X	X		X	X		X	X	✓
8	✓	X	X	X	?	?	?	?	?	?	?		✓	X		✓	✓	X	✓	✓		✓	✓		X	✓	✓
9	✓	X	X	X	?	?	?	?	?	?	?		X	X		X	✓	✓	X	X		✓	✓		✓	X	✓
10	✓	X	X	X	✓	X	✓	?	?	?	?		✓	X		X	X	X	✓	X		X	X		X	X	X
11	✓	X	X	X	✓	X	✓	✓	✓	X	X		✓	✓		✓	✓	✓	✓	✓		✓	✓		X	X	X
12	✓	X	X	X	X	X	✓	X	✓	X	X		X	X		✓	X	X	X	X		X	✓		X	X	X
13	✓	X	X	X	?	?	?	?	?	?	?		X	X		X	✓	X	✓	X		X	X		X	X	X
14	X	X	✓	✓	✓	✓	✓	✓	✓	✓	✓		X	X		✓	X	X	X	X		✓	✓		✓	X	X
15	✓	X	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	X		✓	✓	✓	✓	✓		✓	✓		✓	✓	X
16	X	X	X	✓	X	✓	✓	?	✓	✓	?		X	X		X	X	X	X	✓		X	X		X	X	X
17	✓	X	X	✓	X	X	✓	✓	X	X	X		?	?		?	?	?	?	?		X	X		X	X	X
18	✓	X	X	?	?	?	?	?	?	?	?		X	X		X	X	✓	✓	X		X	X		X	X	✓
19	X	X	X	✓	X	X	✓	?	?	?	?		X	X		X	X	✓	X	X		X	X		X	X	X
20	X	X	X	✓	X	X	✓	?	?	?	?		X	X		X	✓	X	X	✓		X	X		X	X	✓
	Other Related and Selected Studies																										
30	✓	X	X	X	?	?	?	?	?	?	?		✓	✓		✓	X	✓	✓	X		✓	X		X	✓	✓
13	✓	X	X	X	?	?	?	?	?	?	?		✓	X		X	X	X	X	X		X	X		✓	X	X
32	✓	X	X	X	?	?	?	?	?	?	?		X	X		X	X	X	✓	X		X	X		X	X	X
33	✓	X	X	X	?	?	?	?	?	?	?		✓	X		X	X	✓	✓	X		X	X		✓	X	X
34	✓	X	X	✓	?	?	?	?	?	?	?		X	X		✓	X	✓	X	X		X	X		X	X	✓
75	✓	X	X	X	?	?	?	?	?	?	?		X	X		X	X	?	X	X		X	X		X	✓	✓
76	✓	X	X	X	?	?	?	?	?	?	?		X	X		X	X	X	X	X		X	X		X	X	X
104	✓	X	X	✓	?	?	?	?	?	?	?		X	X		X	X	X	X	X		X	X		X	X	✓
78	✓	X	X	✓	✓	✓	✓	✓	✓	X	X		X	✓		X	✓	X	X	X		✓	✓		X	X	X
38	X	X	✓	✓	✓	✓	✓	✓	✓	✓	✓		X	✓		X	X	X	X	✓		X	X		✓	X	X
105	✓	X	X	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓		X	X	X	X	✓		X	X		✓	X	✓
80	✓	X	✓	✓	✓	✓	✓	✓	✓	✓	✓		X	X		X	X	X	X	X		X	X		✓	X	✓

82	✓	X	X	✓	✓	✓	✓	✓	✓	X	X		X	✓		X	X	X	X	X		X	X		X	X	X
85	✓	X	✓	X	✓	✓	✓	✓	✓	✓	✓		✓	✓		X	✓	✓	X	X		✓	✓		✓	X	✓
83	✓	✓	X	✓	✓	✓	✓	?	?	?	?		X	X		✓	✓	X	X	X		X	X		X	X	✓
84	✓	X	X	X	✓	✓	✓	?	?	?	?		✓	X		✓	✓	X	X	X		✓	✓		X	X	✓
85	✓	✓	X	X	✓	✓	✓	?	?	?	?		✓	X		✓	X	X	X	X		✓	✓		✓	X	X
42	✓	X	X	X	✓	✓	✓	X	X	X	X		X	X		✓	✓	✓	✓	X		X	X		X	✓	✓
46	✓	X	X	X	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	X	X	X	✓		✓	✓		✓	X	✓
27	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓		X	X	X	X	✓		✓	✓		X	X	✓
37	X	X	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	X		✓	✓	✓	X	X		✓	✓		✓	X	✓
81	✓	X	X	X	✓	✓	✓	✓	✓	✓	✓		X	X		✓	✓	X	X	X		X	X		X	X	X
116	✓	X	✓	✓	✓	✓	X	✓	✓	X	X		X	✓		✓	X	X	✓	✓		X	X		✓	X	X
117	✓	X	✓	X	✓	X	✓	✓	✓	X	X		✓	✓		✓	X	X	✓	✓		X	X		X	X	X

C. Result Analysis and Conclusions

As a brief summary to the explored research challenges identified in the previous chapter, we emphasize major concerns and express a strong interest in the trusted distributed AI mechanism with the identified justification features. The features and indicators of the end-to-end trust mechanism can be listed as below.

- **Performance**
  - **Scalability, Elasticity, and Connectivity Limits:** Total number of nodes and users in the system
  - **Energy Efficiency:** Average energy consumption of nodes and transactions
  - **Energy Efficiency:** Average EMF (v/m), SAR(w/kg), Power(W) environmental friendliness
- **Run-time Monitoring**
  - **Data-flow monitoring:** Data-state monitoring between applications
  - **Transaction-Flow Monitoring**: Transaction lifecycle monitoring
  - **Trust Monitoring**: Periodical trust verification
  - **AI/ML Pipelining**: Dynamic knowledge-base monitoring and update
- **Security**
  - **Context-aware dynamic adaptiveness:** Event-based secure connection policies and protocols
  - **Context-aware dynamic adaptiveness:** Dynamic package checksum verification
- **Test**
  - **Verification (survey, benchmarking, expert):**
    - **Dynamic check-point locating with feedback controllers and optimization**
    - **Resiliency and robustness monitoring with holistic views and feedback controllers**
  - **Confidence-Building:** Trust and confidence measurement/quantification in smart systems
  - **User-level continuous trust measurement**: Face mimics/body language, behavioral anomalies.

The main categories of these features and indicators identified are: (1) Performance (2) Run-Time Monitoring (3) Security (4) Test-based dynamic metrics. These can build the end-to-end trust mechanism with technical computing paradigms and user-level concerns. The rest of the chapter will provide information on the main related works identified that introduce the emerging challenges.

The reviewed literature shows that distributed AI has been investigated in many domains, such as telecommunication technologies [16]. It has been merged with multi-agent systems as a joint approach for complex system [17] design. Trust features are also explored for the AI/ML paradigms, which are [18] Fair, Explainable, Auditable and Safe (FEAS), to be explored in different stages of a system lifecycle, with each stage forming part of a Chain of Trust. Formal definitions of trust are also elaborated widely in literature [19,4]. The mechanisms are improved with ML and statistical perspectives to cover data management challenges [2] with end-to-end pipelining mechanisms [3]. Programming paradigms, such as concurrent [20], probabilistic/dynamic/differential [1] are also explored to adapt the mechanisms to change in a dynamic context.

Performance modeling paradigms are widely discussed in the literature. For instance, an acceleration framework is proposed for the performance increase of distributed machine-learning algorithms. Noises, such as straggler nodes, system failures, or communication bottlenecks are identified and elaborated with a coding theory technique to provide resiliency in different engineering contexts [11]. The authors state a bandwidth reduction gain of O(1/n) from the fundamental limit of communication rate for coded shuffling. Another current problem identified is to find an information-theoretic lower boundary for the rate of coded shuffling. AI methods have been reviewed for robot teaming and human cooperation methodologies. Mobile robotic communication and swarm UAVs will be explored with CNN and RNN methods for the data processing of the obtained image/video data [78].

AI techniques are proposed for challenges of mission-critical autonomous software. Novel abstraction paradigms are identified as a requirement in order to reduce the complexity of swarm systems. The heterogenous structure of emerging intelligent swarms is identified as a challenge for system behavior monitoring and verification. Requirements in engineering, nontrivial learning and planning, agent technology, self-modifying systems, and verification

technologies are emphasized as future challenges for critical swarm mission autonomous software [79]. Formal methods and techniques are explored for the verification, validation, and assurance of future swarm-based missions, such as the ANTS (Autonomous Nano Technology Swarm) mission. It has 1,000 autonomous robotic agents designed to cooperate in asteroid exploration [80]. Its non-deterministic nature, high degree of parallelism, intelligent behavior, and emergent behavior, and new kinds of verification methods remain to be explored. Formal specification language to predict and verify the emergent behavior of future NASA swarm-based systems is currently being designed and developed.

In order to have a comprehensive overview of the challenges, it is proposed to limit the scope to the system-level thinking process for AI systems [83]. Comprehensive transdisciplinary approaches are proposed, which include Axiomatic Design (AD), AI/ML techniques, and Information Theoretic Methods (ITM) to reduce risks and complexities by improving cyber-system adaptiveness, enhancing cyber-system learning, and increasing the cyber-system prediction and insight potential [84]. The growth perspectives of the mechanisms are another research direction [27] in context awareness surveyed from an IoT perspective, and include techniques, methods, models, functionalities, systems, applications, and middleware solutions. The growth progress of context-aware computing, from desktop applications, web applications, mobile computing, pervasive/ubiquitous computing to the Internet of Things (IoT) is explained. Therefore, trusted computing paradigms can help to ensure the behavioral integrity of the mechanisms, in which trusted computing, trusted platforms, and trusted systems are defined as the system components which behave as expected for all transactions [37].

These challenges for the mechanisms can be identified in a nutshell as major points. Key challenges for emerging smart-system mechanisms are identified as computing, communication, and control. In order to ensure resilience against manipulation threats, the other research directions concern end-to-end trust mechanisms (integrated view of the three pillars: networking, processing/optimization, as well as security) and swarm controller methods guaranteeing safety, which aim to enable the trusted scalability of the swarm systems. These features are called CCAM Connected, Cooperative, Autonomous Mobility) as generalized use/business cases [15]. Chapter. V briefs on these emerging features and discusses the challenges with a focus on the last ten years between 2011-23.

Chapter IV: Trusted Distributed Artificial Intelligence (TDAI) Methodology

IV.I Introduction

Intelligent systems are becoming more complex and diverse as the amount of data exponentially increase. Since, the system nodes and components are diverse and complexity exponentially grows, which is not feasible to ensure trusted scalability of the system and algorithms running in real time [140]. Fortunately, widely accepted learning representation approaches with the data such as back propagation [124] can help to formally state the environment and interaction within that. In order to be able to track sequences and state transitions, end-to-end pipeline modelling and differentiation approaches [125,126] can be implemented. However, to be able to keep the critical systems constraints and trusted scalability of the algorithms between the cooperating components, robust distributed check-point mechanisms and [127] computationally scalable mathematical/system models [129,130] are required.

On the other hand, holistic abstraction paradigms can help to extend ACID (atomicity, consistency, isolation, durability) features of database systems to higher system level by extending data locality [130] to the edges in trusted scalable manner. Cooperation and task data sharing between the components can be accelerated with SDN (software defined networking) [131] features of the components. However, as the system functions virtualizes, and number of transactions increases, behavioral integrity of the system is also become controversial due to exponential growth in error rates in task sharing. In order to maximize the performance of task cooperation and minimize the error rates, trust assurance methodologies can help to ensure the behavioral integrity of the system with TEE (trusted execution environment) utilization such as open-TEE [132].

Furthermore, holistic views [130] are required as critical constraints for dynamic package transmission and task sharing between these units. In order to tackle the challenge, we propose a methodology called Trusted Distributed AI (TDAI) in this study to ensure end-to-end trust and built a software driven trusted execution environment to maximize performance of task cooperation and minimize error rates. So that, behavioral integrity of a growing intelligent system can be assured with maximum performance and dynamic feedback structures, which are utilized via the trusted holistic views.

In addition to the holistic views, behavioral integrity and trusted scalability issues of intelligent systems are widely explored in literature with many perspectives. In [133], authors integrate resources virtualization approaches as SuperCloud and publishes initial data sets for performance evaluations. In [134], 75,000,000,000 streaming inserts/second using hierarchical matrices with bindings to a variety of languages (Python, Julia, and Matlab/Octave) are experimented. In [126], Spatial Temporal Analysis of 40,000,000,000,000 Internet Darkspace packets are observed to analyze internet traffic. Improvements of edge devices enabled to extract more features to implement recent end-to-end paradigms.

Reuther et.al. [136], explores the ways of Interactive supercomputing on 40,000 cores for machine learning and data analysis. The authors targets to overcome old fashion compute bound design limitations of HPC (High Performance Computing) with interactive approaches. Kepper et.al. [137] explores better representation of data in AI systems with associative arrays. More interestingly Tataria et.al. [138,139, 142] makes holistic discussions by covering communication and networking perspectives also with a focus on 6G wireless components of the emerging intelligent systems. Thereby, we can see the increasing need to end-to-end TEE and behavioral integrity assurance with TDAI methodology in the state of the art.

From the standardization side, many initiatives (like EU ones) are claiming about the need of introducing certification for "trusted AI" systems which can delivered by independent bodies after testing the products for key trust features. This is also true for AI products that are distributed by system characteristic requirements [130, 141,143,144]. In all standards and assessment related activities, a reference model is always required to monitor with dynamics holistic views during all stages of a system life-cycle. Thereby, in this study we propose a new methodology called Trusted Distributed AI (TDAI) to ensure end-to-end trust and built a software driven trusted execution environment to maximize performance of task cooperation and minimize error rates. By that means, behavioral integrity of a growing intelligent distributed system can be assured with maximum performance thanks to dynamically justified features of a system as explained in rest of the paper.

Rest of the chapter is organized as follows: Chapter. IV.II defines trusted distributed AI methodology (TDAI), Chapter IV.III defines distributed AI system architectures and explains

the need for distributed architectures and articulates increasing interest to TDAI in literature. Furthermore, gives details about the security, privacy, trust metrics, and regulative constraints considered in this study by asserting the methodology and contributions of TDAI in detail. Additionally, the chapter compares the behavior monitoring application in CCAM (Connected Cooperative Autonomous Mobility) domain to comparatively analyze TDAI with other SOTA methodologies; such as, centralized, decentralized/autonomous ones, non-trusted approaches etc. Chapter IV.IV evaluates the contributions of this study and discusses about the future potentials. Finally, Chapter. IV.V concludes the chapter.

IV.II Methodology Components

A. Methodology Overview

This paper is introducing a novel methodology that offers explicit means to justify trust in distributed intelligent systems with operational features (TDAI-OM). In fact, as the number of required critical justified features increases to ensure trusted interactivity [5], trust cost also increases to be able to justify the trust in dynamic context in (near) real time as illustrated in Figure 10.a TDAI-OM framework helps finding the optimal trust zone based on the balance that any systems operation can leverage in the targeted distributed nodes design and deployment. This way, depending on the context of use case, the reachable trust level can be managed based on the cost available and required features. Next sub- chapter explains the trust levels and introduce TDAI formulation.

B. TDAI Taxonomy

B.I. TDAI-Om Trust Levels:

TDAI-OM is designed in a way that it can be used in most cases compared to similar complex methodologies that exist in the literature like Common Criteria standard [140,141,143]. TDAI-OM is actually based on 5 trust levels that can be accessible. TDAI-OM aim is to estimate the trust value of each node based on key features do identify trust levels in Table.4 and critical metrics/parameters listed in Table.5. So that, each node will be included in available category, are 5 main levels associated to the TDAI taxonomy based on the following:

- TL 1 - Trust Level 1 (0): System nodes not trusted
- TL 2 - Trust Level 2 (0.25): System nodes insufficiently trusted
- TL3 - Trust Level 3 (0.50): System nodes sufficiently trusted
- TL 4 - Trust Level 4 (0-75): System nodes partially trusted
- TL 5 - Trust Level 5 (1.0): System nodes fully trusted

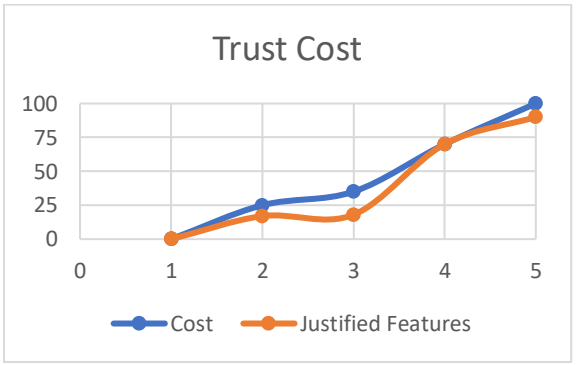

Figure 10.a Trust justification cost.

In order to be able to identify a system or it's node as trusted, it has to be justified [140]. However, for each use case, there is trust cost for each justification feature as illustrated in the example reflected in Figure 10.a. As the number of justification features increases, the trust cost also increases exponentially. Furthermore, to be able to ensure the required minimum throughput of a system, the trust cost worth to pay [130] but can be kept at optimal level with right dynamic strategy mechanisms. In the case of the example of Figure 10.a, Trust level 4 seems to correspond be the optimal TDAI use in the actual context.

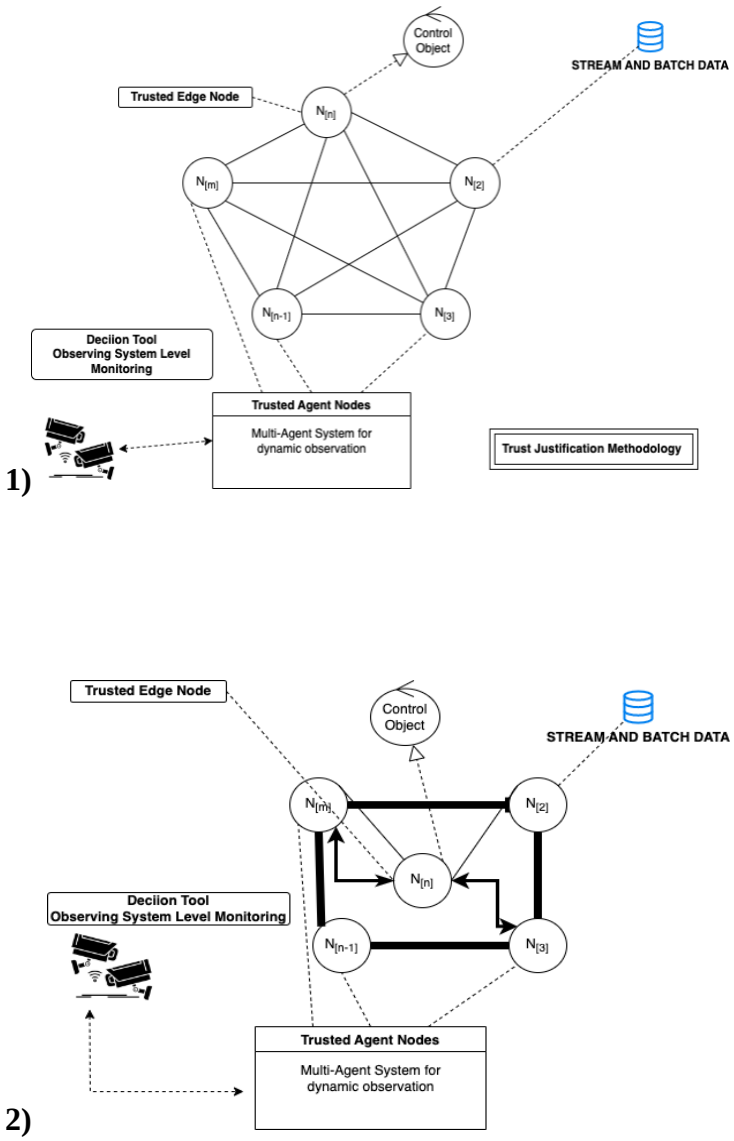

Figure 10.b: (1) A distributed system with set of nodes (2) Justified channels for trusted interactivity with TDAI.

Dynamic strategy plans can be updated in real time by ensuring interactivity of the all components of a system within the observed context. For an optimal level of trust in the context critical nodes iN_i are monitored in (near) real time by ensuring interactivity of trusted agents attached to the nodes as illustrated in Figure 10.b. Next sub chapter explains classes and critical identified features to defined trust levels of TDAI methodology.

B.II. TDAI Classes

Table.4 represents a summary of the defined TLs of nodes N {}. The columns represent an ordinal set of TLs, while the rows represent the trust classes and features of criteria, we use in order to express the requirements for the various trust levels with respect to the trust level taxonomy defined above. Each number in the resulting matrix identifies a specific trust component where higher numbers imply increased requirements. Thereby, the most critical features of a system node can be identified and default values of trust level (TL 1-5) are empirically assigned considering several uses cases in order to make it more operational in a sense that it can capture the maximum measurable values of the features. These can be adjusted dynamically during the observations of many other environment with set of nodes E{N[*]}. Main objective is to maximize trust values and ensure growth-flow in the observed context to be monitored with trusted intelligence-flow between the nodes [140].

Class	Features	Trust Levels				
		1	2	3	4	5
CL1- Performance	PERF_SE: Scalability/Elasticity	1	2	3	3	3
	PERF_EE: Energy Efficiency	1	1	2	3	4
CL2- Runtime	RT_T: Throughput	1	1	2	3	4
	RT_C: Capacity	1	2	3	3	3
	RT_U: Utilization	1	2	3	3	3
CL3- Security	SEC_C: Checksum	1	2	3	3	3
	SEC_SP: Security Protocol	1	1	2	3	4
Cl4 - Test	TST_V: Verification	1	1	2	3	4
	TST_C: Confidence	1	2	3	3	3
	TST_UB: User Behaviors	1	2	3	3	3

Table 4. TDAI Classes

a) Class PERF: Performance

Feature		Description
PERF_SE	Scalability/Elasticity	Maximum number of nodes and user in the observed context

PERF_EE	Energy Efficiency	Average energy consumption of the node.

b) Class RT: RunTıme

Feature		Description
RT_T	Throughput	Expected throughput values of the node.
RT_T	Capacity	Storage capacity of the node.
RT_T	Utilization	Completion of the assigned tasks in expected timelines.

a) Class SEC: Security

Feature		Description
SEC_C	Checksum	Checksum values of the monitored packages.
SEC_SP	Security Protocol	Dynamic secure interactivity protocols.

b) Class RT: Test

Feature		Description
TST_V	Verification	Verification of observed package checksum values.
TST_C	Confidence	User level confidence feedback observations.
TST_UB	User Behaviors	User behavior normal or not.

Each feature within the classes helps to identify trust level of a node and a system composed by the nodes and to justify it dynamically. Next sub chapter explains the weight update and error minimization strategies of TDAI methodology and introduces TDAI formal statement.

B.III. TDAI Formulation

B.III.I TDAI Taxonomy and System Modelling (Trusted Neuron and Trust Measurement Method):

Generic systems are represented by a dynamic model as illustrated in Figure 10.b, where nodes $N_{0..n}\{\}$, are connected to neighbors for cooperation purposes. The aim of the TDAI-OM is to justify channels for trusted interactions among the connected nodes. Each node can be considered as an agent or so-called trusted neuron N_i {}, see Figure 11.a.

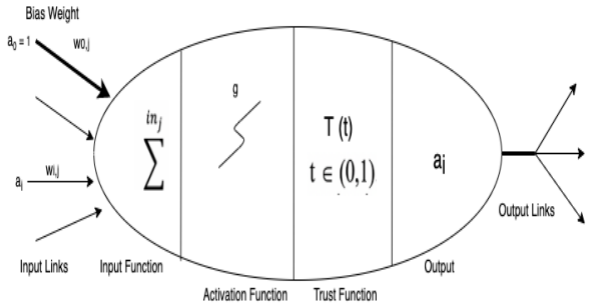

Figure 11.a Trusted neuron

The neurons interact with the environment E {} via the linked nodes and utilizes its' functions dynamically to pursue continues growth-flow within the observed context. Next sub chapter gives details of the TDAI formal statements.

B.III.II Trusted Value Formulation:

Stage 1: Trusted Neuron Formulation:

As formalized in equation.3 below, a **neuron and a trusted neuron, which have input function** N_i, gives the weighted sum of the unit's input values, that is, the sum of the input activations multiplied by their weights w_{ij}:

$$N_i = \sum_{j=0..N} w_{ij}\, a_j \quad (3)$$

Stage 2: Function:

In the second stage, the activation function, g, takes the input from the first stage as argument and generates the output, or activation level, a_i:

$$a_i = g(N_i) = g\left(\sum_{j=0..N} w_{iJ}\, a_j\right) \quad (4)$$

Stage 3: Neurons' Trust Value:

Trust value t_{a_i} (0,1) of output a_i;

$$t_{a_i} = Average\left(\sum_{0}^{i} N_i\right), N\ \epsilon\ transaction\ flow\ (5).$$

Transaction flow repeats continuously with holistic feedback controller mechanism [130], which assures continuous growth of an intelligent system. Learning systems of neural networks can be improved with further justification features and iteratively updated. The frequency of updates improves the total performance of the system, but limited with available resources. By that means, growth flow in dynamic context is observed and updated dynamically. Next sub chapter introduces weight update strategy and error minimization methodology.

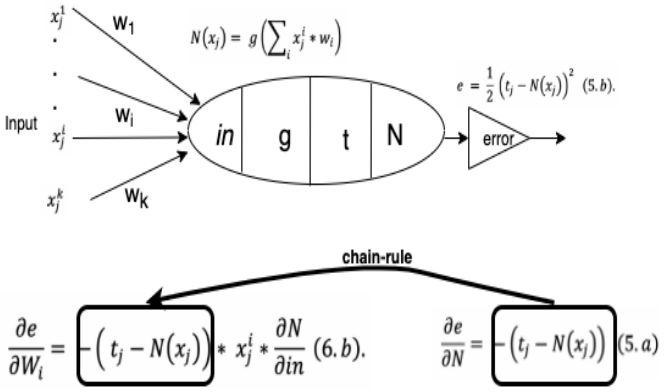

Figure 11.b Single neuron network with error calculation

B.III.III Weight Update Strategy and Error Minimization:

For a given set of representative input and output pairs, $\left(\langle x_j\ , t_j\rangle\right)_{j=1}^{k}$, consider a network with just one neuron N directly connected to the inputs. The inputs x_j can be thought of as a vector with k components.

Let x_i^j be the i^{th} component in the j^{th} training input. Random weights are assigned for each input to initiate training. Output and total errors are computed based on these inputs. Single

neuron n gives output n (x_j) with the training data x_j that has ideal output o_j. **Error e**, on a single input j is usually defined as $\frac{1}{2}\left(o_j - N(x_j)\right)^2$. The network computes the error periodically as indicated in the **Figure 11.b** below**.**

Gradient descent training moves the weights in the direction that they have greatest impact on the error. The weights are then moved in the direction that the error reduces most. **Equation.6** below formulates changing the weights in round r+1.

$$W_i\,(r+1) = W_i\,(r) - \epsilon \frac{\partial e}{\partial W_i}\ (6).$$

If the **function g is differentiable, chain-rule can be applied** for derivation. The chain rule application can enable to **compute the rate of change of the error function with respect to the weights from the rate of change of the error with respect to the output.** For an input x_j, the derivative of the error with respect to the output is below **equation.7a and 7.b**;

$$\frac{\partial e}{\partial N} = -\left(t_j - N\left(x_j\right)\right)\ (7.a).$$

$$e = \frac{1}{2}\left(t_j - N\left(x_j\right)\right)^2\ (7.b).$$

Chain rule can be used to get the derivative of the error with respect to any weight.

$$\frac{\partial e}{\partial W_i} = \frac{\partial e}{\partial in}\frac{\partial in}{\partial Wi} \qquad (8a).$$

$$= \frac{\partial e}{\partial N}\frac{\partial N}{\partial in}\frac{\partial in}{\partial Wi}$$

$$= -\left(t_j - N\left(x_j\right)\right) * x_i \frac{\partial N}{\partial in}$$

$$\frac{\partial e}{\partial W_i} = -\left(t_j - N\left(x_j\right)\right) * x_j^i * \frac{\partial N}{\partial in}\ (8.b).$$

Equation.6 can be plugged to equation.8 to get a rule to calculate how the weights should be updated. Figure.11.bu. illustrates the single neuron network error calculation strategy.

Chain-rule can be applied to multi-layer neural networks as well to train the network. Backpropagation method [124] **(Rumelhart et al. 1986)** is proposed for the error derivative with respect to the weight from layer i to layer $i+1$. Derivatives of the errors used with respect to the inputs in layer $i+1$. The approach is emerging point for automatic differentiation methods in machine learning [125] **(Baydin et al. 2018)**. The methods enable end-to-end training of differentiable pipelines across machine learning frameworks **[126] (Milutinovic et al. 2017).**

Name of Parameters/ Metrics	**Abbrv.**	**Label Descriptions**
Parameters:		
Node ID (IMSI/IMEI)	n_{id}	Defines unique ID for the node in the well-defined context.
Status of Node	n_{st}	Defines whether the node is in active state (value 0) or in dead state (value 1) depending on the battery level.
Type of Event	n_e	Indicates the type of event.
Node Location	n_l	Indicates the node location.
Identification	n_{id}	Represents the unique IP address of a node.
Security Key	n_k	Represents the security key.
Node Behavior	n_B	Represents node behavior (1: normal, 0: not)
Risk Alert	n_R	Represents Risk Alert of a node (1: yes, 0: no)
Metrics:		
Throughput GBPs	n_T	Represents maximum bandwidth capacity of a node
Latency	n_L	Gives the latency value of a node.
Checksum	n_{chsum}	Gives the checksum of datum in a node.
Capacity	n_c	Gives the capacity value of a node.
Utilization	n_ρ	Gives the utilization of a node.
Trust Factor	n_t	Gives the trust factor of a node.
Regulative Constraints:		
Power	n_p	Gives the power value of a node.
EMF	n_{emf}	Gives EMF (Electro Magnetic Field) value of a node.
SAR	n_{sar}	Gives SAR (Specific Absorption Rate) value of a node.

Table.5. Monitoring metrics/parameters of a node.

Backpropagation algorithm is a special case of automatic differentiation [127] **(Griewank 2000).** The method computes a program P' for the derivative of a function f' of a function f given a program P for a function f. Univariate Taylor series with suitable degree is proposed for the problem of evaluating all pure and mixed partial derivatives of some vector function defined by an evaluation procedure. Possibility of derivatives calculation only in some directions instead of the full derivative tensor is explained. Estimates for the corresponding computational complexities are given.

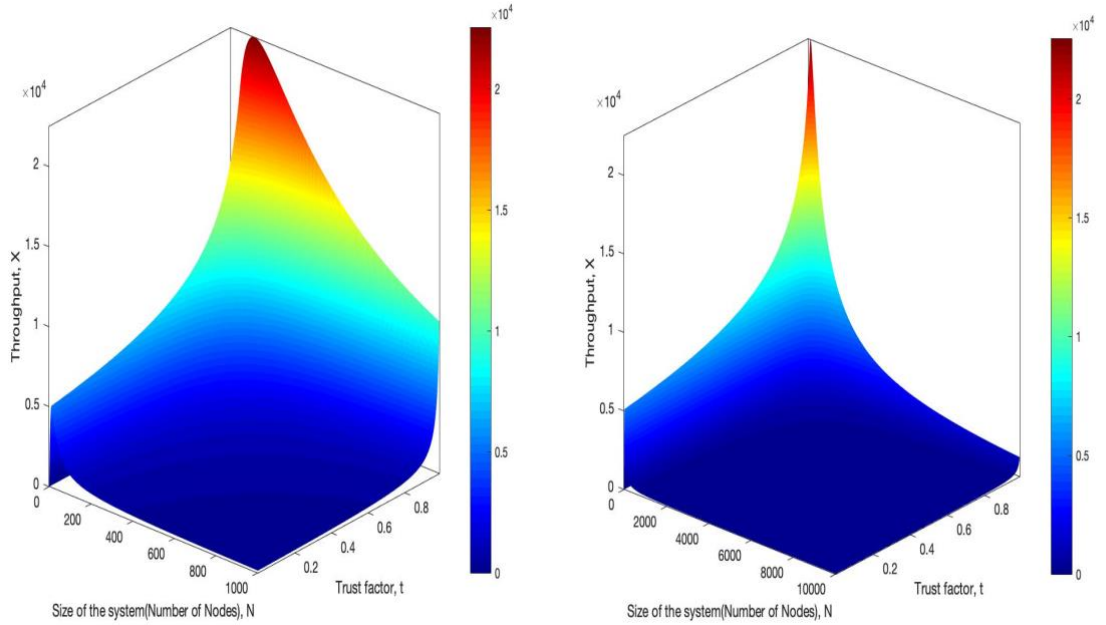

Figure.12. a/b. Trust factor coefficient-based throughput maximisation approach [5].

Computational differentiation is useful for gradient error calculation and single/multi-layer neural network training can be improved with other features. However, computing the rate of change is restricted with **computational scalability** limitations. Furthermore, it inflates the memory resources and require larger memory resources. Algorithm 799 [127] (**Griewank et.al. 2000**) implements a checkpointing for the reverse or adjoint mode of computational differentiation. The authors develop a check-point schedule as an explicit "controller" to reduce the storage requirements and to run a time-dependent applications program. However, differential sequences require (near) real time dynamic holistic views to be able to ensure the validity of the control mechanisms. Thereby, scalability of a system can be considered with the dynamics feedback structures as critical performance metrics.

Scalability modelling metrics and parameters are key performance indicator for any system performance evaluation process. Many aspects can be observed to indicate desired outputs. Main bottleneck for the emerging systems and neural networks is computational scalability constraints. Amdahl law [128] (Amdahl 1967) considers sequential and parallelizable portions of the programs. General theory of computational scalability [129] (Gunther 2008) extends Amdhall law with queuing theory approach. The theory proves that computational capacity is equivalent to the synchronous throughput bound for a machine-repairman with state-dependent service rate.

On the other hand, decentral and distributed architectures are preferred for emerging systems. Scheduling and control approach can be improved with MEMCA (Memory Centric Analytics) holistic abstraction and distributed check-pointing/control mechanism [130]. Check-point locations are optimized with a hierarchical structure, which have TI-Cloud, TII-Gateway, TIII-Fog, TIV-Edge layers. It can be applied to end-to-end AI/ML pipelines to monitor transaction flows also.

State-of-the-art design and holistic abstraction extend data locality to the edges in trusted scalable manner, , it can further improve neural network training with other justified features and total performance with a holistic view to the system. The holistic abstraction provides end-to-end trust justification features for decision mechanism with lineage graph recording of transaction-flows. Trust indicators defined in different system layers can maximize the targeted throughput and minimize crosstalk and latency penalties in hybrid designed architectures. Figure.12 illustrates the correlation of trust factor coefficient with respect to growth of a system.

The study shows that, if a system is trusted, same value of throughput can be obtained with less or same number of nodes in an intelligent mechanism. That is, we can say that in order to maximize throughput of a context, making it trusted is more efficient approach rather than the increasing number of nodes. The holistic abstraction can help to define the features sets of a trusted agent as a system node abstract component, which interacts dynamically with the environments, as formally defined and stated in detail in next sub chapter.

The feature sets are fetched dynamically as inputs to the train sets of data models and structures with a feedback controller mechanism. Thereby, checkpoints can be defined as trusted execution environment (TEE) for critical package context extract/embed in set of flowing network packets **p**<>. Next sub chapter explains the trusted agent and interaction with the environment.

C. *TDAI Work-flow: Trusted Agent Interaction with Environment*

Behavior of an agent N_i can be described as system node abstract component in the environment E (from a class E of environments), and which produces a sequence of states or snapshots of that environment. A performance measure U () evaluates this sequence; see the box labelled "Performance Measure" in Figure.13. Let $V(f, \boldsymbol{E}, U)$ denote the expected utility according to U () of the agent function f () operating on E {}.

Behavior of the trusted agent A{} is monitored within set of nodes N{} in the Environment E ∈ E {} with four steps below;

Step 1: Trusted Agent A{} produces a sequence of states or snapshots of that environment.

Step 2: Performance measure U () evaluates this sequence

Step 3: Let $V(f, \boldsymbol{E}, U)$ denote the expected utility according to U of the agent function $f_{opt}()$ operating on E{}.

Step 4: Monitors System/User Behavior with the Performance Indicators

a. Quantifies and measures trust in system within set of nodes Ny}
b. Maximizes expected utility V()

$$f_{opt} = \arg\max_{f} V(f, E, U).$$

Each Node X(N); Defined as Trusted Agent = { N_i and with activation function a_i }
Trusted Agent as N_i and activation function a_i

$$N_i = \sum_{j=0..N} w_{ij}\, a_j$$

$$a_i = g(N_i) = g\left(\sum_j w_{ij}\, a_j\right) \quad (9).$$

Each Environment E has set of nodes; N_E : {N_1, N_2, N_3, …, N_n }. Each environment can be monitored with set of trusted agents or nodes. Each node can be defined as a trusted agent or agents can be defined as system nodes depending on the context.

Let $V(f, \boldsymbol{E}, U)$ denote the expected utility according to U of the agent function f operating on E. We identify rational agent with an agent function:

$$f_{opt} = arg \max_{f} V(f, \boldsymbol{E}, U) \quad (10).$$

Throughput of each Node X(N); monitored via trusted Agent A {} and nodes N{}

Trusted Agent A {} =
{iN_i and with activation function a_i }

The goal for the set of agents A {} and nodes N {} are to maximize the expected utility V () of the set of environments E {} by monitoring behaviors with $f_{opt}()$ function via trusted channels. Set of dynamics packets **p** <> are observed in distributed checkpoints within the trusted execution environments (TEE). Thereby, learning goals can be accelerated with dynamic feature vectors as feedbacks to assure continuous growth of the agents and the environments.

Trusted Agent iN_i in an environment interaction workflow 1-4 is illustrated in Figure.13. Rationality of agents can enable to interact with the environment in dynamic context via trusted channels.

Transaction flow 1 to 4 repeats for continuous growth-flow of intelligent-system. (1) Trusted Channel Builder starts transaction-flow (2) Sensors interacts with environment, (3) Actuators monitors/detects from environment, (4) Performance element updates/trains the agents. Throughput values of each node X(N) is monitored dynamically with expected average threshold limits. Table.5 in next chapter illustrates the selected metrics and regulative constraints identified in this study. The algorithm called Trusted Distributed AI (TDAI) runs as above pseudocode.

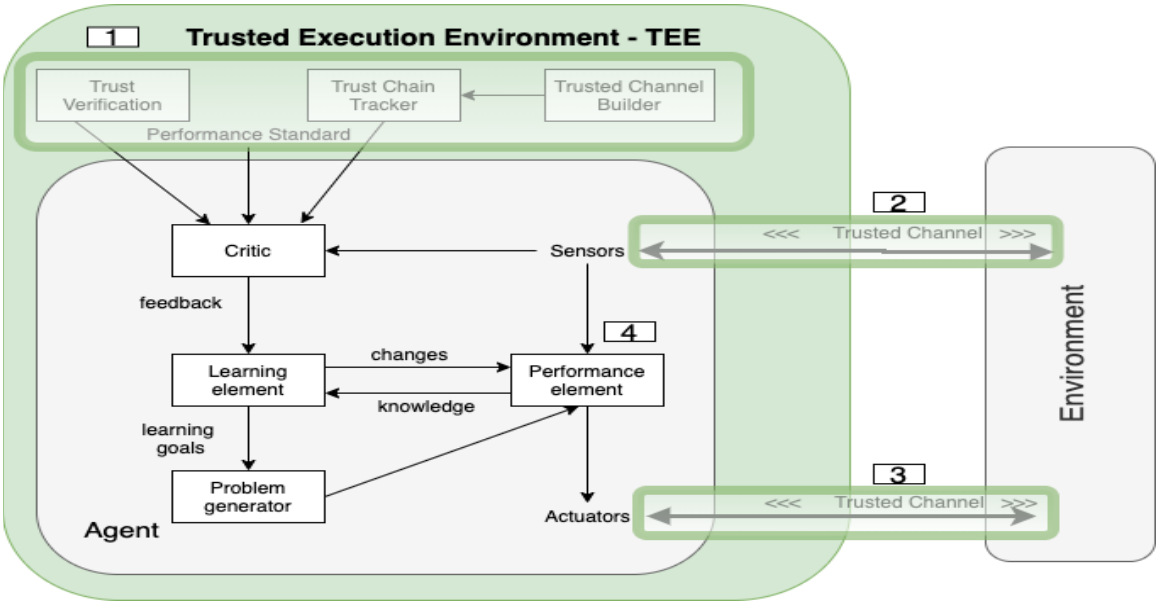

Figure.13. Trusted Agent Environment Interaction Workflow 1 to 4.

Set of Trusted Agent N_i in an environment gets the feature sets dynamically as input and produces set of targeted outputs; such as, risk alerts for the environment dynamically. Figure.14.a illustrates the pseudocode and TEE based interaction in set of environments E{}. The loop enables continuous growth of the system with feedback controller and holistic view to the context with an end-to-end TEE (Trusted Execution Environment). In order to be able to interact with each node, system level design perspectives are required, since the interactions with set of nodes N_E: {N_1, N_2, N_3, …, N_n } in a context are also dynamic and it is not only data dependent but also other dependencies arises up to context features. However, data is the key component to track transaction states and required knowledge-bases to assure the integrity and growth of the mechanism.

Chapter.III explains the **TDAI system components and nodes in detail**. Figure.14.a introduces the basic pseudocode for TEE based interaction with the environment. Each Environment E{} has set of snapshots for selected time spans of the observed context. Set of nodes N{} in a context are observed

```
# Environment E {"Scenario: Risk Detection,
Time: DD:MM:YYYY,HH:MM:SS", Nodes[*],
FeatureVectors v <*>};

Input: Environment E {Nodes[*]};
Output: Environment E {Nodes[*][ 'Alerts']} ;

BEGIN

While ( E{} has active nodes )

        Maximize V(f,E,U)
        Update U ( "Feedback Controllers")
}

END
```

Figure 14.a Pseudo code for TEE based interaction with the environment

dynamically with set of Trusted Agents A{}, which are embedded to OBU (On Board Units/Computers) of each node.

Figure.15 in chapter illustrates the main components of the OBU nodes and interaction with the dynamic context with (1) Central (2) Decentral/autonomous (3) Distributed/Hybrid system architectural design perspectives. Thereby, we can say that the generic and dynamic holistic abstraction [128] can be applied to obtain dynamic holistic view of the context with trust factor coefficient-based throughput maximization approach.

Performance measure U () function is dynamically merged from the dynamic context with an expected utility maximization function $V(f,\boldsymbol{E},U)$. Furthermore, within the well-defined parameters of the Environment E {}, optimization function $f_{opt}()$ is also defined dynamically to improve the knowledge base built with feature vector functions v<*>. So that, the agent function $f_{Trusted\ Agent\{A\{\}\}}(E\{N[*]\})$ can be operated dynamically in set of environments within end-to-end Trusted Execution Environment (TEE) to maximize the dynamically defined improvement parameters with a dynamic optimizer $f_{opt} = arg\max_{f} V(f,E,U)$ to ensure the continuous growth-flow of the observed context as introduces in below pseudocode of Figure.14.b. Next chapter introduces the TDAI system nodes and the components, which has a dynamic growth-flow mechanism based on continuously justified feedback-structures for optimal trust level of the trusted system.

```
# Environment E {"Scenario: Risk Detection, Time:
DD:MM:YYYY,HH:MM:SS", Nodes[*],
FeatureVectors v <*>};

Input: Environment E {Nodes[*]};
Output: Environment E {Nodes[*][ 'Alerts']} ;

BEGIN

Maximize V(ta.f (true),E{},U[])
{

for ( i from 0 to N: number of trusted agents)

  while (data sensors fetch new data)
  {
        Decode Package p <*>;
        Extract Package p <*>;
        Extract Feature Metrics v <*>;
        Update Trust Values of Nodes [*];
        Embed Feature Metrics v <*>;
        Embed Package p <v>;
        Encode Package p <*>;
        Update Environment E {
              Nodes[p<*>][ 'Alerts']} ;

  }

}

END
```

Figure 14.b Pseudo code for TDAI trust verification for continuous growth-flow

IV.III Trusted Distributed AI System Architectures and Components

In this sub-chapter, we introduce three categories of systems (1) Centralized (2) Decentralized/autonomous (3) Distributed/Hybrid design perspectives and hypothesis with theorems regarding the applicability of TDAI methodology to such systems. In fact, the first category of systems enables fully connected context but it faces connectivity and bandwidth limitations, while the second category enable to design fully decentralized autonomous nodes but capacities are limited with edge node feature sets. The third one can maximize connectivity and interactivity with distributed nodes N{} and hybrid system design paradigms. Following sub-chapter will describe the main architectural and component features required for the applicability of the TDAI methodology described in the previous sub-chapter and applied in different contexts.

A. System nodes and components

Trusted agent structure iN_i and interaction flow with the environment is formally stated in previous sub-chapter II. It introduces TDAI system node main components and architectural perspective differences. Throughput values of the nodes X(N) are monitored dynamically via interaction units called OBU (On Board Units) in the context of mobile nodes and other linked nodes when required. The node can be any kind of edge device/computers such as: mobile/smart phones/watches, servers, storages, networking/communications gateways etc.

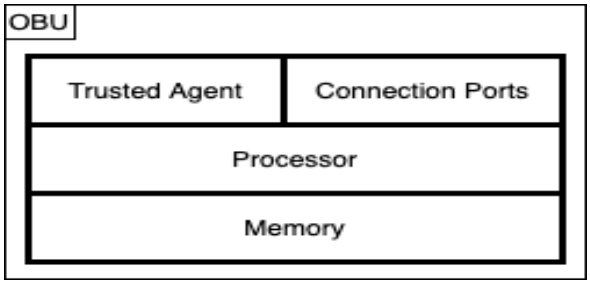

Figure 15. OBU system components

Basic components of an OBU device is illustrated in Figure.15, which are: trusted agent (for TDAI needs), connection ports, processor, and memory. Interaction intra-nodes and the environment can be iterated with (1) central (2) decentral **(**Autonomous/Embedded/Local**), (**3) distributed/hybrid mechanisms. Rest of the sub-chapter introduces the main design paradigms

Critical Features	**Centralized** (Fully connected)	**Decentral (**Autonomous /Embedded/Local**)**	**Distributed (**Edge /Hybrid**)**
Memory/ Storage	+: Easy to maintain and scale up. -: Mobility and accessibility limited.	+: Mobility can be maximized. -: Capacity is limited with edge node feature sets.	+: Data accessibility can be maximized by extending it to the edge in trusted scalable manner. -: Multi-layer system maintenance and data caching is required.
Computation	+: Data can be mapped in real time to maximize the computing speed. - Edge nodes have limited access to the computational devices.	+: Computation can be done in any location. -: Edge devices have limited access to the computational resources can capacities are limited with edge nodes.	+: Computational algorithm can be decentralized with task cooperation approaches and data caching policies. -: System maintenance and deployment is more time consuming.
Power/ Energy	+: Each unit can be connected to central power units. -: Health risks increases due to emission impacts.	+: Mobile batteries can be used. -: Battery life times are limited and causes toxic garbage.	+: Power and energy can also be transferred with wireless energy transfer approaches to maximize mobility of the components. -: Human health and environmental impact risks increases.

Table 6. Distributed design critical features advantages/disadvantages comparison ("+": advantages, "– "= disadvantages).

and hypothesis proposed regarding the methodology. Each node N{} in different contexts interacts with set of Environment E{} via TDAI methodology and embeds the critical metrics and parameters dynamically to the packages p<> as stated in equation.11. The packages are dynamically monitored in critical checkpoints and detect/react mechanisms are triggered depending on the threshold values of the alerts (trust values ae under the expected value). The metrics listed in Table.5 is transmitted from each node as a packet;

$$p =< n_{id}, n_{st}, n_e, n_l, n_{id}, n_k, n_T, n_{chsum}, n_c, n_\rho, n_d >, \qquad (11)$$

The packages p<> are monitored dynamically and related feature sets are vectorized within the continues growth-flow mechanism as represented in Figure.14.b. Each architectural design has critical advantages and limitations ad summarized in Table.6. It is clear that distributed design is required to be able to ensure trusted interactivity, which will be formally proved in next sub-chapters. Next chapter gives details on centralized design and its limitations with regard to the applicability of TDAI methodology.

*B. Centralized (**Fully connected**)*

Centralizing interactions of the nodes in a system has advantages; such as, integrity of the design, accessibility of resources, assuring trusted connectivity of components. However, increasing diversity of the components and decentralization of data/memory resources require system level design reconsideration. Due to computational scalability limit of algorithms and control structures, it is not feasible to centralize the resources [130]. Figure.16 illustrates is a typical example showing the basic components, which are a central cloud, mobile nodes, and fog layer-based networking and communication components.

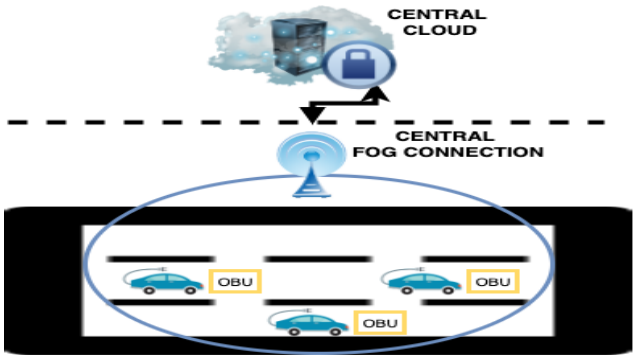

Figure 16. Centralized (Fully connected**) System**

As the number of the node $N_{E:}$ {N_1, N_2, N_3, …, N_n} in the context increases, throughput of the system decreases as illustrated in Figure.12.a. Fortunately, making the system trusted can help maximize the total throughput of the system with less number of nodes, see Figure.12.b. Thereby, we can assume that in order to be able to make the system trusted and scalable, the nodes have to cooperate and share the tasks to be able to ensure the integrity. Next sub chapter briefs about decentralized design basics as another approach, which enables to maximize capacities of each node.

*C. Decentralized (**Autonomous/ Embedded/Local**)*

Decentralized design enables each node to have memory/storage and networking/communication components independently as seen in Figure.17. As the Moores's law disappears with emergence of 3D-Stack memory and storage components, data capacities can be maximized within the autonomous nodes as decentralized mechanism. By that means, the interactions intra set of nodes $N_{E:}$ {N_1, N_2, N_3, …, N_n } can be minimized and networking and communication bottlenecks are minimized as

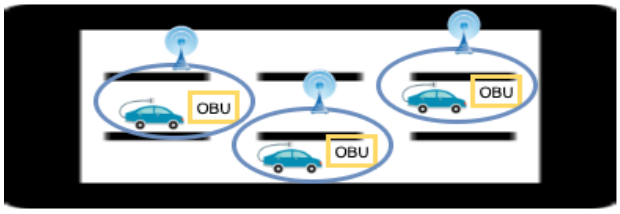

Figure 17. Decentral (Autonomous/ Embedded /Local **) System**

well. However, data intensive nature of emerging intelligent systems generates peta-scale data and require real-time massive analytics. Therefore, distributed design is required to be able to assure required throughput (as one of the TDAI metric) of each node and minimize crosstalk and contention bottleneck in total system [130].

Next sub chapter introduces basics of the distributed and hybrid design approach and compares advantages and disadvantages of each approach by correlating with expected throughput values (as one of the TDAI metric) of the nodes X(N) within the observed environment E {}.

*D. Distributed (**Edge/Hybrid**)*

Previous two approaches (1) centralized (2) decentralized ones can enable to build a joint knowledge base, is enough for most cases as compared in above Table.6. However, as the growth acceleration of the emerging intelligent systems increases exponentially, the third approach is important since distributed design becomes de facto paradigm for throughput level requirement of each node and the total system.

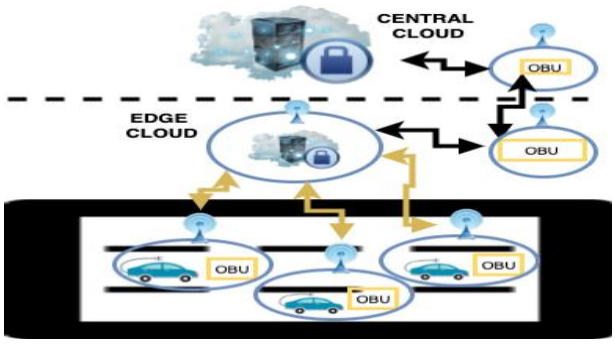

Figure 18. Distributed (Edge/Hybrid**) System**

Since the diversity of the components and number of interacted nodes increases exponentially, behavioral integrity of total system and transactions have to be assured in real-time to be able to keep the critical system constraints. Figure.18 illustrates multi-layer abstraction approach and distributed connectivity channels between the components. Each node has an on-board unit and an embedded trusted agent (out of the TDAI methodology requirement) to interact with the environment. Trusted channels ensure and maximizes connectivity of the components.

Trust factor coefficient-based throughput maximization methodology [130] can enable to build end-to-end trusted scalable channel within the components and total system. Throughput value

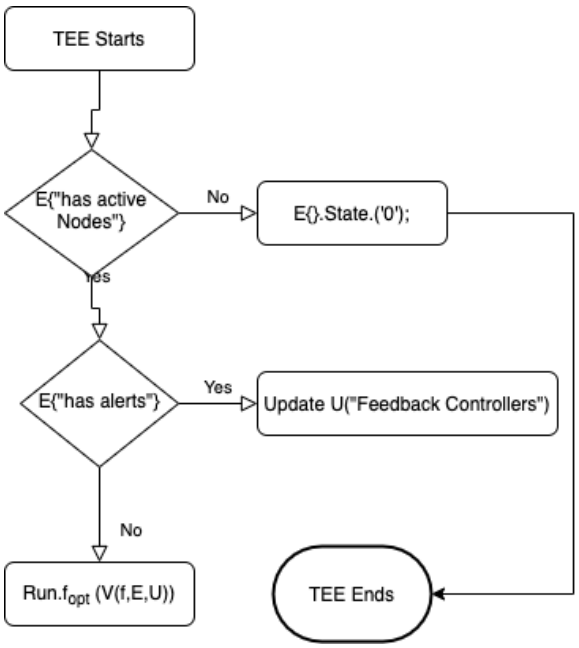

Figure 19.a End-to-end TEE Flow Diagram

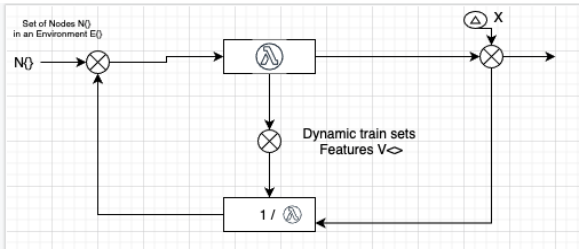

Figure 19.b High-level view of proposed Mechanism and learning approach for continuous growth

of each node X(N) is observed dynamically and the agent function $f_{opt} = arg\,\max_{f} V(f, E, U)$ operates continuously to assure coherency and continuous growth of a system.

Table.5 Introduces metrics, parameters, and regulative constraints observed in this study. Table.6 introduces basic comparative analysis and advantages/disadvantages of distributed design over other approaches with identified system features. Next sub chapter articulates how this approach of TDAI can be utilized to enable TEE based networking for trusted channel as SDN (Software Defined Networking) feature of the growth-flow mechanism as visualized in Figure.19.a/b.

E. TEE based networking for trusted channels as SDN (Software Defined Networks)

Increasing diversity of the components of the nodes $N_{E:}\{N_1, N_2, N_3, \ldots, N_n\}$ and data intensive nature of the systems require critical updates in networking paradigms also. Data-flow processes are improved with virtualized network functionalities, which have software-controller based switch and router design approaches [131], which is called software defined networking (SDN). The innovation enables dynamism for the growth-flow mechanisms of the emerging intelligent systems, which require (near) real-time interactivity constraints of the trusted agents. As the virtualized components increase in the systems, abstraction paradigms are also rethought such as holistic views [130]. The innovations enable to design end-to-end Trusted Execution Environment (TEE) as illustrated in Figure.13. to ensure interactivity within

the environment E {} more coherently. The proposed approach can enable to ensure network scalability and throughput maximization of emerging software defined networking-based systems.

Furthermore, it can be utilized to monitor systems and user behavior to minimize misbehaviors with a holistic view to the system [130]; such as, emission generated by cars and EMF generated by emerging computational/memory units of intelligent-systems with set of the nodes $N_{E:}$ {N_1, N_2, N_3, …, N_n } as dynamically controlled features of the growth-flow mechanism. Next sub chapter introduces security, privacy, trust metrics and package transmission approach of feature vector structures with TDAI and introduces theorems and hypothesis of TDAI methodology with distributed design paradigms.

F. Security, Privacy and Trust Metrics

As the diversity of the components increase, security and privacy are also considered as key trust metrics within TDAI. Security by design principles enable to design more robust and secure intelligent mechanisms. However, security constraints still cause dependency to a custom hardware design and limits software driven dynamic reconfiguration for adaptive systems.

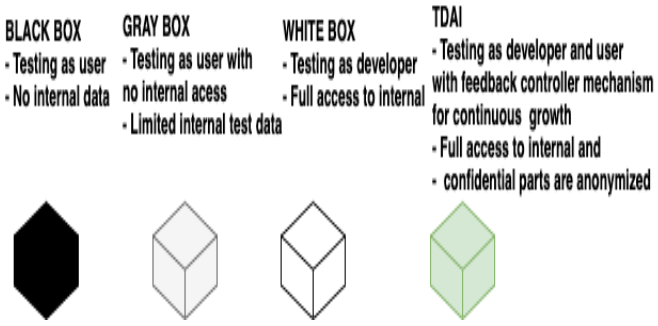

Figure.20. TDAI vs Black, Gray, and White Boxes

Fortunately, emerging TEE mechanisms like Open-TEE [132] can help to make the system software driven trusted mechanisms with dynamic compiling structures to any platform. Thereby, we can obtain measurable dynamic trust metrics as feature vectors within the transaction flow and package transmission processes; such as, checksum values of packages, trust factor of the nodes in the system, latency values of the transactions. See Table.4 and Table.5 for the identified metrics and features of TDAI for a package p<> based observation within the dynamic context.

Virtualized functionalities of software driven TEE ecosystems, can help to overcome limitations of hardware isolated TEE mechanism in state-of-the-art designs. Dynamic compiling and testing approaches, which are (1) black box (2) gray box (3) white boxes are

widely implemented for cryptography and testing features. Main differences are summarized in Figure.20.

However, emerging paradigms triggers architectural improvement need concerns also. For instance, decentralization of resources requires updates in many system layers in real time, which is only possible with distributed and hybrid design approaches. Thereby, TDAI can enable dynamic testing as user and developer with risk prediction and minimization via monitored set of network packages p<>, which have embedded feature to be compressed/decompressed in available check-points.

Distributed check-point mechanisms are dynamically correlated with throughput values of each node X(N), with respect to the regulative constraints identified in Table.5 as critical features of the observed environment E{} with set of nodes $N_{E:}$ {N_1, N_2, N_3, …, N_n }.

So that, we can state the hypothesis and theorems with TDAI with a comparative analysis on growth acceleration of the mechanisms, which have

(1) Centralized.

(2) Decentralized/autonomous/embedded.

(3) Distributed/hybrid architectural perspectives.

Figure.19.a illustrates the end-to-end TEE workflow diagram. The loop repeats continuously while the monitored set of Environments E{} has active nodes to ensure the growth-flow of the context.

Furthermore, dynamic optimizer function $f_{opt} = arg\max_{f} V(f, E, U)$maximizes expected utilities of each node via trusted agents $f_{Trusted\ Agent\{A\{\}\}}(E\{N[*]\})$.

However, interactivity of these agents is strongly dependent of embedded feature transmission via set of network packages p<>. Fortunately, these features can be compressed/decompressed

within reasonable latency thresholds of the emerging OBU mechanisms as feature vectors V<> via dynamically observed packages p<>.

$Theorem$:

So that, we can claim that it is possible to ensure the growth-flow of a dynamic environment with set of nodes $E\{N[*]\}$ with

- Centralized,
  - $N_{E:}$ {N_1, N_2, N_3, …, N_n } are fully connected to master node.
- Decentralized/autonomous/embedded
  - $N_{E:}$ {N_1, N_2, N_3, …, N_n } are not connected but has (near) real time connectivity feature to each node and master when required
- Distributed/hybrid-design perspectives of TDAI methodology.
  - $N_{E:}$ {N_1, N_2, N_3, …, N_n } are fully-connected to each other and master node also in real-time via edge node or any other available node to the master node when required.

The claim can be formalized as below in Equation 12.

Growth-flow of (1) Decentralized (2) Centralized (3) Distributed design can be arranged as below equation.12. Since, distributed design can enable to maximize total throughput of each node X(N) and total system.

Next chapter evaluates the methodology and introduces validation for the proposed theorem after a short proof of the statement below.

$$f_{DecentralizedGrowth\ of\ \mathrm{E[N]}}(N_1, N_2, \dots, N_n) < f_{CentralizedGrowth\ of\ E[N]}(N_1, N_2, \dots, N_n) < f_{DistributedGrowth\ of\ E[N]}(N_1, N_2, \dots, N_n) \quad (12)$$

$Proof$:

The proof for TDAI methodology can be correlated with the risk alerts minimized in the monitored context. It is observed that TDAI can enable to minimize the alerts in a dynamic context, for which the results are briefly introduced in next chapter and will be discussed in details within the related future works.

$$f_{NumberOfAlerts\ of\ Decentralized\ \mathrm{E\{N[*]\}}}(N_1, N_2, \dots, N_n) > f_{NumberOfAlert\ Centralized\ \mathrm{E\{N[*]\}}}(N_1, N_2, \dots, N_n) > f_{NumberOfAlert\ DistributedGrowth\ of\ \mathrm{E\{N[*]\}}}(N_1, N_2, \dots, N_n) \quad (13)$$

As formulated in above equation 13, we can say that number of alerts are minimized via TDAI, which have distributed/hybrid design, it outperforms other centralized and decentralized/autonomous design perspectives.

Minimized risk alerts of dynamic context, is illustrated above equation.13, proves that interactivity of the nodes and growth-flow of the context can be maximized with a distributed design rather than central and autonomous ones.

So that,

Throughput of each Node X(N); in an Environment E {} can be monitored in (near) real-time via trusted Agent A{} and set of nodes N{} can be improved continuously with dynamic growth-flow mechanism

$$f_{Trusted\ Agent\{A\{\}\}}(E\{N[*]\}) \quad (14)$$

Trusted Agent A {}

$\{iN_i$ and with activation function a_i $\}$

Each feature of the linked nodes is improved with activator level $a_{i,j}$ dependency;

Trusted Agent as N_i and linked activation function g () are triggered depending on the activation level of the a_i where set of nodes N {} monitored continuously via package p <> within the environment $E\{N[*]\}$,

$$N_i = \sum_{j=0..N} w_{ij}\, a_j \quad (15).$$

$$a_i = g(N_i) = g\left(\sum_j w_{ij}\, a_j\right).$$

Trust value t_{a_i} (0,1) of activator threshold level a_i monitored continuously where;

$$t_{a_i} = Average\left(\sum_0^i N_i\right), N\ \epsilon\ transaction\ flow$$

The activation levels are linked to optimal trust level of the nodes within the observed $E\{N[*]\}$ as dynamic growth-flow mechanism of TDAI methodology as stated in above equation.14 and equation.15.

IV.IV Conclusion

To sum up, we can state that the TDAI utilizes the MEMCA holistic abstraction with AI systems perspectives. Trust factor theorem [5] introduced novel holistic abstraction as extension to Amdahl's notation to improve the available abstraction approaches like neural networks. So that we can obtain dynamic holistic views of observed environment with set of nodes E{N[*]}, which have dynamic vector structure for input/output data steams to embed required features of the nodes N[*] via streaming packages p<*>. The abstraction and system node definition can be applied to many domains like networks and 5G technologies as compute intensive computer systems. Thereby, network functions can be virtualized (NFV) and utilized to improve SDN (Software Defined networking) features of the network systems, which are critical fog-layer components of holistic abstraction paradigms. So that latency risks can be eliminated and (near) real-time threshold limits can be succeeded by trusted interactivity between the nodes thanks to the maximized trust values of the nodes and the observed environment.

Next chapter introduces initial experimental validation results with trust factor coefficient theorem [130] based total system throughput maximization approach. In which, we experimentally validated and demonstrated promising performance of TDAI methodology within real-life use-cases of critical and autonomous applications to minimize the risk alerts within the observed context. The trust is measured with initial assumption and a default value assigned to the system node between 0-1 to identify the initial trust level-TRL [1-5]. Depending on the fulfillment of required expectations in the observed environment E{N[*]} the values are adjusted dynamically within the observed time span. The more features are justified in the context the more trust level is obtained. Furthermore, it is utilized to ensure the sufficiently trusted interactivity within the observed context. Main objective in real-life experimental validations is to maximize trust values and ensure growth-flow in the observed context with trusted intelligence-flow between the nodes [140,147], in which we demonstrated promising performance of TDAI methodology.

Chapter V: Evaluations and Experiments

Introduction

This chapter presents the thesis results in terms of evaluation of the TDAI as well as with the experiments conducted during the time frame of the thesis. The experiments were driven with some scenarios showing some critical use cases and applicable as much as possible to autonomous and distributed systems like in the context of smart city business cases.

A. Use-Cases Specifications and Descriptions

The scenarios targeted in this thesis is related to smart city use-cases that considers a mix of connected autonomous or semi-autonomous fixed or mobile nodes. In order to reflect the TDAI evaluation method, we focus on the main scenario use case describing the mobility and the financial related use cases.

Mobility Scenario Description:

In the context of mobility for instance, each node is represented by a vehicle (autonomous: SAE level 4/5 or semi-autonomous) which is deployed with a sensing unit as system node that has a processing and reasoning capability – to process the raw data collected from a vehicle's sensors and subsequently interpret them into useful outcomes of emerging AI systems. In this context, we can imagine a given number of nodes/vehicles on the road moving from a starting point to a destination with all vehicles connected to each other and sharing some data measured and/or interpreted locally, using their sensing capabilities, or collaboratively, using each other's knowledge. In such a situation, each node could implement its own AI module to estimate/predict or learn, not only from what has happened in the past, but also from what the other nodes are sharing with it (using its reasoning capability). In this context, we assume that for some tasks, a given node might rely on the processing power offered (vs allowed) by a remote node due to the lack of processing power or learning capability of the original node. In other words, vehicles that are not equipped with this AI feature or with a too weak computing capability can potentially rely on other vehicles' modules by using low-latency communication capabilities provided by P2P or cellular communication networks. This can be achieved via a collaboration feature that a distributed system can offer to give all mechanisms for the ability to run in real time. This heterogeneity (in the nature and capacities of vehicles) makes this collaborative approach particularly efficient, allowing a single node to benefit from the overall knowledge and processing capabilities. In such a scenario, we see a double (win/win) benefit, where a local node

can profit from neighboring nodes to help make decisions locally and anticipate decisions for the next steps.

The resulting distributed architecture can become complex and may involve self-organizing techniques with multiple hierarchical layers to better manage the decisions between several nodes. A node which might play the role of master would benefit from an overall view for global decision-making. Many applications might be related to this, especially those related to mobile, edge and ubiquitous computing where vehicles are equipped with context-aware and user-centric technologies.

Cross-Border Mobility Applications:

Applications that cover this could be related to driver behavior profiling, with different possible outcomes, like low-emission driving where the user or the car (if fully autonomous) would need to follow precise instructions depending on the way that is driving and, on the environment, (i.e. other vehicles).

One challenge of particular interest is when a mix of fully autonomous and semi-autonomous vehicles are collaborating. This specific scenario involves different behaviors and ways of sending, processing, and reacting to a given situation. This might of course lead to conflicting decisions, with semi-autonomous vehicles, still operated by a driver, sending requests or information that might be badly interpreted by the other vehicles. This type of scenario, when coupled with the complexity of the underlying distributed architecture, can lead to trust issues. This is even more true when AI algorithms are distributed over several disparate entities, since one of them may misinterpret an outcome interpreted by another vehicle. Another possible scenario is a complementary mobility application being related to an emergency where an ambulance can process data (related to early medical diagnosis) of the patient being transported, to be transmitted in real time before arriving at the hospital. In such a case, a patient's mobile device (e.g., smartphone or smartwatch) could be used for such a transmission via a roadside unit node (like a 5G edge node), which is utilized for transmitting the packets to the destination.

This process involves low latency and a high quality of service, as well as cooperation. Sometimes, it might indeed occur that we rely on such a node to request ad hoc computation if

an edge node (hospital edge node) is not available or not trusted anymore. In such a scenario, the main challenge is ensuring that the actors in the value chain trust the services at the top of such a distributed system since they mainly rely on AI systems capabilities: sensing, processing and reasoning features. In other words, what are the measurable trust indicators that can be considered to gauge confidence and correlate with trust in the overall services offered by such a distributed system?

Financial/Banking Monitoring Systems Related Use Case:

The scenario use-case is about global digital asset monitoring within a bank and accounts on top of which financial transactions are performed. These ones present some risks especially when the users are in mobility mode.

The mobility of the environment with set of nodes $E\{N[*]\}$ is observed to minimize the risk alerts and maximize the trust factor of the nodes with a focus on throughput values of each one to ensure expected throughput levels for required sufficient trust levels of the observed context.

Key Trust Indicators:

Looking at the trusted conceptual background provided by the literature, we can already specify the following key trust indicators that are specific to the distributed nature of a given intelligent and growing system:

- AI and the underlying distribution/architecture (versus organization) of its analytics functions: local learning (like in centralized systems), federated learning (like in distributed systems), etc.
- Processing over the nodes: processing power and organization within the distributed system
- Reputation of the distributed nodes, meaning how can we exclude any of the nodes if the overall reputation is degraded and how can we detect such a failure?
- Impact assessment of the overall trust value at both the node and global level.
- Measure/quantify/justify the trust factor coefficient of each node and the context dynamically in different time spans by observing the throughput levels expected for them.

B. Evaluations and Discussions

Previous chapters presented a comprehensive scientific background on emerging intelligent systems with a focus on TDAI concept and trust justification features. Details regarding the challenges and required main feature sets of the identified trust justification features are explained in details in Table.3. In this chapter, we focus on experimenting these features with regard to the challenges reflected in Table 7, these features are focused on the challenges between 2011-23.

Based on the explored literature, we can say that the system resource management principles and AI system perspectives introduced in previous chapters can enable continues growth for smart-system mechanisms with the set of nodes $N_i : \{N_o, N_1, N_2, \dots, N_n, \}$. Thus, in the TDAI evaluation we consider some critical features like robustness, resilience, reliability, and trust of the nodes to be ensured that are compliant to the observed AI systems research studies and aligned with the four main evaluation questions below:

- Is trust in distributed systems measured/quantified/justified?
- Is trusted scalability of autonomous systems achieved?
- Is trust for swarm intelligence mechanisms ensured?
- Is swarm system units is manipulated to implement trusted distributed AI methodology in (near) real time?

These questions can help us to understand how to build a growth-flow mechanism for emerging intelligent systems, which have dynamic and untrusted contexts. For this reason, the trusted distributed AI methodologies need to be implemented to maximize confidence and accelerate the growth of intelligent systems within the observed environment $E\{N[*]\}$. Table.7. gives details regarding the challenges and required main feature sets of the identified trust justification features are (1) Trusted scalability and elasticity for throughput maximization (2) Adversarial/un-adversarial thread monitoring and transaction approval (3) Simulation-based validation and verification (3) Monitoring (4) Detection and reaction with dynamic feedback-controllers for continuous growth-flow, for which experimental validation observations are explained in next sub-chapter.

RELATED WORKS AND MAIN FEATRUES	Trusted scalability and elasticity for throughput maximization	Adversarial/ un-adversarial threads	Simulation-based validation and verification	Monitoring	Detection and reaction with dynamic feedback-controllers for continuous growth-flow
[106] Data-Centric Operating System	✗	✗	✗	✓	✗
[107] A protected data-plane operating system for high throughput and low latency	✓	✓	✓	✗	✗
[108] A device for secure transaction approval	✗	✓	✓	✓	✓ Limited with end-to-end latencies
[5] A holistic abstraction to ensure trusted scaling and memory-speed trusted analytics	✓	✓	✓	✓	✓
[109] White Paper – Artificial Intelligence	✗	✓	✗	✓ Limited evaluation for abstraction levels	✗
[110,111,112,113] Adversarial/un-adversarial thread monitoring for transaction approvals.	✗	✓	✓	✓	✗
[114] Quantum cryptography features	✗	✓	✗	✓	✗
[115] BioDynaMo_ a general platform for scalable agent-based simulation	✗	✗	✓	✓	✗
[116] Survey on AI/ML accelerators.	✗	✗	✓	✓	✗
[117] SuperCloud Data sets and categorization of AI/ML workloads.	✓	✗	✓	✓	✗

Table 7. Artificial Intelligence System state-of-the-art and main research challenges between 2011-23.

In order to demonstrate the proof of concept for TDAI methodology, each critical metrics and parameters up to impact on throughput level of each node's trust factor [130] listed in Table.5 are correlated dynamically with expected throughput values of the nodes and the context. The correlations are utilized dynamically with TDAI risk minimization approach to ensure growth-flow within the observed environment $E\{N[*]\}$. Thereby, set of trusted agents $f_{Trusted\ Agent\{A\{\}\}}(E\{N[*]\})$ operates in dynamic context of the observed Environment E {} to monitor misbehaviors and maximize trust factor of each node and total system and generates risk alerts when there is impact on throughput levels of the monitored nodes N {}.

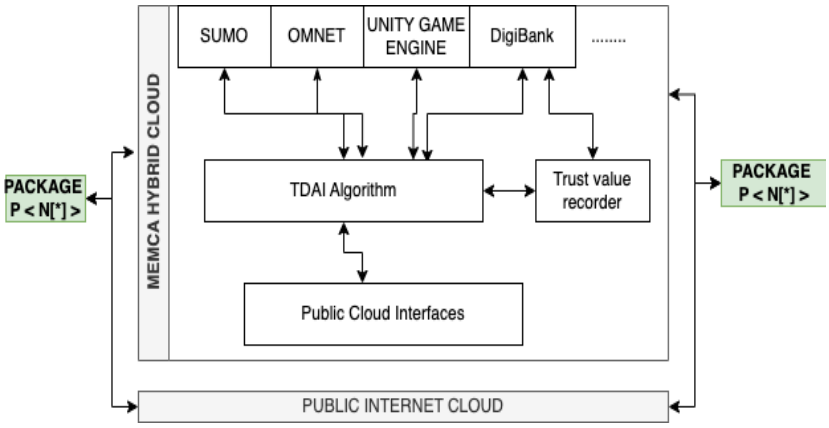

Figure.21.a High-level architecture of TDAI experimental validation setup and data streams.

Data streams are dynamically mapped to control nodes N [*] via package p<*> in unified vectorized structure as illustrated in Figure 21.a. Main objective is to maximize trust values of the observed context within the monitored time-span. Training data sets are mapped dynamically to transaction-flows as summarized in Table 8.

Data set are standardized as unified vector structure for experimental validations of TDAI in observed environment $E\{N[*]\}$ as below;

- E{"Scenario: Cross-Border Risk Monitoring, Time: DD.MM.YYYY", Nodes[*], FeatureVectors <*>}; .

Transmission capacities of the monitored nodes N{}. Critical metrics and parameters of package $p =< n_{id}, n_{st}, n_e, n_l, n_{id}, n_k, n_T, n_{chsum}, n_c, n_\rho, n_d, n_R >$, such as, throughput GBPs, EMF v/m, latency ms, are defined as critical risk resources n_R and activation threshold a_i within the EN[*]} and dynamic feature vectors

FeatureVectors<*>. Figure 22 summarizes the observation metrics/parameters and Table.8 have the results for alert sources in a dynamic environment with set of nodes $E\{N[*]\}$.

Trust factor coefficient theorem is utilized with a focus on 16.02.21-21.12.22 time-span for observations. Transaction flows linked to the nodes N [*] are observed via package p<*> analyzes with respect to it's impact to throughput and correlate with risk alerts to identify trust value of the linked nodes with a value between 0-1. The more risk the less throughput and the less trust for the nodes. MEMCA 5G connected hybrid cloud edge nodes have approximately 10GBPs max and 4G ones about 100 MBPs throughput levels.

The main objective is to maximize trust values of the critical nodes and the observed environment within the monitored time-span with a focus on detection of critical risk alerts. FIGURE 23 illustrates MEMCA-Hybrid cloud for generic smart city emulation and TDAI trusted interactivity experimental observation with sample DigiBank monitoring application [7].

Package $p =< n_{id}, n_{st}, n_e, n_l, n_{id}, n_k, n_T, n_{chsum}, n_c, n_\rho, n_d, n_R >$, transmissions are observed via trsuted agents of the monitored nodes N{}. Any metric can be measured and extracted to detect risk resources n_R and depending on the activation threshold a_i within the EN{[*]} via available physical sensors of the nodes. 1 physical MEMCA mobile node can record required observations. Table 8 summarizes experimental validation results and summarizes the data sets of the observation results.

So that dynamic vector structure can enable to merge the observed features and correlate with risk alerts as visualized in Figure 22 TDAI Sample experimental observation. Thereby, the impacts on throughput level of the nodes and the context are monitored with dynamic holistic views. Experimental validation setup is also visualized in Figure 23 where we evaluated experimentally validated TDAI [5,147].

The experimental results are obtained based on the deployment of some commercial tools with focus on DigiBank hybrid cloud, in which we experimentally validated TDAI methodologies as well. The tests were achieved following the scenario use cases that we considered. The obtained results are reflected in the following part of this sub-chapter.

Experimental Validation of the Financial/Banking Monitoring Systems Use Case:
The time span and observation metric/parameters can be limited to scope of Figure 22 to demonstrate and experimentally validate TDAI performance as in the defined cross-border smart-city financial transactions focused scenario use case as described in the previous section.

The scenario for the risk observations are based on a DigiBank Intracontinental hybrid-cloud use-case for global digital asset monitoring within a bank and an account based on package p<> transmission traffic-based risk alerts within the environment E {}. The mobility of the environment with set of nodes $E\{N[*]\}$ is observed to minimize the risk alerts and maximize the trust factor of the nodes with a focus on throughput values of each one to ensure expected throughput levels for required sufficient trust levels of the observed context [5].

The monitoring periods are in limited time-span and utilized between the USA-TÜRKİYE-EU during cross-border mobility evaluations of the bank account transaction flow analysis as summarized in Figure 22 holistic views.

Configuration of the Experiment:
The number of nodes considered in the experimentation has been fixed to 29 which is sufficient for scale up and therefore meets the TDAI evaluation objectives. The observation has been achieved with a focus on the period from 16.02.2021 to 21.12.2022. Out of the 29 nodes, 1 node is a physical mobile node which is linked to the other nodes via public internet cloud. The 28 other nodes are simulated with custom designed smart city emulator which might be similar or different depending on the system functional requirements.

From the functional view point, such a distributed system composed of the 29 nodes is focused on financial banking transactions such a money transfer via for instance banking clearing systems. Such a use case is a perfect scenario for TDAI evaluation, since it requires a high level of trust by nature due to its global legal constraints. Such nodes are basically exchanging some data on the business needs based on some clearing protocols being deployed on fixed or mobile nodes. The observation of such a system in operation requires the application of TDAI nodes monitoring which is done dynamically in (near) real-time basis using package p<> transmission among the different nodes.

The main objective is to maximize trust values of the critical nodes and the observed environment within the monitored timespan with a focus on detection of critical risk alerts. The number of transactions exchanged among the nodes might have an impact on the TDAI features like throughput which has a direct correlation with risk alerts to identify trust level of the linked nodes with a value between 0-1. In fact, the more risk is high the less throughput value is and therefore the less trust value is on the nodes. It's impact to throughput level of the node is observed with respect to critical metrics and parameters of package $p =< n_{id}, n_{st}, n_e, n_l, n_{id}, n_k, n_T, n_{chsum}, n_c, n_\rho, n_d, n_R >$, such as, throughput (in GBPs), EMF (in v/m), latency (in ms). Here we have some contextual data that are considered in the case of MEMCA 5G: the connected hybrid cloud edge nodes have approximately 10GBPs max and 4G ones about 100 MBPs throughput levels.

TDAI Features Data Collection:

The TDAI observing system exploits the concept of *package p<> transmission* implementable via trusted agents transmission capacities of the monitored nodes N{}. Critical metrics and parameters of *package* $\boldsymbol{p =< n_{id}, n_{st}, n_e, n_l, n_{id}, n_k, n_T, n_{chsum}, n_c, n_\rho, n_d, n_R >}$, such as, throughput (in GBPs), EMF (in v/m), latency (in ms), are defined as critical risk resources in the experimented case. Any metric can be extracted via available physical sensors located at the nodes level. The 1 physical MEMCA mobile node records the required observed data.

From the technical viewpoint, TDAI observation system is exploiting the concept of probes that are linked to sensors associated to the physical or virtual nodes. The observation is rather dynamic which can allow real time detection of risks during the system in operation within the specified timespan. TDAI methodology requires that all metrics are embedded to the nodes of *package p* <> and each one has its specific embedded detector e.g. SAR, power, etc.

Experimental Validation of the Mobility Use Case:

The risks metrics and parameters are broadened with further parameters of a custom designed simulator for the simulation of accident detection and signal propagation analysis as represented in Figure 23 with dynamic holistic views of the observed environment $E\{N[*]\}$.

Dynamic holistic views are updated dynamically with data sensor fetching as illustrated in Figure.19 a with dynamically updated feedback controllers. Furthermore, features embedding mechanism also ensure the continuous growth of the context depending on throughput values of each node X(N) with respect to $1/\lambda$ as indicated in Figure.19. b.

The Architecture Developed:

Furthermore, Figure 21.b illustrates the TDAI microservice architecture, is designed to ensure and maximize interactivity of the trusted agents. It introduces the critical parameters and the correlation approach to minimize the false-positive alerts of the dynamic environment to maximize growth-flow of the observed context.

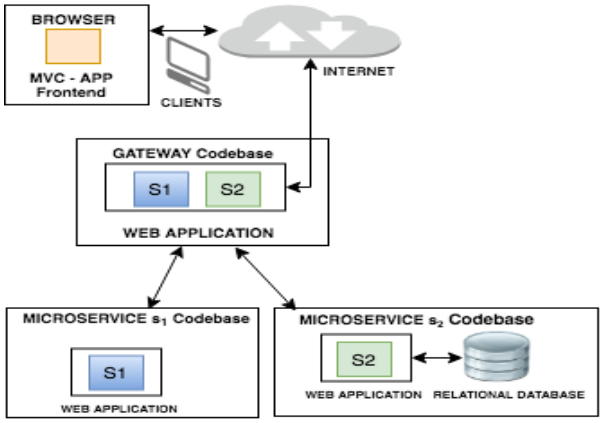

Figure.21b. TDAI Micro-service architecture for trusted interactivity.

The target for the observation within the $E\{N[*]\}$ is defined as detection of critical risk alerts with TDAI utilization and porting those to growth-flow mechanism of TDAI approach via package p<> transmission to dynamically train the DigiBank 5G connected hybrid-cloud system as sample critical commercial technology tool we used during the thesis experimentation. It has 5G connectivity features also for (near) real-time package transmission capacities of the monitored nodes N{}, where we observe package p<> transmission to extract the critical metrics and parameters when required.

Critical metrics and parameters of package $p =< n_{id}, n_{st}, n_e, n_l, n_{id}, n_k, n_T, n_{chsum}, n_c, n_p, n_d >$; such as, throughput GBPs, EMF v/m, latency ms, are defined as critical risk resources n_R and activation threshold a_i within the $E\{N[*]\}$. Next sub chapter articulates the correlation and alerting approach utilized for the observations.

As stated in trust factor coefficient theorem [130], trust value of each node is correlated dynamically to throughput and the metrics, which have direct impact on it. Furthermore, it is proportional to generated risk alerts n_R from the nodes as formulated in equation 16.

Additionally, Table.5 illustrates the monitoring metrics/parameters of each node and critical identified regulative constraints, which are EMF, SAR and Power values of each node and the observed dynamic context. Low emission driving constraint defines a trusted system as a **system that cannot exceed/generate a certain level of emissions**.

To do so, it has to take care of the following elements, and find countermeasures, when necessary (e.g. with optimization), so that overall trust value of each node controlled and correlated to risk alerts in (near) real time. Equation.16 below formulates the risk monitoring model and trust value correlation of an environment $E\{N[*]\}$.

$$t_{N_i} = Average\left(\sum_{0}^{i} \frac{N_{i.t\ trust\ value\ t \in \{0\ to\ 1\}}}{N_{i.R\ (total\ risk\ alert\ number)}}\right),$$

$$N\epsilon\ the\ nodes\ in\ observed\ Environmment\ \boldsymbol{E}\{\boldsymbol{N}[*]\}\ (16).$$

As stated in above Equation.16, the trust values are directly proportional to risk alerts of resources as set of nodes N {}. Thereby, the metrics have impact on the risk alerts can be defined as other critical constraints of the observed context. By that means, critical alerts from legal constraints violations are the major trust indicators and threshold bounds a_i have direct impact of throughput level of the nodes and linked trust values. Table.8 visualizes list of alert resources within the observed 4G and 5G traffics.

TDAI performs promising performance of detection of the legal constraint based critical alerts, which results in account termination and bank investigations with %100 ratio for the observations of the main transactions linked to monitored nodes. On the other hand, further parameters and metrics can be defined with custom designed simulator as visualized in Figure 23. Below are the some of the other critical observation metrics for TDAI methodology for further risk analyzes and observations;

- The **behavior of the driver**, which directly impacts the emission level. It can be measured by inferring the **acceleration** of the car, which obviously depends on the way the user accelerates or decelerates. With this value it is possible to deduce how much a driver is aggressive, slow, etc. This acceleration profile can be computed with GPS, RPM (Revolution Per Minute) or the smartphone's accelerometer for instance depending on the position of the device in the car. The driver him/herself is also important, since the age, driving habits and experience are all factors that obviously influence the behavior.
- The **vehicle itself**, and most specifically its maintenance and type (e.g., age, engine, etc.) – A old vehicle for instance usually generates more emissions that a recent one.
- **Environmental conditions**: weather, road traffic, etc. can affect the emissions generated by tires, brakes and exhaust emissions.
- The **profiling and recommendations systems** that are embedded in the app and that are only using local routines, so only a limited information knowledge that can easily be influenced negatively – thus biasing the recommendations.
- The **connectivity part** used to transmit the data – if false data is sent, then the models will not be accurate.

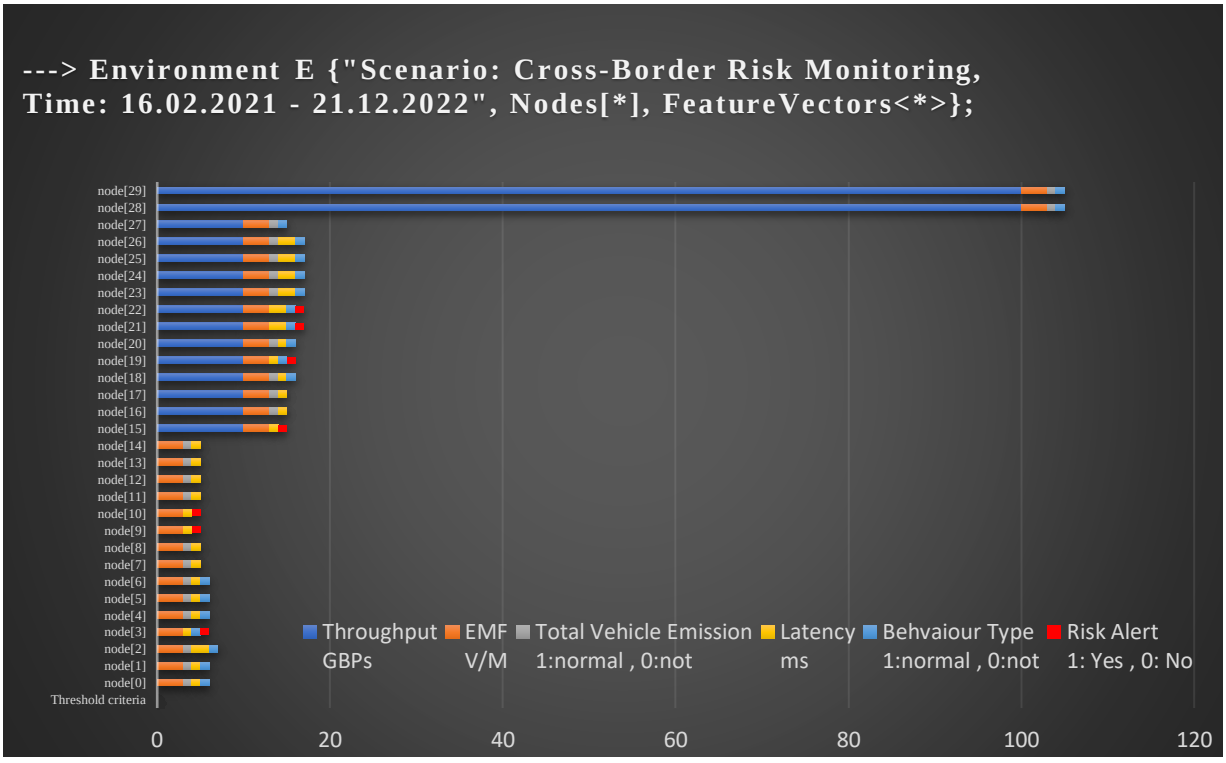

Figure.22. TDAI sample experimental observation.

Furthermore, other metrics from data sensors e.g. data collected through the OBD dongle includes Gas pedal position%, RPM (Revolution Per Minute), Gear position, Fuel consumption, Mass Air Flow (MAF), NOx sensor, Vehicle speed, Engine Coolant Temperature, Steering wheel angle, Catalyst Temperature Banks & sensors, Air pressure, Engine out NOx emission are defined as critical constraints of TDAI.

The more metrics are observed the more trust level between TRL 1-5 can be assured and justified proportionally in a dynamic context with trusted agents based continuous growth-flow mechanism in the monitored context via $f_{Trusted\ Agent\{A\{\}\}}(E\{N[*]\})$. However, each feature causes a justification process and increases the trust cost exponentially as formulated in Figure1.a. Therefore, TDAI limits the critical constraint and metrics as selected in Table 8. The more trust value is depending on the risk alerts in the context which have impact on required throughput values of the nodes as summarized in Figure 22 with a holistic view to the context within the selected time frame.

Dynamic holistic views can help to accelerate the growth-flow with trusted feedback structures, which have (1) Centralized (2) Decentralized/autonomous (3) Distributed/Hybrid design perspectives. Since, distributed design can enable to maximize total throughput of each node

Figure.23. MEMCA-Hybrid Cloud for generic smart city emulation and TDAI trusted interactivity experimental observation with sample DigiBank monitoring application [5].

X(N) and total system with utilization of TDAI with maximized growth-flow of the observed environment $E\{N[*]$.

To sum up, it is observed that in initial simulations of TDAI, as briefed in Figure 23, which has trusted interactivity since it can be assured at massive scale with minimum risk alerts as summarized in Table.8. TDAI can merge the feedbacks in (near) real time and detect the risk with 70% true positive detection performance with scaled up alerts and trustfully utilized MEMCA holistic views [130] for accelerated growth-flow mechanism.

Alert Sources	**Intersection/Convergence Points Accident Risk Alerts for Snapshot.E{"16.02.2021 – 21.12.2022"};**	
EMF	**Transaction Flow Traffic Size:** 4G/ ~100 MBPs – 5G/ max. 10GBPs **Fatal Alerts:** 7	
Emission	**True Positive**	**False Positive**
Bandwidth Congestion	**70%**	**20%**
System Throughput Limit	**False Negative**	**True Negative**
Saturation Effect	**5%**	**5%**
Legal Constraints		

Table 8. Alert sources in a dynamic environment with set of nodes $E\{N[*]\}$.

MEMCA 5G connected hybrid-cloud is utilized to develop generic smart city emulation and ensure trusted interactivity with TDAI for a sample DigiBank financial transaction monitoring application. So that, we can ensure optimal trust in observed Environment E {Nodes[*]['Alerts']} and correlate potential risks with trust values of each nodes.

Thereby, minimized risk alert and maximum growth-flow performance as summarized in Figure 22 and visualized in Figure 23, in which each node and total system can succeed the expected throughput levels in (near) real time thanks to TDAI based trusted interactivity of the context and maximized growth-flow performance with dynamic holistic views to the system. The more we justify trust the more accelerated growth-flow can be succeeded. Next sub-chapter concludes the chapter before the thesis conclusions and future works introductions.

B.III Chapter Conclusion

To sum up, we can state that three main architectures (central, decentral/autonomous, distributed/hybrid) can enable to build basis for TDAI methodology to ensure end-to-end trust in holistic AI system life cycle. Distributed/hybrid designs can enable to improve total system performance and ensure growth-flow in dynamic context. So that, computational algorithm can be decentralized with task cooperation approaches and data caching policies with trust factor coefficient based total system throughput maximization approach and TDAI based multi-agent system with maximized interactivity.

Any other critical constraints, which have impact on expected throughput values of the nodes are defined as risk alerts and critical constraints of TDAI. Thereby, main sources of the alerts can be detected in (near) real time and ported to growth-flow paths with trusted feedback controller structures. So that defined critical constraints summarized in Table.8, e.g. EMF, bandwidth congestion, system throughput limits, saturation effects can be detected and false-positives values can be minimized for optimal system resource management. The more trust in the context, the less alerts generated in observed context and continuous growth-flow can be assured for maximum trust values of an environment with set of nodes $E\{N[*]\}$.

Chapter VI: Conclusion and Future Work

A. Summary of The Main Findings

As a brief conclusion of the research challenges explored, we ascertain that TDAI is seen as a missing, and yet, largely unexplored area. TDAI core component of the methodology has three main architectural perspectives (1) Centralized. (2) Decentralized/autonomous/embedded (3) Distributed/hybrid architectural perspectives. So that we can assure the trusted interactivity of the nodes within the observed environment $E\{N[*]\}$ with optimal trust cost. Critical feature sets of the TDAI can be summarized as summarized in Table.2, that is:

(1) trust measurement, quantification, and justification
(2) trusted scalability
(3) trust assurance
(4) swarm manipulation
(5) system and user behavior monitoring.

The feature sets elaborated in this thesis help to ensure the continuous growth of the intelligent systems, which are core mechanisms of emerging smart ecosystems, with software-driven dynamic systems to ensure adaptiveness and flexibility in a dynamic context, rather than hardware-dependent designs. So that we can obtain a dynamic end-to-end trust mechanism to justify the required level of trust with optimal cost. Thereby, the trust factor of the system, $P(x^*) \propto t$ with the set of nodes $N_E : \{N_1, N_2, N_3, \ldots, N_n\}$ can be increased to maximize the throughput in the well-defined dynamic context with the adaptive agent function $f_{opt}()$. The critical feature sets selected in Table.3.B can be considered as the key elements of the end-to-end trust mechanism for the continuous growth of intelligent systems. The better the features justified, the faster the growth for the dynamic objectives of the mechanisms is ensured.

As the TDAI experimentally validated and evaluated in previous Chapter V, we can rely on the promising performance of the novel methodology to ensure interactivity and sufficient level of trust in a dynamic environment E{}. Architectural design principles are critical concerns for the novel innovations in the growing context. Decentralized approaches can build autonomous/embedded/local components with basic functionalities like swarm manipulation. In order to improve the components with trusted scalability and monitoring features, end-to-end fully connected channels are required. Centralized designs can guarantee these features

with respect to end-to-end latency limits. Identified advanced justified functionalities are required for the novel futuristic designs, which are possible with the distributed design paradigms, which include edge/hybrid/hierarchical/multi-layer features with emerging holistic abstraction principles [31,14,15].

These features and research challenges are also included in the growth-flow of the emerging intelligent systems with advanced trusted AI capabilities. Next chapter introduces potential research challenges and future directions within a summary table. The challenges summarized in Tables.2 and 3 briefly introduced future directions to be explored in related future works.

B. Potential TDAI Research Fields and Future Directions

Disruptive innovations proposed for growing intelligent systems trigger acceleration to obtain an end-to-end fully trusted execution environment, which can be operated in the distributed context within the limits of critical systems constraints. However, the limitations of the decentralized components can only provide basic functionalities, like swarm manipulation features. These features can be improved with decentralized designs and hybrid mechanisms for the recent challenges. Table.9 introduces major points from the related works and reviewed literature between 1950 and 2023, with a focus on recent years. These challenges remain open issues to be explored in detail to obtain a fully trusted execution environment for growing intelligent systems with dynamically correlated and observed socio-dynamic features, which mainly focus on the regulative legal measurement metrics of alerting methodologies. These identified challenges will be investigated in detail in future related works.

In the TDAI research field, we identify the following emerging areas as being of increasing interest within the **distributed computing communities:**

- Trust measurement, quantification, and justification in distributed systems and its underlying diverse components.
- Trusted scalability of autonomous systems with algorithms and system levels with end-to-end holistic views.

- Trusted architecture mechanisms (e.g. Machine Learning) with novel abstraction approaches of end-to-end paradigms.
- (Near) Real-Time swarm manipulation to implement trusted distributed AI methodology.

The challenges and future directions identified in this thesis can be defined as key features and milestones for massive scale trusted AI. Thereby, an intelligence flow can be assured for growing intelligent mechanisms via end-to-end trust mechanisms, which have TEE based trusted interaction with the environments within the smart-ecosystems and dynamic contexts. The more features justified within the critical systems constraints, the more trust can be obtained with TDAI for the growing intelligent systems.

On the other hand, in spite of the major progress made in computing systems with distributed design innovations, there are still challenges for the critical system constraints for the continuous growth of the intelligence systems. For instance, in [37] experiments have recently reported with trusted computing paradigms in real life use-cases. In [81] high throughout mobile and wireless communication technologies have recently been replaced with tethered ones. More critically, [117] HPC limitations are still set to be improved with modern AI/ML frameworks with hybrid cloud design paradigms.

Fortunately, defined architectural perspectives (central, decentral/autonomous, distributed/hybrid) for emerging trusted distributed AI mechanisms can enable to ensure resiliency and robustness in a dynamic context with an end-to-end TEE for growing intelligent mechanisms and systems. Furthermore, the trust measurement, quantification, and justification methodologies can be applied in emerging distributed systems and their underlying diverse application domains with TDAI.

As conclusion we can summarize the future perspective of the thesis as follows;

1. TDAI Experimentation and Application contexts: TDAI approach can further improve the distributed ML systems and justify the trust features for such applications, which has not been fully tested in all contexts during this thesis. Examples of experimentations contexts are detailed in the following:

- Smart city use-cases (such as driver behaviors for smart mobility) with AI systems considerations like consciousness of machines, as well as end-to-end trust mechanism for swarm systems.
- Development and operation of AI systems with increasing trust for holistic DevOpS lifecycle which expect to bring together the Larger-scale data amounts and high number of users.
- Coordination-based systems, caching policies, and middleware mechanisms to be investigated. Security/Privacy/Trust support are also limited aspects of proposed solutions.
- Further trust formalizations and understandings to be identified for the TDAI use-cases specific experimental validation in different domains. Other approaches where feature interaction problems are important like interactive multiagent systems to be further improved with the TDAI justification features.

2. TDAI Framework Design and Tooling: Framework design and experimentation are also ongoing activity and TDAI methodology can support the frameworks and state of the art system with the identified novel features. This can enable further application fields of TDAI as a justification feature sets within the holistic views with trust factor coefficient theorem and pertaining abstraction paradigms. Frameworks can be improved and tested with large scale use-cases for experimental validation activities in related applications. Furthermore, DAI and (MAS) multi-agent systems can also be improved with TDAI feature sets with improved abstraction levels. So that, agent features to be broadened and distributed design paradigms to with improved abilities.

3. TDAI Standardization: TDAI Impact can be at the standardization which might require its evaluation in many industrial applications and therefore the trust metrics can be adapted to fit to the standardized requirements. More particularly the development perspectives can bring some benefits as listed in the following:
   - TDAI improvement with customized trust metrics and performance indicators.
   - TDAI method and tools could be further developed with the perspectives of exploiting them for Auditing and Certification potential where fair, explainable, auditable, and

safe futures to be explored in different stages of a system lifecycle, with each stage forming part of a Chain of Trust.

- Improvement of the TDAI evaluation matrix associated to the TDAI taxonomy via further trust formalization implementations and understandings.

4. TDAI Automation for Decision Making: As another aspect, TDAI can help minimization of humans in loop and real-time decision-support limitations to increase automated features of growing AI systems. There are strong requirements for TDAI to be utilized in automation decision making processes, which is a part of TDAI trust justification process that will be experimented in related future works.

Further details of the additional parts for the identified future directions, which will be explored and experimented in our related future works justified in details in [140], where there are strong requirements for further TDAI implementations.

References

[1] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2017). Automatic differentiation in machine learning: a survey. *The Journal of Machine Learning Research*, *18*(1), 5595-5637.

[2] Kraska, T., Talwalkar, A., Duchi, J. C., Griffith, R., Franklin, M. J., & Jordan, M. I. (2013, January). MLbase: A Distributed Machine-learning System. In *Cidr* (Vol. 1, pp. 2-1).

[3] Milutinovic, M., Baydin, A. G., Zinkov, R., Harvey, W., Song, D., Wood, F., & Shen, W. (2017). End-to-end training of differentiable pipelines across machine learning frameworks.

[4] Cohen, R., Schaekermann, M., Liu, S., & Cormier, M. (2019, May). Trusted AI and the contribution of trust modeling in multiagent systems. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems* (pp. 1644-1648). International Foundation for Autonomous Agents and Multiagent Systems.

[5] Agca, M. A. (2019, December). A Holistic Abstraction to Ensure Trusted Scaling and Memory Speed Trusted Analytics. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 1428-1434). IEEE.

[6] Nassar, M., Salah, K., ur Rehman, M. H., & Svetinovic, D. (2020). Blockchain for explainable and trustworthy artificial intelligence. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *10*(1), e1340.

[7] Kumar, A., Braud, T., Tarkoma, S., & Hui, P. (2020). Trustworthy AI in the Age of Pervasive Computing and Big Data. *arXiv preprint arXiv:2002.05657*.

[8] Hummer, W., Muthusamy, V., Rausch, T., Dube, P., El Maghraoui, K., Murthi, A., & Oum, P. (2019, June). Modelops: Cloud-based lifecycle management for reliable and trusted ai. In *2019 IEEE International Conference on Cloud Engineering (IC2E)* (pp. 113-120). IEEE.

[9] Martínez-Fernández, S., Franch, X., Jedlitschka, A., Oriol, M., & Trendowicz, A. (2020). Research Directions for Developing and Operating Artificial Intelligence Models in Trustworthy Autonomous Systems. *arXiv preprint arXiv:2003. 05434*.

[10] Bottou, L., Curtis, F. E., & Nocedal, J. (2018). Optimization methods for large-scale machine learning. *Siam Review*, *60*(2), 223-311.

[11] Lee, K., Lam, M., Pedarsani, R., Papailiopoulos, D., & Ramchandran, K. (2017). Speeding up distributed machine learning using codes. *IEEE Transactions on Information Theory*, *64*(3), 1514-1529.

[12] Hull, B., Bychkovsky, V., Zhang, Y., Chen, K., Goraczko, M., Miu, A., ... & Madden, S. (2006, October). Cartel: a distributed mobile sensor computing system. In *Proceedings of the 4th international conference on Embedded networked sensor systems* (pp. 125-138).

[13] Russell, S., & Norvig, P. (1995). A modern, agent-oriented approach to introductory artificial intelligence. *ACM SIGART Bulletin*, *6*(2), 24-26.

[14] Survey, van Steen, M., & Tanenbaum, A. S. (2016). A brief introduction to distributed systems. Computing, 98(10), 967-1009.

[15] Agca, M. A. et al. (2020, December). Challenges for Swarm of UAV/Drones Based Intelligence. In PDPTA'20- The 26th Int'l Conference on Parallel and Distributed Processing Techniques and Applications

(on press), Springer Nature - Book Series: Transactions on Computational Science & Computational Intelligence.

[16] Velthuijsen, H. (1993). Distributed artificial intelligence for runtime feature-interaction resolution. *Computer*, *26*(8), 48-55.

[17] Stone, P., & Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, *8*(3), 345-383.

[18] Toreini, E., Aitken, M., Coopamootoo, K., Elliott, K., Zelaya, C. G., & van Moorsel, A. (2020, January). The relationship between trust in AI and trustworthy machine learning technologies. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (pp. 272-283).

[19] Marsh, S. (1992, July). Trust in distributed artificial intelligence. In *European Workshop on Modelling Autonomous Agents in a Multi-Agent World* (pp. 94-112). Springer, Berlin, Heidelberg.

[20] Agha, G. A., & Jamali, N. (1999). 12 Concurrent Programming for Distributed Artificial Intelligence.

[21] Alan, M. (1950). Turing. Computing Machinery and Intelligence. Mind LIX, 2236: 433-460.

[22] McCarthy, J., Minsky, M., Rochester, N., & Shannon, C. (1956). Dartmouth conference. In *Dartmouth Summer Research Conference on Artificial Intelligence*.

[23] McCarthy, J. J., Minsky, M. L., & Rochester, N. (1959). *Artificial intelligence*. Research Laboratory of Electronics (RLE) at the Massachusetts Institute of Technology (MIT).

[24] McCarthy, J. (1989). Artificial intelligence, logic and formalizing common sense. In *Philosophical logic and artificial intelligence* (pp. 161-190). Springer, Dordrecht.

[25] McCarthy, J. (1998). What is artificial intelligence?.

[26] McCarthy, J. (2000). Concepts of logical AI. In *Logic-based artificial intelligence* (pp. 37-56). Springer, Boston, MA.

[27] Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2013). Context aware computing for the internet of things: A survey. *IEEE communications surveys & tutorials*, *16*(1), 414-454.

[28] Mitchell, C. (Ed.). (2005). *Trusted computing* (Vol. 6). Iet.

[29] Ağca, M. A., Ataç, Ş., Yücesan, M. M., Küçükayan, Y. G., Özbayoğlu, A. M., & Doğdu, E. (2016). Opinion mining of microblog texts on Hadoop ecosystem. *International Journal of Cloud Computing*, *5*(1-2), 79-90.

[30] Gadepally, V., Goodwin, J., Kepner, J., Reuther, A., Reynolds, H., Samsi, S., ... & Martinez, D. (2019). AI Enabling Technologies: A Survey. *arXiv preprint arXiv:1905.03592*.

[31] Eberhart, R. C., Shi, Y., & Kennedy, J. (2001). *Swarm intelligence*. Elsevier.

[32] Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., & Zhu, J. (2019, October). Explainable AI: A brief survey on history, research areas, approaches and challenges. In *CCF International Conference on Natural Language Processing and Chinese Computing* (pp. 563-574). Springer, Cham.

[33] Wing, J. M. (2020). Trustworthy AI. *arXiv preprint arXiv:2002.06276*.

[34] Sileno, G., Boer, A., & van Engers, T. (2018). The role of Normware in Trustworthy and Explainable AI. *arXiv preprint arXiv:1812.02471*.

[35] Oriesek, D. F. (2022). The Potential Impact of Artificial Intelligence on Preventive Diplomacy from a Balance-of-Threat Perspective (Doctoral dissertation, Harvard University).

[36] Martínez-Fernández, S., Franch, X., Jedlitschka, A., Oriol, M., & Trendowicz, A. (2020). Research Directions for Developing and Operating Artificial Intelligence Models in Trustworthy Autonomous Systems. *arXiv preprint arXiv:2003.05434*.

[37] Chen, L., Mitchell, C. J., & Martin, A. (Eds.). (2009). Trusted Computing: Second International Conference, Trust 2009 Oxford, UK, April 6-8, 2009, Proceedings (Vol. 5471). Springer.

[38] Kuo, B. C., & Golnaraghi, F. (2010). Automatic control systems. JOHN WILEY & SONS, INC.

[39] (Gambetta90) D. Gambetta. Can We Trust Trust?. In, Trust: Making and Breaking Cooperative Relations, Gambetta, D (ed.). Basil Blackwell. Oxford, 1990, pp. 213-237

[40] (Dorodchi 2016) M Dorodchi, M Abedi, B CukicTrust-based development framework for distributed systems and IoT - 2016 IEEE 40th Annual, 2016

[41] (Shrien 2011) Guido Shrien, Melanie Volkamer, Sebastian Ries, A formal approach towards measuring trust in distributed systems; SAC '11: Proceedings of the 2011 ACM Symposium on Applied Computing; March 2011 Pages 1739–1745

[42] Proper, H. (2020). Data Ecosystems-Fuelling the Digital Age. In *VMBO* (pp. 1-4).

[43] Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific american*, *284*(5), 34-43.

[44] Antoniou, G., & Van Harmelen, F. (2004). *A semantic web primer*. MIT press.

[45] Lehr, T., Lorenz, U., Willert, M., & Rohrbeck, R. (2017). Scenario-based strategizing: Advancing the applicability in strategists' teams. *Technological Forecasting and Social Change*, *124*, 214-224.

[46] Aksit, M. (2019). SOFTWARE ENGINEERING CHALLENGES IN REALIZING SMART-CITY SYSTEMS.

[47] Taylor, M. B., Kim, J., Miller, J., Wentzlaff, D., Ghodrat, F., Greenwald, B., ... & Agarwal, A. (2002). The raw microprocessor: A computational fabric for software circuits and general-purpose programs. *IEEE micro*, *22*(2), 25-35.

[48] Agarwal, A., Mitra, A., Joshi, P., & Rastogi, K. (2019). *U.S. Patent No. 10,248,566*. Washington, DC: U.S. Patent and Trademark Office.

[49] Wentzlaff, D., Griffin, P., Hoffmann, H., Bao, L., Edwards, B., Ramey, C., ... & Agarwal, A. (2007). On-chip interconnection architecture of the tile processor. *IEEE micro*, *27*(5), 15-31.

[50] Mutlu, O., & Moscibroda, T. (2007, December). Stall-time fair memory access scheduling for chip multiprocessors. In *40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007)* (pp. 146-160). IEEE.

[51] Mutlu, O., Ghose, S., Gómez-Luna, J., & Ausavarungnirun, R. (2020). A Modern Primer on Processing in Memory. *arXiv preprint arXiv:2012.03112*.

[52] Agarwal, A. (1991). Limits on interconnection network performance. *IEEE transactions on Parallel and Distributed Systems*, *2*(4), 398-412.

[53] Mutlu, O. (2020). Intelligent Architectures for Intelligent Computing Systems. *arXiv preprint arXiv:2012.12381*.

[54] Agarwal, A., Hennessy, J., & Horowitz, M. (1988). Cache performance of operating system and multiprogramming workloads. *ACM Transactions on Computer Systems (TOCS)*, *6*(4), 393-431.

[55] Agarwal, Anant (Weston, MA, US) 2013 ONLINE DISTRIBUTED INTERACTION United States edX Inc. (Cambridge, MA, US) 20130204942, https://www.freepatentsonline.com/y2013/0204942.htmlaaa

[56] Alizadeh, M., Greenberg, A., Maltz, D. A., Padhye, J., Patel, P., Prabhakar, B., ... & Sridharan, M. (2010, August). Data center tcp (dctcp). In *Proceedings of the ACM SIGCOMM 2010 Conference* (pp. 63-74).

[57] Tekinerdoğan, B., Özcan, K., Yağız, S., & Yakın, İ. Systems Engineering Architecture Framework for Physical Protection Systems. In *2020 IEEE International Symposium on Systems Engineering (ISSE)* (pp. 1-8). IEEE.

[58] Ahoa, E., Kassahun, A., & Tekinerdogan, B. (2020). Business processes and information systems in the Ghana cocoa supply chain: A survey study. *NJAS-Wageningen Journal of Life Sciences*, *92*, 100323.

[59] Mao, H., Alizadeh, M., Menache, I., & Kandula, S. (2016, November). Resource management with deep reinforcement learning. In *Proceedings of the 15th ACM workshop on hot topics in networks* (pp. 50-56).

[60] Agarwal, A., Bianchini, R., Chaiken, D., Johnson, K. L., Kranz, D., Kubiatowicz, J., ... & Yeung, D. (1995). The MIT Alewife machine: Architecture and performance. *ACM SIGARCH Computer Architecture News*, *23*(2), 2-13.

[61] Dustdar, S., Mutlu, O., & Vijaykumar, N. (2020). Rethinking Divide and Conquer—Towards Holistic Interfaces of the Computing Stack. *IEEE Internet Computing*, *24*(6), 45-57.

[62] Ağca, M. A. (2015). Özelleştirilmiş analitik bulut mimarilerinde dağıtık dosya sistemleri ile performans iyileştirilmesi (Master's thesis, TOBB ETÜ Fen Bilimleri Enstitüsü).

[63] Negi, P., Marcus, R., Kipf, A., Mao, H., Tatbul, N., Kraska, T., & Alizadeh, M. (2021). Flow-Loss: Learning Cardinality Estimates That Matter. *arXiv preprint arXiv:2101.04964*.

[64] Baghdadi, R., Debbagh, A. N., Abdous, K., Benhamida, F. Z., Renda, A., Frankle, J. E., ... & Amarasinghe, S. (2020). Tiramisu: A polyhedral compiler for dense and sparse deep learning. *arXiv preprint arXiv:2005.04091*.

[65] Han, D., Tople, S., Rogers, A., Wooldridge, M., Ohrimenko, O., & Tschiatschek, S. (2020). Replication-Robust Payoff-Allocation with Applications in Machine Learning Marketplaces. *arXiv preprint arXiv:2006.14583*.

[66] Gutierrez, J., Najib, M., Perelli, G., & Wooldridge, M. (2020). On computational tractability for rational verification.

[67] Han, D., Harrenstein, P., Nugent, S., Philpott, J., & Wooldridge, M. (2021). Behavioural strategies in weighted Boolean games. *Information and Computation*, *276*, 104556.

[68] Wooldridge, M. (2020). Artificial Intelligence requires more than deep learning

[69] Wooldridge, M. (1999). Intelligent agents. *Multiagent systems*, *6*.

[70] Chih-Lin, I., Kuklinski, S., Chen, T. C., & Ladid, L. L. (2020). A perspective of O-RAN integration with MEC, SON, and network slicing in the 5G era. *IEEE Network*, *34*(6), 3-4.

[71] Ladid, L., Karagiannis, G., & Potts, R. M. (2019). D3. 4: Harmonisation of Standards for 5G Technologies.

[72] Chaccour, C., Soorki, M. N., Saad, W., Bennis, M., Popovski, P., & Debbah, M. (2021). Seven Defining Features of Terahertz (THz) Wireless Systems: A Fellowship of Communication and Sensing. *arXiv preprint arXiv:2102.07668*.

[73] Ağca, M. A., Khadraoui, D., & Faye, S. (2019). Persisting Trust In Intrusted Varying Resilient Citv Context V. In *Proceedings on the International Conference on Artificial Intelligence (ICAI)* (pp. 312-318). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

[74] Krienke, C., Kolb, L., Diken, E., Streuber, M., Kirchhoff, S., Bukur, T., ... & Sahin, U. (2021). A noninflammatory mRNA vaccine for treatment of experimental autoimmune encephalomyelitis. *Science*, *371*(6525), 145-153.

[75] Tomsett, R., Preece, A., Braines, D., Cerutti, F., Chakraborty, S., Srivastava, M., ... & Kaplan, L. (2020). Rapid trust calibration through interpretable and uncertainty-aware AI. *Patterns*, *1*(4), 100049.

[76] Smuha, N. A. (2019). The eu approach to ethics guidelines for trustworthy artificial intelligence. *CRi-Computer Law Review International*.

[77] Veale, M. (2020). A critical take on the policy recommendations of the EU high-level expert group on artificial intelligence. *European Journal of Risk Regulation*.

[78] Alsamhi, S. H., Ma, O., & Ansari, M. S. (2019). Survey on artificial intelligence based techniques for emerging robotic communication. *Telecommunication Systems*, *72*(3), 483-503.

[79] Vassev, E., Sterritt, R., Rouff, C., & Hinchey, M. (2012). Swarm technology at NASA: building resilient systems. *IT Professional*, *14*(2), 36-42.

[80] Rouff, C., Vanderbilt, A., Truskowski, W., Rash, J., & Hinchey, M. (2004, April). Verification of NASA emergent systems. In *Proceedings. Ninth IEEE International Conference on Engineering of Complex Computer Systems* (pp. 231-238). IEEE.

[81] Cherry, S. (2004). Edholm's law of bandwidth. *IEEE spectrum*, *41*(7), 58-60.

[82] Machovec, Dylan, et al. " Dynamic Heuristics for Surveillance Mission Scheduling with Unmanned Aerial Vehicles in Heterogeneous Environments." Colorado Engineering Inc. Whitepaper. 2020.

[83] Crowder, J., et al. " The Systems AI Thinking Process (SATP) for Artificial Intelligent Systems." Colorado Engineering Inc. Whitepaper. 2020.

[84] Carbone, John. (2020). Artificially Intelligent Cyber Security: Reducing Risk & Complexity.

[85] Stuckman et al., EU Commission, The European 5G Annual Journal.2021.

[86] Balakrishnan, H., Padmanabhan, V. N., Seshan, S., & Katz, R. H. (1997). A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM transactions on networking*, *5*(6), 756-769.

[87] Martinet, P., Gallice, J., & Djamel, K. (1996, May). Vision based control law using 3d visual features. In *World Automation Congress, WAC'96, Robotics and Manufacturing Systems* (Vol. 3, pp. 497-502).

[88] Shukur, H., Zeebaree, S. R., Ahmed, A. J., Zebari, R. R., Ahmed, O., Tahir, B. S. A., & Sadeeq, M. A. (2020). A State of Art Survey for Concurrent Computation and Clustering of Parallel Computing for Distributed Systems. *Journal of Applied Science and Technology Trends*, *1*(4), 148-154.

[89] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., & Balakrishnan, H. (2001). Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review*, *31*(4), 149-160.

[90] Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000, January). Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd annual Hawaii international conference on system sciences* (pp. 10-pp). IEEE.

[91] Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on wireless communications*, *1*(4), 660-670.

[92] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., ... & Turner, J. (2008). OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM computer communication review*, *38*(2), 69-74.

[93] Heinzelman, W. R., Kulik, J., & Balakrishnan, H. (1999, August). Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking* (pp. 174-185).

[94] Chen, B., Jamieson, K., Balakrishnan, H., & Morris, R. (2002). Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless networks*, *8*(5), 481-494.

[95] Andersen, D., Balakrishnan, H., Kaashoek, F., & Morris, R. (2001, October). Resilient overlay networks. In *Proceedings of the eighteenth ACM symposium on Operating systems principles* (pp. 131-145).

[96] Balakrishnan, Hari (Belmont, MA, US), Arun, Venkat (Cambridge, MA, US) 2020 SURFACE FOR CONTROLLED RADIO FREQUENCY SIGNAL PROPAGATION United States Massachusetts Institute of Technology (Cambridge, MA, US)20200350693https://www.freepatentsonline.com/y2020/0350693.html

[97] Meklenburg, J., Specter, M., Wentz, M., Balakrishnan, H., Chandrakasan, A., Cohn, J., ... & Weitzner, D. (2020). SonicPACT: An Ultrasonic Ranging Method for the Private Automated Contact Tracing (PACT) Protocol. arXiv preprint arXiv:2012.04770.

[98] Goyal, P., Agarwal, A., Netravali, R., Alizadeh, M., & Balakrishnan, H. (2020). {ABC}: A Simple Explicit Congestion Controller for Wireless Networks. In *17th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 20)* (pp. 353-372).

[99] Arun, V., & Balakrishnan, H. (2019). RFocus: Practical beamforming for small devices. arXiv preprint arXiv:1905.05130.

[100] Cho, I., Saeed, A., Fried, J., Park, S. J., Alizadeh, M., & Belay, A. (2020). Overload Control for µs-scale RPCs with Breakwater. In *14th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 20)* (pp. 299-314).

[101] Ye, F., Zhou, S., Venkat, A., Marucs, R., Tatbul, N., Tithi, J. J., ... & Gottschlich, J. (2020). MISIM: An End-to-End Neural Code Similarity System. *arXiv preprint arXiv:2006.05265*.

[102] Balakrishnan, H., Girod, L. D., & Ossin, I. (2015). *U.S. Patent Application No. 14/529,812*.

[103] Mannoni, V., Berg, V., Sesia, S., & Perraud, E. (2019, April). A comparison of the V2X communication systems: ITS-G5 and C-V2X. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)* (pp. 1-5). IEEE.

[104]Scherer, M. U. (2016). Regulating Artificial Intelligence Systems: Risks, Challenges, Competencies, and Strategies (SSRN Scholarly Paper No. ID 2609777). Rochester, NY: Social Science Research Network.

[105]Truszkowski, W. & Hinchey, M. & Rash, J. & Rouff, Christopher. (2004). NASA's swarm missions: The challenge of building autonomous software. IT Professional. 6. 47 - 52. 10.1109/MITP.2004.66.

[106]Cafarella, M., DeWitt, D., Gadepally, V., Kepner, J., Kozyrakis, C., Kraska, T., ... & Zaharia, M. (2020). DBOS: A Proposal for a Data-Centric Operating System. *arXiv preprint arXiv:2007.11112*.

[107]Belay, A., Prekas, G., Klimovic, A., Grossman, S., Kozyrakis, C., & Bugnion, E. (2014). {IX}: A Protected Dataplane Operating System for High Throughput and Low Latency. In *11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14)* (pp. 49-65).

[108]Athalye, A., Belay, A., Kaashoek, M. F., Morris, R., & Zeldovich, N. (2019, October). Notary: a device for secure transaction approval. In *Proceedings of the 27th ACM Symposium on Operating Systems Principles* (pp. 97-113).

[109]REIFF et al., ARTIFICIAL INTELLIGENCE TECHNOLOGY, USE CASES AND APPLICATIONS, TRUSTWORTHINESS AND TECHNICAL STANDARDIZATION White Paper, https://portail-qualite.public.lu/dam-assets/publications/normalisation/2021/ilnas-white-paper-artificial-intelligence.pdf, February 2021.

[110]Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

[111]Goldblum, M., Tsipras, D., Xie, C., Chen, X., Schwarzschild, A., Song, D., ... & Goldstein, T. (2020). Data Security for Machine Learning: Data Poisoning, Backdoor Attacks, and Defenses. *arXiv preprint arXiv:2012.10544*.

[112]Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., ... & Kurakin, A. (2019). On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*.

[113]Salman, H., Ilyas, A., Engstrom, L., Vemprala, S., Madry, A., & Kapoor, A. (2020). Unadversarial Examples: Designing Objects for Robust Vision. *arXiv preprint arXiv:2012.12235*.

[114]ENISA, Post-Quantum Cryptography: Current state and quantum mitigation, https://www.enisa.europa.eu/publications/post-quantum-cryptography-current-state-and-quantum-mitigation, February 09, 2021

[115]Breitwieser, L., Hesam, A., de Montigny, J., Vavourakis, V., Iosif, A., Jennings, J., ... & Bauer, R. (2021). BioDynaMo: a general platform for scalable agent-based simulation. *bioRxiv*, 2020-06.

[116]Reuther, A., Michaleas, P., Jones, M., Gadepally, V., Samsi, S., & Kepner, J. (2020, September). Survey of machine learning accelerators. In *2020 IEEE High Performance Extreme Computing Conference (HPEC)* (pp. 1-12). IEEE.

[117] Samsi, S., Weiss, M. L., Bestor, D., Li, B., Jones, M., Reuther, A., ... & Gadepally, V. (2021). The MIT Supercloud Dataset. arXiv preprint arXiv:2108.02037.

[118] Sabt, M., Achemlal, M., & Bouabdallah, A. (2015, August). Trusted execution environment: what it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/ISPA* (Vol. 1, pp. 57-64). IEEE.

[119] Chen, J., Li, K., Bilal, K., Li, K., & Philip, S. Y. (2018). A bi-layered parallel training architecture for large-scale convolutional neural networks. IEEE transactions on parallel and distributed systems, 30(5), 965-976.

[120]Chen, J., Li, K., Tang, Z., Bilal, K., & Li, K. (2016). A parallel patient treatment time prediction algorithm and its applications in hospital queuing-recommendation in a big data environment. IEEE Access, 4, 1767-1783.

[121]Goldblum, M., Tsipras, D., Xie, C., Chen, X., Schwarzschild, A., Song, D., ... & Goldstein, T. (2020). Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. arXiv preprint arXiv:2012.10544.

[122]Win, K. N., Li, K., Chen, J., Viger, P. F., & Li, K. (2020). Fingerprint classification and identification algorithms for criminal investigation: A survey. Future Generation Computer Systems, 110, 758-771.

[123] Chen, J., Li, K., & Philip, S. Y. (2021). Privacy-Preserving Deep Learning Model for Decentralized VANETs Using Fully Homomorphic Encryption and Blockchain. *IEEE Transactions on Intelligent Transportation Systems*.

[124]Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533-536.

[125]Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2018). Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, *18*.

[126]Milutinovic, M., Baydin, A. G., Zinkov, R., Harvey, W., Song, D., Wood, F., & Shen, W. (2017). End-to-end training of differentiable pipelines across machine learning frameworks.

[127]Griewank, A., & Walther, A. (2000). Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Transactions on Mathematical Software (TOMS)*, *26*(1), 19-45.

[128]Amdahl, G. M. (1967, April). Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, spring joint computer conference* (pp. 483-485).

[129]Gunther, N. J. (2008). A general theory of computational scalability based on rational functions. *arXiv preprint arXiv:0808.1431*.

[130]**Ağca, M. A. et.al. (2021, December). Persisting Trust in Emerging Hybrid-Clouds and 6G Systems. In 2021 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 379-385). IEEE.**

[131]Kirkpatrick, K. (2013). Software-defined networking. *Communications of the ACM*, *56*(9), 16-19.

[132]McGillion, B., Dettenborn, T., Nyman, T., & Asokan, N. (2015, August). Open-TEE--An Open Virtual Trusted Execution Environment. In *2015 IEEE Trustcom/BigDataSE/ISPA* (Vol. 1, pp. 400-407). IEEE.

[133]Samsi, S., Weiss, M. L., Bestor, D., Li, B., Jones, M., Reuther, A., ... & Gadepally, V. (2021). The MIT Supercloud Dataset. *arXiv preprint arXiv:2108.02037*.

[134]Kepner, J., Davis, T., Byun, C., Arcand, W., Bestor, D., Bergeron, W., ... & Reuther, A. (2020, May). 75,000,000,000 streaming inserts/second using hierarchical hypersparse graphblas matrices. In *2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)* (pp. 207-210). IEEE.

[135]Kepner, J., Jones, M., Andersen, D., Buluc, A., Byun, C., Claffy, K., ... & Michaleas, P. (2021). Spatial Temporal Analysis of 40,000,000,000,000 Internet Darkspace Packets. *arXiv preprint arXiv:2108.06653*.

[136]Reuther, A., Kepner, J., Byun, C., Samsi, S., Arcand, W., Bestor, D., ... & Michaleas, P. (2018, September). Interactive supercomputing on 40,000 cores for machine learning and data analysis. In 2018 IEEE High Performance extreme Computing Conference (HPEC) (pp. 1-6). IEEE.

[137]Kepner, J., Gadepally, V., Jananthan, H., Milechin, L., & Samsi, S. (2020). AI Data Wrangling with Associative Arrays. arXiv preprint arXiv:2001.06731.

[138]Tataria, H., Shafi, M., Molisch, A. F., Dohler, M., Sjöland, H., & Tufvesson, F. (2021). 6G wireless systems: Vision, requirements, challenges, insights, and opportunities. *Proceedings of the IEEE*.

[139]Dorri, A., Kanhere, S. S., & Jurdak, R. (2018). Multi-agent systems: A survey. *Ieee Access*, *6*, 28573-28593.

[140]Ağca, Muhammed Akif, Sébastien Faye, and Djamel Khadraoui. "A survey on trusted distributed artificial intelligence." *IEEE Access* 10 (2022): 55308-55337.

[141] Available at [4th August 4, 2023], '**https://www.eesc.europa.eu/en/news-media/news/eesc-proposes-introducing-eu-certification-trusted-ai-products',**

[142]Ouedraogo, M., Khadraoui, D., De Rémont, B., Dubois, E., & Mouratidis, H. (2008, November). Deployment of a security assurance monitoring framework for telecommunication service infrastructures on a VoIP service. In 2008 New Technologies, Mobility and Security (pp. 1-5). IEEE.

[143]Available at [4th August 4, 2023], '**https://www.ai.gov/strategic-pillars/advancing-trustworthy-ai/#Research_and_Development_for_Trustworthy_AI**'

[144]Kim, Y., Eddins, A., Anand, S. et al. Evidence for the utility of quantum computing before fault tolerance. Nature 618, 500–505 (2023).

[145]Manuel, P. (2015). A trust model of cloud computing based on Quality of Service. Annals of Operations Research, 233, 281-292.

[146]Pathak, A., Al-Anbagi, I., & Hamilton, H. J. (2022). An adaptive QoS and trust-based lightweight secure routing algorithm for WSNs. IEEE Internet of Things Journal, 9(23), 23826-23840.

[147]Ağca, M. A., Faye, S., & Khadraoui, D. (2023). Trusted Distributed Artificial Intelligence (TDAI). IEEE Access.

Annexes

Publications

- **1st Paper:** CSCI'19- 21st ICAI 2019 – International Conference on Artificial Intelligence: **Persisting trust in untrusted resilient city context**
    - Ağca, M. A., Khadraoui, D., & Faye, S. (2019). Persisting Trust In Intrusted Varying Resilient Citv Context V. In Proceedings on the International Conference on Artificial Intelligence (ICAI) (pp. 312-318). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).
- **2nd Paper:** CSCI'19 - 6th Annual Conf. on Computational Science & Computational Intelligence: **A holistic abstraction to ensure trusted scaling and memory speed trusted analytics.**
    - Agca, M. A. (2019, December). A Holistic Abstraction to Ensure Trusted Scaling and Memory Speed Trusted Analytics. In 2019 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 1428-1434). IEEE.
- **3rd Paper:** CSCE'20 - PDPTA'20 - The 26th Int'l Conference on Parallel and Distributed Processing Techniques and Applications: **Challenges for Swarm of UAV/Drone Based Intelligence**.
    - Agca, M. A., et.al (2020). Challenges for Swarm of UAV/Drone Based Intelligence. In the 26th Int'l Conference on Parallel and Distributed Processing Techniques and Applications.

- **4th Paper:** CSCI'21 - 8th Annual Conf. on Computational Science & Computational Intelligence: **Persisting Trust in Emerging Hybrid-Clouds and 6G Systems.**
    - Agca, M. A. et.al. (2022, December). Persisting Trust in Emerging Hybrid-Clouds and 6G Systems. In 2021 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 1428-1434). IEEE.

- **5th Paper: A Survey on Trusted Distributed Artificial Intelligence**, IEEE Access, Published
    - Ağca, M. A., Faye, S., & Khadraoui, D. (2022). A survey on trusted distributed artificial intelligence. IEEE Access, 10, 55308-55337.

- **6th Paper:** CSCE'22 - PDPTA'22 - The 28th Int'l Conference on Parallel and Distributed Processing Techniques and Applications: **End-to-end Trust Mechanism with Dynamic Holistic View Based Throughput Maximization Approach**.
    - Agca, M. A., (2022). End-to-end Trust Mechanism with Dynamic Holistic View Based Throughput Maximization Approach. In the 26th Int'l Conference on Parallel and Distributed Processing Techniques and Applications.

- **7th Paper:** CSCE'22 - 24th ICAI 2022 – International Conference on Artificial Intelligence: **brainIT: A generic IT Core Mechanism for Continuous Growth-Flow in Dynamic Context.**
    - Ağca, M. A., (2022). brainIT: A generic IT Core Mechanism for Continuous Growth-Flow in Dynamic Context. In Proceedings on the International Conference on Artificial Intelligence (ICAI) (pp. ..). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

- **8th Paper:** Invited Talk: **End-to-end Trusted Execution Environment (TEE) with Dynamic Holistic View Based Throughput Maximization Approach for Open-Source Systems**, LFMS 2022 The Linux Foundation Member Summit 2022 Invited Speaker, https://lfms22.sched.com/speaker/agca.akif
    - https://lfms22.sched.com/event/1BKCf/video-broadcast-only-end-to-end-trusted-execution-environment-tee-with-dynamic-holistic-view-based-throughput-maximization-approach-muhammed-akif-agca-list-phd-tobb-etu

- **9th Paper: IEEE Access - Trusted Distributed Artificial Intelligence (TDAI)**.
    - Ağca, M. A., Faye, S., & Khadraoui, D. (2023). Trusted Distributed Artificial Intelligence (TDAI). *IEEE Access*.

Patents: Submitted

1. An algorithm to assure trust and bring confidence in smart systems.
    - **P-2022-019-AGCA-trusted autonomous interaction nodes (ITIS) - updated: METHOD FOR TRUSTED DISTRIBUTED AUTONOMOUS SYSTEMS, PCT Application Number: PCT/EP2023/070587, Submission Number: 12273270, submitted: 25th July 2022, extended at: 25thJuly,2023.**
2. An algorithm to ensure network scalability and to maximize throughput for 5G
3. An algorithm to ensure connectivity in CCAM and optimize energy consumption by emission minimization

- **MUHAMMED AKİF AĞCA (**Member, IEEE) received the degree from Middle East Technical University (METU), in 2011, master's degree in computer engineering from Turkish Stock Market Union Economy and Technology University (TOBB ETU), Ankara, TÜRKİYE, in 2015, and PhD Degree from University of LUXEMBOURG in Computer Science and Computer Engineering in 2023. He joined HAVELSAN Aerospace and Defense Company, in 2014, as a Software Engineer and worked as a NATO Coordinator for two years. He is pursuing his research and development studies in various groups as a Full-Stack Systems Architect and a Researcher: https://orcid.org/0000-0001-5986-3829.