IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 September 2023, accepted 21 September 2023, date of publication 6 October 2023, date of current version 18 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3322568

METHODS

Trusted Distributed Artificial Intelligence (TDAI)

MUHAMMED AKİF AĞCA[1,2], (Member, IEEE), SÉBASTIEN FAYE[2], (Member, IEEE), AND DJAMEL KHADRAOUI[2], (Member, IEEE)

[1]Computer Engineering Department, TOBB University of Economics and Technology (TOBB ETÜ), 06560 Ankara, Türkiye
[2]Luxembourg Institute of Science and Technology—LIST, 4362 Esch-Sur-Alzette, Luxembourg

Corresponding author: Muhammed Akif Ağca (akif.agca@etu.edu.tr)

This work was supported in part by the Trusted Distributed Artificial Intelligence (TDAI) Project through the Luxembourg Institute of Science and Technology (LIST) Ph.D. Grant 10.13039/100016172.

ABSTRACT As the diversity of components increases within the intelligent systems, trusted interactivity also becomes critical challenge for the system components and nodes. Furthermore, emerging SDN (Software Defined Networking) features are also utilized to assure its resiliency and robustness in a dynamic context and monitored by trusted multi-agents' system to maximize trustworthiness of the system components and the deployed context. However, it is not feasible to deploy the intelligent mechanisms at massive scale with the state-of-the-art architectural design paradigms. Therefore, we define three main architectures (central, decentral/autonomous/embedded, distributed/hybrid) as a basis for TDAI methodology to ensure end-to-end trust in holistic AI system life-cycle. Thanks to such a trusted multi-agents-based trust monitoring mechanism, we will be able to overcome hardware limitations and provide flexible and resilient end-to-end trust mechanism for trusted AI models and emerging massive scale intelligent systems. Finally, we evaluated our TDAI Methodology in CCAM (Connected, Cooperative, Autonomous Mobility) domain of a smart-city to monitor its system trust and user behaviors. By that means, it is exploited as a mean of decision-making mechanism to be deployed either manually or automatically (example of anomalies detection etc.). Such a mechanism improves total system performance and behavioral anomaly detection and risk minimization algorithms over the distributed nodes of a given AI system. Furthermore, smartness features are also improved with human-like intelligence abilities at massive scale thanks to the promising performance of TDAI at real-life deployment experiments to maximize trust factor of the dynamically observed context of the smart-cities during the monitored time-span.

INDEX TERMS Trusted AI, distributed computing, software defined networking (SDN), multi-agent systems (MAS), trusted execution environment (TEE).

I. INTRODUCTION AND RELATED WORK

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 [17]. Fortunately, widely accepted learning representation approaches with the data such as back propagation [1] 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 [2], [3] 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 [4] computationally scalable mathematical/system models [6], [7] 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 [7] to the edges in trusted scalable manner. Cooperation and task data sharing between the components can be accelerated with SDN (software defined networking)

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

[8] 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 [9].

Furthermore, holistic views [7] 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 [10], authors integrate resources virtualization approaches as SuperCloud and publishes initial data sets for performance evaluations. In [11], 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 [3], 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. [13], 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. [14] explores better representation of data in AI systems with associative arrays. More interestingly Tataria et.al. [15], [16], [19] 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 [7], [18], [20], [21]. 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.

Paper is organized as follows: Section. II defines trusted distributed AI methodology (TDAI), Section 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 section 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. Section IV evaluates the contributions of this study and discusses about the future potentials. Finally, Section. V concludes the paper.

II. METHODOLOGY

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 [17], trust cost also increases to be able to justify the trust in dynamic context in (near) real time as illustrated in Figure 1.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

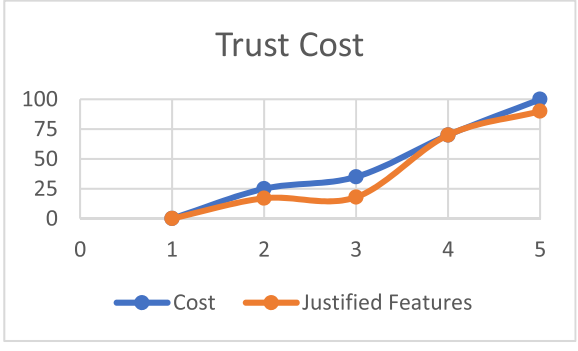

FIGURE 1. a Trust justification cost.

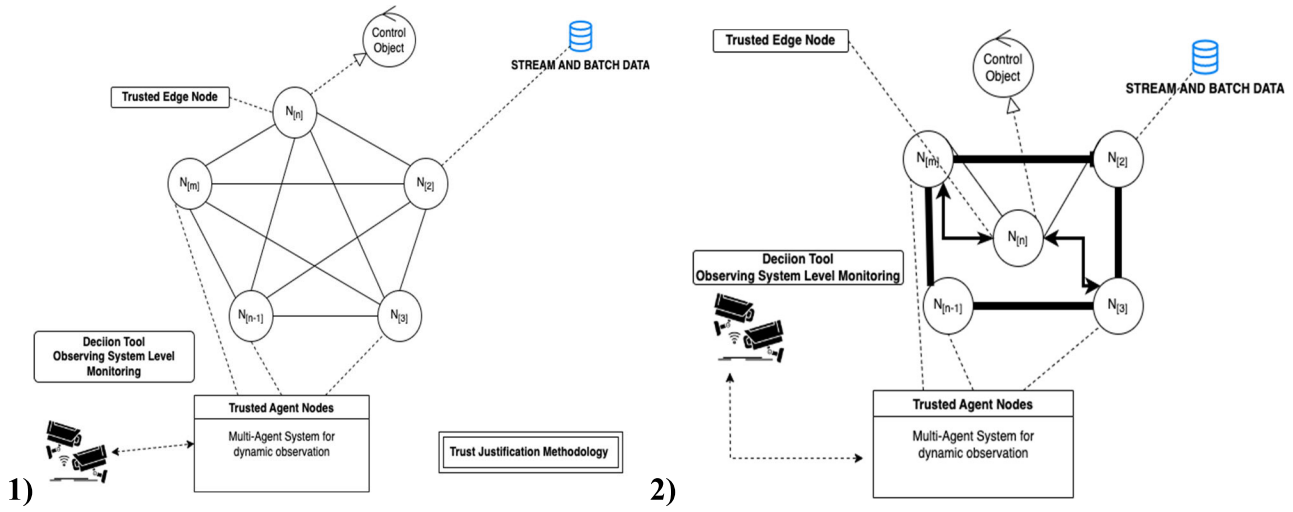

FIGURE 1b. (1) A distributed system with set of nodes (2) Justified channels for trusted interactivity with TDAI.

TABLE 1. TDAI classes.

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

on the context of use case, the reachable trust level can be managed based on the cost available and required features. Next sub-section explains the trust levels and introduce TDAI formulation.

B. TDAI TAXONOMY

1) 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 [18], [20]. 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 1 and critical metrics/parameters listed in Table 2. 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

In order to be able to identify a system or it's node as trusted, it has to be justified [17]. However, there is trust cost for each justification feature as illustrated in Figure 1.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 [7] but can be kept at optimal level with right dynamic strategy mechanisms. 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 1.b. Next subsection explains classes and critical identified features to defined trust levels of TDAI methodology.

2) TDAI CLASSES

Table 1 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.

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: RUN-TIME

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.

c: CLASS SEC: SECURITY

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

d: 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 subsection explains the weight update and error minimization strategies of TDAI methodology and introduces TDAI formulation.

C. TDAI FORMULATION

1) TDAI TAXONOMY AND SYSTEM MODELLING (TRUSTED NEURON AND TRUST MEASUREMENT METHOD)

Generic systems are represented by a dynamic model as illustrated in Figure 1.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 2.a. The neurons interacts with the environment E{} via the linked nodes and utilizes its' functions dynamically to pursue continues growth-flow within the observed context. Next subsection gives details of the TDAI formal statements.

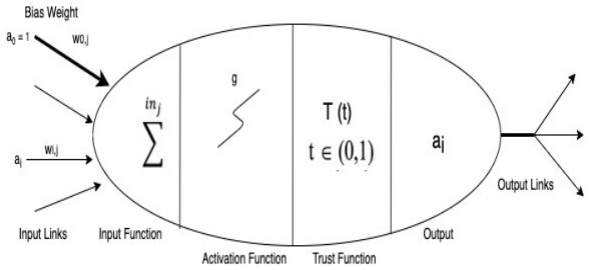

FIGURE 2. a Trusted neuron.

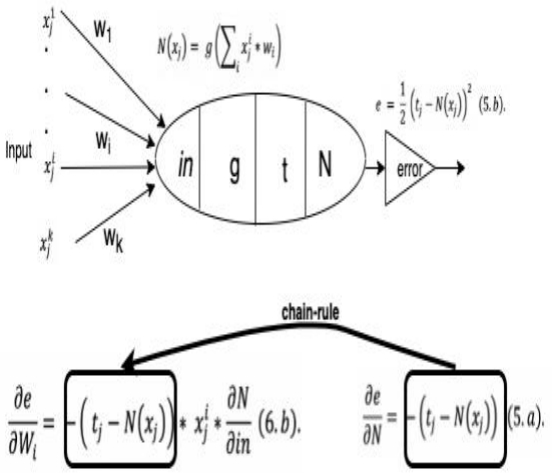

FIGURE 2. b Single neuron network with error calculation.

2) TRUSTED VALUE FORMULATION

Stage 1: Trusted Neuron Formulation:

As formalized in equation (1) 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}:

$$N_i = \sum_{j=0..N} w_{ij} a_j \tag{1}$$

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:

$$a_i = g\,(N_i) = g\left(\sum_{j=0..N} w_{iJ} a_j\right) \tag{2}$$

Stage 3: Neurons' Trust Value:

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

$$t_{a_i} = Average\left(\sum_{0}^{i} N_i\right), N \epsilon transaction\ flow. \tag{3}$$

Transaction flow repeats continuously with holistic feedback controller mechanism [7], which assures continuous growth of an intelligent system. Learning systems of neural networks are 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 subsection introduces weight update strategy and error minimization methodology.

3) 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 2.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.4** below formulates changing the weights in round r+1.

$$W_i\,(r+1) = W_i\,(r) - \epsilon \frac{\partial e}{\partial W_i}. \tag{4}$$

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.5.a and 5.b**;

$$\frac{\partial e}{\partial N} = -\left(t_j - N\left(x_j\right)\right) \tag{5.a}$$

$$e = \frac{1}{2}\left(t_j - N\left(x_j\right)\right)^2. \tag{5.b}$$

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

$$\begin{aligned}\frac{\partial e}{\partial W_i} &= \frac{\partial e}{\partial in}\frac{\partial in}{\partial Wi} \\ &= \frac{\partial e}{\partial N}\frac{\partial N}{\partial in}\frac{\partial in}{\partial Wi} \\ &= -\left(t_j - N\left(x_j\right)\right) * x_i \frac{\partial N}{\partial in} \qquad (6.a)\end{aligned}$$

$$\frac{\partial e}{\partial W_i} = -\left(t_j - N\left(x_j\right)\right) * x_j^i * \frac{\partial N}{\partial in} \tag{6.b}$$

Equation.4 can be plugged to equation.6.a to get a rule to calculate how the weights should be updated. Figure.3 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 [1] Rumelhart et al. 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 [2] Baydin et al. The methods enable end-to-end training of differentiable pipelines across machine learning frameworks [3] Milutinovic et al.

Backpropagation algorithm is a special case of automatic differentiation [4] Griewank and Walther. 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.

Computational differentiation is useful for gradient error calculation and single/multi-layer neural network training. However, computing the rate of change is restricted with **computational scalability** limitations. Furthermore, it inflates the memory resources and require larger memory resources. Algoritqhm 799 [4] Griewank et.al. 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 [5] Amdahl considers sequential and parallelizable portions of the programs. General theory of computational scalability [6] Gunther 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 checkpointing/control mechanism [7]. 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 maximize neural network training 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.

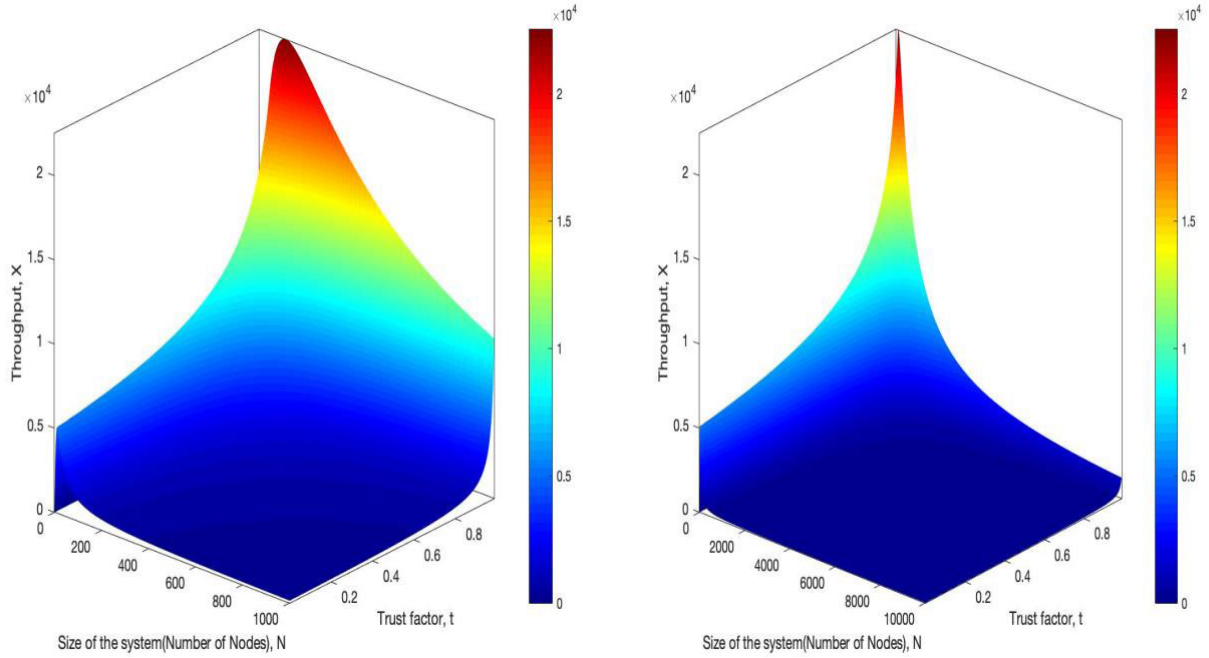

FIGURE 3. a/b. Trust factor coefficient-based throughput maximisation approach [7].

Trust indicators defined in different system layers can maximize the targeted throughput and minimize crosstalk and latency penalties in hybrid designed architectures. Figure.3 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 subsection.

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 subsection explains the trusted agent and interaction with the environment.

D. 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.4. Let $V(f, \boldsymbol{E}, U)$ denote the expected utility according to U () of the agent function f () operating on E { }.

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

$$N_i = \sum_{j=0..N} w_{ij} a_j$$
$$a_i = g\,(N_i) = g\left(\sum_j w_{ij} a_j\right) \tag{7}$$

Each Environment E has set of nodes; $N_{E:}\{N_1, N_2, N_3, \ldots, N_n\}$. Each environment can be monitored with set of trusted agents or nodes. Each node can be defined as a trusted agentor 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:

$$f_{opt} = arg \max_f V(f, \boldsymbol{E}, U) \tag{8}$$

Throughput of each Node X(N); monitored via trusted Agent

$$A\{\}\text{and nodes N}\{\}$$
$$\text{Trusted Agent A}\{\}$$
$$= \{iN_i \text{ and with activation function } a_i\}$$

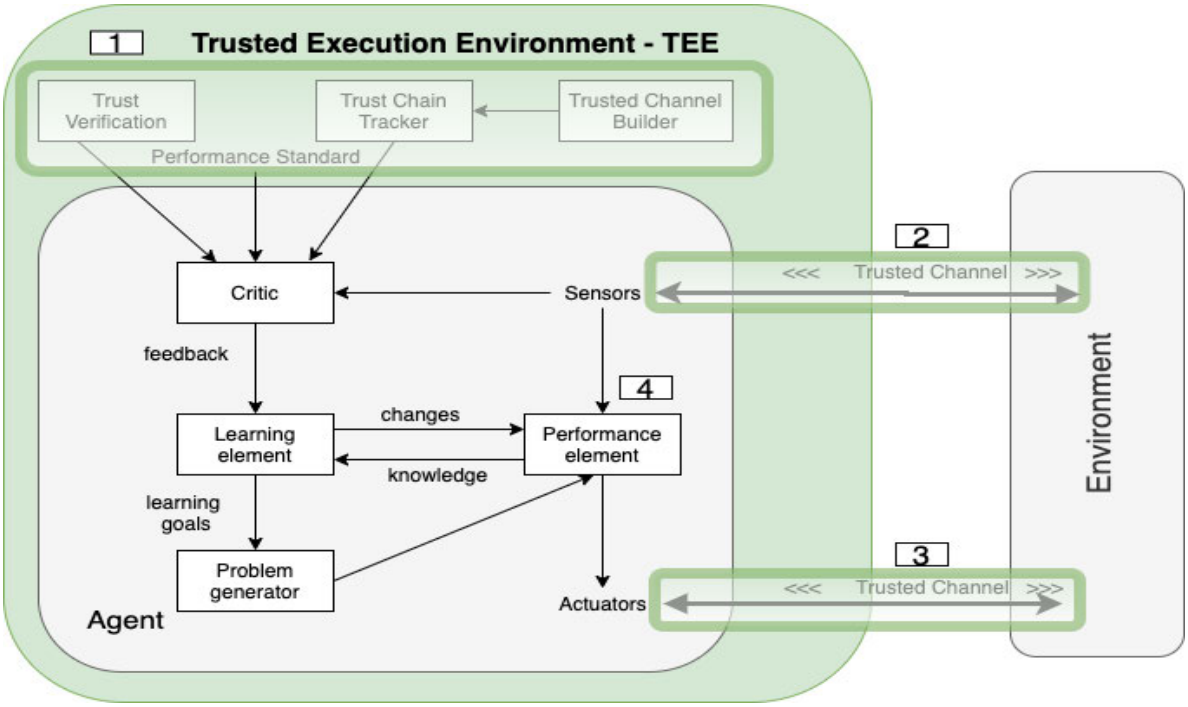

FIGURE 4. Trusted agent environment interaction workflow 1 to 4.

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).

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()

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

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.4. 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.2 in next chapter illustrates the selected metrics and regulative constraints identified in this study. The algorithm called Trusted Distributed AI (TDAI) runs as below steps.

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.5 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 $N_{E:}\{N_1, N_2, N_3, \ldots, N_n\}$ 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 5.a introduces the basic pseudocode for TEE based interaction with the environment. Each

TABLE 2. Monitoring metrics/parameters of a node.

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.

Environment E{ } has set of snapshots for selected time spans of the observed context. Set of nodes N{ } in a context are observed dynamically with set of Trusted Agents A{ }, which are embedded to OBU (On Board Units/Computers) of each node.

Figure.6 in chapter III 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 [5] can be applied to obtain dynamic holistic view of the context with trust factor coefficient-based throughput maximization approach.

```
# 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 5. a Pseudo code for TEE based interaction with the environment.

```
# 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 5. b Pseudo code for TDAI trust verification for continuous growth-flow.

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 $f_{TrustedAgent\{A\{\}\}}(EN[*]\})$ 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 5.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.

III. TRUSTED DISTRIBUTED AI SYSTEM ARCHITECTURES AND COMPONENTS

In this section, 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-sections will describe the main architectural and component features required for the applicability of the TDAI methodology described in the previous section 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 section II. This section 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.

Basic components of an OBU device is illustrated in Figure.6, 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 section introduces the main design paradigms and hypothesis proposed regarding the methodology.

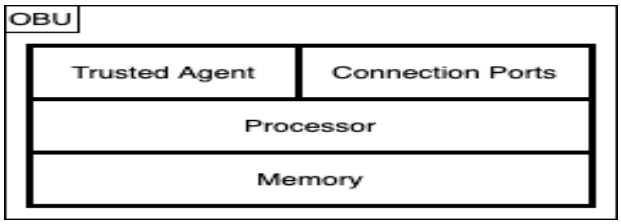

FIGURE 6. OBU system components.

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 9. 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.2 is transmitted from each node as a packet;

$$p =< n_{id}, n_{st}, n_e, n_l, n_{id}, n_k, n_T, n_{chsum}, n_c, n_\rho, n_d >, \quad (9)$$

The packages p<> are monitored dynamically and related feature sets are vectorized within the continues growth-flow mechanism as represented in Figure 5.b. Each architectural design has critical advantages and limitations ad summarized in Table 3. It is clear that distributed design is required to be able to ensure trusted interactivity, which will be formally proved in next subsections. Next section 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 [7]. Figure 7 illustrates is a typical example showing the basic components, which are a central cloud, mobile nodes, and fog layer-based networking and communication components.

As the number of the node $N_{E:}\{N_1, N_2, N_3, \ldots, N_n\}$ in the context increases, throughput of the system decreases as illustrated in Figure.3.a. Fortunately, making the system trusted can help maximize the total throughput of the system with less number of nodes, see Figure.3.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 subsection 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

TABLE 3. Distributed design critical features advantages/disadvantages comparison ("+": advantages, "– "= disadvantages).

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.

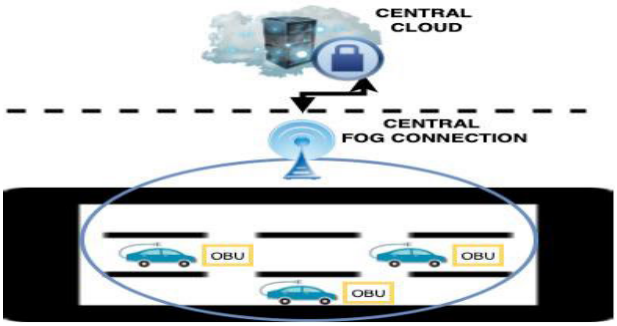

FIGURE 7. Centralized (Fully connected).

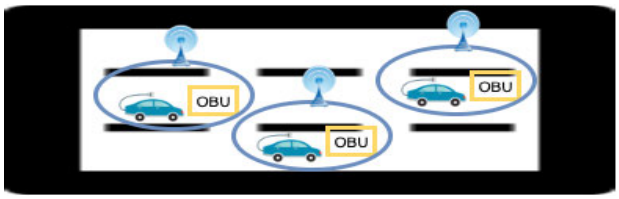

FIGURE 8. Decentral (Autonomous/ Embedded /Local).

independently. 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 $N_{E:}\{N_1, N_2, N_3, \ldots, N_n\}$ can be minimized and networking and communication bottlenecks are minimized as 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 [7].

Next subsection 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 3. 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.

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 9 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 [7] can enable to build end-to-end trusted scalable channel within the components and total system. Throughput value of each node X(N) is observed dynamically

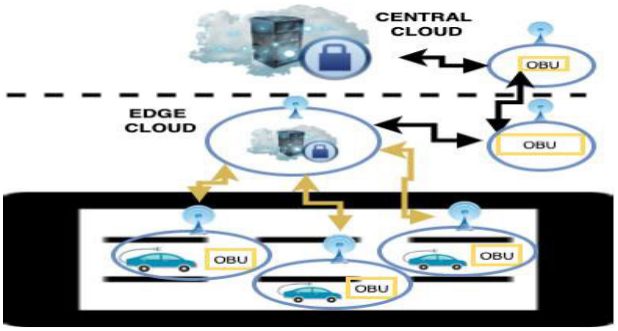

FIGURE 9. Distributed (Edge/Hybrid).

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 2 Introduces metrics, parameters, and regulative constraints observed in this study. Table 3 introduces basic comparative analysis and advantages/disadvantages of distributed design over other approaches with identified system features. Next subsection 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.10.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 [8], 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 [7]. The innovations enable to design end-to-end Trusted Execution Environment (TEE) as illustrated in Figure.4 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 [7]; such as, emission generated by cars and EMF generated by emerging computational/memory units of intelligent-systems with set of the nodes $N_{E:}\{N_1, N_2, N_3, \ldots, N_n\}$ as dynamically controlled features of the growth-flow mechanism. Next subsection introduces security, privacy, trust metrics and package transmission approach of feature

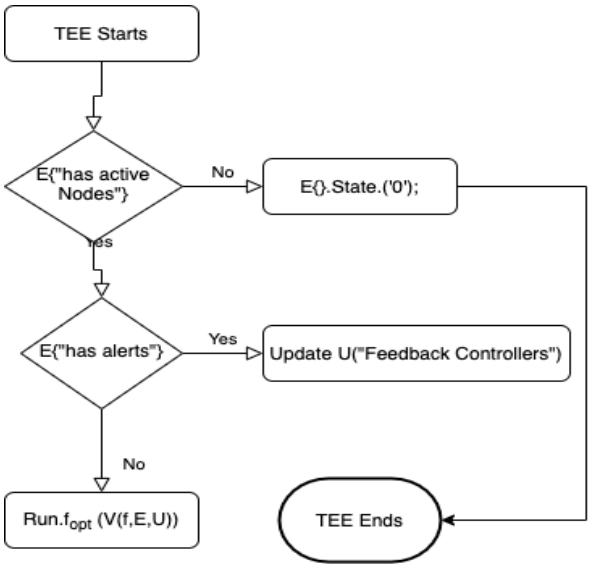

FIGURE 10. a End-to-end TEE Flow Diagram.

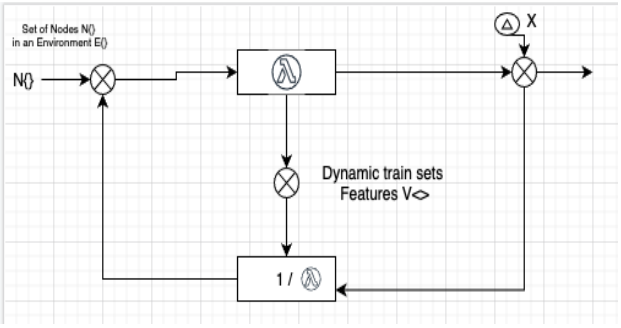

FIGURE 10. b High-level view of proposed Mechanism and learning approach for continuous growth.

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.

Fortunately, emerging TEE mechanisms like Open-TEE [9] 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 1 and Table 2 for the identified metrics

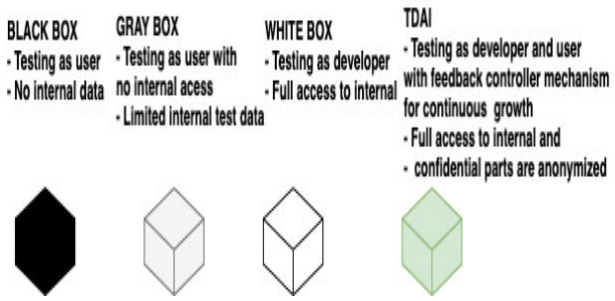

FIGURE 11. TDAI vs other approaches.

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 11.

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 2 as critical features of the observed environment E{} with set of nodes $N_{E:}\{N_1, N_2, N_3, \ldots, N_n\}$.

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 10.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 $\boldsymbol{f_{opt} = \arg\max_{f} V(f, E, U)}$ maximizes expected utilities of each node via trusted agents $\boldsymbol{f_{TrustedAgent}\{A\{\}\}(EN[*]\})}$.

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 $\boldsymbol{EN[*]\}}$ with

- Centralized,
  - $N_{E:}\{N_1, N_2, N_3, \ldots, N_n\}$ are fully connected to master node.
- Decentralized/autonomous/embedded
  - $N_{E:}\{N_1, N_2, N_3, \ldots, N_n\}$ are not connected but has (near) real time connectivity feature to each node and master when required
- Distributed/hybrid-design perspectives of TDAI methodology.
  - $N_{E:}\{N_1, N_2, N_3, \ldots, N_n\}$ 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 10. Growth-flow of (1) Decentralized (2) Centralized (3) Distributed design can be arranged as below equation 10. Since, distributed design can enable to maximize total throughput of each node X(N) and total system.

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

$$\begin{aligned} &f_{DecentralizedGrowth\ of\ \mathrm{E[N]}}(N_1, N_2, \ldots, N_n) \\ &\quad < f_{CentralizedGrowth\ of\ E[N]}(N_1, N_2, \ldots, N_n) \\ &\quad < f_{DistributedGrowth\ of\ E[N]}(N_1, N_2, \ldots, N_n) \end{aligned} \tag{10}$$

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 section and will be discussed in details within the related future works.

$$\begin{aligned} &f_{NumberOfAlerts\ of\ Distributed\ \mathrm{E\{N[*]\}}}(N_1, N_2, \ldots, N_n) \\ &\quad < f_{NumberOfAlert\ Centralized\ \mathrm{E\{N[*]\}}}(N_1, N_2, \ldots, N_n) \\ &\quad < f_{NumberOfAlert\ Decentralized\ \mathrm{E\{N[*]\}}}(N_1, N_2, \ldots, N_n) \end{aligned} \tag{11}$$

As formulated in above equation 11, 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 11, 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

$$f_{\text{Trusted Agent}\{A\{\}\}}(EN[*]\}) \quad \text{Trusted Agent A}\{\} = \{iN_i \text{and with activation function } a_i\} \tag{12}$$

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 $EN[*]\}$,

$$N_i = \sum_{j=0..N} w_{ij} a_j$$
$$a_i = g(N_i) = g\left(\sum_j w_{ij} a_j\right). \tag{13}$$

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 trust level of the nodes within the observed $EN[*]\}$ as dynamic growth-flow mechanism of TDAI methodology as stated in above equation 12 and equation 13.

Next section introduces initial experimental validation results with trust factor coefficient theorem [7] based total system throughput maximization approach. In which, we experimentally validate and demonstrate promising performance of TDAI methodology within real-life use-cases of a smart-city to minimize the risk alerts within the observed context.

IV. EVALUATION AND DISCUSSION

A. EXPERIMENTAL VALIDATION ACTIVITIES

In order to demonstrate the proof for TDAI methodology, each critical metrics and parameters up to impact on throughput level of each node's trust factor [7] listed in Table 2 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 $EN[*]\}$.

Thereby, set of trusted agents $f_{Trusted\ Agent\{A\{\}\}}(EN[*]\})$ 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{}. The time span and observation metric/parameters can be limited to scope of Figure.13 to demonstrate TDAI performance as in the defined cross-border smart-city scenario.

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{}.

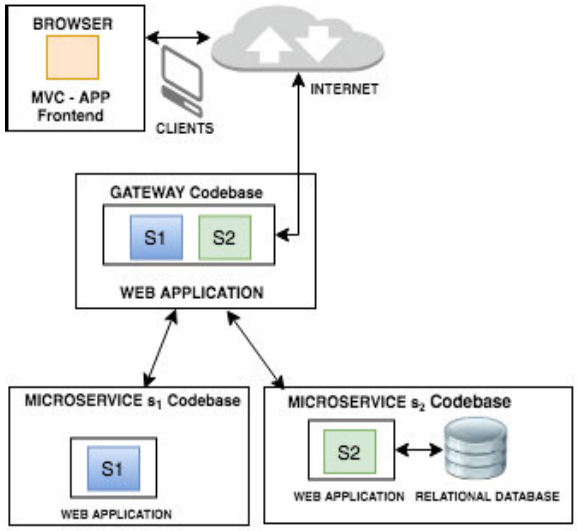

FIGURE 12. TDAI micro-service architecture for trusted interactivity.

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 13.

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 14 with dynamic holistic views of the observed environment $EN[*]\}$.

Dynamic holistic views are updated dynamically with data sensor fetching as illustrated in Figure 10.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.10.b. Furthermore, Figure 12 illustrates the TDAI microservice architecture, is designed to ensure and maximize interactivity of the trusted agents. Next chapter 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.

The simulation scenario for experimental validation, is visualized at Figure.14, targets trusted observation within the $EN[*]\}$ is defined to detect critical risk alerts with TDAI utilization. The alerts are ported to growth-flow mechanism of TDAI approach via package p<> transmission dynamically to train the DigiBank hybrid-cloud system, which have 5G connectivity features also for (near) real-time package transmission capacities of the monitored nodes N{}.

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

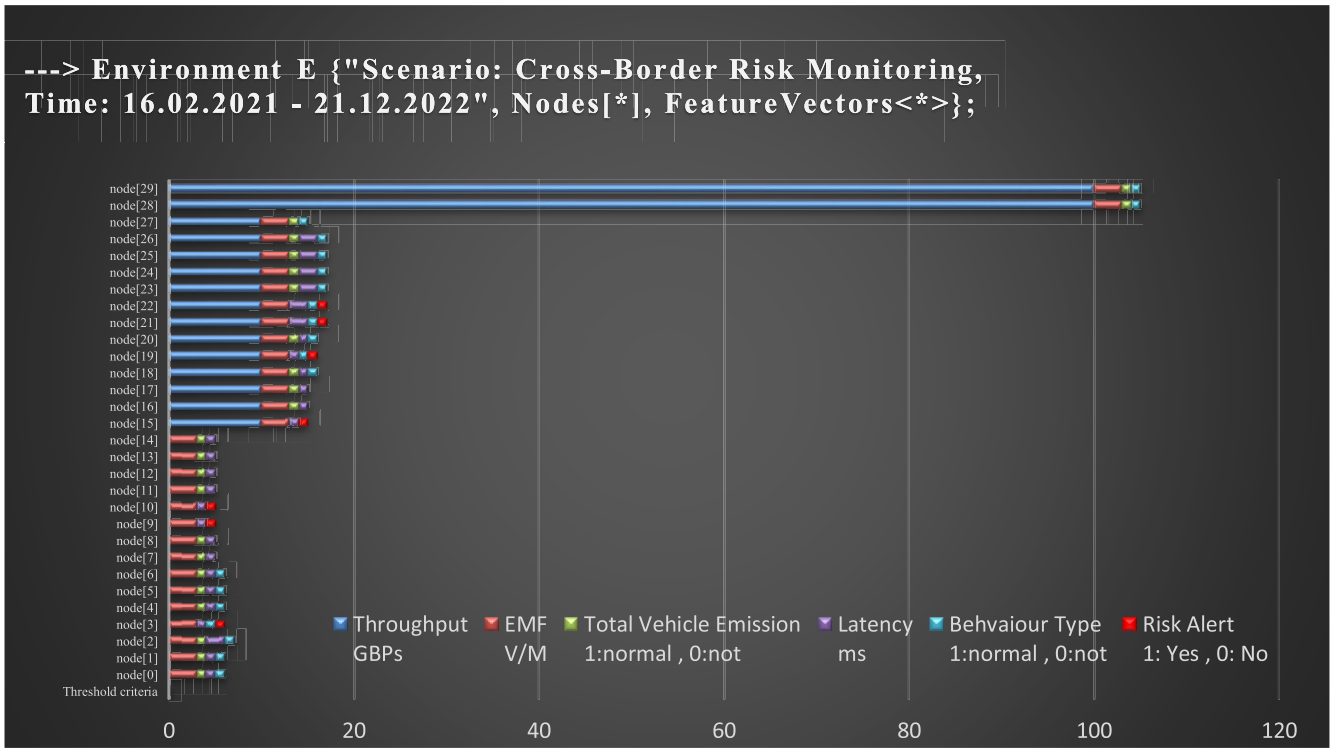

FIGURE 13. TDAI sample experimental observation.

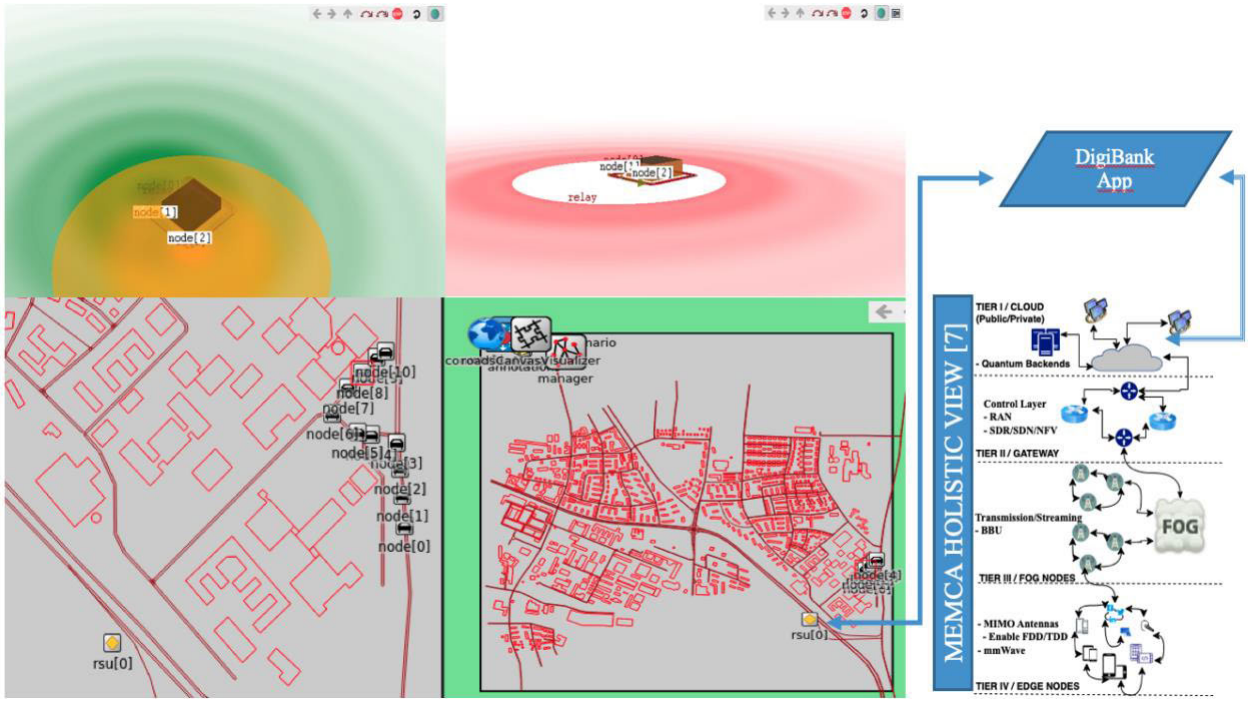

FIGURE 14. MEMCA-Hybrid cloud for generic smart city emulation and TDAI trusted interactivity experimental observation with sample DigiBank monitoring application [7].

B. CORRELATION AND ALERTING APPROACH

As stated in trust factor coefficient theorem [7], 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 14.

TABLE 4. Alert sources in a dynamic environment with set of nodes $E\{N[*]\}$.

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		

Additionally, Table 2 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 14 below formulates the risk monitoring model and trust value correlation of an environment $EN[*]\}$.

$$t_{N_i} = Average\left(\sum_{0}^{i} \frac{N_{i.ttrustvaluet\in\{0to1\}}}{N_{i.R(total\ risk\ alert\ number)}}\right), \quad N \epsilon the\ nodes\ in\ observed\ Environmment\ \boldsymbol{E}\{\boldsymbol{N}[*]\} \quad (14)$$

As stated in above Equation 14, 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.4 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.14. 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.
- **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.

Furthermore, other metrics from data sensors e.g. data collected through the OBD dongle includes Gas pedal position (%), RPM, 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 $f_{Trusted\ Agent\{A\{\}\}}(EN[*]\})$.

However, each feature causes a justification process and increases the trust cost exponentially as formulated in Figure.1.a. Therefore, TDAI limits the critical constraint and metrics as selected in Table.1. 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.13 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 X(N) and total system with utilization of TDAI with maximized growth-flow of the observed environment $EN[*]$.

To sum up, it is observed that in initial simulations of TDAI, as briefed in Figure 14, which has trusted interactivity since it can be assured at massive scale with minimum risk alerts as summarized in Table 4. 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 [7] for accelerated growth-flow mechanism.

MEMCA 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 with minimized risk alert and maximum growth-flow performance as summarized in Figure 13, 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.

V. CONCLUSION

To sum up, we can state that three main architectural perspectives (central, decentral/ autonomous/ embedded, 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 4, 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 $EN[*]\}$.

REFERENCES

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[2] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *J. Mach. Learn. Res.*, vol. 18, pp. 1–43, Apr. 2018.

[3] M. Milutinovic, A. G. Baydin, R. Zinkov, W. Harvey, D. Song, F. Wood, and W. Shen, "End-to-end training of differentiable pipelines across machine learning frameworks," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, 2017. [Online]. Available: https://openreview.net/pdf?id=ryh7qqGRZ

[4] A. Griewank and A. Walther, "Algorithm 799: Revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation," *ACM Trans. Math. Softw.*, vol. 26, no. 1, pp. 19–45, Mar. 2000.

[5] G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," in *Proc. Spring Joint Comput. Conf. AFIPS (Spring)*, 1967, pp. 483–485.

[6] N. J. Gunther, "A general theory of computational scalability based on rational functions," 2008, *arXiv:0808.1431*.

[7] M. A. Agca, "A holistic abstraction to ensure trusted scaling and memory speed trusted analytics," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Dec. 2019, pp. 1428–1434.

[8] K. Kirkpatrick, "Software-defined networking," *Commun. ACM*, vol. 56, no. 9, pp. 16–19, Sep. 2013.

[9] B. McGillion, T. Dettenborn, T. Nyman, and N. Asokan, "Open-TEE—An open virtual trusted execution environment," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, vol. 1, Aug. 2015, pp. 400–407.

[10] S. Samsi et al., "The MIT supercloud dataset," 2021, *arXiv:2108.02037*.

[11] J. Kepner, T. Davis, C. Byun, W. Arcand, D. Bestor, W. Bergeron, V. Gadepally, M. Hubbell, M. Houle, M. Jones, A. Klein, P. Michaleas, L. Milechin, J. Mullen, A. Prout, A. Rosa, S. Samsi, C. Yee, and A. Reuther, "75,000,000,000 streaming inserts/second using hierarchical hypersparse GraphBLAS matrices," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. Workshops (IPDPSW)*, May 2020, pp. 207–210.

[12] J. Kepner et al., "Spatial temporal analysis of 40,000,000,000,000 internet darkspace packets," 2021, *arXiv:2108.06653*.

[13] A. Reuther, J. Kepner, C. Byun, S. Samsi, W. Arcand, D. Bestor, B. Bergeron, V. Gadepally, M. Houle, M. Hubbell, M. Jones, A. Klein, L. Milechin, J. Mullen, A. Prout, A. Rosa, C. Yee, and P. Michaleas, "Interactive supercomputing on 40,000 cores for machine learning and data analysis," in *Proc. IEEE High Perform. Extreme Comput. Conf. (HPEC)*, Sep. 2018, pp. 1–6.

[14] J. Kepner, V. Gadepally, H. Jananthan, L. Milechin, and S. Samsi, "AI data wrangling with associative arrays," 2020, *arXiv:2001.06731*.

[15] H. Tataria, M. Shafi, A. F. Molisch, M. Dohler, H. Sjöland, and F. Tufvesson, "6G wireless systems: Vision, requirements, challenges, insights, and opportunities," *Proc. IEEE*, vol. 109, no. 7, pp. 1166–1199, Jul. 2021.

[16] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28573–28593, 2018.

[17] M. A. Agca, S. Faye, and D. Khadraoui, "A survey on trusted distributed artificial intelligence," *IEEE Access*, vol. 10, pp. 55308–55337, 2022.

[18] (Aug. 4, 2023). *The EESC Proposes Introducing EU Certification for 'Trusted AI' Products*. [Online]. Available: https://www.eesc.europa.eu/en/news-media/news/eesc-proposes-introducing-eu-certification-trusted-ai-products

[19] M. Ouedraogo, D. Khadraoui, B. De Remont, E. Dubois, and H. Mouratidis, "Deployment of a security assurance monitoring framework for telecommunication service infrastructures on a VoIP service," in *Proc. New Technol., Mobility Secur.*, Nov. 2008, pp. 1–5.

[20] (Aug. 4, 2023). *Overseeing and Implementing The United States National AI Strategy*. [Online]. Available: https://www.ai.gov/#Research_and_Development_for_Trustworthy_AI

[21] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. van den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, and A. Kandala, "Evidence for the utility of quantum computing before fault tolerance," *Nature*, vol. 618, no. 7965, pp. 500–505, Jun. 2023.

MUHAMMED AKİF AĞCA (Member, IEEE) received the degree from Middle East Technical University (METU), in 2011, the master's degree in computer engineering from Turkish Stock Market Union Economy and Technology University (TOBB ETU), Ankara, Türkiye, in 2015, and the Ph.D. degree in computer science and computer engineering from the University of Luxembourg, in 2023.

He joined HAVELSAN Aerospace and Defense Company, in 2014, as a Software Engineer and a NATO Coordinator for two years. He is currently doing his research and development studies in various groups as a full-stack systems architect and a researcher.

SÉBASTIEN FAYE (Member, IEEE) received the Graduate degree in telecommunication networks from the University of Picardie Jules Verne, Amiens, France, in 2009, and the Ph.D. degree from Telecom ParisTech, Paris, France, in 2011. During the M.S. thesis, he carried out several studies on wireless sensor networks and the security mechanisms they offer. Between 2009 and 2014, he was a web entrepreneur and created several web services and two startups. During the Ph.D. degree, he focused on distributed systems applied to intelligent transportation systems. His studies focused on how sensors equipped with magnetometers and short-range wireless radio interfaces can be deployed along with the road networks to detect the passage of vehicles and exchange traffic count information, investigating their deployment, and performance in the area of traffic light management. Between 2014 and 2017, he was a Research Associate with the SnT/University of Luxembourg and responsible for the VehicularLab Team. He is currently a Research and Technology Associate with the ITIS Department, Luxembourg Institute of Science and Technology (LIST), and works on projects related to connected mobility, the IoT, wireless sensor networks, and 5G. He is an expert in mobile computing and has already worked on several national and European initiatives related to implementing embedded intelligence to address mobility, traffic flow optimization, and human activity detection. In particular, he has implemented a series of demonstrators designed to assist end-users in their daily mobility choices. He has recently started working in the 5G domain, with active involvement in several national and European initiatives. He is also the author of two recently filed patents about location profiling based on networks traces. He has authored or coauthored about 50 journals and refereed conference papers.

DJAMEL KHADRAOUI (Member, IEEE) is currently the Head of the Reliable Distributed Systems Research Unit (READY), LIST, Luxembourg. His current research interests include trusted distributed and optimized systems. These systems can be physical infrastructures (buildings, roads, vehicles, and drones), but they are also more and more increasingly IT infrastructures like data centers, networks, and clouds. It is important to collect and manage data about the functioning of these infrastructures in order to guarantee the quality of the delivered services. This is the role of service systems to transform these data into information and decisions guaranteeing the alignment of the infrastructures with the business services delivered. Examples of such systems include: supply chains, mobility systems, ICT/Telco systems and communication networks, remote manufacturing, and space systems. The reliable distributed systems research activities are dealing with the design, the security, and the optimization of service systems enabled by data-intensive infrastructures engineering and aligned with the creation of business impact. The actual research activities will target the following topics, such as optimization and decision-making systems (OPTIMISE), edge computing and networks (EDGE), and trustworthy data systems (TRUST).