

Trustworthy, Explainable, and Sustainable Decentralized Intelligence for 6G Networks

Giovanni Perin, Michele Rossi, Enrique Tomás Martínez Beltrán, Fernando Torres-Vega, José María Jorquera Valero, Manuel Gil Pérez, Eunjeong Jeong, Nikolaos Pappas, Farah Abed Zadeh, Chamara Sandeepa, Bartłomiej Siniarski, Madhusanka Liyanage, Betül Güvenç Paltun, Leyli Karaçay, Ioannis Pitsiorlas, and Marios Kountouris

As 6G networks transition from theoretical frameworks into operational realities, artificial intelligence (AI) evolves from an add-on optimization tool into a distributed and interconnected structural layer. Unlike previous network generations that mostly relied on centralized cloud analytics platforms, AI-native 6G networks operate across a dynamic, multi-domain edge-cloud continuum where data originates from heterogeneous sources including user devices, radio access networks, sensing infrastructures, and vertical applications. Centralizing this massive volume of data creates severe communication overhead, unacceptable latency bottlenecks, single points of failure, and complex cross-domain governance challenges.

Consequently, *decentralization becomes a fundamental architectural requirement* for future 6G network intelligence and zero-touch operations. Security serves as the primary enabler of this decentralized paradigm. Critical security functions, such as real-time threat detection, physical-layer attack mitigation, slice protection, and intrusion detection, require immediate access to local context and telemetry before operational data loses its value. However, moving intelligence to the edge via collaborative paradigms like federated learning (FL) and decentralized FL (DFL) introduces complex trade-offs. System security cannot be addressed in isolation; it is deeply intertwined with equally important aspects like *trustworthiness*, *explainability*, and *energy sustainability*. To reliably operate 6G networks within automated control loops, distributed security models must prove resilient against input manipulation, poisoning attacks, and privacy leakage. Explainable AI (XAI) must evolve from post-hoc explanations intended for human inspection into actionable, machine-readable operational signals that quantify model uncertainty, detect concept drift, and validate peer alerts across administrative domains. Simultaneously, security mechanisms must respect strict energy constraints at edge and Internet of Things (IoT) nodes, but also at base stations and network controllers at the edge boundary, which represent a massive portion of localized energy consumption and

This work has been funded by the European Commission through the Horizon Europe/JU SNS project ROBUST-6G (grant no. 101139068) and through the Italian Ministry of University and Research under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, project SERICS/ISP5G+ (CUP D33C22001300002).

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

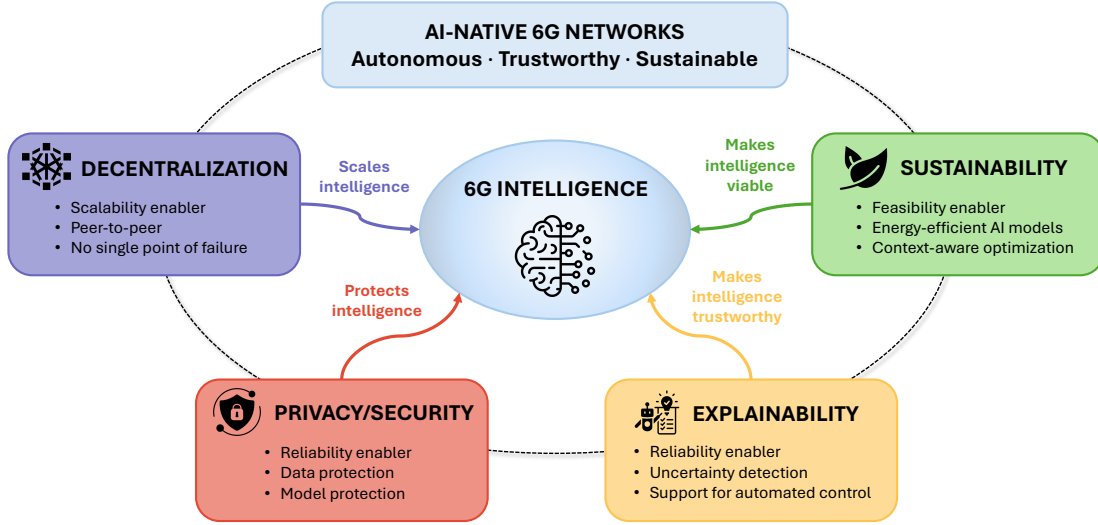


Fig. 1. Interdependent design dimensions of intelligence in AI-native 6G networks. Decentralization enables scalable distributed intelligence, privacy/security and explainability provide complementary reliability mechanisms, and sustainability ensures its operational feasibility.

hardware footprint. To achieve true edge-to-access sustainability, we advocate combining joint energy- and information-aware distributed training schedulers for AI models and implementing such models in dedicated ultra-low-power hardware that is energy efficient by design. For this, we identify neuromorphic platforms involving spiking neural networks (SNNs) and reservoir computing as the computational means. Figure 1 summarizes this vision: decentralization provides the scalability required to distribute intelligence across 6G networks, privacy/security and explainability provide complementary forms of reliability, and sustainability ensures its long-term operational feasibility.

Taking these aspects into account, this paper develops a unified perspective on decentralized intelligence for 6G, arguing that decentralization, trustworthiness, explainability, and sustainability must be designed jointly rather than treated as independent requirements. We start by establishing the architectural transition from centralized machine learning to peer-to-peer decentralized FL, formulating graph-regularized consensus mechanisms that support low-latency threat mitigation across dynamic multi-domain topologies. We then analyze security threats across the AI model lifecycle alongside privacy-preserving cryptographic and hardware-based techniques tailored to distributed environments. Moving beyond traditional explanations, we argue that feature attributions (anomaly scores) and uncertainty metrics can serve as actionable control-plane signals to automatically detect and gauge security threats, driving model self-optimization, policy enforcement, and cross-peer trust verification. Finally, we address energy efficiency as a core design objective, advocating dedicated neuromorphic hardware and semantics-aware scheduling strategies that balance battery states with information freshness in energy-harvesting scenarios.

DECENTRALIZED INTELLIGENCE IN AI-NATIVE 6G SYSTEMS

We start our discussion by underlining the architectural role of decentralized intelligence in 6G security, discussing why centralized AI pipelines are insufficient, how federated and decentralized learning enable collaborative model generation without centralizing raw data, and how distributed intelligence can support closed-loop security management.

Limitations of Centralized AI for 6G

Centralized AI simplifies the learning pipeline by consolidating data collection, model training, validation, and deployment within a limited number of controlled infrastructure environments. This approach has been effective in many cloud-based analytics systems, but it becomes increasingly restrictive in the context of AI-native 6G security.

Its first limitation is *scalability*. Future 6G systems will generate large volumes of heterogeneous data from radio, transport, core, edge, application, and sensing domains. Moving this data continuously to a central collector can create excessive communication overhead and energy consumption. The problem becomes particularly relevant for services that require continuous monitoring, rapid model adaptation, or the integration of data from a large number of distributed nodes.

A second limitation is *latency*. Many applications requiring real-time decisions must be executed close to the point of observation. Examples in the domain of security include detecting abnormal traffic in a network slice, identifying suspicious radio fingerprints, reacting to jamming or spoofing attempts, or adapting an intrusion detection model at the network edge. In these cases, centralized analysis may introduce delays that are incompatible with the control timescale of the security function. Intelligence that is not timely enough to support mitigation becomes primarily forensic, rather than operational.

A third limitation is *governance*. 6G infrastructures will be inherently multi-domain and multi-stakeholder. Operators, vendors, verticals, edge/cloud providers, and service tenants may all contribute to data and computational resources while retaining distinct privacy, regulatory, and business constraints. Under these settings, a centralized data collector or model owner is often unrealistic. Decentralized intelligence enables generating collaborative security models while preserving domain autonomy.

Centralized AI also introduces points of concentration in the learning and decision-making pipeline. Data repositories, model-training platforms, aggregation servers, and model registries become critical assets whose failure or compromise may affect large portions of the system. Decentralization reduces such dependency by distributing learning and decision support across multiple nodes, provided that the learning process is designed to operate under heterogeneous, intermittent, and partially trusted participation.

Distributed Model Training: Federated and Decentralized Learning

In the path towards decentralization, FL provides a natural mechanism for distributed model training in 6G security. Instead of transferring raw data to a centralized repository, each participant trains a local model using its own observations and exchanges model-related information, such as parameters or gradients. Collaborative learning can therefore take place across devices, edge nodes, or administrative domains, while *data remain within their original domains*.

For 6G security services, FL can support the collaborative training of intrusion detection models, anomaly detectors, traffic classifiers, radio-frequency (RF) fingerprinting models, and attack prediction mechanisms. Local nodes learn from their own operational context, while the overall system benefits from knowledge distributed across multiple domains. This approach is particularly relevant when data sharing is restricted by privacy regulations, commercial policies, or security requirements.

Conventional FL, however, commonly relies on a *central aggregator*, often referred to as the parameter server (PS). The PS coordinates the training process: it receives local updates, combines them into a global model, and redistributes the resulting model to the participants. Although effective in many settings, the implied star topology is not always suitable for 6G. In fact, the PS may become a communication bottleneck, a privacy-sensitive entity, an attack target, and a *single point of failure*. Moreover, conventional FL assumes a level of centralized trust that may not exist in multi-domain deployments.

DFL addresses these limitations by replacing the central aggregator with peer-to-peer collaboration [1]. The nodes that take part in the distributed training exchange model updates with selected neighbors according to a potentially time-varying topology. Model training is governed by local optimization, neighborhood-level aggregation, and progressive agreement among the involved peers. The architectural transition from centralized machine learning (ML) to FL and DFL is summarized in Fig. 2.

To describe the decentralized learning process mathematically, consider a network of K learning nodes represented at communication round t by the weighted graph $\mathcal{G}^t = (\mathcal{V}, \mathcal{E}^t, \mathbf{A}^t)$, where $\mathcal{V} = \{1, \dots, K\}$ denotes the set of participating nodes and $\mathcal{E}^t$ the set of active communication links. An edge $(k, j) \in \mathcal{E}^t$ indicates that nodes k and j can exchange model information. The matrix $\mathbf{A}^t = [a_{kj}^t]$ contains the non-negative coupling weights associated with the active links, where a_{kj}^t controls the strength of model agreement between nodes k and j . Node k stores a local dataset $\mathcal{D}_k$, maintains a model $\mathbf{w}_k^t$ (a set of “weights” describing the model being trained), and minimizes the local objective $F_k(\mathbf{w}_k \mid \mathcal{D}_k)$. The set $\mathcal{S}_k^t$ contains the neighboring nodes whose updates are available to node k during round t . A graph-regularized DFL objective function is

$$\Phi^t(\mathbf{W}) = \sum_{k=1}^K p_k F_k(\mathbf{w}_k \mid \mathcal{D}_k) + \frac{\lambda}{2} \sum_{(k,j) \in \mathcal{E}^t} a_{kj}^t \|\mathbf{w}_k - \mathbf{w}_j\|_2^2, \quad (1)$$

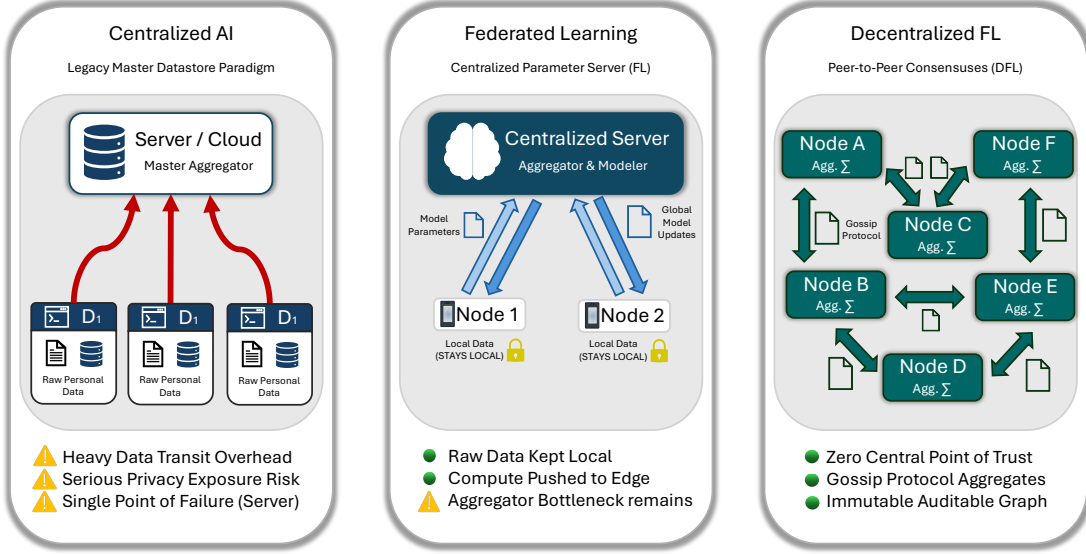


Fig. 2. Evolution of AI paradigms from centralized to fully decentralized architectures.

where $\mathbf{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_K\}$ denotes the collection of local models. The second term in (1) promotes consensus among neighboring models. The coefficients p_k weight the local objectives, whereas λ controls the overall strength of the consensus regularization. The edge-specific coefficients a_{kj}^t determine how strongly the models associated with individual active links are encouraged to agree. Representative local-adaptation and robust neighborhood-aggregation steps consistent with this decentralized learning architecture are

$$\text{local adaptation: } \mathbf{z}_k^t = \mathbf{w}_k^t - \eta_t \tilde{\nabla} F_k(\mathbf{w}_k^t | \mathcal{B}_k^t), \quad (2)$$

$$\text{neighborhood aggregation: } \mathbf{w}_k^{t+1} = (1 - \rho_t) \mathbf{z}_k^t + \rho_t \text{RobAgg}(\{\mathbf{z}_j^t\}_{j \in \mathcal{S}_k^t}, \{\alpha_{kj}^t\}_{j \in \mathcal{S}_k^t}).$$

The stepsize η_t controls the local adaptation of the weights, where $\mathcal{B}_k^t \subseteq \mathcal{D}_k$ is the data (mini-batch) available at node k to compute the local stochastic gradient $\tilde{\nabla} F_k$. The local adaptation step can be repeated for multiple local iterations with different mini-batches before setting the value $\mathbf{z}_k^t$ and using it for the aggregation step. The coefficient ρ_t balances local adaptation and neighborhood collaboration, whereas the operator $\text{RobAgg}(\cdot)$ represents a robust aggregation mechanism selected according to the deployment and threat model. More generally, the aggregation mechanism in (2) provides a natural interface between decentralized learning and the trustworthiness and sustainability requirements discussed throughout this paper. The weights α_{kj}^t and the operator $\text{RobAgg}(\cdot)$ need not depend solely on network connectivity, but may account for the *reliability* of individual peers, *uncertainty or anomaly indicators* associated with their updates, *consistency of their explanations* with locally observed evidence, and their current energy and computational *resources*. Hence, neighborhood collaboration can be made *context-aware*: model

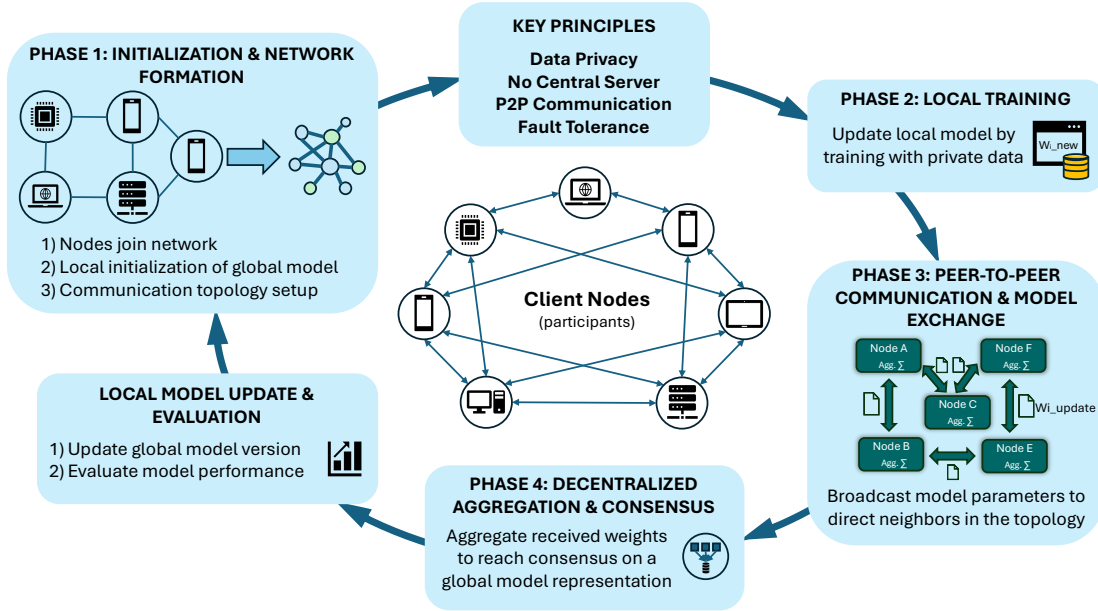


Fig. 3. Schematic overview of the DFL closed-loop.

updates are not necessarily treated equally, but can be weighted, filtered, or discarded based on their estimated trustworthiness, informational value, and cost. The DFL closed-loop shown in Fig. 3 allows for heterogeneous participation, intermittent connectivity, and multi-domain operation, while adapting the learning process to the topology and operational constraints of the underlying infrastructure.

This peer-to-peer paradigm is particularly well-suited to 6G, where intelligence is expected to span the cloud, edge, far-edge, and device layers without relying on a single coordination point. Model-sharing relationships may be horizontal, e.g., among edge nodes from different domains, or vertical, e.g., between cloud platforms, edge nodes, and constrained devices. DFL thus enables the learning process to follow the topology and operational constraints of the infrastructure, while reducing dependence on centralized aggregation and preserving the autonomy of participating entities.

Distributed Intelligence and Zero-Touch Management

Importantly, the value of decentralized intelligence extends beyond collaborative model training. Distributed learning can *become part of the network control fabric*, supporting continuous observation, adaptation, and mitigation. While conventional ML pipelines often treat model training, deployment, monitoring, and orchestration as separate stages, AI-native 6G systems can benefit from tighter interactions among these functions: local observations update distributed models, model outputs support network decisions, actions modify the network state, and the resulting changes generate new data distributions that may spur further adaptation.

A *zero-touch architecture* therefore places decentralized intelligence at the interface between learning and control across the edge–cloud continuum. Edge and far-edge nodes not only execute inference tasks, but also monitor their local environments, exchange model information and learning evidence with peers, and provide operational signals to orchestration functions. Model updates, anomaly scores, confidence indicators, resource constraints, and trust-related metadata can serve as inputs to zero-touch management mechanisms, triggering actions such as attack mitigation, network reconfiguration, slice protection, access-control enforcement, or network resource allocation.

However, before the outputs of distributed entities can be incorporated into automated control loops, distributed models and learning processes must be assessed against several trustworthiness dimensions. Relevant properties include robustness, privacy, fairness, explainability, accountability, and sustainability, as well as the reliability of nodes and model updates during training. These requirements establish the connection between decentralized intelligence and the trustworthiness mechanisms discussed in the following section.

Realizing this vision under realistic 6G conditions remains challenging. Nodes are heterogeneous in computational capacity, data is non-IID, and channel quality varies over time and across nodes. Participation may be intermittent, adversaries may be adaptive, and privacy- and security-preserving mechanisms may introduce additional overhead. Future FL and DFL systems must therefore jointly address learning accuracy, communication cost, energy consumption, privacy leakage, robustness, and operational transparency.

TRUSTWORTHINESS & EXPLAINABILITY OF AI MODELS

Robustness to Manipulated Input

AI models in 6G may be exposed to security threats that target different stages of the model lifecycle, from data collection and training to deployment and inference [2].

During *training*, poisoning attacks can corrupt the data or gradients to create backdoors or degrade model performance. Poisoning attacks are training-time attacks in which an adversary deliberately manipulates training data, labels, gradients, or model updates to influence the learned model [3]. These attacks are especially challenging to detect in large-scale, heterogeneous 6G environments, where data originates from multiple, potentially untrusted sources with varying quality and statistical properties. At *inference* time, adversarial attacks can subtly manipulate inputs through carefully crafted perturbations that remain imperceptible to humans but cause incorrect model predictions. In the context of 6G, such mispredictions may lead to critical failures, such as incorrect resource allocation, degraded quality of service, or even large-scale network disruptions.

Mitigating poisoning and adversarial input attacks in 6G AI systems requires a holistic defense strategy that combines data-centric and model-centric mechanisms across the entire learning pipeline. For poisoning attacks, robust data validation and sanitization techniques are key to filtering out anomalous or malicious inputs before they influence the training process. Anomaly detection methods, including statistical outlier detection, explainability-guided detection, and learning-based approaches, can identify inconsistent updates or suspicious patterns during distributed training. In federated settings, reputation-based filtering and trust scoring of participating nodes can help isolate unreliable contributors, while robust aggregation techniques—such as median-based or trimmed-mean methods [4]—limit the impact of corrupted updates on the global model. At the model level, adversarial robustness can be enhanced through techniques such as adversarial training [5], [6], where models are explicitly trained with perturbed inputs to improve resilience. Additionally, uncertainty estimation and ensemble learning [7] approaches can reduce overconfidence in predictions, enabling the system to flag potentially manipulated or out-of-distribution data.

Ensuring Data Privacy

Privacy-preserving AI refers to a set of techniques designed *to limit the exposure of sensitive data* and protect confidential information throughout the whole AI lifecycle [8]. This protection is particularly important in AI-enabled 6G networks, where massive volumes of sensitive data are continuously generated by users, devices, applications, and network operations. In this context, model privacy refers to protecting the AI model itself and *the information that may be inferred from it*. Even when raw training data are not directly accessible, an adversary may exploit model outputs, parameters, gradients, updates, or prediction interfaces to infer sensitive information about the training data, determine whether a specific record was used during training, reconstruct private features, or extract and replicate the model. Thus, both training-data privacy and model confidentiality may be required, depending on the deployment scenario.

Unlike centralized AI systems, 6G environments are inherently distributed and multi-stakeholder. Data is generated and stored across edge devices, network functions, and service providers, making direct data sharing impractical or undesirable. As a potential solution, privacy-preserving AI techniques aim to enable collaborative learning and inference without exposing raw data and ensuring that sensitive information remains protected. However, achieving strong privacy guarantees while maintaining model accuracy, efficiency, and scalability remains a key challenge. To address privacy risks, a range of complementary techniques is being explored for 6G AI systems. For example, federated learning enables distributed model training by keeping data localized on devices, reducing the direct exposure of sensitive information, while secure aggregation protocols prevent leakage through shared model updates. Cryptographic methods

such as homomorphic encryption (HE) and secure multiparty computation (SMC) allow computations to be performed without exposing raw data to ensure confidentiality even in untrusted environments, although often at the cost of increased computational overhead. Trusted execution environments (TEEs) provide hardware-based protection for secure model execution and data processing. In addition, differential privacy introduces controlled noise into data or model updates to limit the risk of information leakage about individual data points. Despite these advances, privacy-preserving mechanisms introduce trade-offs in accuracy, latency, use of computational resources, and scalability, requiring careful integration and optimization to meet the high performance requirements of 6G networks.

While FL avoids sharing raw data, model updates themselves can still leak sensitive information about the underlying training data. This risk has motivated the development of integrated privacy-aware learning frameworks that combine federated learning with complementary protection mechanisms. At the same time, a key challenge arises from the need to perform model validation, robustness analysis, and attack detection *without exposing model updates in plaintext*. For instance, some approaches leverage encryption techniques to encode model updates and compute similarity between them, enabling anomaly detection while preserving confidentiality. These mechanisms are particularly useful to identify adversarial behaviors, such as poisoning attacks, in collaborative learning environments. Referring to DFL architectures, privacy-preserving mechanisms can also be incorporated into peer-to-peer learning to protect model updates without relying on a central aggregator. In this case, to increase privacy, participants can encrypt local model updates in a peer-to-peer manner and jointly perform secure aggregation through collaborative cryptographic protocols. Although this improves privacy and eliminates the single point of failure, it requires coordination and techniques such as key sharing to manage cryptographic keys between peers and enable secure aggregations.

Overall, implementing privacy-preserving AI requires striking a careful balance between protecting sensitive information and meeting the security, scalability, energy, and low-latency requirements of 6G networks.

Explainability, Transparency, and Trust in 6G AI

As AI becomes a core part of 6G, the ability to understand and trust automated AI decisions is essential. In 6G networks, AI and ML models are used across the entire system, from detecting attacks at the network edge to managing threats at the core with little human involvement. When the reasoning behind a model cannot be examined, it is not possible to detect its errors or understand how it could be misled. It is important to distinguish two related notions: *interpretability* is the degree to which a model is inherently understandable, as with a decision tree or a linear model whose logic can be read

directly, whereas *explainability* refers to producing human-understandable, usually post-hoc, accounts of why an otherwise opaque model reached a decision. Put simply, an interpretable model explains itself, while an opaque model requires external explanation. XAI provides this external explanation, making the decisions of AI and ML models understandable and verifiable.

Two main types of XAI methods are popular today. The first type is called *model-agnostic XAI*, which works with any model, regardless of its internal structure. Examples include Shapley additive explanations (SHAP), which assigns a score, or *attribution*, to each input feature showing how much it contributed to a prediction; the resulting per-feature scores form an *attribution vector*. Another popular model-agnostic method called local interpretable model-agnostic explanations (LIME) explains individual predictions by testing how small changes to the input affect the output. The second type is *model-specific*. For instance, the layer-wise relevance propagation (LRP) method traces a prediction back through the layers of a neural network to identify which inputs mattered the most.

Among 6G security functions, intrusion detection is where explainability becomes especially relevant. Intrusion detection systems (IDSs) operate continuously at the network edge, processing large volumes of traffic flows, often via ML models, with minimal human oversight. Here, the ability to explain why a flow was flagged or cleared serves two purposes: catching model errors early and maintaining trust in automated security operations. Most XAI research in intrusion detection only focuses on explaining decisions after a model has made them, primarily for human review. While useful, this is not enough for 6G security, where detection must be fast and consistent across multiple operators. Instead, XAI should be built into the full life cycle of an IDS, namely: selecting the most relevant features, reducing model size, detecting when the model starts behaving differently, and deciding when retraining is needed. Beyond supporting the IDS itself, XAI outputs can serve as structured evidence for automated security systems such as near-real-time RAN intelligent controller applications (xApps), non-real-time RAN intelligent controller applications (rApps), and policy engines in open radio access network (O-RAN) to trigger responses without human input. However, when explanations are used to make security decisions, they also become a target for attacks. Methods such as SHAP and LIME can be manipulated to hide malicious behavior behind misleading explanations. For this reason, the explanation system must be designed to be just as secure and reliable as the detection model it supports. As 6G moves toward fully automated operation, explainability *must go beyond identifying why a decision was made*. It must also express how confident the model is in that decision, so that automated systems can act safely and appropriately.

Accordingly, explanations can serve a much larger purpose. They can be used not only to justify decisions but also to support model optimization, trust assessment, and automated security operations. The following section develops these ideas further, examining how explanations and the corresponding

attributions can function as operational signals that support trustworthy and autonomous IDS operation in 6G networks.

Explanations as Actionable, Trustworthy Signals

As observed above, most XAI-for-IDS work stops at post-hoc justification, presenting an attribution vector for a human analyst to read. Yet, an attribution vector is itself a compact, machine-readable summary of the model behavior, and can act not only as an explanation for a human, but as an operational signal inside and around the IDS.

To support autonomous operation, however, explainability must be paired with a *calibrated* assessment of prediction reliability. In fact, an explanation identifies the factors that influenced an IDS decision, but it does not establish whether that decision is sufficiently dependable to justify a subsequent automated network action. This distinction is critical in AI-native 6G environments, where IDS outputs may directly trigger mitigation procedures, traffic isolation, or resource reallocation with limited human intervention. Feature attributions should therefore be accompanied by confidence or uncertainty estimates that characterize the reliability of the underlying prediction. A promising approach is to derive such uncertainty estimates from the latent representations learned by deep anomaly detection models, for example by evaluating latent-space density, similarity to known training samples, or distance from the learned normal-behavior manifold. Samples located in well-represented regions of the normal manifold may support high-confidence benign decisions, whereas samples clearly separated from it may support high-confidence anomaly decisions. By contrast, observations near the decision boundary or in sparsely represented regions should be treated as uncertain and may require additional validation or human escalation. Combined with feature attributions, these indicators would enable xApps, rApps, and orchestration components to determine not only why a decision was made, but also whether it is sufficiently reliable to justify an automated response.

As a signal, these attributions can drive the detector's own lifecycle. Because they rank features by their contribution to a decision, they provide a principled criterion for *feature reduction*. For example, retaining only the features that genuinely drive detection can shrink the model while removing inputs an attacker could cheaply manipulate, thereby reducing the attack surface. The same idea extends inward to *pruning* in artificial neural networks (ANNs), where neurons that contribute little to the important attributions are removed to meet the memory, latency, and energy budgets of edge hardware. Attributions also give an early-warning channel for *concept drift*, the situation in which the statistical relationship between the inputs and the quantity being predicted changes over time, so that a model trained on past traffic gradually becomes stale as attack behavior and normal usage evolve. Such drift is usually caught

only indirectly: accuracy-based tests react only after misclassifications have accumulated, and tests on the raw input distribution are blind to shifts that leave the marginal feature statistics unchanged while altering the input–output relationship the model has learned. Attributions offer a more direct view. Tracking the *distribution* of attributions over a sliding window of traffic, and flagging when its statistical distance from a stable baseline grows large, can reveal that the features the model relies on—its reasoning—have shifted before accuracy visibly degrades, prompting timely retraining.

Exposed outside the detector, the same signal becomes evidence for automated security control. A binary verdict authorizes only a coarse response, block or allow, whereas a verdict carrying its top contributing features supports a *fine-grained* response. In the latter case, the mitigation measure can be targeted at the features driving an alert, access decisions conditioned on whether those features are user-supplied or operator-controlled, and human escalation occurs when the attribution profile matches no known threat or when an attribution provides evidence of out-of-distribution behavior, but its confidence is not sufficiently high to justify an automated decision. Consumed as a compact, schema-conformant record, such explanations let xApps, rApps, and policy engines in the O-RAN architecture act without a human reading them. This matters most when detection is decentralized: as discussed for DFL, when models are trained across operators or edge domains, an attribution record becomes a portable unit of trust, letting a peer’s alert be independently checked against a recipient’s own model and policy before it is acted upon.

Once explanations and attributions drive security actions, *the explainer becomes a target in its own right*, and post-hoc attribution methods are not adversarially robust by construction. For example, in *explanation manipulation* attacks, crafted inputs yield arbitrary attributions while leaving the model prediction unchanged. In *fairwashing*, a model is made to present benign-looking explanations while relying on other features; and *backdoor* attacks embed a trigger during training so that the explainer reports an innocent rationale exactly when the model misbehaves. The consequence is that the trustworthiness properties demanded of the detector, i.e., fidelity, stability, adversarial resilience, privacy, and access control, apply just as strongly to the explainer. Robustness, privacy, and faithfulness, long studied in isolation, become inseparable: before any automated policy is allowed to depend on an attribution, that attribution must be made robust to adversarial inputs, controlled in what it discloses, and faithful to the model at once. Together with the confidence estimates noted above, this yields explanations that are not merely readable but safe to act on.

Two hypotheses follow that we believe deserve investigation. First, because a shift in a model’s reasoning manifests as a shift in its attributions, tracking the *distribution of attributions* over a window of traffic may enable *earlier* and *more reliable* detection of concept drift than tests on the raw input signal.

TABLE I
THREE ROLES OF EXPLANATIONS IN A TRUSTWORTHY, DECENTRALIZED 6G IDS.

Role of XAI	Explanation signal	What it enables
Self-optimization	Feature and neuron attributions	Compact, drift-aware detectors tailored to edge constraints
Control-plane evidence	Machine-readable attribution record	Differentiated, automated response across xApps, rApps, and peers
Assurance	Attribution stability and integrity	Explanations that are robust, private, and faithful before policy acts on them

This particularly applies to drifts that leave the input statistics unchanged while altering the learned input–output relationship. The second hypothesis concerns detecting attacks on the explainer itself. Because tampering perturbs the attribution distribution in ways benign traffic does not, monitoring that distribution may expose manipulation once a trustworthy baseline and realistic tamper models have been formalized. Together, these claims reframe XAI as a measurable, defensible method rather than a presentation layer, which is most needed by a decentralized, trustworthy 6G security stack.

SUSTAINABLE INTELLIGENCE AT THE EDGE

Trustworthy decentralized intelligence is operationally meaningful only if its learning, inference, and security mechanisms can be sustained within the energy budgets of edge and far-edge devices. Sustainability must therefore be treated as a system-level design constraint rather than as a post-hoc optimization objective.

Traditionally, energy minimization in networks is treated as the result of an optimization problem, once the system architecture is established. Considering modern 6G systems, this is deemed insufficient. Energy efficiency and sustainability are, in coherence with the sustainable development goals (SDGs), objectives that must be pursued right from the design phase in modern networks empowered by AI and ML. For this reason, two main research directions that consider the decentralization of in-network processing are identified:

- The design of AI/ML models that are inherently energy-efficient and their deployment in ultra-low-power dedicated hardware;
- The design of networks whose computing facilities are made sustainable through the exploitation of renewable energy resources (RERs) and a wise management of energy storage systems and computation scheduling.

Natively Energy Efficient AI Models

Recently, numerous approaches have been devised to make deep learning models more energy-efficient. Among the most notable context-agnostic methods, we mention *pruning* and *quantization*.

Pruning. Iterative deletion of model parameters (and of the corresponding edges) with absolute values lower than a user-defined threshold, followed by retraining.

Quantization. Reduction of the number of bits used to store a model parameter after training.

The state-of-the-art shows that, often, (i) deleting redundant neurons can yield improvements in accuracy and generalization to unseen samples, reducing overfitting risks, and (ii) even by using 8-bit floating-point numbers, the model accuracy is not significantly degraded, while obtaining, on the other hand, significant advantages in terms of memory footprint and computational power. While these methods are, to a certain extent, effective, they belong to the class of *post-training optimization approaches* to make memory and computation more efficient. As such, even when carefully engineered, these approaches can only provide limited improvements.

In contrast, we argue for a paradigm shift toward deep learning models that are energy-efficient by design, and not only through post-training operations. For this, we deem *neuromorphic computing* especially suitable, specifically (i) *reservoir computing* and (ii) *SNNs*.

Reservoir computing: It is a neuromorphic computing paradigm particularly suited for temporal signal processing. Unlike conventional deep neural networks, where all parameters are optimized through backpropagation, reservoir computing relies on a fixed recurrent dynamical system, the *reservoir*, that projects the input into a high-dimensional temporal representation. Only a shallow output layer is trained, usually via linear regression, by significantly reducing computational complexity and energy consumption. The reservoir naturally retains temporal information through its internal dynamics, making it especially effective for sequential data and time-varying signals. Typical applications include sensor data analysis, traffic forecasting, anomaly detection, and edge intelligence.

Among the most notable approaches are *echo state networks* and *liquid state machines*. The latter establishes a direct connection with spiking neural networks, since the reservoir itself is composed of recurrently connected spiking neurons.

Spiking neural networks: SNNs are dynamical systems that closely mimic how the human brain works [9]. They propagate signals through the emission of *spikes* of current, encoded in the digital domain by binary values, where “1” corresponds to the emission of a spike and “0” leads to no signal propagation.

From a physical perspective, the computationally tractable and most common spiking neuron, the leaky

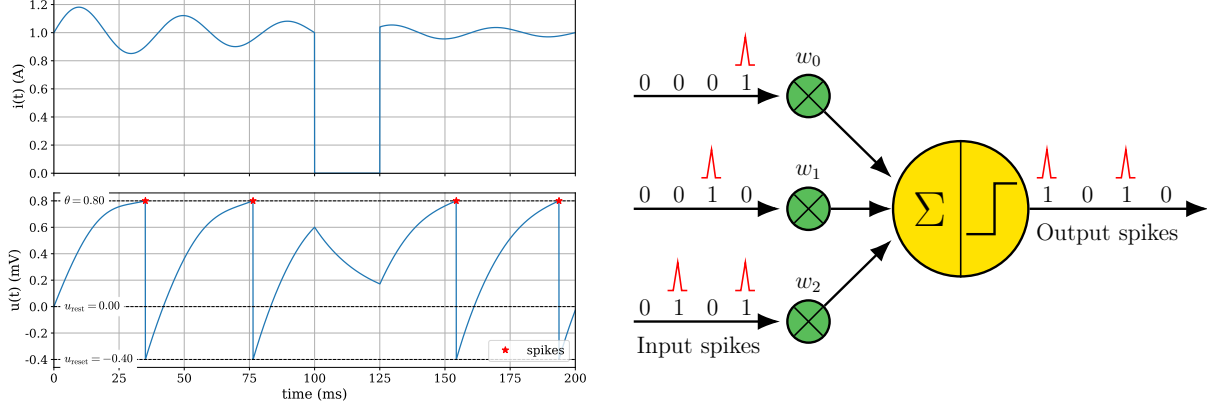


Fig. 4. On the left: input current $i(t) = 1 + A \sin(2\pi \frac{t}{B}) \exp(-Ct)$ (with $A = 0.2$ A, $B = 40$ ms, and $C = 0.01$ ms $^{-1}$), suppressed for $100 \leq t \leq 125$ and the corresponding membrane potential $u(t)$ obtained from Eq. (3) and with the additional mechanisms of the firing and reset thresholds. Here, we used $\tau_m = 20$ ms. On the right: schematic representation of a spiking neuron with three inputs (multiplied by the network weights) and the resulting output current.

integrate-and-fire (LIF) neuron, models an RC electric circuit, i.e., the parallel between a resistance R and a capacitor C . Neurons in SNNs are *stateful*: they dynamically update their *membrane potential* $u(t)$, which represents the neuron's state, through the differential equation

$$\tau_m \frac{du}{dt} = -[u(t) - u_{\text{rest}}] + R i(t), \quad (3)$$

where $\tau_m = RC$ is the membrane time constant, affecting the duration of the potential discharging process in time, u_{rest} is the baseline (rest) potential, and $i(t)$ is the input current. This equation tells us that a LIF neuron charges its membrane potential when an input current is provided, and exponentially discharges over time (it forgets information). In models used in deep learning, two other parameters are defined: a *firing threshold* θ , whose reaching by the membrane potential induces a *spike* and a discharge (in the form of a jump) to a *reset threshold* u_{reset} . A graphic representation of the working principles of SNNs is shown in Fig. 4: on the left, the membrane potential dynamics as a function of the input current; on the right, the working principle of the spiking neuron in a network, with step function activation. Notably, convolutional, graph, and gated spiking networks can also be implemented.

Hardware Implementations: Deploying SNNs in specifically tailored hardware involves realizing chips using dedicated digital, analog, or mixed-signal circuits, with complementary metal-oxide semiconductor (CMOS) and emerging memory technologies such as memristive devices providing different realizations of stateful neuronal and synaptic computation. Photonic architectures are also used to implement the paradigm of reservoir computing: photonic reservoir computing exploits the ultra-fast dynamics and low propagation losses of optical systems to enable high-throughput and energy-efficient processing of temporal signals.

Recently, neuromorphic SNN hardware has attracted substantial interest in both industry and academia, with prominent platforms including IBM TrueNorth, Intel Loihi, and the academic SpiNNaker architecture. The neuromorphic hardware revolution is also led by startups, such as Innatera, Neuronova, and SynSense. Studies conducted on an Intel Loihi processor show that SNNs running on dedicated hardware can improve the energy-delay-product (EDP) by up to *three orders of magnitude*, making them a promising candidate to break the traditional energy vs. latency tradeoff [10].

Energy- and Context-Aware Distributed Learning

To conclude our discussion, we now consider embedding energy considerations into the learning phase. To start with, we observe that energy efficiency in FL and DFL is often framed as a communication problem: how to compress, quantize, or schedule model updates so that less data crosses communication links [1], [11]. On edge devices running deep neural networks, however, this framing overlooks a second, often dominant cost. Local training can consume an amount of energy that is larger than that required to transmit the parameters, so the design objective is not only how to move model updates cheaply, but also how to avoid producing updates that were not worth being computed in the first place.

This second consideration becomes central when devices rely on intermittent, ambient energy, such as solar or RF energy harvesting. In these systems, every joule spent on a redundant local training operation is a joule unavailable for a future, more informative one. Scheduling participation on battery availability alone can therefore waste harvested energy on updates that contribute little to the global model. This motivates scheduling policies that are informed of both the energy state of a device and the operational context in which its update will be used.

One line of work addresses the energy dimension through *the timing of local computation*. Rather than training as soon as a device has the energy for it, devices can be organized into groups that take turns across a learning cycle, with training deferred until shortly before a group's transmission opportunity. Postponing computation in this way avoids training prematurely on stale global models and then idling until the transmission slot. This regularizes the number of active participants per round, smoothing the otherwise erratic dynamics of energy-harvesting systems. The tradeoff is a controlled amount of staleness that the scheduler must keep bounded.

Energy-awareness alone, however, does not tell the scheduler whether an update is worth computing. A device may be well-charged and well-timed yet *carry an update that adds little to the global model*, for instance because its local data distribution is already well-represented or because its model has not drifted significantly since the last round. Assessing this requires looking beyond battery level to the informational content of the prospective update.

A useful lens here is information freshness. The age of information (AoI) and its variants provide a principled method for quantifying how stale a remotely observed process has become and how much value a fresh observation carries, and version-based notions of age extend naturally to distributed learning by increasing only when an update would carry significantly new information. Guiding participant selection with this metric biases the system toward contributions that are fresh and substantive; the practical approach is that measuring divergence directly in parameter space is itself costly, so lightweight proxies are needed to keep the check affordable on constrained hardware. This can be done, for example, by comparing compact intermediate representations rather than full parameter vectors [12].

Taken together, these observations point to a broader design approach for sustainable distributed learning at the wireless edge: scheduling decisions should be made *jointly in the energy and information domains*. Pure energy-awareness conserves resources but risks spending them on updates of marginal value, whereas pure information-awareness identifies valuable updates but cannot guarantee they are feasible to produce. The two perspectives are complementary, and their benefits are most pronounced precisely where distributed learning is hardest: under severe data heterogeneity and scarce energy. A natural extension carries the same logic from *whether* a device should train to *how much* it should train, measuring the intensity of each contribution to the expected value of the update. Combined with inherently efficient model architectures, such energy-proportional training would close the loop between model-level and system-level sustainability.

CONCLUDING REMARKS

This paper analyzed the steps required to realize AI-native 6G intelligent functions under strict requirements in terms of security, latency, performance, and energy consumption. This necessitates going beyond isolated optimizations, adopting a unified, multi-dimensional design strategy. Such a strategy entails moving intelligence from centralized clouds to peer-to-peer decentralized architectures to overcome bottlenecks in latency, scalability, and cross-domain data governance. Moreover, decentralization alone is deemed insufficient if the underlying intelligence lacks trust. Thus, appropriate security means must be in place to protect decentralized training, provide algorithmic trustworthiness, and ensure data confidentiality. Building a dependable security infrastructure requires treating trust as a core design requirement rather than a post-deployment adjustment.

Moreover, to support automated decision-making across zero-touch control loops, explainability tools must evolve beyond human-readable summaries into actionable operational signals that quantify prediction uncertainty, detect concept drift, and validate alerts across peers.

At the same time, this distributed security stack must operate within strict energy budgets across the entire edge-to-access continuum. To achieve this, we propose a twofold approach: we suggest the adoption of low-power neuromorphic computing architectures and the use of *joint* energy- and age-of-information-aware training schedulers to ensure that security mechanisms remain computationally viable without draining excessive resources.

The future 6G security stack cannot be designed as decentralized AI complemented by separate security, explainability, and green-computing mechanisms. These dimensions interact and should jointly determine who participates in learning, whose updates are trusted, when models are adapted, whether their decisions are sufficiently reliable to trigger automated actions, and how much energy should be spent producing those decisions. Their joint design is therefore key to realizing autonomous, trustworthy, and sustainable next-generation networks.

REFERENCES

- [1] E. T. Martínez Beltrán, M. Quiles Pérez, P. M. Sánchez Sánchez, S. López Bernal, G. Bovet, M. Gil Pérez, G. Martínez Pérez, and A. Huertas Celdrán, “Decentralized Federated Learning: Fundamentals, State of the Art, Frameworks, Trends, and Challenges,” *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2983–3013, 2023.
- [2] N. Papernot, P. McDaniel, A. Sinha, and M. P. Wellman, “Sok: Security and privacy in machine learning,” in *2018 IEEE European Symp. on Security and Privacy (EuroS&P)*, pp. 399–414, IEEE, 2018.
- [3] Z. Wang, J. Ma, X. Wang, J. Hu, Z. Qin, and K. Ren, “Threats to training: A survey of poisoning attacks and defenses on machine learning systems,” *ACM Comput. Surv.*, vol. 55, no. 7, pp. 1–36, 2022.
- [4] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, “Byzantine-robust distributed learning: Towards optimal statistical rates,” in *Int. Conf. Mach. Learn. (ICML)*, pp. 5650–5659, PMLR, 2018.
- [5] J. Yang, A. A. Soltan, D. W. Eyre, Y. Yang, and D. A. Clifton, “An adversarial training framework for mitigating algorithmic biases in clinical machine learning,” *NPJ Digit. Med.*, vol. 6, no. 1, p. 55, 2023.
- [6] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, “Adversarial training for free!,” *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019.
- [7] C. Zhao, D. Wu, J. Huang, Y. Yuan, H.-T. Zhang, R. Peng, and Z. Shi, “Boosttree and boostforest for ensemble learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 7, pp. 8110–8126, 2022.
- [8] R. Xu, N. Baracaldo, and J. Joshi, “Privacy-preserving machine learning: Methods, challenges and directions,” *arXiv preprint arXiv:2108.04417*, 2021.
- [9] A. Fono, M. Singh, E. Araya, P. C. Petersen, H. Boche, and G. Kutyniok, “Mathematical foundations of spiking neural networks: Strengths, challenges, and computational paradigm potential [special issue on the mathematics of deep learning],” *IEEE Signal Process. Mag.*, vol. 43, no. 2, pp. 64–76, 2026.
- [10] B. Rueckauer, C. Bybee, R. Goettsche, Y. Singh, J. Mishra, and A. Wild, “NxTF: An API and Compiler for Deep Spiking Neural Networks on Intel Loihi,” *J. Emerg. Technol. Comput. Syst.*, vol. 18, Jan. 2022.
- [11] N. Jia, Z. Qu, B. Ye, Y. Wang, S. Hu, and S. Guo, “A comprehensive survey on communication-efficient federated learning in mobile edge environments,” *IEEE Commun. Surveys Tuts.*, vol. 27, no. 6, pp. 3710–3741, 2025.

- [12] E. Jeong, G. Perin, H. H. Yang, and N. Pappas, “Feature-based semantics-aware scheduling for energy-harvesting federated learning,” in *2026 IEEE Int. Conf. Mach. Learn. Commun. Netw. (ICMLCN)*, 2026.

BIOGRAPHIES

Giovanni Perin (giovanni.perin@unibs.it) received his Ph.D. degree in Information Engineering from the University of Padova, Italy. He is an Assistant Professor at the University of Brescia, Italy. His research focuses on distributed learning, optimization, and sustainable AI and networks. He is a Member of IEEE.

Michele Rossi (michele.rossi@unipd.it) received his Ph.D. degree in Information Engineering from the University of Ferrara, Italy. He is a Full Professor at the University of Padova, Italy. His research focuses on sustainable AI, wireless sensing, and next-generation networking. He is a Senior Member of IEEE.

Enrique Tomás Martínez Beltrán (enriquetomas@um.es) received his Ph.D. degree in Computer Science from the University of Murcia, Spain, where he is currently a Postdoctoral Researcher within the CyberDataLab. His research focuses on collaborative learning, large language models, and cybersecurity.

Fernando Torres-Vega (torresvegaf@um.es) is a B.Sc. student in Cybersecurity at the International University of La Rioja, Spain. He is currently a software developer at the University of Murcia, Spain. His research interests include distributed systems, decentralized architectures, and cybersecurity.

José María Jorquera Valero (josemaria.jorquera@um.es) received the M.Sc. and Ph.D. degrees in Computer Science from the University of Murcia. He is currently a Postdoctoral Researcher with the CyberDataLab, University of Murcia. His scientific research interests include trust management, cybersecurity, 5G networks, intent-based management, and continuous authentication.

Manuel Gil Pérez (mgilperez@um.es) received the M.Sc. and Ph.D. degrees in Computer Science from the University of Murcia. He is an Associate Professor with the Department of Information and Communication Engineering, University of Murcia, Spain. His scientific activity focuses mainly on cybersecurity, trust and reputation management, and security operations.

Eunjeong Jeong (eunjeong.jeong@liu.se) received her Ph.D. degree in Communication Systems from EURECOM, France. She is currently a Postdoctoral Researcher at Linköping University, Sweden. Her research focuses on federated learning and energy-efficient scheduling. She is a Member of IEEE.

Nikolaos Pappas (nikolaos.pappas@liu.se) received his Ph.D. degree in Computer Science from the University of Crete, Greece. He is an Associate Professor at Linköping University, Sweden. His research interests include semantic communications, age of information, network-level cooperative networks, and performance analysis and stochastic modeling. He is a Senior Member of IEEE.

Farah Abed Zadeh (farah.abedzadeh@ucdconnect.ie) received her M.Sc. degree in Data and Computational Science from University College Dublin, Ireland, where she is currently a Ph.D. student in Computer Science. Her research focuses on explainable AI in next-generation networks.

Chamara Sandeepa (abeysinghe.sandeepa@ucd.ie) is a Postdoctoral Research Fellow at University College Dublin, Ireland. His research interests include AI security, explainable AI, federated learning, and intelligent security solutions for next-generation networks.

Bartłomiej Siniarski (bartlomiej.siniarski@ucd.ie) received his Ph.D. degree from University College Dublin, Ireland. He is a Researcher and Co-Director of the Network Softwarization and Security Labs (NetsLab), and is involved in several EU-funded research projects. His research interests include 5G/6G networks, IoT, network security, and network management.

Madhusanka Liyanage (madhusanka@ucd.ie) is a Professor and Ad Astra Fellow at the School of Computer Science, University College Dublin, Ireland, and Director of the Network Softwarization and Security Labs (NetsLab). His research interests include 5G/6G security, artificial intelligence, explainable AI, federated learning, blockchain, and edge computing. He is a Senior Member of IEEE.

Betül Güvenç Paltun (betul.guvenpaltun@ericsson.com) is a Senior Researcher at Ericsson Research, Turkey. Her research focuses on trustworthy AI and intrusion detection systems.

Leyli Karaçay (leyli.karacay@ericsson.com) received her Ph.D. degree in Computer Science and Engineering from the University of Sabanci, Turkey. She is a Senior Security Researcher at Ericsson Research, Turkey. Her research focuses on distributed learning, network security, and trustworthy AI.

Ioannis Pitsiorlas (ioannis.pitsiorlas@eurecom.fr) received his M.Sc. degree in Data Science and Engineering from EURECOM, France, where he is currently a Ph.D. candidate. His research focuses on trustworthy and explainable AI.

Marios Kountouris (marios.kountouris@eurecom.fr) received the Ph.D. degree in Electrical Engineering from Télécom Paris, France. He is a Full Professor at EURECOM, France, and a Distinguished Researcher at the University of Granada, Spain. His research focuses on communications theory and machine learning for communications. He is a Fellow of IEEE.