Received 27 August 2021; accepted 26 September 2021. Date of publication 1 October 2021; date of current version 13 October 2021.

Digital Object Identifier 10.1109/OJCOMS.2021.3116437

Enabling and Leveraging AI in the Intelligent Edge: A Review of Current Trends and Future Directions

TOM GOETHALS, BRUNO VOLCKAERT (Member, IEEE), AND FILIP DE TURCK (Fellow, IEEE)

IDLab, Department of Information Technology, Ghent University—imec, 9052 Ghent, Belgium

CORRESPONDING AUTHOR: T. GOETHALS (e-mail: togoetha.goethals@ugent.be)

This work was supported by Vlaio by means of the FlexNet Research Project.

ABSTRACT The use of AI on Smart applications and in the organization of the network edge presents a rapidly advancing research field, with a great variety of challenges and opportunities. This article aims to provide a holistic review of studies from 2019 to 2021 related to the Intelligent Edge, a concept comprising both the use of AI to organize edge networks (Edge Intelligence) and Smart applications in the edge. An introduction is given to the technologies required to understand the state of the art of AI in edge networks, and a taxonomy is provided with “Enabling Technology” for Edge Intelligence, “Organization” of the edge using AI, and AI “Applications” in the edge as its main topics. Research trend data from 2015 to 2020 is presented for various subdivisions of these topics, showing both absolute and relative research interest in each subtopic. The “Organization” aspect, being the main focus of this article, has a more fine-grained subdivision, explaining all contributing factors in detail. The trends indicate an exponential increase in research interest in nearly all subtopics, but significant differences between them. For each subdivision of the taxonomy a number of selected studies from 2019 to 2021 are gathered to form a high-level illustration of the state of the art of Edge Intelligence. From these selected studies and the trend data, a number of short-term challenges and high-level visions for Edge Intelligence are formulated, providing a basis for future work.

INDEX TERMS Fog computing, fog networks, edge networks, edge computing, artificial intelligence, review, trends.

I. INTRODUCTION

IN RECENT years, many of the computational workloads previously associated with the cloud have moved into fog networks, or even to the network edge [1], where they are run as distributed or decentralized tasks. This migration is a necessary step in the emergence of various “Smart” application domains, and eventually Smart Cities, in which Artificial Intelligence (AI) is to be deployed exactly where and when it is required.

There are several reasons for the computational migration to fog and edge networks. For example, to run software services closer to end-users in order to reduce latency, or to pre-process data instead of gathering all data to the cloud, thereby avoiding bandwidth issues or undue pressure on cloud resources. Additionally, the number of devices in fog and edge networks increases at an accelerated pace, while the hardware resources of the average device keep increasing. As such, there is ever more task offloading capacity available in the fog and in the edge.

However, there are also disadvantages to offloading tasks to geographically widespread fog and edge networks. In cloud data centers, hardware resources and network technologies are homogeneous, and properly managed using planned upgrades, typically resulting in high availability of services and systems. In the edge, networks are heterogeneous and unpredictable, and hardware resources and capabilities are extremely varied. As the scale of edge networks and the variety of devices they comprise increases, these factors make it increasingly more difficult to manage software services and organize traffic flows. For example, gathering all the required data for service orchestration in the cloud becomes infeasible due to network bandwidth saturation and memory requirements.

Additionally, tasks such as data placement and service migration are more difficult to orchestrate in edge networks than they are in the cloud. In cloud data centers, or a limited number of fog data centers, the target nodes for service deployments can be optimally calculated, and migrations can be executed quickly over high bandwidth connections. The network edge, however, is a volatile environment with a continually changing topology. In such an environment, calculating the optimal nodes to deploy data or services on is nearly impossible, and limited network bandwidth reduces the potential for service migration.

Finally, there are also various security risks that present themselves when running software services in the fog or edge. As opposed to the strictly controlled environment of a cloud data center, edge networks are largely comprised of unknown devices in networks with unknown, and often insufficient security measures. Such environments make it difficult to detect issues such as unauthorized access, data loss, privacy infringement and malicious injections of data or code, and nearly impossible to avoid them.

AI can solve many of these issues. For example, some classes of AI algorithms can learn from data gathered in the cloud and from the edge in order to recognize network intrusions, route traffic around faulty nodes, or quickly determine suitable nodes to deploy software and data on. However, AI algorithms can be resource intensive, and edge devices are often resource constrained and low-powered. Until recently, most edge devices were incapable of running any containerized services or advanced AI applications. Advances in both software and hardware, specifically related to Artificial Neural Networks (ANN), have commoditized AI in the edge. Although many devices have different priorities, e.g., extremely low-power sensors, all data is usually gathered at local gateway devices in the edge, or edge servers, which have the appropriate hardware resources to run complex AI algorithms. These advances have enabled AI to play an increasingly important role in properly organizing the network edge, orchestrating software services in the edge, and in software services themselves, which use AI to optimize end-user experience. Edge Intelligence (EI) [2] arises from any use of AI to enhance the organization or operation of software services in the edge, while the whole EI and the AI-powered end-user applications it enables results in the Intelligent Edge.

In this article, several important types of AI for the edge are explained, along with the concept of edge computing itself. The synthesis of edge computing and AI is discussed to show how the Intelligent Edge emerges from it, and what the general areas of end-user AI-powered applications in Smart Cities are. After the introductions to these topics, a taxonomy of EI is presented, and the state of the art of each category is discussed by charting research trends, and by presenting a selection of recent studies. The main topics are **enabling technologies** for AI in the edge, AI approaches to **organize** various aspects of edge networks, and finally AI-assisted **applications** running in edge networks. From the presented studies, future challenges for each topic are drawn, along with some long-term visions for the use of AI in the edge.

TABLE 1. **Abbreviations and acronyms used in the text.**

AI	Artificial Intelligence
AIoT	Artificial Intelligence of Things
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
DNN	Deep Neural Network
DRL	Deep Reinforcement Learning
EI	Edge Intelligence
FL	Federated Learning
FPGA	Field Programmable Grid Array
GRU	Gated Recurrent Units
IIoT	Industrial Internet of Things
IoMT	Internet of Medical Things
IoV	Internet of Vehicles
LSTM	Long Short Term Storage
MDP	Markov Decision Process
MLP	Multi-Layer Perceptron
NFV	Network Function Virtualization
NPU	Neural Processing Unit
PSO	Particle Swarm Optimization
PU	Processing Unit
RL	Reinforcement Learning
RNN	Recurrent Neural Network
SDN	Software Defined Networking
SI	Swarm Intelligence
VANet	Vehicular Ad-hoc Network

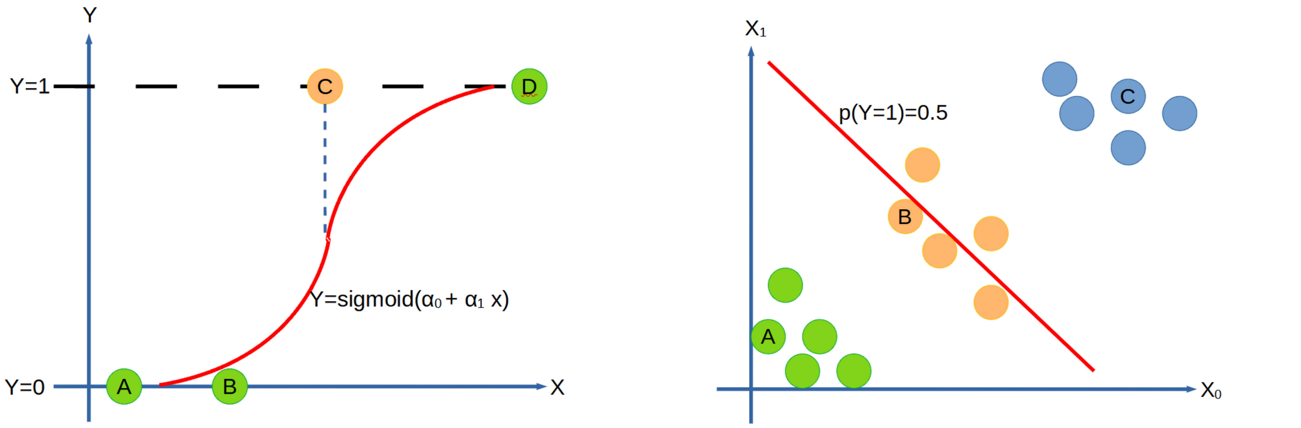
In short, the contributions of this article are:

- A high-level introduction to the topics required to understand the state of the art of the Intelligent Edge.
- A holistic taxonomy of AI in the Intelligent Edge, including enabling technologies, organization of the network edge, and applications.
- Discussion of research trends and recent advances in every aspect of the Intelligent Edge, based on the taxonomy.
- Suggestions for short-term and long-term research directions, compiled from trends and selected studies.

The rest of this article is organized as follows. Section II explains the different types of AI touched upon in this article, while Section III describes the emergence of fog and edge computing, and the potential of combining AI and the edge. Section IV explains the motivation for this review, and lists related work for the main topics. In Section V, the taxonomy of the review is elaborated, and a number of recent studies for each topic are examined in Sections VI, VII and VIII. Finally, future challenges and vision papers are presented in Section IX and conclusions are drawn in Section X. Table 1 lists recurrent abbreviations and acronyms used in the text.

II. COMMON TYPES OF AI IN THE EDGE

In this section, the most common types of AI mentioned in this article are introduced. Although sufficient explanation is given for the purposes of this review, the goal is only to introduce each of the topics, with more comprehensive works included as references. There are numerous studies and books that explain the general principles of AI, for example Hunt [3] or Brewka [4].



(a) Logistic regression for one input dimension (X). The points A, B and D can be reliably classified, but the model will give unreliable output for C, which has $p(Y = 1)_C = 0.5$ and may belong to either class.

(b) Logistic regression for two input dimensions. The red line describes the "border" between the output classes, or $p(Y = 1) = 0.5$. The points A and C can be reliably classified, but again the model will give unreliable output for points such as B, which have high probability of belonging to either class.

FIGURE 1. Classification in 1 and 2 dimensions using logistic regression.

A. STATISTICAL

Statistical approaches can be used to solve (binary) classification problems. The most popular algorithm of this type is logistic regression [5], a special case of binary regression which results in binary classifiers that output probabilities rather than a hard classification. This algorithm uses supervised learning [6], a method which "trains" a model on an initial data set containing expert-labeled outputs for known sets of inputs. After training, the statistical patterns in the data learned by the model are used to predict the probability of new inputs belonging to either class.

Assume that for an input with values x_i the output Y is required, with $Y = 0$ meaning that the input belongs to class A and for $Y = 1$ it belongs to class B. In logistic regression, the log-odds of an input is calculated using a linear combination of its values:

$$\log \frac{p}{1-p} = \alpha_0 + \sum \alpha_i x_i \tag{1}$$

with α_0 and α_i being learned parameters. To recover the probability p from this, a Sigmoid function is applied to the right side of the equation, giving the probability that the output Y belongs to class B, or $p(Y = 1)$. Generally, $p < 0.5$ means the input is likely to belong to class A, and for $p > 0.5$ to class B.

During training, the difference between $p(Y = 1)$ and the expert-labeled output is used to adjust the parameters α, improving the model. This is achieved through gradient descent [7], which calculates the impact of each input value x_i on the final output, and adjusts its parameter α_i to better match the output that is required. Generally, a learning rate $l << 1$ is used to modify the weights only slightly for each input, to avoid undoing the effects of previous adjustments. The model resulting from this training process is visualized in Fig. 1. Fig. 1(a) shows how the model discriminates between points in one dimension. The red curve represents a model trained on the input dimension (X), giving the probability of a point belonging to $Y = 1$. The points A and D are classified with absolute certainty, and B almost certainly belongs to $Y = 0$, as $p(Y = 1)_B$ is only 0.05. However, the model has difficulty classifying points such as C, which may belong to either class as $p(Y = 1)_C$ is 0.5. Similarly, Fig. 1(b) shows the classification of points with two input dimensions. In this figure, the red line represents the "border" of the two classes as determined by the model, or $p(Y = 1) = 0.5$. Points A and C can be classified with high certainty, but again there are points such as B which could belong to either class. Such data points may exist for any trained model; no training set can contain all possible data points as that would defeat the purpose of training a model.

However, this approach means that the accuracy of the final model generally increases with the amount of training data, as long as the inputs and outputs are properly distributed over all possible values. A downside of logistic regression is that the algorithm can get stuck in a local optimum or oscillate between several local optima, depending on initial parameters and the available training data.

Although logistic regression can be applied to any number of inputs, it can only discriminate between two output classes, limiting its usefulness. However, it is often used as a base model in more complex systems.

Some problems can be modeled as a Markov Decision Process (MDP) [8], a discrete-time stochastic process model. Such a process defines a state space S, an action space A, and a probability function $P_a(s_\alpha, s_\beta, t)$ which describes the likelihood of transitioning from state s_α to state s_β through action a at timestep t. A reward function $R_a(s_\alpha, s_\beta, t)$ provides the relevant reward for any state transition. Using the reward function, Reinforcement Learning (RL) [9], which is further explained in Section II-C, can be used to learn the optimal action policy for an MDP. This policy, once

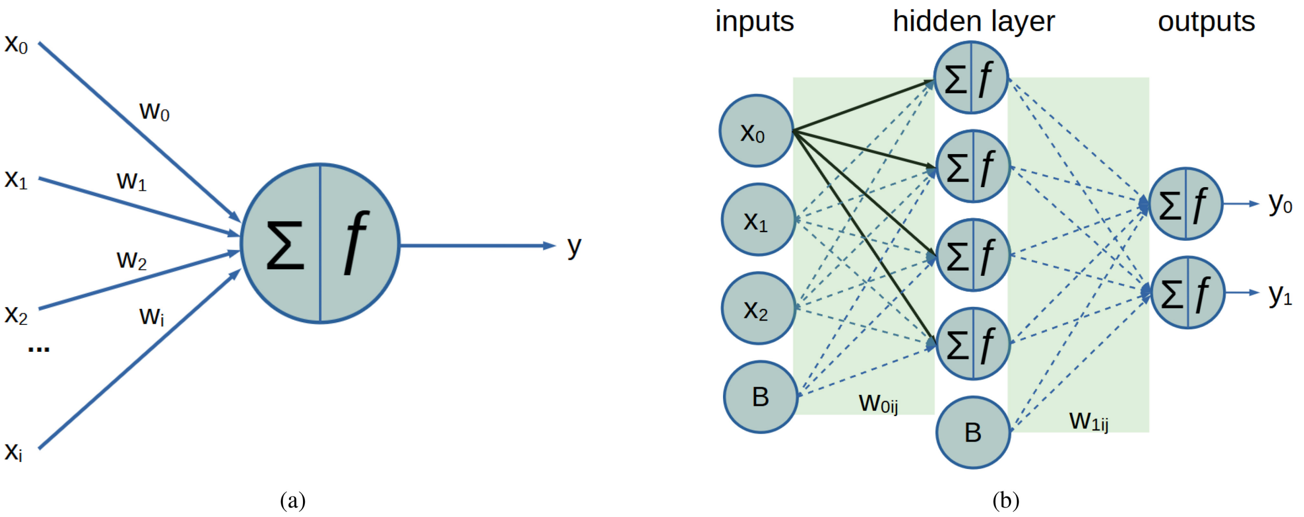


FIGURE 2. Visual representations of (a) a single neuron (perceptron) and (b) layered neural network (Multi-Layer Perceptron, MLP). Neurons generally output the result of a function *f* on the weighted w_i sum Σ of their inputs. Neurons without inputs represent input values x_i or bias neurons *B*.

learned, decides which action to take in any state, reducing $P_a(s_\alpha, s_\beta, t)$ to a straightforward probabilistic state transition $P(s_\alpha, s_\beta)$.

B. EVOLUTIONARY

Evolutionary or genetic algorithms [10] are modeled after the process of evolution in biological organisms, and can be applied to a wide range of problems. Technically, they can solve any problem that can be represented using a fitness function [11], whose minimum value over a search space should be minimized. This basic property makes them well-suited for scheduling problems and organizational problems.

An evolutionary algorithm starts by randomly generating n genomes (potential solutions), each of which contains all the values necessary to construct a concrete solution to the problem at hand. Each genome can be evaluated by the fitness function, thus ranking them by effectiveness. The algorithm then runs for a predetermined number of epochs (iterations), with two actions being performed in each epoch. First, n new genomes are generated by combining the values of parent genomes from the previous epoch, taking into account restrictions on the search problem. The chance of a genome being selected is proportional to its fitness value. In the second step, the new genomes are mutated by randomly changing values in order to introduce randomness in the search process. After the last epoch, the genome with the best fitness value is selected as the solution.

This type of algorithm relies on examining many potential solutions simultaneously and introducing randomness in the search process to cover as much of the search space as possible, while using unsupervised learning [12] in the form of a fitness function to guide the process in the direction of optimal solutions. However, the solution is not guaranteed to be optimal, and the algorithm may need to run for an undetermined amount of time before arriving at an acceptable solution, while the end result may not be explainable through math or logic.

Multi-objective optimization algorithms [13] such as MOGA [14] and NSGA-II [15] are a popular subset of evolutionary algorithms in the fog and edge. These algorithms find Pareto optimal solutions [16] for multiple optimization parameters by encoding data points, parameters and restrictions in genomes. As an example, consider finding the optimal computational nodes to deploy a number of software services on, depending on end-user latency and available resources. A multi-objective optimization algorithm will integrate the relevant properties of services and nodes into the genome, and both latency and available resources will be combined in a fitness function resembling Pareto search. The output is a genome that encodes the optimal node for each service to deploy.

C. ARTIFICIAL NEURAL NETWORKS

Like evolutionary algorithms, ANNs [9] are biologically inspired, simulating computation as it occurs in biological brains. The base element of an ANN is the neuron, which in its most basic form is described by Eq. (2). It accepts a number of inputs x_i, weighted by factors w_i. The sum of these values is used as input for the activation function f, the result of which is the output y of the neuron:

$$y = f\left(\sum x_i * w_i\right) \tag{2}$$

This equation is visualized in Fig. 2(a). Geometrically, a neuron represents a hyperplane dividing an i-dimensional space, which can be interpreted as performing binary classification of the points in the space. There are many variations on the basic neuron, with binary or floating point input/output values, a great variety of activation functions [17], [18], and the option of adding a "bias" value; a static input that is always active. For example, a single neuron can perform logistic regression by choosing a Sigmoid activation function and including a bias with weight α_0, resulting in 1.

ANNs learn patterns in a data set by using a backpropagation algorithm (gradient descent) [9] to modify their weights, similar to how parameters are updated in logistic regression. This can be done with either supervised (pre-labeled data) or unsupervised learning (loss function, automated feedback). However, backpropagation can be computationally intensive depending on the choice of activation function, as the complexity of the loss function involved in gradient descent depends largely on the activation function used.

In the case of supervised learning [6], a training data set is used with inputs and expert-labeled outputs. The output of the ANN for a given input is compared to the expert-labeled output, and the difference is used to update the weights. This type of learning is usually reserved for classification.

Unsupervised learning or Reinforcement Learning [12] is used for tasks where labeling outputs is infeasible, either due to the volume of data involved or because the correct output is not known. In these cases, a reward function is constructed which returns higher values for "more correct" outputs, and a modified backpropagation algorithm is used.

Although a single neuron (or perceptron) can emulate basic algorithms such as logistic regression, multiple neurons can be combined into neural networks (or Multi-Layer Perceptron, MLP) to solve a wide range of problems. An example of a basic neural network is shown in Fig. 2(b), with neurons organized into several layers, each processing the outputs of the previous layers using a weights tensor w_{kij}, between neuron i in layer k and neuron j in layer $k+1$. In this figure, the middle layer is a "hidden layer", only used for computation rather than writing input values x_i or reading output values y_i. The concept of bias neurons B is also illustrated here. Constructing neural networks in layers allows each layer to process progressively more complex features in the input data, with the final layer being able to classify intricate, abstract shapes or patterns. Note that the example model is fully connected, with each neuron in any layer being connected to each neuron in the previous one, but in practical applications this is rarely the case.

Stacking layers of neurons results in a more complex, recursive learning process. Neural networks and backpropagation require a lot of parameters to work correctly, such as initial input weights and learning rates. Sub-optimal choices often result in the failure to train a network, and as such many studies have focused on choosing correct initialization values for these parameters, and if and how they should be modified throughout the training process [19]. Regularization and specialized activation functions are also used to reduce model size and improve the learning process [20], [21].

The goal of gradient descent can be interpreted geometrically as finding the lowest point in the hyperplane formed by the loss function used during backpropagation. By itself, a static learning rate results in only a minor improvement in a specific direction for each training input, which may not be entirely in line with the true gradient of the hyperplane. Alternatives include using decaying momentum [22] to guide the backpropagation algorithm into a general direction over multiple training inputs. Other methods, using second-order derivatives of the loss function, are more computationally intensive and not always applicable, but produce excellent results with less training data [23]. Finally, training samples are often processed in batches to optimize both performance and training results.

In the last decade, hardware acceleration and architectural improvements [24] have made it possible to create and train neural networks with dozens or even hundreds of layers, now known as Deep Neural Networks (DNN) [9]. Combined with other advances, this has led to many specialized, highly efficient innovations. For example, Convolutional Neural Networks (CNN) [25] contain layers that have a similar function to image kernels, and are currently the most effective classification networks for visual input. Recurrent Neural Networks (RNN) [26], in which certain layers feed their outputs back into their own inputs, can use memory strategies such as Long Short Term Storage (LSTM) [27] or Gated Recurrent Units (GRU) for natural language processing or translation, or other types of tasks with tokenized, unbounded input where the state depends on previous inputs. In terms of training, Deep Reinforcement Learning (DRL) [28] has enabled the unsupervised training of deep networks, and advanced RL approaches such as Q-learning [9] can take into account expected and future rewards, rather than immediate returns from a reward function.

D. DISTRIBUTED

Distributed and decentralized algorithms are designed to run on a large number of computational nodes simultaneously. Whereas distributed algorithms generally still have a centralized goal and combine their outputs, decentralized algorithms simply divide a problem into small, independent parts without the need to merge the outputs. Distribution and decentralization are applicable to a wide range of algorithms, although some types of problems are easier to partition, such as ANN training. Distributed algorithms have two important advantages compared to monolithic, centralized algorithms. First, the complexity of a problem can often be greatly reduced by splitting it up into smaller tasks. Second, a distributed algorithm is generally far more scalable, e.g., grid computing projects such as Boinc [29].

A popular distributed learning algorithm in the fog and edge is Federated Learning (FL) [30], in which the training of a neural network is split up into parts. This may be a straightforward division of labor, or it can be organized hierarchically. After each node finishes its part of the training, the resulting model updates are integrated into a centralized model, usually in the cloud. The main advantage of FL is that it can offload model training from the cloud to fog and edge devices depending on the computational capacity of each node. A further advantage is the reduction of network traffic by processing training data at the network edge, and that local processing of training data can avoid privacy issues related to sending data to the cloud. However, depending on the model involved, training may be unfeasible on resource-limited edge hardware, and a long-term disadvantage is that FL has to update a centralized model and distribute it to fog and edge nodes from the cloud. More advanced approaches try to eliminate the cloud part, and fully decentralize the weight updates through peer-to-peer updates. Hierarchical Federated Learning (HFL) solves some of the communications issues of vanilla FL by introducing a hierarchical structure into the process of consolidating weight updates, usually through a middle layer in which cluster heads perform intermediate model integration.

Swarm Intelligence (SI) [31] is a general class of distributed algorithms in which large numbers of independently functioning nodes or particles perform localized improvements, resulting in a globally optimal solution. The logic of this approach is that an improvement for any node is also an overall improvement, and the optimal solution is simply that in which each node can find its own optimal state. Although this can result in acceptable solutions, the lack of global coordination does not often result in a theoretically optimal solution. SI is usually applied to problems that are easy to handle for a single node, but which are intractable on a larger scale. Particle Swarm Optimization (PSO) is a subclass of SI, but it usually simulates all particles and generally does not run as a distributed algorithm. PSO finds optimal solutions in a search space by simulating the movement of large numbers of particles, gravitating them towards each other as they find optimal states in the search space.

E. BLOCKCHAIN

Blockchains are relatively new, originally introduced as the technology behind various digital currencies, but increasingly popular in research for their potential as secure, distributed storage. While a blockchain is not an AI concept in itself, it is interesting to introduce it here because of its popularity in AI-related studies. Although variations exist, blockchains in general have interesting properties, but also significant challenges for their widespread adoption [32].

Generally, blockchains are transaction-based, and they operate through a number of decentralized, non-hierarchical nodes known as miners. Each node in the network has a copy of the blockchain, a collection of "blocks", each of which in turn contains a number of transactions. Whenever a node in the blockchain network creates a transaction, it is spread throughout the network on a peer-to-peer basis, and processed by the miners into a new block at the end of the chain. For the blockchain to be reliable, all miners must reach a consensus on the transactions processed, and the order blocks are processed in. However, this process is quite computationally intensive, and a (financial) reward for miners is usually attached in the form of digital currency, either through the process of mining itself or by demanding a transaction cost. This digital currency is tracked by the blockchain itself, avoiding fraud and contested transactions.

The most popular alternative currently used in studies is Ethereum [33], an open source blockchain using Ether as currency. Ethereum implements smart contracts, allowing code to be embedded into transactions, and enabling its execution whenever the requirements are met. Due to the nature of the blockchain, all parties agree by definition on the contents and execution of the smart contracts.

Although the decentralization of blockchain solutions offers some intrinsic security and reliability, and smart contracts are a flexible and reliable approach to digital transactions, there are also some challenges to widespread adoption these technologies. Most importantly, the energy use of blockchain solutions is generally known to be excessively high, although various solutions have been presented to alleviate this issue, for example Proof-of-Stake consensus. However, the current state of the art still requires orders of magnitude more energy per transaction than classical systems [34]. Because of its distributed, peer-to-peer nature, it also takes far longer for transactions to be processed by a blockchain solution than by a classical, centralized system. Whereas a single node can process a transaction in just a few milliseconds, the need for a network-wide consensus can increase the total transaction processing time to minutes. Some blockchain implementations are susceptible to manipulation if any single party controls over 50% of the mining capacity, giving that party a monopoly on the consensus mechanism. This risk can be mitigated with both technical and practical measures. Finally, the distributed and open nature of the blockchain means that anyone can view its contents. Although they can not be changed, the plain readability of transactions presents severe privacy issues. As such, extra security measures will be needed for most concrete blockchain solutions, or off-chain storage solutions may be needed to augment the blockchain.

F. OTHER

AI is not limited to the types previously listed in this section. It can take many forms, especially when applied in a new environment such as edge computing. For the purposes of this review, any method or algorithm is considered a form of AI as long as the base problem is intractable, the algorithm runs in the fog or edge, and predictive outputs are generated based on any number of input dimensions. Note that this does not necessarily mean that the algorithm has learning capabilities.

III. EDGE COMPUTING

This section details how fog and edge computing arise from a growing number of devices and the demand of consumers for more functionality and better responsiveness. It also explains some of the base technologies that are used to deploy and manage software in the fog and edge, and factors that can enable the use of AI in the edge.

A. FOG AND EDGE

In the last decade, the Internet of Things (IoT) paradigm has become popular in any number of products, from basic sensors for home or process automation to intelligent

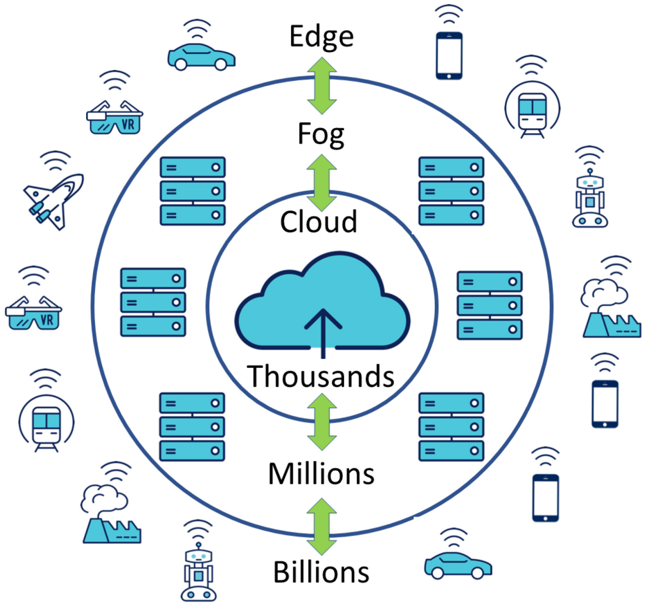


FIGURE 3. Representation of the difference of scale and device types in cloud, fog and edge networks.

appliances such as smart light bulbs and washing machines with self-adjusting programs.

These devices regularly send operational data and metadata to data centers in the cloud for several reasons. For one, a centralized access point for software services makes it easier for consumers to control their devices from any physical location. Another reason is that gathering all this data allows manufacturers to further improve their devices and services. However, in the last few years the growth of network traffic and processing power required to support the increasing number of smart devices is too high for centralized data centers to keep pace with.

Fog computing [1] offers a solution to this problem by decentralizing data centers. Although cloud data centers are often geographically distributed, they usually service entire countries or large geographical regions. Fog data centers, on the other hand, only service small geographical areas such as (parts of) cities. As a result, they are more numerous by orders of magnitude, but also less powerful because each of them must process less data. This concept is illustrated in Fig. 3.

Edge computing is an additional solution to reduce the load on cloud data centers and end-user devices alike, enabled by ever-improving hardware, increasing energy efficiency, and the proliferation of powerful handheld devices and IoT gateways. The essence of edge computing is that much of the work that is normally performed in the cloud can be broken up into small tasks which are then performed in the network edge. The network edge, shown in Fig. 3, consists of billions of low-power, resource constrained devices which are nonetheless highly programmable. To alleviate the workload of the cloud, the spare capacity of these devices is leveraged to pre-process data and provide basic, highly responsive end-user services.

The combination of cloud, fog and edge has resulted in many studies into tiered service architectures, where intensive data processing, big data analysis and the learning phase of AI models generally take place in the cloud, and the fog and edge provide responsive services and run the inference stage of AI models. Tiered architectures can be achieved with offloading, which uses real-time monitoring and migration of micro-services to move workloads to the fog and edge whenever possible, and back to the cloud if necessary. The term "necessary" may involve a combination of many parameters, from end-user proximity to packet loss and battery life. However, the volatile and heterogeneous nature of the edge and fog are problematic for scalable, optimal software deployment.

B. SERVICE DEPLOYMENT

Both the fog and edge are composed of a wide range of network technologies and various types of devices, leading to volatile conditions for software deployment and communication. Edge computing also presents new security issues. For example, edge devices are often less secure than cloud servers, and traffic over public networks may be intercepted.

In order for a piece of software to work on a wide range of devices, various types of virtualization can be used. Virtual machines are commonly used in cloud data centers, but since they contain an entire operating system, they are bulky and slow to migrate. Container technology [35] on the other hand uses the host operating system for kernel functions, and enables flexible but lightweight software deployment on any device that can run the required parts of the Linux kernel. Combined with container engines that manage the containers on a device, and container orchestrators which distribute deployments and tasks among computational nodes, containers are an essential enabling technology for edge computing.

Another recent virtualization technology is unikernels [36], which are essentially virtual machines that avoid context switches by always running in kernel mode. They also minimize their memory use and image size by only including the kernel functions required by the software they are compiled for. Finally, there are hybrid technologies such as Kata containers, which combine lightweight virtual machines with containers for increased security, at the cost of some performance.

As mentioned in Section III-A, offloading often takes into account many factors related to device resources and user experience. This is true for service deployment in general, especially in the fog and edge. For example, Kubernetes [37] accepts any number of custom plug-ins for its scheduler, and many studies are directly concerned with measuring the effectiveness of combinations of specific parameters for AI models to optimally assign services to fog and edge nodes.

C. SOFTWARE DEFINED NETWORKS

In some cases computational nodes are located in different physical networks and may not be able to reach each other directly, but have to be able to work together as if they are in the same logical network. Software Defined Networking (SDN) [38] is a flexible solution to such situations. Using Network Function Virtualization (NFV), SDN can create a highly flexible logical network on top of any physical network infrastructure. Because the network is completely virtual, IP addresses can also be assigned to interfaces used by virtual machines and containers.

Note that while a Virtual Private Network (VPN) is an example of an SDN, the latter is usually more lightweight and does not require features such as traffic encryption by default. In a container network, the most basic function of an SDN is to create an overlay network using a single IP address pool, from which addresses are assigned to nodes and containers.

SDNs can be used both in the cloud and in fog networks, and many networking features can be implemented on top of them using AI. For example, because the entire network is software-managed, much of the information required for service discovery, DNS, intrusion detection and intelligent traffic routing is available by default. The difficulty in building such features is finding the right parameters to use, and running them as fault-tolerant, distributed services over many nodes.

D. INTEGRATING INTELLIGENCE

As the scale of fog and edge networks grows, they eventually contain so many computational nodes that classical, centralized algorithms cannot scale sufficiently to manage them. In networks containing millions of nodes, it is impossible to gather network information and changing node statuses in real-time to a single location, nor is it feasible for a single algorithm instance to orchestrate services, detect malicious traffic, and route traffic within an acceptable time frame. Even in applications where timing is not an issue, the scale of any problem combined with the computational complexity of any classical, cloud-based algorithm will quickly overwhelm the hardware resources of a single cloud node, or even a few cloud nodes. This problem of scalability can be solved by decentralizing such algorithms and deploying them in the edge, and by integrating AI into them. As discussed in Section II, some types of AI algorithms have a training phase, and as such they can determine the important parameters for a problem during the training phase and produce results quickly at inference. Furthermore, AI algorithms can be designed to either send data to the cloud for use in further training, or even to keep executing training rounds themselves using gathered data, and merging the resulting weight updates through federated training. In all cases, AI algorithms can keep improving their efficiency. Finally, neural networks are very computationally intensive, but using multiple, specialized layers they can discover complex, nonlinear relations between parameters that classical algorithms would not be programmed to take into account.

Apart from decentralizing cloud algorithms and imbuing them with AI, there are also cases where processing data locally is the most logical choice. Reasons for this may include minimizing end-user latency, providing functionality even when connections to the cloud fail, privacy issues with sending data to the cloud (e.g., GDPR), or various other legal requirements or user preferences. As such, it is unavoidable to increasingly use decentralized intelligent algorithms to manage any and all aspects of organization and orchestration in the edge. Combining this infrastructural intelligence with AI applications featuring direct user interaction results in the Intelligent Edge, opening up the way to concepts such as Smart Cities [39]. Smart Cities have various application areas where AI can be useful. There are general Smart City applications, as well as Smart Homes, Industry 4.0, Internet of Vehicles (IoV) and Smart Health Care. Each of these domains will be further explained in Section V-A.

E. STANDARDS

Several recent IEEE standards and active projects focus on various aspects of EI, or can be taken into account when creating EI solutions. For example, IEEE 1934-2018 [40] adopts the OpenFog architecture as a standard, providing a framework for distributed computing, control and networking functions in an IoT environment on which EI can be built. Sub-projects of P2805 aim to establish intelligent protocols for self-managing edge computing nodes [41] and cloud-edge collaboration for machine learning [42], while P2961 [43] is to provide a framework for distributed, collaborative machine learning in an edge-cloud environment. Finally, there are also projects explicitly aimed at Smart Cities applications, e.g., P2979 [44] which aims to provide a framework for intelligent cooperation of edge devices in various IoV use cases.

IV. MOTIVATION AND RELATED WORK

Although the use of AI in the edge is relatively new, a vast body of work is related to it directly or indirectly. Existing surveys and reviews often focus on an extremely narrow aspect of AI in the edge (e.g., specific enabling hardware, only deep learning applications), without providing a larger context and assuming advanced knowledge of the reader in all the discussed topics. While such works are undeniably useful, the continued expansion of the research field and the divergence of its constituent topics make it ever more difficult to form a high-level overview.

This work aims to provide a holistic overview of what constitutes, and is necessary for, the Intelligent Edge, and to provide a variety of useful, recent studies in this wide area of research. However, this article does not include an exhaustive list of studies for each topic discussed, preferring instead to provide a high-level summary of the state of the art. The topics in this article require a deep understanding of AI, cloud technology, and fog and edge networking. As such, all these concepts are first introduced to the required degree, and references are provided for further exploration. Furthermore, the concept of the Intelligent Edge, being based on two large and rapidly-changing fields of research, is itself volatile and

TABLE 2. Comparison of scope of general edge intelligence surveys.

Article	AI types	Focus	Additional features
Deng et al., 2020	Deep learning	Wireless networking	Taxonomy
Shi et al., 2020	Neural networks	Communication	/
Zhou et al., 2019	Deep learning	Enabling/integrating AI	EI rating
Zhang et al., 2020	General	IoT sensors	Taxonomy
Wang et al., 2020	Deep learning	General	Taxonomy
This work	General	Organization/applications	Taxonomy

constantly progressing. Therefore, periodical reviews can aid in the continued discovery of research in the field.

The rest of this section presents related work, starting with general reviews and surveys of AI in the edge and continuing with more specific areas of research, such as enabling technology and Smart City AI applications.

A. EDGE INTELLIGENCE

The work of Deng *et al.* [45] provides a taxonomy of AI in the edge which focuses mostly on AI for wireless networking, improving service placement using AI and enabling AI, specifically in the context of DNNs. Other aspects of AI in the edge, such as security and reliability, are only summarily explored in favor of a more in-depth technical explanation of the main topics.

A survey by Shi *et al.* [46] considers the communication efficiency of AI in the edge. The premise of the study is that AI algorithms on edge devices should sparingly use the limited bandwidth available. As such, they present studies ranging from the training of communication efficient models to optimizing communication between algorithms on different nodes during inference.

Zhou *et al.* [2] provide a broad overview of studies related to both the training and inference stages of deep learning for EI. In addition, they also provide a rating system for the amount of integration of intelligence in the edge, ranging from cloud-only AI to edge-only AI.

In their survey on Artificial Intelligence of Things, Zhang and Tao [47] present a detailed taxonomy on enabling, designing and using intelligence for edge IoT sensors. The article provides a wide range of relevant studies, mostly related to the main topic of learning methods and perception models for IoT.

Wang *et al.* [48] provide a taxonomy and works related to the various stages of enabling and using deep learning models in the edge, ranging from hardware innovations to actual inference on the edge and relevant applications.

A comparison of these works is found in Table 2.

B. ENABLING AI IN THE EDGE

Much effort has gone into enabling DNNs on edge hardware. CNNs in particular have very deep and computationally intensive architectures, but the operations involved are highly modular and repetitive, making them excellent candidates for acceleration through custom hardware. A survey by Véstias [49] focuses specifically on accelerating CNNs using reconfigurable computing hardware, while another from Véstias [50] focuses on hardware acceleration of deep learning in general.

In a more general study, Zou *et al.* [51] list various hardware technologies that enable or accelerate specific types of AI in the edge. Most of these are designed for CNNs or deep learning in general, but some are aimed at Support Vector Machines (SVM). For each technology, the envisioned machine learning tasks and energy efficiency are reported.

In a survey by Nazir *et al.* [52], a holistic pipeline model for the compression and distribution of deep learning tasks in the edge is presented. As an introduction, this study lists various types of neural networks commonly used in the edge, and links to studies with concrete applications of each type. For the main part, it provides selected studies for each of the stages of the presented pipeline: model compression, (hardware) acceleration and parallelization. Importantly, the authors discuss the various types of model, data, and architectural parallelism that can be exploited to run complex neural networks in the edge.

C. ORGANIZING THE EDGE THROUGH AI

The importance of AI for security in the edge is highlighted in a survey by Mohanta *et al.* [53]. In this study, they list potential attacks on IoT devices, and refer to studies showing how AI can be applied to prevent attacks (e.g., intrusion detection, malicious app code). Additionally, studies are cited that show how blockchain technology can be used to enable distributed intelligence and ensuring smart contracts.

The use of AI in vehicle-to-everything (V2X) networks is highlighted by Rihan *et al.* [54]. In their survey, they provide an overview of the potential of AI to both enable next-generation V2X networks, and the future applications utilizing those networks.

More applications of AI for edge networks are provided by Wang *et al.* [55]. In this article, studies are listed that use AI to enable or improve various aspects of 5G and Beyond-5G (B5G) networks. The authors argue that AI can be used to solve currently intractable problems in the design and optimization of wireless edge networks.

The importance of AI for reliable edge networks is covered by Gupta *et al.* [56], specifically arguing for the synergy of EI and next-generation 6G networks to enable advanced, low-latency, ultra-reliable applications (e.g., IoV, drones, holographic communication).

D. END-USER APPLICATIONS USING AI

In a general survey of the application domains of AI in the edge, Huh and Seo [57] provide a number of studies related to often-referenced domains such as Smart Homes, autonomous vehicles, Smart Factory and Smart City, but also cite studies related to the more general domains of cloud offloading, video content analysis and Mobile Edge Computing (MEC).

An overview of AI applications in the Smart City is provided in a survey by Ullah *et al.* [58]. This article considers AI in Intelligent Transport Systems (ITS), Smart power grids, and cyber-security of Smart City systems. Additionally, the topic of UAV-based communication in 5G and B5G networks is discussed.

In their Smart Grid review, Gilbert *et al.* [59] list various studies in three distinct categories: the current requirements and uses for smart grid applications, which smart grid applications benefit from edge computing, and the future challenges for smart grid applications in the edge.

A survey by Sepasgozar *et al.* [60] provides an overview of AI in Smart Homes and Energy Management Systems. The article presents a deep statistical analysis of the studies found, including co-author connectivity and lexical analysis. Selected papers for each domain vary greatly, but are discussed in detail.

The potential of AI in Internet of Medical Things (IoMT) based health care is illustrated in a survey by Greco *et al.* [61]. Examples of IoMT-specific devices are health monitoring wearables and field sensor networks, which can be organized in edge networks. Greco *et al.* provide studies that combine AI and IoT in a wide range of medical aspects, including physiological monitoring, rehabilitation, dietary assessment and epidemic diseases.

In their survey, Angelopoulos *et al.* [62] provide studies related to the use of AI in Industry 4.0 and Industrial Internet of Things (IIoT). The article provides a taxonomy for AI in Industry 4.0, listing studies for each category with a focus on the link between functionality and the type of AI algorithm used.

E. BLOCKCHAIN

A survey by Singh *et al.* [63] provides the required background knowledge on blockchains as distributed, public databases in the context of Smart Cities. A number of studies are provided that combine AI with blockchain for security aspects of Smart Cities, while discussing how blockchains can improve privacy and trust, and analyzing potential issues with blockchain solutions.

In their survey, Yang *et al.* [64] provide a complete roadmap to the integration of blockchain and edge computing, starting with the motivation for the integration of both technologies, and moving on to frameworks, potential functions of blockchain in the edge, and challenges to the widespread adoption of blockchain technology.

Mohanta *et al.* [53] list various studies that show how blockchain technology can be used to enable distributed intelligence and ensuring smart contracts.

A more specific survey by Wu *et al.* [65] considers the combination of blockchain and edge computing to improve the security and scalability of IIoT. This survey identifies potential issues with critical infrastructures in Industry 4.0, and argues for the convergence of blockchain and edge computing to tackle these issues, providing various supporting studies.

Finally, Nguyen *et al.* [66] discuss the potential of the blockchain combined with FL (FLchain) for edge computing, listing opportunities and challenges for various edge applications such as crowdsensing and edge content caching.

TABLE 3. Query keywords per taxonomy (sub)category.

(Sub)category	Keywords
Enabling Technology	GPU, NPU, FPGA, hardware acceleration, partitioning, inference, offloading
Orchestration	deployment, provisioning, scheduling, optimization, energy efficient
Scalability	scalability, decentralized, hierarchical, discovery, offloading
Security	security, anomaly detection, intrusion detection, adversarial, blockchain security
Reliability	reliability, resilience, fault tolerant
Network	networking, discovery, SDN, routing, 6G
Applications	smart city, smart home, industry, iiov, iov, vanet, iomt, health care

V. METHODOLOGY

This section describes the methodology used in constructing a taxonomy and discovering the relevant subcategories for each top-level category. Note that while many of the individual low-level aspects may be applicable to other forms of computing, including general cloud computing, the taxonomy concerns only how they affect AI-related edge computing specifically. This section also elaborates how queries are formed from taxonomy-related parameters to find relevant studies, and how studies are categorized based on current research trends.

A. TAXONOMY

Fig. 4 show the taxonomy used for this review. Although the main focus of the article is the "Organization" category, both "Enabling Technology" and "Applications" are useful to include because they are closely related to, and often mesh with, proposed frameworks and solutions to organize the Intelligent Edge.

The subcategories are discovered by performing searches on Google Scholar with various relevant keywords, and then grouping the results by recurring subjects. The keywords and subcategories are refined iteratively, until each subcategory contains at least 3 sample studies, but no more than 10, preferably with one recent, dedicated review or survey indicating further research. Table 3 shows the final list of keywords, which are also used to construct queries to find the individual studies and articles listed in Sections VI through VIII. Although each (sub)category is elaborated in those sections, a short introduction to each is given here to fully explain the taxonomy.

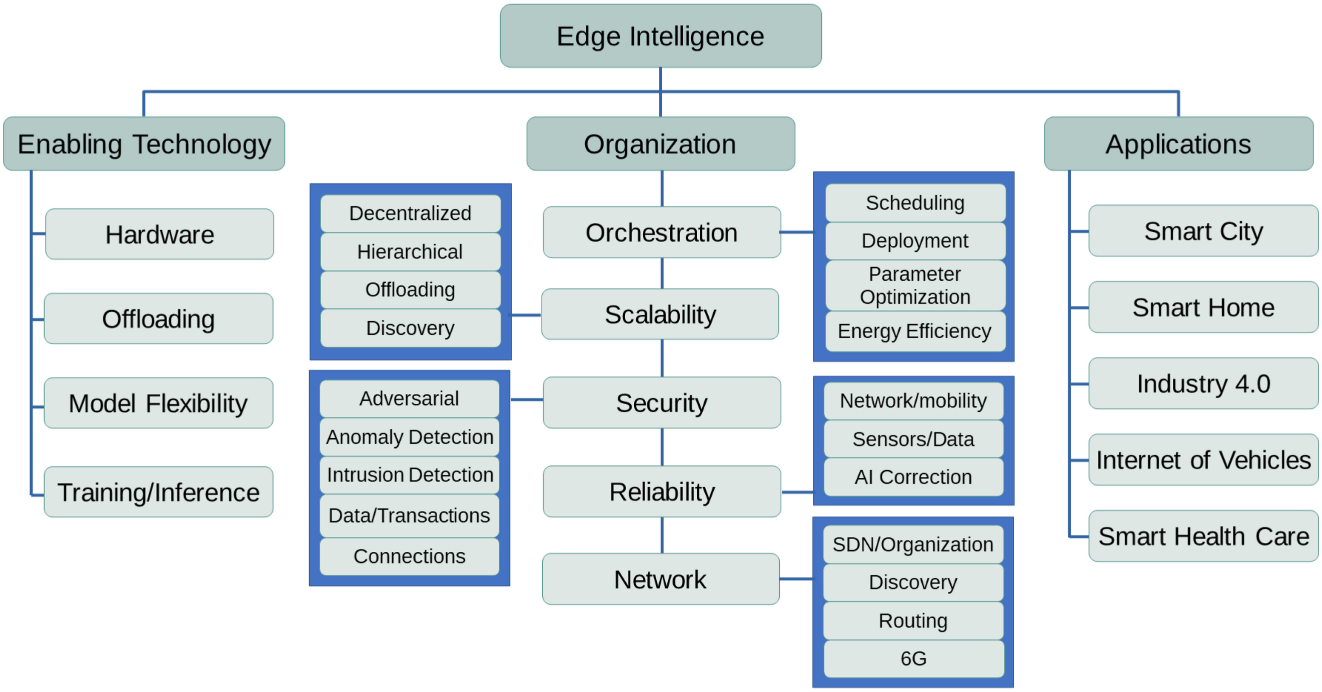


FIGURE 4. Taxonomy of the review.

1) ENABLING TECHNOLOGY

In the context of this article, enabling technology is defined as any hardware or software improvement that enables or improves the use of AI on an edge device. In other words, this category entails improvements to AI itself, rather than improvements in edge networks achieved through the application of AI.

There are four popular areas of research in this category.

- **Hardware** improvements or new specialized types of hardware generally increase the performance of AI, although they also indirectly result in new functionality and the ability to use more accurate AI models. Examples of this are the Neural Processing Unit (NPU) [67], which can be optimized for the type of repetitive calculation used in ANNs, and Field Programmable Grid Arrays (FPGA), which in their most basic version can be modified at the hardware level to quickly execute any algorithm without the need for software programming. Any edge device is capable of running neural networks without an NPU, but such a specialized processor can increase performance and practical model size by orders of magnitude without increasing the power requirements of a device.
- **Offloading** is not strictly an enabling factor of AI in the edge, but improves it nonetheless. Its original intent was to move certain well-delineated tasks from the cloud to the edge or vice versa. As such, a lot of research in offloading is related to being able to run AI in the edge in the first place.
- **Model flexibility** relates to different factors that allow the modification of AI models for low-resource edge devices. For example, model compression is used to reduce the size of a model, and to improve its performance, at the cost of a small loss in accuracy. Other approaches involve creating incrementally smaller but less accurate models for different classes of hardware, or using modular models, although the latter is closely related to offloading.
- **Training** and inference phases are often split up, as training a model is very computationally intensive and usually done in the cloud. Because all training data has to be gathered in the cloud, this approach is not scalable. Moreover, it enforces a single model for all devices, even if individual, localized learning may result in more accurate results. Existing techniques such as FL aim to solve this issue by integrating the changes learned by each device into a central model, but training in the edge is still affected by energy efficiency, scalability and processing power.

2) ORGANIZATION

This category entails studies that use AI to improve the infrastructure of the fog and edge. More specifically, this includes organizing software services, data and (software defined) networks, and ensuring their security and reliability. Note that many of the aspects of “Organization” are closely related. For example, a study may involve a novel method of combining service scheduling and aspects of SDN to improve service reliability or scalability.

There are a number of subcategories that studies can contribute to:

- **Orchestration** is the optimization of service scheduling and deployment, and the study of various parameters involved. Many studies focused on this involve optimizing QoS or end-user experience, balanced against a number of other factors such as energy efficiency, or minimization of resource use or network traffic. Effective algorithms for orchestration in the cloud exist, but in the fog and edge they are complicated by scale, heterogeneous hardware and the need for real-time adjustments due to mobile nodes. This problem can be further complicated by also taking data placement into account, although most studies focus on either data placement or service placement alone.

- **Scalability** focuses specifically on the problems imposed on service and network management by the geographical scale of the edge, and the sheer number of devices in it. Scalability can be achieved by decentralizing frameworks or algorithms, but also by organizing them hierarchically or through modularization. In the first case, self-organizing networks and service architectures can be designed, while in the others the cloud is usually employed as the highest, centralized level of the service architecture. Offloading, as discussed in "Enabling Technology", can be used to move parts of cloud workloads to the edge, and as such represents a limited form of scalability. Finally, automated discovery of nodes and (service) resources is an important step in effective and efficient self-organization on the scale of edge networks.
- **Security** of data and network traffic in the edge is complicated by the increased exposure compared to cloud data centers, and because of the scale of the edge. Research into adversarial attacks attempts to solve security issues with AI itself, in particular DNNs which can be "tricked" into incorrect classification. Anomaly detection and intrusion detection using AI are popular research topics in the security of edge networks, although there are other aspects in securing networks. Similarly, blockchain technologies are gaining a lot of attention for scalable and secure transaction systems, and where data in the edge is concerned, privacy is the most significant aspect of security. Finally, AI can also be used to secure connections through authentication or authorization.
- **Reliability** of software, networks and data (integrity) ensures the continued and seamless functioning of fog and edge services from an end-user viewpoint. Reliability of software includes seamlessly failing over to other service instances when any instance becomes unreachable, in addition to taking steps that services do not end up in invalid states to begin with. Network reliability involves finding new routes around unreachable nodes or subnetworks, and discovering and maintaining redundant routes. Both network and service reliability require real-time monitoring of nodes and services to enable AI optimization, and they are often used in combination to ensure QoS targets. For data, reliability means not only availability and redundancy, but also the integrity of the data itself. This is different from data security in that data may become unintentionally corrupted due to hardware or software errors. The latter case can also be caused by problems with AI systems, in which case redundancy and correction are needed.
- **Network** organization in the edge is usually done through SDNs, imposing a virtual, software-controlled layer of IP addresses and network functions (e.g., NFV) on top of the physical networks comprising the fog and edge. Apart from SDNs, many studies focus on network resource discovery and application traffic routing in the edge, either as NFV or as part of a holistic approach to edge networking. Finally, 6G networking has recently emerged as a research topic, aiming to integrate AI directly into various aspects of next-generation network management and operation.

3) APPLICATIONS

"Applications" in the Intelligent Edge differ from the topics listed in "Organization" in that they are AI applications that interact with end-users, running on top of the AI-organized edge. As such, these applications represent the end goal of creating the Intelligent Edge: intelligent applications running autonomously on AI-managed infrastructure, enabled by AI specific technology.

The Intelligent Edge applications discussed in this review are:

- **Smart City** is a collective term for all applications using AI in the context of cities. In this article, only applications that employ EI are considered, although many can also be realized through AI in the cloud, albeit at the cost of increased communication overhead and higher response times. There are a number of popular research topics in this area, such as inner-city traffic and parking management. Other topics include public health monitoring (e.g., fall detection), and security (e.g., surveillance of specific areas). Scaling Smart City applications to manage entire cities poses challenges in terms of service deployment, traffic routing, resource monitoring and real-time reaction to changes in service and network topology (e.g., movement of nodes, redistributing load).
- **Smart Home** applications aim to gradually improve all aspects of homes, from basic automation to fully AI-assisted living. Similar to Smart City applications, many recent Smart Home studies also focus on health monitoring and security, although there is less need for scaling and more focus on privacy and personalization. Scalability is also an important requirement, but only to be able to deploy the appropriate services to individual homes when required, rather than forming a collaborative service mesh across an entire city.
- **Industry 4.0** aims to improve various aspects of industry and manufacturing through AI. For example, blockchain and AI combinations can reliably log information, which can later be used to track down production chain issues related to faulty manufactured

items. Other technologies such as digital twins promise to optimize manufacturing processes by setting up virtual duplicates and searching for ideal settings and parameters, either for each step or holistically.

- **Internet of Vehicles** or IoV has a wide range of applications. Some studies involve the detection of traffic problems and proactively managing the flow of traffic around affected areas, often using dedicated roadside units as computational nodes. Others focus on inter-vehicle communications for optimized traffic flow, or other network-related aspects of autonomous vehicles. In almost all cases, IoV applications need to work with large numbers of fast-moving, unpredictable vehicles, combining the latest in extremely low-latency communication (e.g., 5G or 6G) with highly flexible network and service management.
- **Smart Health Care** aims to combine IoT and AI for various health related purposes, most importantly preventive health care and efficient, personalized patient monitoring. Applications include, but are not limited to, fall prediction, general elderly care, preventive and chronic health care through monitoring, and epidemic monitoring.

B. QUERY PARAMETERS

The results of this review include both numbers on recent research trends, and selected studies, both of which are gathered by querying Google Scholar. This source is chosen because it is a well-maintained meta-index, linking to studies found in various other indexes. For trends, the following base query is used:

(“*edge network*” *OR* “*edge computing*” *OR* “*fog computing*” *OR* “*fog network*”) *AND* “(*keyword*(*s*))” *AND* “*artificial intelligence*”

where (*keyword*) is replaced by the keywords from Table 3. Note that the keywords are not always directly mapped to taxonomy categories. Rather, they are considered relevant topics which may yield studies that can be mapped onto the taxonomy. The query is crafted to return almost no false positives, and as little false negatives as possible. However, many keywords are mentioned only in passing in loosely related studies, especially as the popularity of any subject increases, so the apparent interest in some topics will be inflated compared to the actual interest.

Historical trends are given from 2015 to 2020. Before 2015, most keywords yield either unreliable results (e.g., less than 5 studies, keywords not yet coined) or irrelevant results, and 2021 is excluded from the trends because extrapolating numbers from an incomplete year is unreliable. In all searches, the Google Scholar options “include patents” and “include citations” are disabled so that the results represent only studies in which the keywords were actually used in the text.

The trends are presented in Sections VI through VIII. Some keywords are inevitably more popular than others, either due to a focus of interest in their specific direction, or due to being often-quoted concepts in studies on EI. Because of this, the results will be presented in two forms; the absolute numbers to indicate the amount of research interest per keyword, and normalized numbers to determine the growth of research interest. In the latter case, the results are normalized to the amount of research interest in 2015 for each keyword. Finally, in the charts for relatively research interest, a “General” trend is added representing the average interest growth in EI.

TABLE 4. Query keywords for types of AI.

AI type	Keywords
Regression	regression
Genetic algorithm	evolutionary, genetic
Unsupervised learning	unsupervised
Supervised learning	supervised
Neural network	neural
Federated learning	federated
Distributed learning	distributed learning
Swarm intelligence	swarm

In addition to the popularity of AI in research topics, the popularity of the types of AI used in the edge, as discussed in Section II, is determined by compiling interest trends using the same methodology as for research topics. The keywords used to gather the data for these trends are shown in Table 4. The results are presented in Section VII.

Further requirements are introduced for the selection of referenced studies from the base query. The studies included in this review range only from 2019 to 2021, and their topics must explicitly relate to a novel application of AI in one or more aspects of edge computing as detailed in the taxonomy. They must also be effectively published in a peer-reviewed journal, barring a limited number of accepted studies from 2021 for which pre-print versions are used.

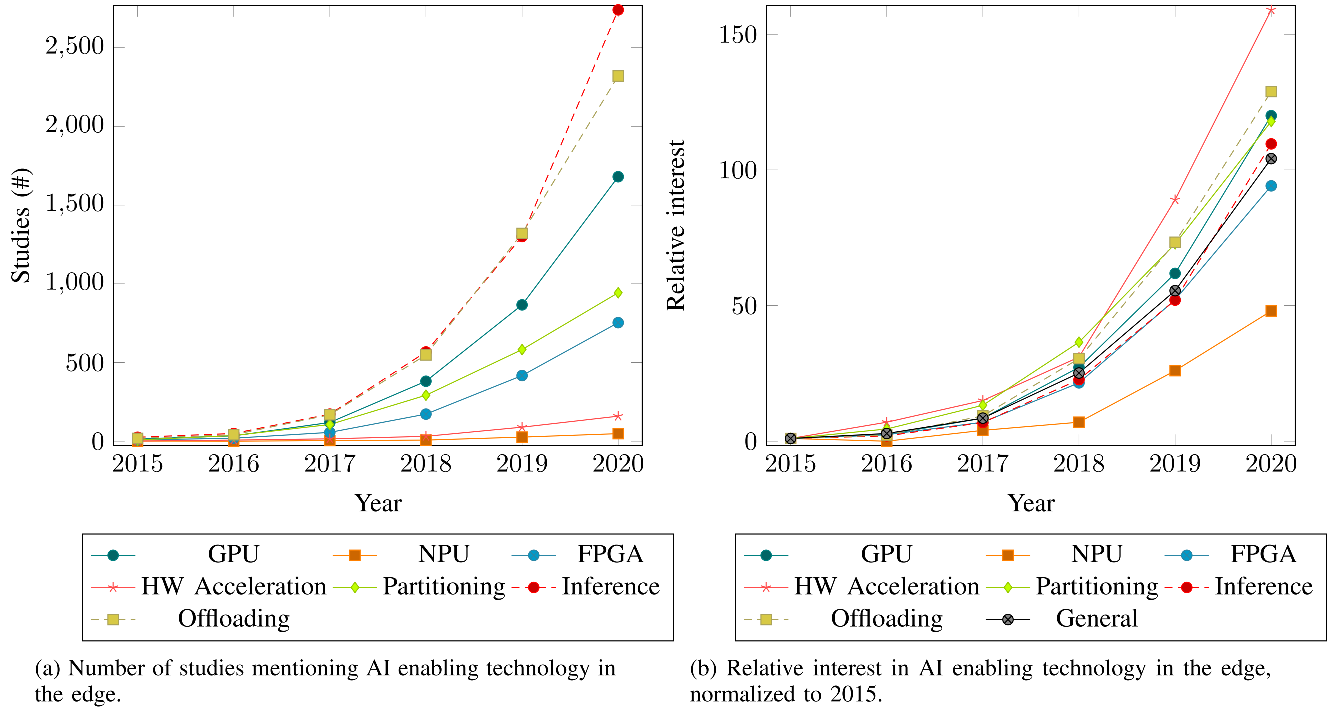
VI. ENABLING THE INTELLIGENT EDGE

This section discusses recent work related to the “Enabling Technology” category of the taxonomy presented in Section V-A, including novel hardware solutions, innovations in offloading, AI model flexibility and important progress in (distributed) training and inference algorithms for EI.

Although this article aims to cover all types of AI, (deep) neural networks are currently the most computationally intensive type of AI, and the least suitable to run on general purpose low-resource edge devices. As such, most of this section covers technologies to improve the inference stage of DNNs in the edge.

Fig. 5(a) shows the number of studies that mention keywords related to enabling AI in the edge since 2015. In absolute terms, the most popular topics are offloading and optimization of inference in the edge, followed closely by GPU acceleration. Considering relative interest, various hardware acceleration methods have gained a lot of interest since 2018, keeping pace with or outpacing interest growth for other keywords.

Various dedicated PUs aim to improve the performance of AI inference on low-powered edge devices. Commercial



(a) Number of studies mentioning AI enabling technology in the edge.

(b) Relative interest in AI enabling technology in the edge, normalized to 2015.

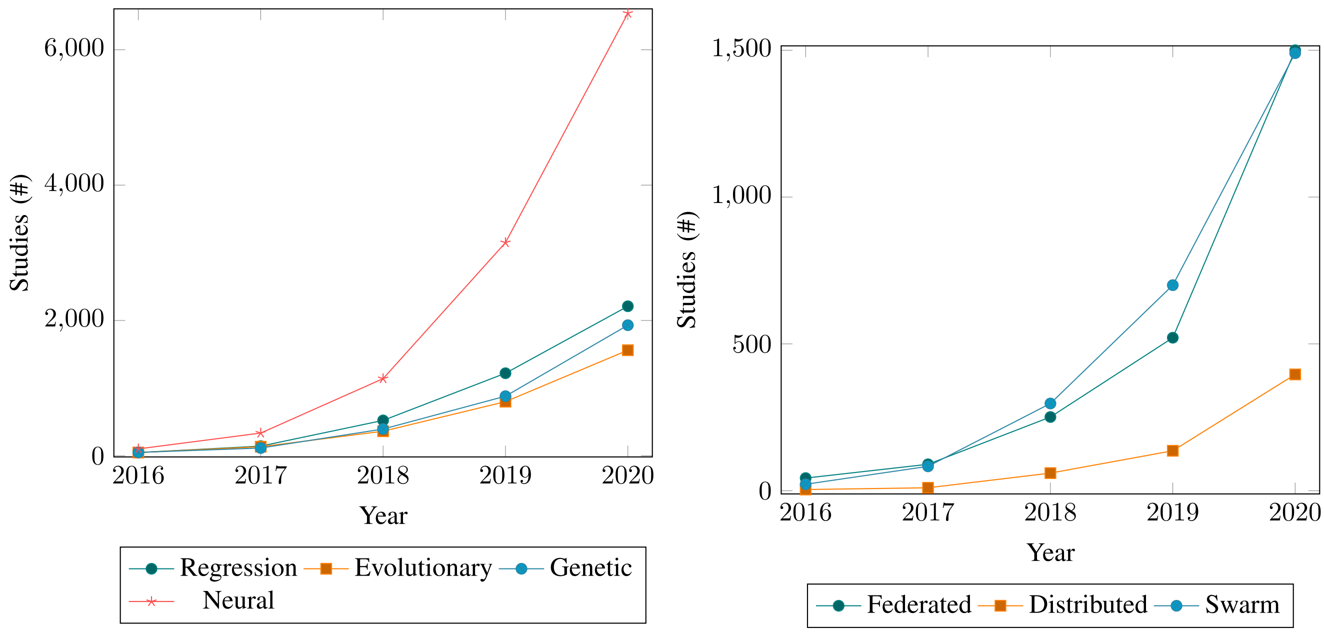
FIGURE 5. **Research interest in AI enabling technology in the edge, compared to general interest in edge AI.**

PUs include Google Edge TPU [68] and Nvidia Jetson Nano, both of which are designed for running DNNs in the edge. FPGAs are often used for the acceleration of repetitive but computationally intensive tasks. As an example, the use of an FPGA System-on-Chip (SoC) with OpenStack [69] allows the ARM CPU of the SoC to run a customized OpenStack worker and task planning, while the FPGA itself executes DNN inference. This particular solution uses Dynamic Partial Reconfiguration (DPR) to continually update the FPGA programming, enabling OpenStack to run a virtual machine on the FPGA. Memory is shared between the CPU and FPGA for performance reasons. This solution manages to run a YOLO implementation at 8fps using merely 6.57W of power, coming close to real-time video stream processing.

At the level of single devices, efficient management of different PUs can significantly improve AI performance. In particular, NeuroPipe [70] is aimed at improving the energy efficiency of DNN inference on edge devices by slicing each layer into chunks suited for the processing capacity of each available PU, and pipelining them independently. By parallelizing execution like this, NeuroPipe manages to reduce energy consumption by 11% compared to a normal inference run.

Moving up to the level of edge networks, efficiency and responsiveness can be improved by intelligent cooperation between devices. In an example of client-server cooperation, Edgent [71] aims to improve the performance of DNN inference on end-user devices by offloading to edge servers, while maintaining a high responsiveness through co-inference. During an offline stage, Edgent partitions a DNN using right-sizing to optimally divide the workload between devices, after which the partitions can be run on-demand on their respective target machines. The framework is optimized for communication efficiency to reduce the required traffic between edge device and server as they run their respective workloads. Another approach to this problem finds the optimal partitioning point in a DNN by considering latencies between devices and the amount of communication between each pair of layers [72]. The algorithm is evaluated using several CNN models, showing that its offloading results in better performance than local inference, given a sufficiently powerful edge server and at least 16Kbps of network traffic.

Instead of two-part co-inference, DNNs can also be divided into (sub)layer tasks. However, the distributed deployment of such tasks is an intractable scheduling problem (NP-hard). One possibility is to optimize task deployment for minimal total task completion delays using Solution Space Tree Pruning (SSTP) [73]. This approach is shown to produce significantly lower delays than Edgent, while both perform better than a cloud-only inference model. The addition of partial execution of the inference phase to layer-wise partitioning and offloading of DNNs can result in lower overall inference delays and lower processing requirements, at the cost of reduced classification accuracy. However, this approach significantly improves the performance of real-time applications (e.g., video analysis) on resource-constrained embedded devices [74]. Another solution is to partition not only into layers, but into sub-units of layers, while using a scalable, distributed algorithm to handle the offloading [75]. The Matching Game-based DINA-O offloads each individual piece to different fog nodes based on factors such as queue length, communication delays and



(a) Number of studies mentioning various types of AI in the edge, "Evolutionary" and "Genetic" representing the same concept.

(b) Number of studies mentioning various types of distributed AI and learning in the edge.

FIGURE 6. Research interest in various types of learning and AI in the edge.

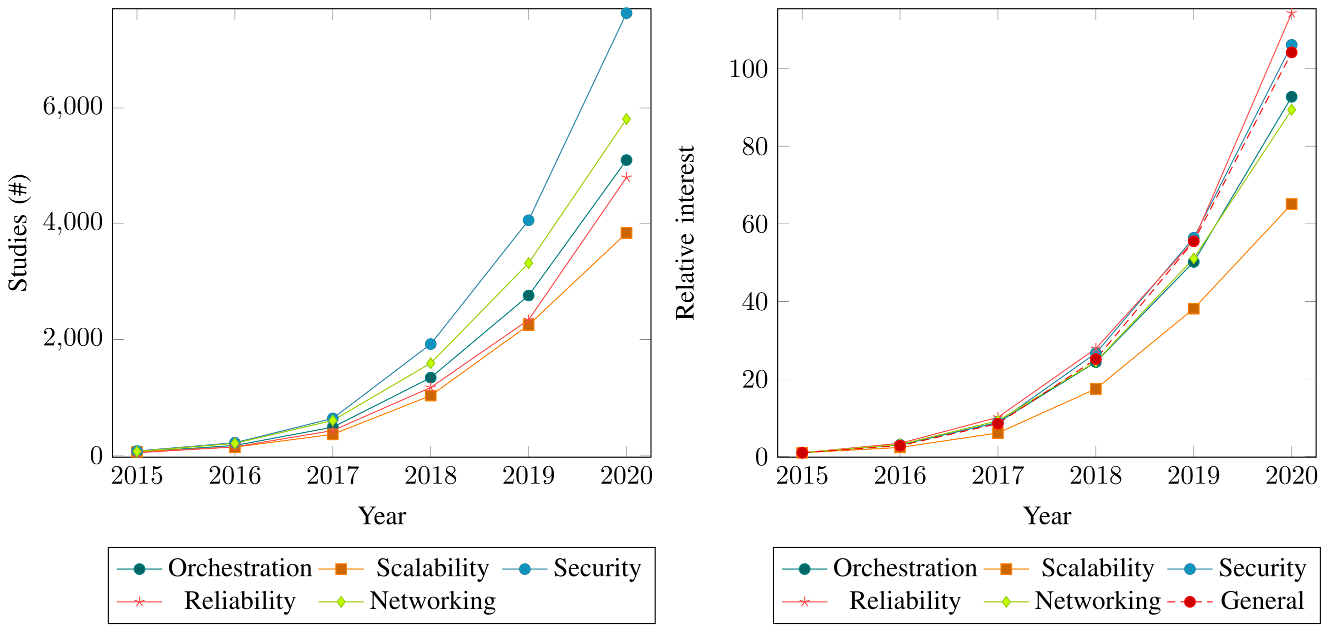
processing delays. This approach is shown to have 2.6 to 4.2 times lower total inference latencies than comparable algorithms.

The learning phase of DNNs is far more computationally intensive than inference, and thus more challenging to efficiently realize in the edge. The offloading of learning tasks from the cloud to the edge can be achieved using a graph-based task representation of a DNN [76]. In this approach, the learning task graph is requested on-demand from the cloud, and divided among nearby, suitable edge servers by the edge server that initiated the learning task using NSGA-II. The result is a collaborative learning scheme for DNNs in the edge with feedback of learned parameters to the cloud. Another solution is to remove the need for a cloud server entirely, by using a voting process to select an appropriate edge node as coordinator for a collaborative learning process [77]. The coordinator node is elected by all nodes through a democratic voting strategy, based on computational capacity and distance from the actual deployments. The learning process itself is twofold: a first training batch is executed on the coordinator node, after which the preliminary model is distributed to all other nodes for further training. The computational and energetic impact of the learning algorithm itself can be optimized by using a ternarized gradient [78]. Ternarized BackPropagation (TBP) uses only the signs of weight differences to update the model weights, rather than calculating whole integer values. Additionally, this method uses L^2 regularization and a mutation rate for weight updates during the training process. The result is increased performance without reducing the accuracy of the resulting trained model, and evaluations show that compared to default backpropagation using 16 bit integers, this method is more energy efficient by two orders of magnitude.

VII. ORGANIZING THE INTELLIGENT EDGE

Fig. 6(a) shows the interest in various types of AI since 2016. This chart does not include 2015 due to unreliable results for several categories. Neural networks are by far the most popular topic, being mentioned or used in around 50% of the studies in 2020. Around 30% of the studies mention genetic algorithms ("Genetic" + "Evolutionary"), while regression methods receive 20% of the total attention. Although neural networks are Turing complete [79], and can thus technically perform any type of calculation, the use of other AI methods makes sense in many situations, for example when the problem is more easily modeled for a different approach, or when hardware requirements are too stringent to run a neural network. The interest in various types of distributed AI is shown in Fig. 6(b). The number of mentions of all keywords has increased over 10 times in just 4 years, showing a strong interest in decentralized AI in the edge, although specific interest in FL and SI significantly outpaces general distributed algorithms.

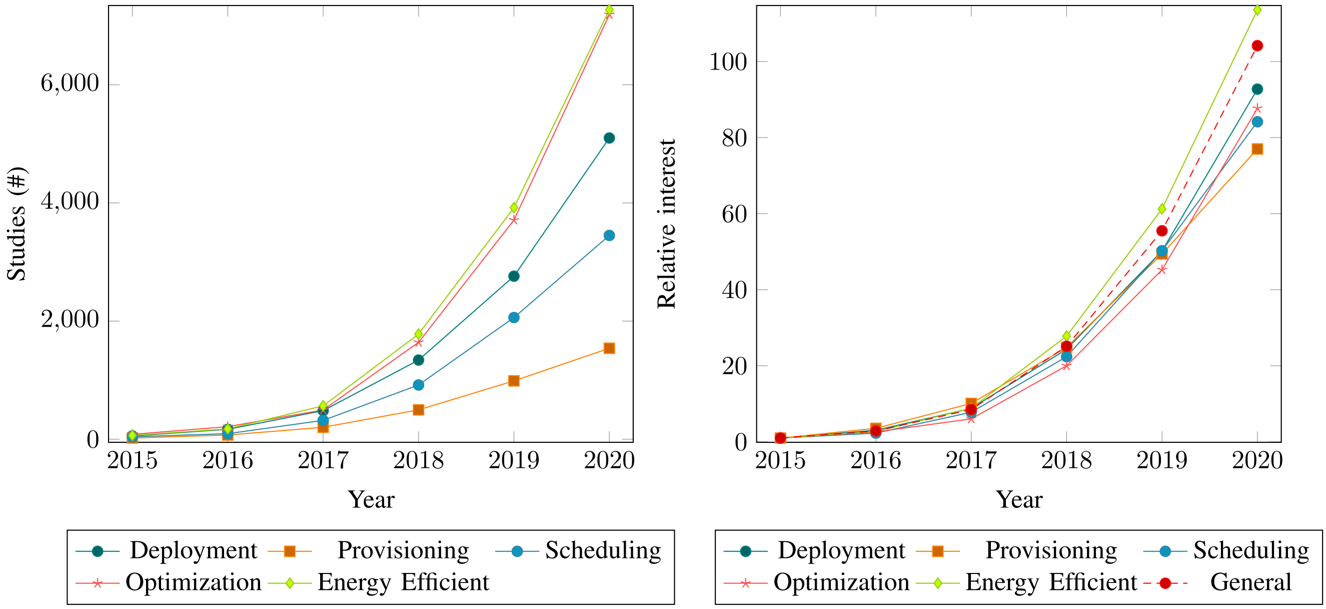
General research trends are presented in Fig. 7, showing that while there is some spread in the numbers of studies mentioning various aspects of organizing the Intelligent Edge, the relative growth is more or less equal for all keywords. The only exception is "scalability", which lags in both absolute and relative interest.



(a) Number of studies mentioning organizational aspects of the Intelligent Edge.

(b) Time-relative interest in organizational aspects of the Intelligent Edge, normalized to 2015.

FIGURE 7. Research interest in high-level organizational aspects of the Intelligent Edge. In the case of "Orchestration", the more common search term "deployment" is used.



(a) Number of studies mentioning software orchestration in the Intelligent Edge.

(b) Time-relative interest in software orchestration in the Intelligent Edge, normalized to 2015.

FIGURE 8. Research interest in software orchestration in the Intelligent Edge.

A. ORCHESTRATION

Fig. 8 shows the number of studies and relative interest in edge orchestration for AI, or using AI. Optimization and energy efficiency attract the most research interest, while provisioning is least mentioned. Despite significant differences in absolute interest, all keywords have a comparable growth in relative interest, indicating significant research potential in any topic.

An example of decentralized AI task orchestration is Cognition-Centric Fog Computing Resource Balancing (CFCRB) [80], which uses a node exploration algorithm and distributed Q-learning to find the optimal nodes to offload computational tasks to. CFCRB consists of three main concepts; **sensing** involves knowledge of node resources and IoT data acquisition, **interacting** involves efficient communication and coordination, and **learning** finds the optimal strategies for dividing workloads over resources.

Self-Optimizing Swirly (SoSwirly) [81] is another distributed edge-oriented orchestrator, using SI to let edge nodes and fog nodes find their own optimal service providers. Nodes run a discovery algorithm to find other nodes in their neighborhoods, requesting services from other nodes based on their available resources and distance. Services are redeployed on-demand in real-time as node positions and resources change, and performance is shown to be two orders of magnitude faster than evolutionary algorithms (NSGA-II).

For a more fine-grained demand-oriented deployment strategy, MDPs and a Dueling-Deep Q-network can be combined [82] to orchestrate services in mobile edge networks based on patterns in end-user service requests. Additionally, the algorithm also decides whether to let an edge service instance handle any request, or to forward it to the cloud. The approach is compared to various other deep learning solutions, showing an improvement in both total response time, and the number of requests executed in the edge rather than forwarded to the cloud.

The decision of whether or not to offload tasks from an edge device to a computational node can also be modeled as an MDP, and optimized using ϵ-greedy Q-learning [83]. This approach takes into account available resources on nodes, as well as communication channel properties and task queue lengths. Evaluations indicate execution times in line with those of offloading everything to edge servers, but higher power requirements than computing everything locally.

Fuzzy Clustering Algorithm with PSO (FCAP) [84] is a combined algorithm in which both fog nodes and computational tasks are represented as standardized resource vectors. In a preliminary phase, Fuzzy Clustering is used to divide fog nodes into computational, storage and network nodes depending on their resources. PSO is used to avoid local optima during this clustering phase. In the scheduling phase, the task resource vector is used to find the best matching class, and the most suitable node to run the task on from that class.

A similar approach, I-FASC [85], clusters the tasks into categories rather than the computational nodes, using the same classes as FCAP. The tasks of each class are scheduled using a modified Fireworks Algorithm (FA), a crossbreed between SI and evolutionary algorithms. Evaluations show that I-FASC has lower execution times than comparable algorithms, while also providing a more stable load across nodes as the number of tasks increases.

Rather than using AI to directly determine the nodes to offload tasks to, AI-Based Task Distribution Algorithm (AITDA) [86] uses a neural network on each computational node to predict the execution time for potential tasks. The predictions are based on task type and task input data, and the results are combined with policies to determine if a task should be run on a fog node or in the cloud. The example policy optimizes both response time and network traffic, and results show a significant advantage over either completely cloud-based or completely-fog based processing.

The use of dual neural networks with RL aims to provide an integral cloud to edge optimization [87]. In this approach, the first network predicts if a specified task is suitable for execution in the fog, while the second distributes fog-allocated tasks among computational nodes. The second network optimizes task placement for evenly distributed resource use and minimal communication, with the explicit goal of clustering interdependent tasks on the same nodes to further reduce network traffic.

LATA, an approach to jointly optimizing communication efficiency and end-user latency specifically for fog nodes connected by a wireless SDN [88], aims to balance the workloads of fog nodes to achieve better global response times. The algorithm itself is distributed over the SDN controller and the fog nodes, and evaluations show consistently lower latencies than comparable solutions.

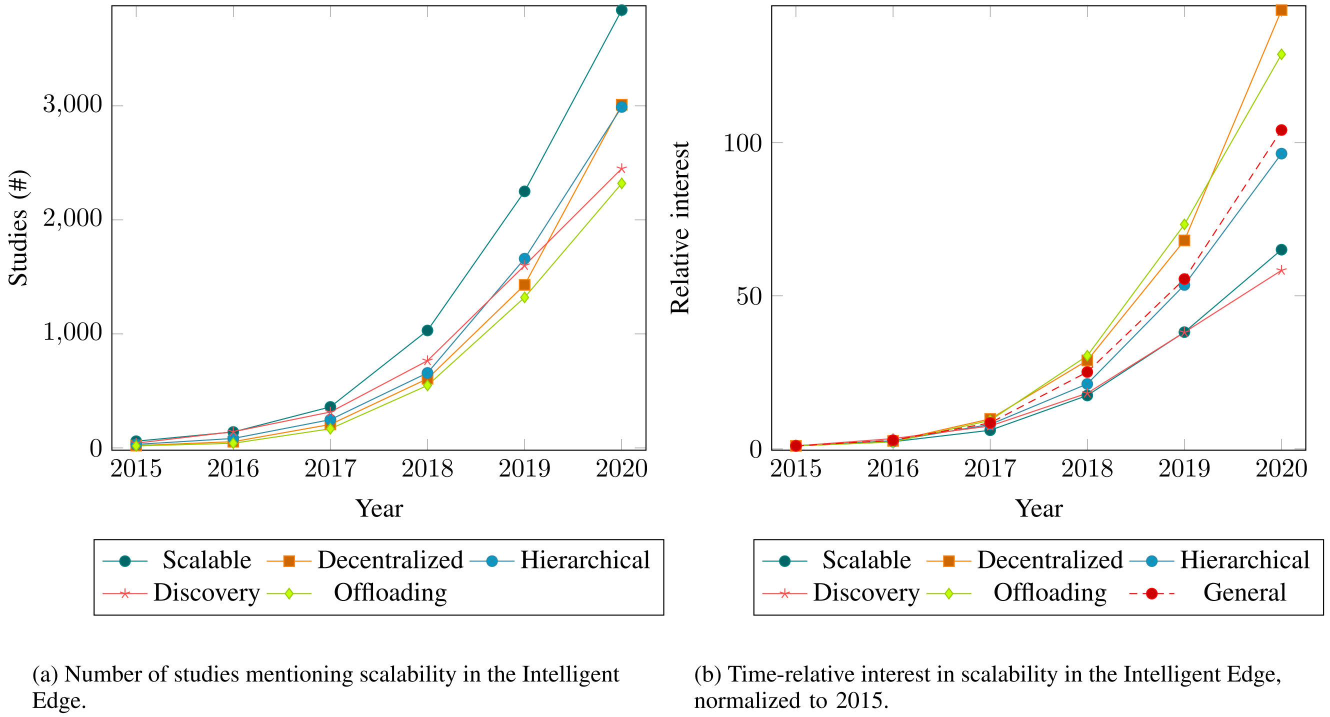
Rather than focusing only on the optimization of latency versus communication efficiency, FairTS [89] uses a resource-centered approach to online task scheduling in the fog. This solution is based on Dominant Resource Fairness (DRF) to ensure that all types of resources are divided fairly among running tasks, using RL to learn the optimal assignments. Comparison to a greedy strategy shows similar average task completion times, but more stable execution times and thus potential QoS guarantees.

Applying a fairness policy to computational nodes rather than resource allocations, Fairness Cooperation Algorithm (FCA) [90] aims to fairly divide tasks between fog nodes based on their available resources, for the joint optimization of global minimal energy consumption and task processing time. To train FCA, an algorithm is presented which converges slower than either Newton Descent or Steepest Descent in early rounds, but results in smaller error rates after only 75 rounds.

In their work, Yang *et al.* [91] present an MDP-based model which attempts to optimize the use of FL in EI. Arguing that while FL preserves privacy, it also has a negative effect on battery-powered and low-resource devices, their algorithm aims to jointly optimize both privacy gains from FL, and resource use on edge devices.

Distributed Artificial Intelligence-as-a-Service (DAIaaS) is a different take on distributed AI task orchestration [92], aiming to provide a standardized framework for distributed intelligent services in Internet of Everything (IoE) environments. Deployment parameters considered by this framework are CPU requirements, network traffic and link latencies, and it is evaluated in terms of energy and financial costs for three distinct use cases.

FogBus [93] provides a Platform-as-a-Service (PaaS) approach to cloud-fog-IoT integration, allowing platform independent deployment of software services. A multi-tiered architecture is used to standardize communication and application behavior, separating IoT devices from communication gateways, computational nodes and the cloud. A blockchain



(a) Number of studies mentioning scalability in the Intelligent Edge.

(b) Time-relative interest in scalability in the Intelligent Edge, normalized to 2015.

FIGURE 9. Research interest in scalability in the Intelligent Edge.

implementation is added in addition to other security features to ensure data integrity when transferring confidential data between nodes. The platform is evaluated in terms of energy efficiency, latency and resource use.

Some orchestrators are designed for specific domains, for example in Mobile Crowdsensing [94] using DRL with a CNN to organize the execution of tasks. The orchestrator is designed to schedule divisible computing tasks generated by edge devices, deploying each subtask in the fog or cloud depending on computational requirements. The scheduler aims to guarantee QoS for each task, and to minimize processing time and network traffic for each task.

Other orchestrators are aimed at databases rather than computational tasks [95], using MDP as a probabilistic method to determine database placement and to guarantee freely definable QoS requirements for application developers. This database orchestrator is evaluated using a Kubernetes-based implementation, and compared to Analytic Hierarchy Process (AHP) in terms of QoS violations.

B. SCALABILITY

The research interest in scalability in the Intelligent Edge is shown in Fig. 9. All keywords are more or less equally mentioned, with the umbrella term “Scalable” occurring more often, although interest in “Discovery” feathers off slightly in 2020. As growth in relative interest is concerned, “Offloading” and “Decentralized” have the fastest growing interest, while “Discovery” again lags.

Scalability and efficient orchestration have largely overlapping requirements. As a result, many of the studies listed in this section are similar to the ones discussed for “Orchestration”, they have been specifically selected to illustrate one or more aspects of scalability for EI.

While offloading is mostly an enabling technology and requires new organizational algorithms, it can also be used as a tool for scalable AIoT (Artificial Intelligence of Things). Splitting neural networks layer-wise and offloading the initial layers to IoT devices [96] has the advantage of not only scaling part of the training process with the number of edge devices, but also that training occurs where the IoT sensor data is most readily available. For the higher layers, less data intensive learned features are communicated to the cloud for further training.

On the level of a single neural network model, scalability can be achieved through the offloading of each layer to different devices. Accelerated Artificial Intelligence for IoT (AAIoT) [97] is one such approach, optimizing the response time of inference versus network traffic and computational effort through dynamic programming. Furthermore, the algorithm can operate in multi-layer IoT architectures, rather than a two-layer cloud-fog architecture.

Intelligent service discovery and integration are an essential part of functionality scaling, providing new functions without the need for manual implementation of suitable interfaces. One approach using Generative Adversarial Networks (GANs) [98] shows the potential of this type of neural network for self-learning service discovery in the edge, specifically in the context of 6G networks. The generators in this approach are trained to produce synthetic data associated with distinct service classes, based on captured data. The discriminators have the dual tasks of recognizing real data from fake data, and associating it with a specific generator. As such, specific traffic flows are discovered by adding generators, whereas the discriminator identifies and classifies them.

Decentralization is an important aspect of scaling EI, as no centralized or offloaded algorithm can scale to the load of exponentially growing edge networks. One approach enables distributed, cloud-cooperative intelligence by combining a Task Model Offloading Algorithm (TMOA) and Adaptive Task Scheduling Algorithm (ATSA) based on Ant Colony [99]. The former assigns nodes to tasks based on computational capacity, latency and energy efficiency, while the latter ensures load balancing of AI tasks between nodes. However, in this approach the scheduler algorithm itself remains a centralized instance. Evaluations show performance comparable to or better than state of the art alternatives.

A study by Lim *et al.* [100] considers the scale limiting bottleneck of communication inefficiency in FL, and resource allocation problems in the more efficient Hierarchical Federated Learning (HFL). They propose a two-level resource allocation solution for HFL. In the lower level, evolutionary game theory is used to model the process of data owners joining cluster heads, based on rewards given for data participation. In the upper level, a deep learning-based auction mechanism is used for cluster heads to service model owners. This added level of indirection is shown by evaluations to lead to stable resource allocation.

SoSwirly [81] uses an approach based on SI to distribute the task and service orchestration process itself. Each end-user device is responsible for finding the nearest suitable fog node for the services it requires, switching to other fog nodes in real-time if QoS requirements are violated. Furthermore, SoSwirly can be layered for a hierarchical architecture, from the IoT sensors through various layers to the cloud.

Recently, the network edge has been used to distribute Data Stream Processing (DSP) [101] for intelligent applications, parallelizing stream processing and increasing scalability. For example, Aggregate End-to-End Latency Strategy with Region Patterns and Latency Awareness (AELS+RP+LA) [102] aims to decrease the processing latency of DSP applications in geo-distributed cloud-edge architectures. The solution analyses DSP application graphs to determine the optimal offloading strategy, and is shown to scale up to 250.000 edge resources (edge nodes and IoT devices). Another edge DSP framework [103] optimizes energy efficiency by reducing network traffic in real-time through two components. The first is an energy-aware IoT data gathering component, using adaptive sampling to reduce its network traffic, while the second is a data prediction model which calculates future data for multiple sensor IoT environments. The data prediction model uses clustering to filter outliers and to generate reliable data, and the framework is shown to be up to 60% more energy efficient for IoT devices than continuous data streams. Finally, Processing Intelligent Agent Running on Fog Infrastructure (PIAF) [104] uses Time Petri Nets to model time-critical DSP in the context of industrial settings, using intelligent agents to distribute DSPs among the available edge nodes.

Another solution for the distributed management of cloud and edge resources for intelligent applications uses modified Virtual Infrastructure Managers (VIMs), specifically OpenStack in the cloud and Docker in the edge [105]. Both OpenStack and Docker are extended with a custom resource management API (DARK), and a Network Function Virtualization Orchestrator (MORCH). The scheduling algorithm in DARK works in real-time, mapping incoming requests in the form of service graphs onto available resources and nodes using a greedy heuristic, taking into account network conditions between nodes. The MORCH component enables network-awareness for a multi-layer architecture.

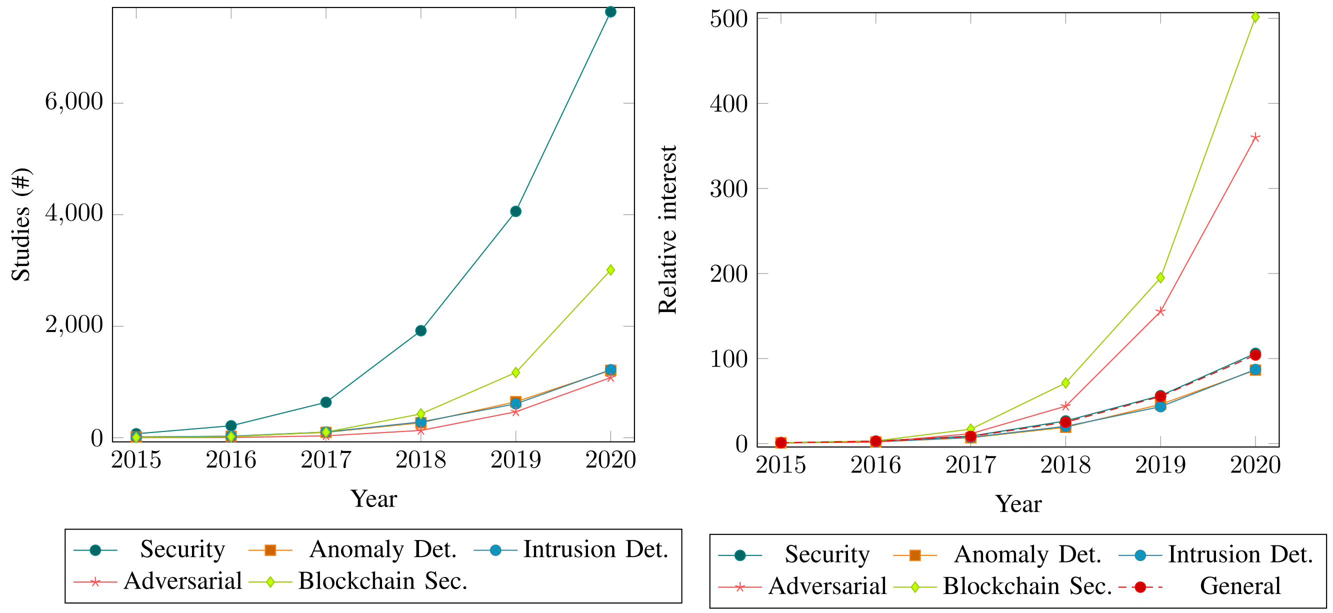
The goal of scalability studies can also be limited to a single aspect of EI. For example, a scalable Intrusion Detection System (IDS) for Smart Cities [106] based on the distributed training and inference of neural networks. Two workflows are presented, a semi-distributed approach in which feature selection is distributed but final classification is performed by a central instance in the fog, and a fully distributed version. While the accuracy of the distributed approach is about 2% less than a centralized algorithm, the Time To Build Model (TTBM) is 64.82 times faster.

C. SECURITY

The research interest in various aspects of security in the Intelligent Edge is shown in Fig. 10. While there is a great interest in security itself, more specific keywords are mentioned far less, possibly indicating that most studies focus on one specific topic, or that general security concerns are a secondary topic of many loosely related studies. While there is a significant interest in challenges such as anomaly detection and intrusion detection, there has been an explosive growth in interest related to adversarial attacks and blockchain-based solutions since 2018.

SecOFF-FCIoT [107] is aimed specifically at secure offloading of computational tasks. The data is secured at the sensor level using a Neuro-Fuzzy Model which predicts device sensitivity to malicious data injection, and offloaded to appropriate fog nodes using PSO, taking into account the processing capacity and energy levels of nodes. Although some tasks are offloaded to the cloud, RL is used to ensure data privacy by offloading tasks with sensitive data only to private clouds. Evaluations show this approach has a significantly lower energy consumption and response latency than comparable solutions.

Secure Mobile Crowdsensing Protocol (SMCP) [108] provides a framework to secure data and ensure privacy for crowdsensing applications in the edge. The framework uses a cloud server to act as a registry for fog and edge nodes, using Extended Triple Diffie–Hellman Key Agreement (X3DHKA) and Advanced Encryption Standard



(a) Number of studies mentioning AI security in the Intelligent Edge.

(b) Time-relative interest in security in the Intelligent Edge, normalized to 2015.

FIGURE 10. Research interest in security in the Intelligent Edge.

(AES) as lightweight algorithms to secure traffic and enable the mutual authentication of nodes.

A general approach to anomaly detection in the fog is provided by Yang *et al.* [109], along with a concrete example of a Deep Network Analyzer (DNA) for 5G networks.

An unnamed holistic framework by Jararweh *et al.* [110] offers a distributed approach for trustworthy and reliable edge services. This framework incorporates custom algorithms which deploy services in the edge and guarantee user privacy. To ensure data and network traffic integrity, a neural network-based IDS is integrated. Evaluations show that the accuracy of this IDS is up to 99.3%, and that response times can be significantly reduced by scaling the number of edge servers.

Another solution for anomaly detection uses a collaborative/transfer learning approach in the fog, using Principal Component Analysis (PCA) for initial feature engineering and using a variety of models (e.g., RL, DNN, SVM) for each node, selecting the optimal one [111]. The fog enabled infrastructure supporting this distributed AI consists of standard software such as Hadoop and Spark, using both batch and streaming modes.

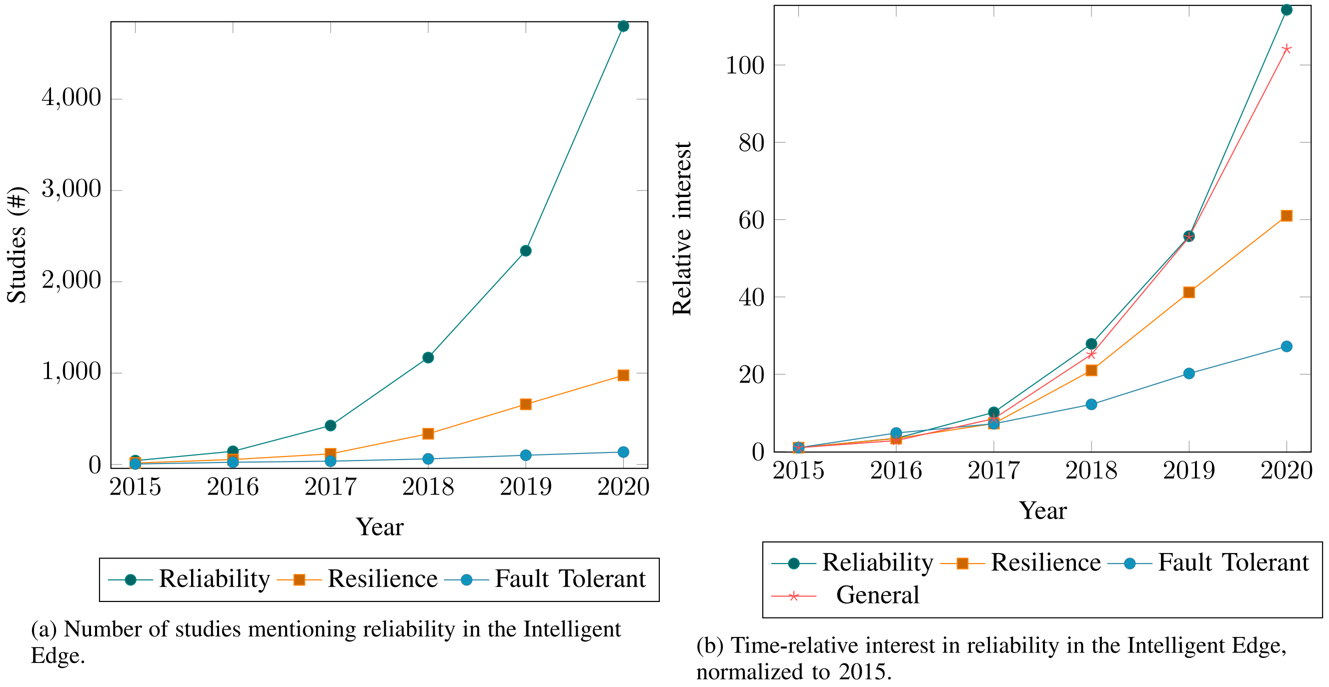
SeArch [112] is a hierarchical IDS for SDN-based cloud IoT, deployed on edge gateways, fog SDN controllers and as a cloud application. Communication channels are restricted to the same level or one level higher in the architecture. The algorithms at each level are restricted by computational power; SVM for node-level detection in the edge, Self-Organizing Maps (SOE) for network-level detection in the fog, and deep learning in the cloud. Evaluations to alternative solutions show that SeArch has similar accuracy, but significantly lower detection times.

In another approach to IDS, a framework using TA-Edge [113] uses Trusted Authority edge nodes to certify other edge devices in their domains, securing communication between them. The second component of the framework, SDN-ADS, is an SDN/Openflow based anomaly detection system which first discovers the topology and SDN data flows of the entire network. This topology is used by a malicious traffic detector to find packets with invalid properties or routed through anomalous flows.

In the context of Smart Homes, AI can be used to monitor smart devices, and to authenticate and authorize them to interact with the cloud [114]. This approach learns the specific behaviors of devices in the home network, which allows the creation of device profiles that can be authorized by end-users.

Adversarial attacks exploit the modeling properties of deep learning networks to cause misclassification or incorrect outputs. Small, but intentional perturbations in the input can reliably cause misclassification, whereas accidental, seemingly random input can sometimes be misinterpreted by a DNN, with high probability outputs. DeSVig [115] is a decentralized approach to correct such problems within milliseconds in Industrial AI systems, using Conditional GANs (CGAN) to verify the inputs and outputs of DNNs against attacks. The CGAN is trained to generate copies of the inputs supplied by DNNs, while a separate discriminator compares the generated copy against the actual input to determine whether it contains signals that indicate an attack. Evaluations indicate 96-99% accuracies for several datasets, and detection within 62ms.

Another strategy aims to construct and execute DNNs safely by ensuring data integrity during both the training



(a) Number of studies mentioning reliability in the Intelligent Edge.

(b) Time-relative interest in reliability in the Intelligent Edge, normalized to 2015.

FIGURE 11. Research interest in reliability in the Intelligent Edge.

(poisoning/backdoor attack) and inference (adversarial) phases, and the security and privacy of data transfers during training [116]. Secure training is achieved by simultaneously using active, pending and secure models for each application to detect suspected hostile data. These data are stored in a "hostile" dataset and used to update the pending model, while eventually a separate detector DNN will recognize the hostile features and integrate them into a new secure model. Security during inference is enforced through a punishment mechanism derived from a game model.

Blockchain technology is often used in conjunction with AI for the secure processing and distributed storage of transaction-like data in edge networks. For example, in Smart Healthcare [117] AI on edge devices can be leveraged for biometric data analysis and feature extraction, the results of which are stored in a blockchain, or enable the execution of smart contracts in the edge. A concrete implementation considers arrhythmia detection with a CNN in the edge, storing the resulting output along with device ID and other transactional metadata in an Ethereum chain.

Similarly, blockchain applications can aid with privacy concerning sensitive data in the edge by processing data locally using AI, and keeping track of all parties accessing the resulting features by using an Ethereum chain [118].

A different application combines blockchain-based smart contracts with trustless smart oracles for trust management in the fog computing platform of DECENTER [119]. This particular framework uses blockchains to register trusted components and users, while smart contracts use data provided by the smart oracles to verify QoS and trust requirements.

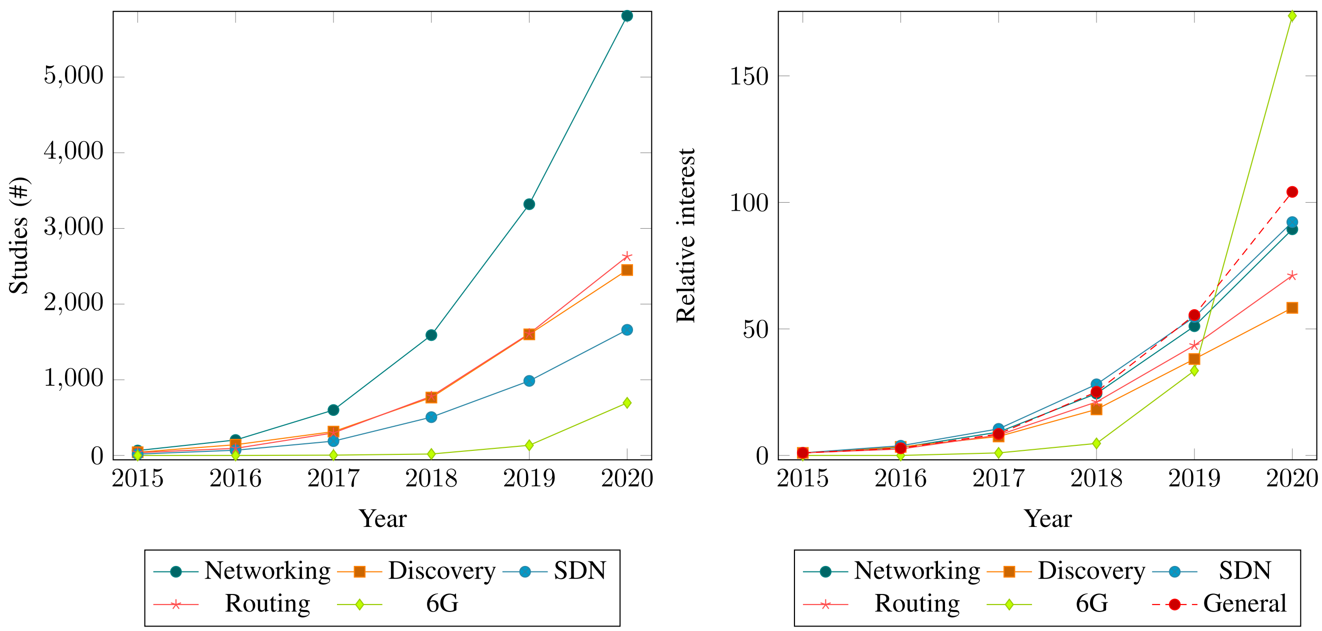
Finally, BlockSecIoTNet [120] provides an example of using blockchain technology as part of an IDS. An SDN based IoT network is used to continually monitor node traffic, allowing ubiquitous and decentralized IDS, while a blockchain ensures decentralized, trusted data storage and logging of the transactions between components. A similar approach can be applied to traffic in Vehicular Ad-Hoc Networks (VANETs) [121], in which the blockchain provides trust between actors and components.

D. RELIABILITY

Fig. 11 shows the research interest in reliability in the Intelligent Edge. The trends presented here are similar to those for the security; great interest in reliability in general, but far less interest in specific aspects, especially fault tolerance. However, the interest in reliability keeps almost perfect pace with the general interest in EI, indicating that it is consistently an important and pervasive topic in studies whose primary subject does not necessarily concern reliability.

As the reliability of systems starts with their input, reliable IoT sensor data is an important enabling factor of EI. One approach towards reliable sensor data uses fog-based validation by combining the output of several physically clustered sensors of different types to detect unreliable outputs [122]. The algorithm is applied to a scenario in which AI detects people through a security camera, showing that false negatives of the AI can be corrected through sensory substitution.

Moving up to the level of reliable AI using IoT data, deepFogGuard [123] is a DNN augmentation scheme which makes distributed inference resilient to failure. The main



(a) Number of studies mentioning networking in the Intelligent Edge.

(b) Time-relative interest in networking in the Intelligent Edge, normalized to 2015.

FIGURE 12. Research interest in networking in the Intelligent Edge.

feature of this scheme is that it relies on skip hyperconnections, which function like residual connections in DNNs, except that they skip entire nodes rather than simply layers. By ensuring at least a minimal data flow from lower layers on node failure, deepFogGuard is shown to significantly improve inference accuracy over default DNN inference, especially for high node failure rates. The mobility of vehicles in IoV can be a detriment to timely and reliable inference, but the application of AI and coded computing can instead exploit this mobility through opportunistic offloading [124]. This solution uses a modified Multi-Armed Bandit (MAB) approach to learn the delay behavior of nodes in real-time, while coded computing is used for redundant offloading, accepting whichever results are received first.

Finally, on the scope of networks, work by Radanliev *et al.* [125] develops a risk-assessment framework for the purpose of creating secure and reliable networks in extreme environments, specifically in the context of edge computing and AI.

For holistic solutions aiming to enhance overall reliability, Elastic Intelligent Fog (EiF) [126] is a general, AI-enabled fog computing framework designed to enable distributed and reliable IoT systems. The approach is similar to offloading, but implemented as PaaS, offering APIs for network, IoT and AI functions for edge deployments. The framework itself uses real-time monitoring to enable Follow-me Moving Edge Cloud functionality, in which services "follow" users in the edge, employing FL to update the deployment strategy.

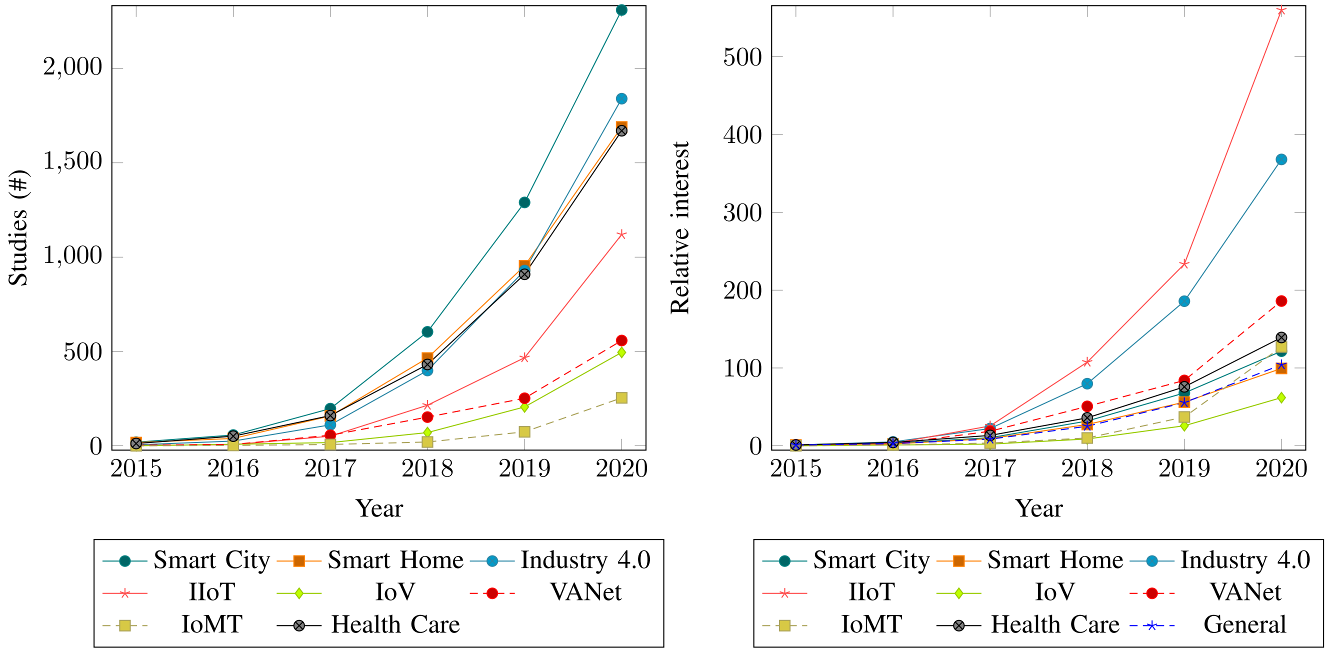
E. NETWORKING

The interest in network-related aspects of the Intelligent Edge is shown in Fig. 12. In absolute terms, discovery and routing have attracted the most interest since 2015, although interest in 6G has skyrocketed since 2018 as the concept of the Intelligent Edge has grown, and at the current growth rate it will become one of the most discussed topics in EI within 2 years.

Work by Xia *et al.* [127] illustrates the effects of AI and Fog Radio Access Networks (F-RANs) on each other. It discusses how the deployment of distributed and hierarchical AI, especially DNNs, is enabled by the properties of F-RANs, while F-RANs themselves are organized more efficiently by AI. A concrete example is given through the use of MAB to solve a caching problem with unknown content popularity.

Offloading can be used to optimize network performance, for example by minimizing communication power consumption in wireless networks [128]. Unlike most similar approaches, this framework uses statistical learning, specifically iteratively reweighted L1 minimization with difference-of-convex functions regularization. Evaluations show that this approach results in a significantly lower power consumption than comparable algorithms.

By dividing large networks into cells and applying a CNNs, cell outages and congestion can be detected and traffic rerouted. The scalability and reliability of this approach can be increased by distributing the CNN over edge servers, each managing 100 cells [129]. Evaluations indicate that this distributed anomaly detection has up to 96% accuracy.



(a) Number of studies mentioning Intelligent Edge applications.

(b) Time-relative interest in Intelligent Edge applications, normalized to 2015.

FIGURE 13. Research interest in Intelligent Edge applications.

Inductive Content Augmented Network Embedding (ICANE) [130] uses a network embedding which preserves higher order (multi-hop) node proximity, aimed to facilitate service deployment in edge networks. The embedding is learned by sampling network nodes for neighbours up to k hops, and transforming proximities and node resources into feature vectors, which are fed to an LSTM based network. Evaluations show that ICANE has significantly higher F1 scores [131] than similar algorithms for various learning datasets.

Because of its virtual nature, SDN allows for new possibilities in ad-hoc network organization. For example, a self-adaptive SDN based solution can organize virtual topologies based on application demands, available resources and physical topologies [132]. A practical implementation uses ONOS SDN controllers and OpenFlow switches, deployed by a self-adaptive framework, to organize the SDN. While this particular approach does not yet employ AI, the authors plan to use machine learning to improve the organizational algorithm.

Another framework combines the flexibility of SDN with extra security [133], with a focus on Smart Healthcare. IoT devices are authenticated by edge servers using a lightweight probabilistic k-nearest neighbor (p-KNN) based algorithm. The edge servers are used for collaborative intelligence, offloading tasks to each other, while the SDN controller is responsible for load balancing and network optimization between them. The offloading algorithm uses a form of SI, with each edge server using Beacons to alert nearby servers if their task queue grows too long.

Intelligent real-time routing decisions can greatly improve network performance. As an example, Smart Edge Broker (SEB) [134] has a dual purpose. Its main purpose is routing Smart Home traffic in edge networks, acting as a broker to organize direct communication between edge nodes instead of routing through the cloud. By keeping all communication between nodes in the edge network, latency and traffic overhead are reduced. It also acts as an edge server, filtering and processing any incoming data instead of forwarding it to the cloud.

AI plays a critical role in most research on next-generation 6G networks. One architecture [135] defines four layers of AI in 6G; intelligent sensing, analytics, intelligent control and smart applications, examining which types of AI would be suitable for each purpose. Further topics discussed include communication spectrum management, AI-empowered MEC, and intelligent mobility management.

VIII. APPLICATIONS IN THE INTELLIGENT EDGE

Fig. 13 shows the research interest in Intelligent Edge applications. In absolute numbers most domains are equally popular, but the relatively few and variable mentions of related abbreviations (e.g., IoMT, IIoT) indicate an uneven terminology. However, the recent relative growth of “IoMT” and “IIoT” may simply indicate some time is required for their widespread adoption. In relative terms, the interest in industrial applications is rising explosively, even compared to the significant growth of other domains.

A. SMART CITY

Apart from the specific domains of health care, IoV, IIoT and Smart Homes, the Smart City comprises a large number of topics and potential AI applications. This section discusses only some of the most recent, AI-based applications as an introduction.

Smart Grids, often regulated by Energy Management Systems (EMSs), play an important role in an ever more fine-grained energy grid, optimizing for demand and minimizing losses and overproduction. Improving EI paves the way for the decentralization of Smart Grid functionality to the edge, such as an AI-oriented Smart Power Meter with edge analytics for use in a cloud-assisted EMS [136]. Another Smart Grid application is the detection of energy fraud using edge-based AI [137]. In this work, data from smart appliances and distributed power sources (e.g., solar panels) is pre-processed using Principal Component Analysis (PCA) and Missing Completely At Random (MCAR), and evaluations show that neural network-based regression shows promise for the classification phase.

Another popular Smart City topic is parking surveillance. The advent of deep learning on edge hardware has enabled real-time intelligent surveillance systems based in the edge. Such systems can combine processing power of IoT devices and edge servers [138], optimizing for performance while minimizing network traffic. The work in this study combines background subtraction with a Single Shot Detector (SSD) on IoT devices, and a tracking algorithm on edge servers to efficiently track multiple vehicles in poor lighting conditions. The work of Mittal *et al.* [139] summarizes the use of and challenges of deep learning in the edge for more general surveillance applications in the Smart City.

B. SMART HOME

Edge-based surveillance systems similar to those in Smart Cities can be applied to crime prevention in Smart Home environments [140]. This work uses an event-driven approach, where edge devices use background subtraction for quick, naive motion detection. Upon motion detection, an edge device forwards video data to fog nodes, which use a CNN (VGGNet) to classify crime objects (e.g., guns). When an object is detected, labeled images tagged with location information can be forwarded to the relevant authorities.

ImPeRIum is a general, fog-enabled Smart Home solution [141]. In its architecture, data is gathered by sensor nodes and forwarded to nearby computation capable devices (defined as fog nodes, e.g., Smart TV, gateway device) for decision making. The decision process uses both an ensemble method and MLP, and the models are distributed over all fog nodes to avoid a single point of failure. Efficient dissemination of data to other devices is achieved through a Publish/Subscribe mechanism (MQTT), only publishing an event when it is dissimilar enough to the previously sent one.

The next section discusses some AI applications in Smart Health Care. Some of these can overlap with Smart Homes, for example a fog-based framework for predictive veterinary health care [142]. This framework uses FogBus [93] as a base platform, along with a WiSense mesh and a health sensor belt. The Probability of Health Vulnerability (PoHV) is calculated in the fog using sensory, environmental, behavioral and dietary data. The PoHV is further processed by a temporal ANN (t-ANN) which predicts a Temporal Sensitivity Measure (TSM), classified into alert levels. Finally, a Self Organized Map is used to create day-to-day visualizations for caregivers.

C. SMART HEALTH CARE

Fall detection is an example IoMT application which can benefit greatly from running in the edge [143]. This type of real-time application requires pervasive sensor and wireless networks, although these are often low-powered and have limited bandwidth. In the proposed architecture, sensor data is sent to a local edge gateway over low-power Bluetooth, where an LSTM RNN performs fall detection in real-time. In case a fall is detected, an event is sent over LoRa to a fog server, which sends the required notifications to caregivers.

One Smart Health Care framework is based on collaborative learning, distributing AI over the edge and fog [144]. A case study on arrhythmia detection has edge devices performing ECG signal pre-processing, feature extraction, and classification with a shallow neural network. If the probability of the classification is too low, a CNN in the fog layer takes over, using the full ECG image as input. Finally, ECG data is also streamed to the cloud, where it is combined with health provider data to train and improve (personalized) models. A similar application [145] uses EEG data to predict seizures in patients, but this approach uses Discrete Wavelet Transform (DWT) as an additional de-noising step and a Kriging model (Gaussian regression) as a classifier. Finally, deep learning can also be used for disease prediction. For example, biometric data from IoMT sensors combined with medical metadata, processed by a Deep Factorization Machine (DeepFM), can predict the presence of hepatitis [146].

D. INDUSTRY 4.0

A general study on distributed AI in IIoT by Queiroz *et al.* [147] lists key concerns for the synergy of distributed AI and Cyber-Physical Systems (CPS) in industrial settings. These concerns are fine-grained, covering every aspect from networking and embedded hardware to human-machine interaction, identifying cross-concerns for the successful cooperation of AI and CPS.

Infrastructurally, IIoT can use the same multi-tier architecture as used in various other applications of AI in the edge, with the edge interfacing with humans and machinery. One such architecture uses a cloud tier for training, fog/edge tiers for distributed inference and an SDN layer to seamlessly connect all devices [148]. An IIoT AI task scheduling algorithm is implemented on top of this architecture, optimizing for latency by taking into account computational capacity and the proximity of edge servers to manufacturing equipment. Evaluations show this approach is significantly more efficient than either cloud computing or ad-hoc, in-place execution.

Similar work also uses a multi-tier architecture, arguing how it solves latency, bandwidth and security problems compared to a purely cloud-based IIoT approach [149]. This approach uses the edge as an interfacing and control layer, and the fog as an information integration layer. The potential for a multi-tier approach to enable Digital Twin Shop-floor (DTS) is discussed, in which the virtual representation of the physical shop-floor can be used to intelligently manage and improve manufacturing processes.

A blockchain-edge framework for IIoT by Kumar *et al.* [150] does not directly involve AI, but is aimed at facilitating AI applications, and the potential for integrating FL into the blockchain is discussed.

Fogsy [151] is a holistic system for the training, deployment and management of AI in industrial settings. It operates as a fog/edge cluster management system, with facilities for data procurement and management in addition to AI model management. It also features AI pipeline management, and explainability of models through causal graphs.

A study from the Smart Maintenance Living Lab presents an approach for Smart Predictive/Preventive Maintenance [152], using a three-tiered platform based on Obelisk, a fog- and cloud-based Smart City framework. Data from IIoT sensors is gathered in the Edge tier and preprocessed (e.g., feature extraction) by gateways, after which it is sent to Obelisk in the Platform tier for ingestion. A collection of machine learning algorithms act on the ingested data to generate dashboard data for a centralized Enterprise tier.

An example AI application in IIoT is smoke detection in foggy environments [153]. This approach uses an energy efficient residual CNN based on MobileNet V2, designed for deployment in Smart City and IIoT settings. Evaluations indicate both higher accuracy and better performance than state of the art solutions.

Another concrete application of AI in IIoT is real-time poultry monitoring using EI [154]. Data from sensors monitoring the atmospheric concentration of gases such as ammonia, methane, and carbon (di)oxide is fed into an RNN with GRUs on an Nvidia Jetson Nano, predicting the evolution of air quality around the poultry farm.

E. INTERNET OF VEHICLES

In F-RANs in an IoV context, the increased wireless network traffic caused by intelligent applications can cause interference on wireless channels. RIMMA (Reliable and Interference-free Mobility Management Algorithm) [155] solves this problem by managing channels based on their characteristics, over AI-driven F-RANs. Furthermore, RIMMA is combined with fog computing to optimize for mobility, reliability and packet loss.

A similar problem on a topological level is efficient caching and communication management in quickly changing topologies with moving nodes (cars) and RSUs (Road-Side Units), especially considering the severe latency constraints on IoV applications. One solution to this problem uses twin timescales for mobility-aware offloading/content requests [156]; a long-term strategy determined by PSO, and a short term strategy determined by deep Q-learning. These strategies consider not only resource use, but also hard deadlines for requests.

Another approach considers the energy efficiency of workload deployment in fog-cloud IoV applications [157]. The algorithm uses a Learning Classifier System (specifically XCS, genetic-based machine learning), optimizing for energy use and workload delay, taking into account battery status of battery powered nodes. Evaluations show that the approach generally results in higher average battery levels than comparable algorithms, and significantly lower execution times.

There is much untapped potential for higher-level IoV applications. For example, Seal *et al.* [158] recognize that in an IoV context, the flood of data from vehicles will soon outstrip the ability of the cloud alone to process it. They develop a benchmark for real-time traffic incidence identification and traffic control, and using a multi-tier testbed, determine that a deep learning approach using (tiny)YOLOv3 is up to 80% faster in an edge-cloud architecture than in the cloud alone.

IX. CHALLENGES AND VISION

In this section, future research directions and challenges are presented for each of the main topics of this review. For a high-level vision, we can look at the highly cyclical history of computing [159], [160], which shows that eventually all functionality will end up as close to end-users as possible. The last decades have seen ever more functionality deployed closer to end-users on increasingly pervasive network infrastructures. As such, it is safe to assume that AI and other resource-intensive tasks will continue to move to the edge. Additionally, the next wave of innovation might very well emerge in a centralized form, but quickly take advantage of the infrastructure provided by the Intelligent Edge. Such next-generation applications could be highly tailored to ubiquitous user interaction, and their concepts far removed from physical systems due to increased virtualization and intelligent management.

A. ENABLING TECHNOLOGY

While common AI frameworks such as TensorFlow are capable of offloading calculations to dedicated hardware, they only offload to one PU per task. In the edge, where there may be many nearby PUs, layer-based slicing provides better overall results, offloading individual layers to different PUs based on their capabilities. This approach can be further optimized through intelligent management of PUs, monitoring their performance for several types of AI tasks. At the local network level, computational tasks are often offloaded to individual devices. A synergy with PU level offloading, either hierarchical or by peer-to-peer sharing of PU details, could provide better performance. Another challenge at this level is the efficient integration of new types of PUs with computing frameworks. Considering the highly customized nature of most hardware (e.g., FPGAs), this will likely remain in the realm of manual work, rather than automated discovery.

Significant progress has been made in low-power inference in the edge, although improvements are still likely due to incremental gains in hardware performance and model efficiency. AI training however still requires immense amounts of computation and power, often beyond the reach of individual devices in the edge, and increasing as neural networks grow deeper. To combat this issue, computational efforts can be offloaded to more powerful layers in the fog, multimode AI models can be deployed which trade accuracy for performance on resource-constrained devices, or the training workload can be spread out over many devices through FL or other cooperative strategies. While further research into truly distributed, cooperative strategies will certainly yield better performance for years to come, the learning process can also be greatly improved by reducing the amount of training required. Contributing factors for this may include improved regularization, fast-converging gradient descent strategies, and zero-shot learning.

B. ORGANIZING THE EDGE

In terms of orchestration, important challenges are real-time redeployment of (AI) services in volatile network environments, and opening up new classes of devices for the flexible deployment of services and AI. For the former challenge, the ideal is to achieve optimal QoS (e.g., availability, latency) for all users at all times, while optimizing any number of other factors (e.g., resource use, network traffic). The latter ensures that more devices can contribute their processing power, and help optimize the general functionality of the Intelligent Edge. This can be partially solved through better hardware, but also through lightweight operating systems capable of suitable virtualization (e.g., containers, unikernels). Energy efficiency is of particular importance, as edge applications increasingly push intelligence to even the most limited IoT devices. Some devices have extremely restricted power supplies, while others are battery powered and may not be (easily) rechargeable. Challenges consist of optimizing network traffic and response times over low-powered protocols, and reducing total CPU use over the lifecycle of a device. As for strategies, some may involve offloading of workloads generated by the devices, while others try to divide a known workload over a pool of devices before their batteries run out.

Significant open challenges for scalable systems are true decentralization of orchestration, and self-organizing service meshes. Almost all recent work relies on a multi-tier architecture, using the cloud as a critical infrastructural component to some degree. However, a truly scalable and flexible architecture can not be dependent on any centralized, resource-bound component (e.g., the cloud) to support an unbounded collection of devices (e.g., the edge). Likely factors to enable new architectures are peer-to-peer weight updates for AI models, local discovery of functionality and resources, and inverted deployment in the sense that an edge node primarily decides where to request/deploy a service, rather than being directed to an instance by load balancers or load balancing (distributed) DNS. As a combination of these factors, an Intelligent Edge could be envisioned in which (AI) services simply “follow” users through nearby computational nodes, pre-emptively moving to other nodes as they learn user behavioral patterns.

While progress in anomaly and intrusion detection is likely to continue, improvements are more likely to be in terms of performance and response time than accuracy, considering the high accuracy of current systems. Recent solutions for adversarial attacks are similarly effective, especially when combined with redundant systems, but adversarial attacks could be severely diminished by studying the fundamental properties of state of the art neural networks that give rise to these vulnerabilities in the first place. The increased popularity of blockchain solutions for distributed, secure transactional storage and smart contracts indicates their usefulness, but widespread adoption requires solving fundamental problems of blockchain technology related to energy consumption, fast consensus protocols, and security in privacy-sensitive applications. Privacy is a significant driving factor for the Intelligent Edge; if data is locally processed it can not be intercepted. However, privacy mechanisms are still important for processed data which is sent to the cloud, and for distributed architectures, particularly the aforementioned blockchain solutions. Some privacy concerns may be alleviated by using different types of sensors (e.g., environmental instead of cameras), encouraging (AI) services based purely on actions and behavior rather than learning properties of individuals.

The latter solution to privacy can aid with reliability; if several types of non-visual sensors are involved in a single decision, a system may be able to detect when a malfunction occurs in one of them. Network reliability in itself is important for many Smart applications, especially in IoV settings where extremely low latencies are required, despite fast-moving network nodes. Solutions to this challenge may include redundant offloading, more effective predictive offloading, and improving the reliability of layer-wise offloaded AI models. Similarly, the resilience of distributed neural networks can be improved through various means, leading to an indirect improvement in overall reliability of EI.

There are many opportunities for network-oriented research using AI in the edge. An important topic is intelligent network management through NFV and SDN to consolidate large edge networks, forming a logical, reliable topology for edge applications. Subtopics include automated discovery and integration of network resources, and redundant routing which adapts in real-time to discover optimal routes. Furthermore, initial work on 6G envisions the integration of AI into every aspect of networking from hardware through connection management to support for applications. The wide-scoped, long-term effort required to form a new, EI oriented, next-generation set of networking standards will undoubtedly provide great opportunities for new forms of intelligent network management.

C. APPLICATIONS

While there are many innovative AI-based applications in the edge, this aspect of the Intelligent Edge is still in its inception, with true Smart Cities still in the distant future. Most studies focus on a single, narrow application within their domain, using little in the way of standards and rarely considering future integration with other applications, but providing valuable proof of the potential of the Intelligent Edge. This is partly due to the constantly changing underlying technologies, which cause rapid obsolescence of existing applications, and give rise to many others. In such a rapidly changing area of research, there are many opportunities. For example, some studies present basic Smart City/Home/Industry EI management frameworks, but standards and integrated, greater scope frameworks (e.g., what TensorFlow did for neural networks) are mostly absent. Existing Smart City frameworks such as FiWare [161] and Obelisk [162] are mostly cloud-based, offering a broad support of IoT communication protocols and scalable data processing, but do not explicitly contain edge-oriented intelligent features. Ongoing IEEE standardization efforts, as presented in Section III-E, are very likely to significantly improve this situation in the near future. However, because of the limited scope of most Smart setting applications, they can be deployed modularly, and this challenge poses no immediate restriction on future innovation. Smart Cities in a broad sense offer many interesting research topics, but traffic and security aspects are likely to receive most attention in the coming years, as they can drastically improve the safety and quality of life in cities through intelligent management. Other topics, such as Smart Grids, are driven by the necessity for intelligent management because of rapidly changing energy grid conditions. In Smart Homes, an interesting topic is the discovery and integration of services and devices, and imbuing discovered devices with (partial, co-operative) AI. Such AI capable networks can then form a solid basis on which to run Smart Home applications that improve the security (e.g., intruder/weapon detection), health (e.g., fall detection, IoMT monitoring at home) and general quality of life of inhabitants. While there are many opportunities in IoV, the research potential in this area is likely to shift from roadside monitoring and traffic flow management in the fog, to inter-vehicle communication and self-organizing traffic flows as vehicles are outfitted with more powerful computational hardware. However, roadside monitoring and city-wide traffic management in the fog will remain important practical topics, especially as self-organizing traffic will remain largely impossible until intelligent, autonomous vehicles outnumber the rest. Smart Healthcare features many highly specialized applications for the prevention or monitoring of diseases and conditions. As such, an interesting challenge is to create a general monitoring and alerting framework, with AI plugins for any number of conditions and diseases. Such a framework could be integrated into the sensory network of hospitals or Smart Homes, taking into account the different types of sensors and networks present in both settings. AI models designed for this architecture should be able to flexibly handle partial or missing sensory information. Smart applications in industrial settings are, more than in any other Smart setting, highly dependent on the situation. However, digital twins are an interesting research topic, especially in terms of automated discovery and digital representation of physical industrial settings, and the subsequent optimization. Finally, human interaction with Smart applications can be used to augment AI, creating Social Edge Intelligence (SEI) [163]. SEI can drastically improve applications in which AI is used to analyze gathered data, but in which some steps benefit from higher cognitive abilities than the state of the art currently offers.

X. CONCLUSION

The use of AI in Smart applications and in the organization of the edge presents a rapidly advancing research field, with a great variety of opportunities. In this article, an introduction is given to the technologies required to understand the state of the art in Edge Intelligence (EI), and the concept of EI is elaborated using a taxonomy with “Enabling Technology”, “Organization” and “Applications” as its main topics. Research trend data from 2015 to 2020 is gathered from Google Scholar for subdivisions of these topics, and presented to show both absolute and relative interest in each subtopic. The “Organization” aspect, being the main focus of this article, has a more fine-grained subdivision, explaining all contributing factors in detail. Related work is presented, comparing it to the work in this article, and for each subdivision of the taxonomy a number of selected studies are gathered to illustrate the state of the art as completely as possible at a high level. From the research trends and selected studies, a number of short-term challenges and high-level visions for EI are formulated, providing a basis for future work.

REFERENCES

[1] A. Yousefpour *et al.*, “All one needs to know about fog computing and related edge computing paradigms: A complete survey,” *J. Syst. Archit.*, vol. 98, pp. 289–330, Sep. 2019.

[2] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, “Edge intelligence: Paving the last mile of artificial intelligence with edge computing,” *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.

[3] V. D. Hunt, “Introduction to artificial intelligence and expert systems,” in *Artificial Intelligence & Expert Systems Sourcebook*. Boston, MA, USA: Springer, 1986, pp. 1–39. [Online]. Available: https://link.springer.com/chapter/10.1007/978-1-4613-2261-0_1

[4] G. Brewka, *Artificial Intelligence—A Modern Approach* (Series in Artificial Intelligence), vol. 11, S. Russell and P. Norvig, Eds. Englewood Cliffs, NJ, USA: Prentice Hall, 1996, pp. 78–79.

[5] A. I. Schein and L. H. Ungar, "Active learning for logistic regression: An evaluation," *Mach. Learn.*, vol. 68, no. 3, pp. 235–265, 2007.

[6] T. Hastie, R. Tibshirani, and J. Friedman, "Overview of supervised learning," in *The Elements of Statistical Learning*. New York, NY, USA: Springer, Dec. 2008, pp. 9–41. [Online]. Available: https://link.springer.com/chapter/10.1007/978-0-387-84858-7_2

[7] S. Ruder, "An overview of gradient descent optimization algorithms," 2016. [Online]. Available: arXiv:1609.04747.

[8] M. L. Puterman, "Chapter 8 Markov decision processes," in *Handbooks in Operations Research and Management Science*. Amsterdam, The Netherlands: Elsevier, 1990, pp. 331–434.

[9] C. C. Aggarwal, *Neural Networks and Deep Learning*. Cham, Switzerland: Springer Nat., 2018. [Online]. Available: https://link.springer.com/book/10.1007/978-3-319-94463-0#about

[10] S. Mirjalili, *Evolutionary Algorithms and Neural Networks*. Cham, Switzerland: Springer, 2019. [Online]. Available: https://link.springer.com/book/10.1007/978-3-319-93025-1#about

[11] L. Kallel, B. Naudts, and C. R. Reeves, "Properties of fitness functions and search landscapes," in *Natural Computing Series*. Berlin, Germany: Springer, 2001, pp. 175–206. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-662-04448-3_8

[12] T. Hastie, R. Tibshirani, and J. Friedman, "Unsupervised learning," in *The Elements of Statistical Learning*. New York, NY, USA: Springer, Dec. 2008, pp. 485–585. [Online]. Available: https://link.springer.com/chapter/10.1007/978-0-387-84858-7_14

[13] L. L. M. Gen and R. Cheng, *Network Models and Optimization: Multiobjective Genetic Algorithm Approach*. London, U.K.: Springer-Verlag, 2008. [Online]. Available: https://link.springer.com/book/10.1007/978-1-84800-181-7#about

[14] T. Murata and H. Ishibuchi, "MOGA: Multi-objective genetic algorithms," in *Proc. IEEE Int. Conf. Evol. Comput.*, 1995, pp. 289–294.

[15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[16] D. A. Van Veldhuizen and G. B. Lamont, "Evolutionary computation and convergence to a Pareto Front," in *Proc. Late Breaking Papers Genet. Program. Conf.*, 1998, pp. 221–228.

[17] B. Karlik and A. V. Olgac, "Performance analysis of various activation functions in generalized MLP architectures of neural networks," *Int. J. Artif. Intell. Expert Syst.*, vol. 1, no. 4, p. 111, 2011.

[18] F. Agostinelli, M. Hoffman, P. Sadowski, and P. Baldi, "Learning activation functions to improve deep neural networks," 2015. [Online]. Available: arXiv:1412.6830.

[19] A. Pavelka and A. Procházka, "Algorithms for initialization of neural network weights," in *Proc. 12th Annu. Conf.*, 2004, pp. 453–459.

[20] C. Louizos, M. Welling, and D. P. Kingma, "Learning sparse neural networks through L_0 regularization," 2018. [Online]. Available: arXiv:1712.01312.

[21] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2015. [Online]. Available: arXiv:1409.2329.

[22] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. 30th Int. Conf. Mach. Learn.*, vol. 28, Jun. 2013, pp. 1139–1147. [Online]. Available: http://proceedings.mlr.press/v28/sutskever13.html

[23] D. Goldfarb, Y. Ren, and A. Bahamou, "Practical quasi-newton methods for training deep neural networks," 2021. [Online]. Available: arXiv:2006.08877.

[24] B. L. Deng, G. Li, S. Han, L. Shi, and Y. Xie, "Model compression and hardware acceleration for neural networks: A comprehensive survey," *Proc. IEEE*, vol. 108, no. 4, pp. 485–532, Apr. 2020.

[25] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Aug. 2017, pp. 1–6.

[26] L. R. Medsker and L. C. Jain, *Recurrent Neural Networks*. Boca Raton, FL, USA: CRC Press, Dec. 1999.

[27] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D, Nonlinear Phenomena*, vol. 404, Mar. 2020, Art. no. 132306.

[28] M. Sewak, *Deep Reinforcement Learning: Frontiers of Artificial Intelligence*. Singapore: Springer, 2019. [Online]. Available: https://www.springer.com/gp/book/9789811382840

[29] D. P. Anderson, "BOINC: A platform for volunteer computing," *J. Grid Comput.*, vol. 18, no. 1, pp. 99–122, 2019.

[30] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," 2019. [Online]. Available: arXiv:1908.07873.

[31] R. C. Eberhart, Y. Shi, and J. Kennedy, *Swarm Intelligence*. San Fransico, CA, USA: Morgan Kaufmann, 2001. [Online]. Available: https://www.sciencedirect.com/book/9781558605954/swarm-intelligence?via=ihub=

[32] J. Golosova and A. Romanovs, "The advantages and disadvantages of the blockchain technology," in *Proc. IEEE 6th Workshop Adv. Inf. Electron. Electr. Eng. (AIEEE)*, Nov. 2018, pp. 1–6.

[33] D. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Zug, Switzerland, Ethereum, Yellow Paper, 2014.

[34] J. Sedlmeir, H. U. Buhl, G. Fridgen, and R. Keller, "The energy consumption of blockchain technology: Beyond myth," *Bus. Inf. Syst. Eng.*, vol. 62, no. 6, pp. 599–608, 2020.

[35] A. Bhardwaj and C. R. Krishna, "Virtualization in cloud computing: Moving from hypervisor to containerization—A survey," *Arabian J. Sci. Eng.*, vol. 46, pp. 8585–8601, Apr. 2021.

[36] T. Goethals, M. Sebrechts, A. Atrey, B. Volckaert, and F. D. Turck, "Unikernels vs containers: An in-depth benchmarking study in the context of microservice applications," in *Proc. IEEE 8th Int. Symp. Cloud Service Comput. (SC2)*, Nov. 2018, pp. 1–8.

[37] (May 2021). *Kubernetes—Production-Grade Container Orchestration*. [Online]. Available: https://kubernetes.io/

[38] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[39] R. Sánchez-Corcuera *et al.*, "Smart cities survey: Technologies, application domains and challenges for the cities of the future," *Int. J. Distrib. Sens. Netw.*, vol. 15, no. 6, Jun. 2019, Art. no. 155014771985398.

[40] *IEEE Standard for Adoption of OpenFog Reference Architecture for Fog Computing*, IEEE Standard 1934-2018, 2018.

[41] *Self-Management Protocols for Edge Computing Node*, IEEE Standard P2805.1, 2019. [Online]. Available: https://standards.ieee.org/project/2805_1.html

[42] *Cloud-Edge Collaboration Protocols for Machine Learning*, IEEE Standard P2805.3, 2019. [Online]. Available: https://standards.ieee.org/project/2805_3.html

[43] *Guide for an Architectural Framework and Application for Collaborative Edge Computing*, IEEE Standard P2961, 2021. [Online]. Available: https://standards.ieee.org/project/2961.html

[44] *Standard for Edge Intelligent Terminal for Expressway Cooperative Transportation*, IEEE Standard P2979, 2021. [Online]. Available: https://standards.ieee.org/project/2979.html

[45] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7457–7469, Aug. 2020.

[46] Y. Shi, K. Yang, T. Jiang, J. Zhang, and K. B. Letaief, "Communication-efficient edge AI: Algorithms and systems," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2167–2191, 4th Quart., 2020.

[47] J. Zhang and D. Tao, "Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 7789–7817, May 2021.

[48] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2nd Quart., 2020.

[49] M. P. Véstias, "A survey of convolutional neural networks on edge with reconfigurable computing," *Algorithms*, vol. 12, no. 8, p. 154, 2019.

[50] M. Véstias, "Processing systems for deep learning inference on edge devices," in *Internet of Things*. Cham, Switzerland: Springer, 2020, pp. 213–240. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-44907-0_9

[51] Z. Zou, Y. Jin, P. Nevalainen, Y. Huan, J. Heikkonen, and T. Westerlund, "Edge and fog computing enabled AI for IoT—An overview," in *Proc. IEEE Int. Conf. Artif. Intell. Circuits Syst. (AICAS)*, Mar. 2019, pp. 51–56.

[52] A. Nazir, R. N. Mir, and S. Qureshi, "Exploring compression and parallelization techniques for distribution of deep neural networks over edge–fog continuum—A review," *Int. J. Intell. Comput. Cybern.*, vol. 13, no. 3, pp. 331–364, 2020.

[53] B. K. Mohanta, D. Jena, U. Satapathy, and S. Patnaik, "Survey on IoT security: Challenges and solution using machine learning, artificial intelligence and blockchain technology," *Internet Things*, vol. 11, Sep. 2020, Art. no. 100227.

[54] M. Rihan, M. Elwekeil, Y. Yang, L. Huang, C. Xu, and M. M. Selim, "Deep-VFog: When artificial intelligence meets fog computing in V2X," *IEEE Syst. J.*, vol. 15, no. 3, pp. 3492–3505, Sep. 2021.

[55] C.-X. Wang, M. D. Renzo, S. Stanczak, S. Wang, and E. G. Larsson, "Artificial intelligence enabled wireless networking for 5G and beyond: Recent advances and future challenges," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 16–23, Feb. 2020.

[56] R. Gupta, D. Reebadiya, and S. Tanwar, "6G-enabled edge intelligence for ultra-reliable low latency applications: Vision and mission," *Comput. Stand. Interfaces*, vol. 77, Aug. 2021, Art. no. 103521.

[57] J.-H. Huh and Y.-S. Seo, "Understanding edge computing: Engineering evolution with artificial intelligence," *IEEE Access*, vol. 7, pp. 164229–164245, 2019.

[58] Z. Ullah, F. Al-Turjman, L. Mostarda, and R. Gagliardi, "Applications of artificial intelligence and machine learning in smart cities," *Comput. Commun.*, vol. 154, pp. 313–323, Mar. 2020.

[59] G. M. Gilbert, S. Naiman, H. Kimaro, and B. Bagile, "A critical review of edge and fog computing for smart grid applications," in *IFIP Advances in Information and Communication Technology*. Cham, Switzerland: Springer, 2019, pp. 763–775. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-030-18400-1_62

[60] S. Sepasgozar *et al.*, "A systematic content review of artificial intelligence and the Internet of Things applications in smart home," *Appl. Sci.*, vol. 10, no. 9, p. 3074, 2020.

[61] L. Greco, G. Percannella, P. Ritrovato, F. Tortorella, and M. Vento, "Trends in IoT based solutions for health care: Moving AI to the edge," *Pattern Recognit. Lett.*, vol. 135, pp. 346–353, Jul. 2020.

[62] A. Angelopoulos *et al.*, "Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects," *Sensors*, vol. 20, no. 1, p. 109, 2019.

[63] S. Singh, P. K. Sharma, B. Yoon, M. Shojafar, G. H. Cho, and I.-H. Ra, "Convergence of blockchain and artificial intelligence in IoT network for the sustainable smart city," *Sustain. Cities Soc.*, vol. 63, Dec. 2020, Art. no. 102364.

[64] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1508–1532, 2nd Quart., 2019.

[65] Y. Wu, H.-N. Dai, and H. Wang, "Convergence of blockchain and edge computing for secure and scalable IIoT critical infrastructures in industry 4.0," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2300–2317, Feb. 2021.

[66] D. C. Nguyen *et al.*, "Federated learning meets blockchain in edge computing: Opportunities and challenges," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12806–12825, Aug. 2021.

[67] L. Fiack, L. Rodriguez, and B. Miramond, "Hardware design of a neural processing unit for bio-inspired computing," in *Proc. IEEE 13th Int. New Circuits Syst. Conf. (NEWCAS)*, Jun. 2015, pp. 1–4.

[68] A. Yazdanbakhsh, K. Seshadri, B. Akin, J. Laudon, and R. Narayanaswami, "An evaluation of edge TPU accelerators for convolutional neural networks," 2021. [Online]. Available: arXiv:2102.10423.

[69] K. Karras *et al.*, "A hardware acceleration platform for AI-based inference at the edge," *Circuits Syst. Signal Process.*, vol. 39, no. 2, pp. 1059–1070, 2019.

[70] B. Kim, S. Lee, A. R. Trivedi, and W. J. Song, "Energy-efficient acceleration of deep neural networks on realtime-constrained embedded edge devices," *IEEE Access*, vol. 8, pp. 216259–216270, 2020.

[71] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge AI: On-demand accelerating deep neural network inference via edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 447–457, Jan. 2020.

[72] S. Dey, J. Mondal, and A. Mukherjee, "Offloaded execution of deep learning inference at edge: Challenges and insights," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2019, pp. 855–861.

[73] W. Jiang and S. Lv, "Inference acceleration model of branched neural network based on distributed deployment in fog computing," in *Web Information Systems and Applications*. Cham, Switzerland: Springer, 2020, pp. 503–512. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-030-60029-7_45

[74] S. Dey, A. Mukherjee, A. Pal, and P. Balamuralidhar, "Embedded deep inference in practice," in *Proc. 1st Workshop Mach. Learn. Edge Sens. Syst. SenSys-ML*, 2019, pp. 25–30.

[75] T. Mohammed, C. Joe-Wong, R. Babbar, and M. D. Francesco, "Distributed inference acceleration with adaptive DNN partitioning and offloading," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Jul. 2020, pp. 854–863.

[76] L. Zhang, J. Wu, S. Mumtaz, J. Li, H. Gacanin, and J. J. P. C. Rodrigues, "Edge-to-edge cooperative artificial intelligence in smart cities with on-demand learning offloading," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[77] T. Zhang, Z. Shen, J. Jin, X. Zheng, A. Tagami, and X. Cao, "Achieving democracy in edge intelligence: A fog-based collaborative learning scheme," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2751–2761, Feb. 2021.

[78] T. Kaneko *et al.*, "A study on a low power optimization algorithm for an edge-AI device," *Nonlinear Theory Appl.*, vol. 10, no. 4, pp. 373–389, 2019.

[79] H. Siegelmann and E. Sontag, "On the computational power of neural nets," *J. Comput. Syst. Sci.*, vol. 50, no. 1, pp. 132–150, 1995.

[80] S. Liao, J. Wu, S. Mumtaz, J. Li, R. Morello, and M. Guizani, "Cognitive balance for fog computing resource in Internet of Things: An edge learning approach," *IEEE Trans. Mobile Comput.*, early access, Sep. 24, 2020, doi: 10.1109/TMC.2020.3026580.

[81] T. Goethals, F. D. Turck, and B. Volckaert, "Self-organizing fog support services for responsive edge computing," *J. Netw. Syst. Manage.*, vol. 29, no. 2, p. 16, 2021.

[82] Y. Zhai, T. Bao, L. Zhu, M. Shen, X. Du, and M. Guizani, "Toward reinforcement-learning-based service deployment of 5G mobile edge computing with request-aware scheduling," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 84–91, Feb. 2020.

[83] X. Liu, Z. Qin, and Y. Gao, "Resource allocation for edge computing in IoT networks via reinforcement learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[84] G. Li, Y. Liu, J. Wu, D. Lin, and S. Zhao, "Methods of resource scheduling based on optimized fuzzy clustering in fog computing," *Sensors*, vol. 19, no. 9, p. 2122, 2019.

[85] S. Wang, T. Zhao, and S. Pang, "Task scheduling algorithm based on improved firework algorithm in fog computing," *IEEE Access*, vol. 8, pp. 32385–32394, 2020.

[86] M. Abedi and M. Pourkiani, "Resource allocation in combined fog-cloud scenarios by using artificial intelligence," in *Proc. 5th Int. Conf. Fog Mobile Edge Comput. (FMEC)*, Apr. 2020, pp. 218–222.

[87] M. K. Pandit, R. N. Mir, and M. A. Chishti, "Adaptive task scheduling in IoT using reinforcement learning," *Int. J. Intell. Comput. Cybern.*, vol. 13, no. 3, pp. 261–282, 2020.

[88] G. S. S. Chalapathi, V. Chamola, C.-K. Tham, S. Gurunarayanan, and N. Ansari, "An optimal delay aware task assignment scheme for wireless SDN networked edge cloudlets," *Future Gener. Comput. Syst.*, vol. 102, pp. 862–875, Jan. 2020.

[89] S. Bian, X. Huang, and Z. Shao, "Online task scheduling for fog computing with multi-resource fairness," in *Proc. IEEE 90th Veh. Technol. Conf. (VTC Fall)*, Sep. 2019, pp. 1–5.

[90] Y. Dong, S. Guo, J. Liu, and Y. Yang, "Energy-efficient fair cooperation fog computing in mobile edge networks for smart city," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7543–7554, Oct. 2019.

[91] W. Yang, Y. Zhang, W. Y. B. Lim, Z. Xiong, Y. Jiao, and J. Jin, "Privacy is not free: Energy-aware federated learning for mobile and edge intelligence," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2020, pp. 233–238.

[92] N. Janbi, I. Katib, A. Albeshri, and R. Mehmood, "Distributed artificial intelligence-as-a-service (DAIaaS) for smarter IoE and 6G environments," *Sensors*, vol. 20, no. 20, p. 5796, Oct. 2020.

[93] S. Tuli, R. Mahmud, S. Tuli, and R. Buyya, "FogBus: A blockchain-based lightweight framework for edge and fog computing," *J. Syst. Softw.*, vol. 154, pp. 22–36, Aug. 2019.

[94] H. Li, K. Ota, and M. Dong, "Deep reinforcement scheduling for mobile crowdsensing in fog computing," *ACM Trans. Internet Technol.*, vol. 19, no. 2, pp. 1–18, 2019.

[95] P. Kochovski, R. Sakellariou, M. Bajec, P. Drobintsev, and V. Stankovski, "An architecture and stochastic method for database container placement in the edge-fog-cloud continuum," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, May 2019, pp. 396–405.

[96] O. Debauche, S. Mahmoudi, S. A. Mahmoudi, P. Manneback, and F. Lebeau, "A new edge architecture for AI-IoT services deployment," *Procedia Comput. Sci.*, vol. 175, pp. 10–19, Aug. 2020.

[97] J. Zhou, Y. Wang, K. Ota, and M. Dong, "AAIoT: Accelerating artificial intelligence in IoT systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 825–828, Jun. 2019.

[98] Y. Xiao, G. Shi, Y. Li, W. Saad, and H. V. Poor, "Toward self-learning edge intelligence in 6G," *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 34–40, Dec. 2020.

[99] S. Xu, Z. Zhang, M. Kadoch, and M. Cheriet, "A collaborative cloud-edge computing framework in distributed neural network," *EURASIP J. Wireless Commun. Netw.*, vol. 2020, Oct. 2020, Art. no. 211. [Online]. Available: https://jwcn-eurasipjournals.springeropen.com/articles/10.1186/s13638-020-01794-2

[100] W. Y. B. Lim *et al.*, "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 536–550, Mar. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9479786

[101] R. Dautov, S. Distefano, D. Bruneo, F. Longo, G. Merlino, and A. Puliafito, "Pushing intelligence to the edge with a stream processing architecture," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jun. 2017, pp. 792–799.

[102] A. da Silva Veith, M. D. de Assuncao, and L. Lefevre, "Latency-aware strategies for deploying data stream processing applications on large cloud-edge infrastructure," *IEEE Trans. Cloud Comput.*, early access, Jul. 20, 2021, doi: 10.1109/TCC.2021.3097879.

[103] E. R. de Oliveira, F. Delicato, A. R. da Rocha, and M. Mattoso, "A real-time and energy-aware framework for data stream processing in the Internet of Things," in *Proc. 6th Int. Conf. Internet Things Big Data Security*, 2021, pp. 17–28.

[104] I. Gharbi, K. Barkaoui, and B. A. Samir, "An intelligent agent-based industrial IoT framework for time-critical data stream processing," in *Mobile, Secure, and Programmable Networking*. Cham, Switzerland: Springer, 2021, pp. 195–208. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-67550-9_13

[105] B. Sonkoly *et al.*, "Scalable edge cloud platforms for IoT services," *J. Netw. Comput. Appl.*, vol. 170, Nov. 2020, Art. no. 102785.

[106] M. A. Rahman, A. T. Asyhari, L. S. Leong, G. B. Satrya, M. H. Tao, and M. F. Zolkipli, "Scalable machine learning-based intrusion detection system for IoT-enabled smart cities," *Sustain. Cities Soc.*, vol. 61, Oct. 2020, Art. no. 102324.

[107] A. A. Alli and M. M. Alam, "SecOFF-FCIoT: Machine learning based secure offloading in fog-cloud of things for smart city applications," *Internet Things*, vol. 7, Sep. 2019, Art. no. 100070.

[108] F. Concone, G. L. Re, and M. Morana, "SMCP: A secure mobile crowdsensing protocol for fog-based applications," *Human-Centric Computing and Information Sciences*, vol. 10, no. 1, p. 28, Jul. 2020.

[109] K. Yang, H. Ma, and S. Dou, "Fog intelligence for network anomaly detection," *IEEE Netw.*, vol. 34, no. 2, pp. 78–82, Mar./Apr. 2020.

[110] Y. Jararweh, S. Otoum, and I. A. Ridhawi, "Trustworthy and sustainable smart city services at the edge," *Sustain. Cities Society*, vol. 62, Nov. 2020, Art. no. 102394.

[111] S. Xu, Y. Qian, and R. Q. Hu, "Data-driven network intelligence for anomaly detection," *IEEE Netw.*, vol. 33, no. 3, pp. 88–95, May/Jun. 2019.

[112] T. G. Nguyen, T. V. Phan, B. T. Nguyen, C. So-In, Z. A. Baig, and S. Sanguanpong, "SeArch: A collaborative and intelligent NIDS architecture for SDN-based cloud IoT networks," *IEEE Access*, vol. 7, pp. 107678–107694, 2019.

[113] K. N. Qureshi, G. Jeon, and F. Piccialli, "Anomaly detection and trust authority in artificial intelligence and cloud computing," *Comput. Netw.*, vol. 184, Jan. 2021, Art. no. 107647.

[114] M. S. Zareen, S. Tahir, M. Akhlaq, and B. Aslam, "Artificial intelligence/machine learning in IoT for authentication and authorization of edge devices," in *Proc. Int. Conf. Appl. Eng. Math. (ICAEM)*, Aug. 2019, pp. 220–224.

[115] G. Li, K. Ota, M. Dong, J. Wu, and J. Li, "DeSVig: Decentralized swift vigilance against adversarial attacks in industrial artificial intelligence systems," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3267–3277, May 2020.

[116] C. Zhou, Q. Liu, and R. Zeng, "Novel defense schemes for artificial intelligence deployed in edge computing environment," *Wireless Commun. Mobile Comput.*, vol. 2020, Aug. 2020, Art. no. 8832697. [Online]. Available: https://www.hindawi.com/journals/wcmc/2020/8832697/

[117] T. N. Gia, A. Nawaz, J. P. Querata, H. Tenhunen, and T. Westerlund, *Artificial Intelligence at the Edge in the Blockchain of Things* (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering). Cham, Switzerland: Springer, 2020, pp. 267–280. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-030-49289-2_21

[118] A. Nawaz, T. N. Gia, J. P. Queralta, and T. Westerlund, "Edge AI and blockchain for privacy-critical and data-sensitive applications," in *Proc. 12th Int. Conf. Mobile Comput. Ubiquitous Netw. (ICMU)*, Nov. 2019, pp. 1–2.

[119] P. Kochovski, S. Gec, V. Stankovski, M. Bajec, and P. D. Drobintsev, "Trust management in a blockchain based fog computing platform with trustless smart oracles," *Future Gener. Comput. Syst.*, vol. 101, pp. 747–759, Dec. 2019.

[120] S. Rathore, B. W. Kwon, and J. H. Park, "BlockSecIoTNet: Blockchain-based decentralized security architecture for IoT network," *J. Netw. Comput. Appl.*, vol. 143, pp. 167–177, Oct. 2019.

[121] J. Gao *et al.*, "A blockchain-SDN-enabled Internet of Vehicles environment for fog computing and 5G networks," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4278–4291, May 2020.

[122] L. Russell, F. Kwamena, and R. Goubran, "Towards reliable IoT: Fog-based AI sensor validation," in *Proc. IEEE Cloud Summit*, Aug. 2019, pp. 37–44.

[123] A. Yousefpour *et al.*, "Guardians of the deep fog: Failure-resilient DNN inference from edge to cloud," in *Proc. 1st Int. Workshop Challenges Artif. Intell. Mach. Learn. Internet Things (AIChallengeIoT)*, 2019, pp. 25–31.

[124] S. Zhou, Y. Sun, Z. Jiang, and Z. Niu, "Exploiting moving intelligence: Delay-optimized computation offloading in vehicular fog networks," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 49–55, May 2019.

[125] P. Radanliev *et al.*, "Design of a dynamic and self-adapting system, supported with artificial intelligence, machine learning and real-time intelligence for predictive cyber risk analytics in extreme environments, cyber risk in the colonisation of Mars," *Safety Extreme Environ.*, vol. 2, pp. 219–230, Feb. 2021.

[126] J. An *et al.*, "EiF: Toward an elastic IoT fog framework for AI services," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 28–33, May 2019.

[127] W. Xia, X. Zhang, G. Zheng, J. Zhang, S. Jin, and H. Zhu, "The interplay between artificial intelligence and fog radio access networks," *China Commun.*, vol. 17, no. 8, pp. 1–13, Aug. 2020.

[128] K. Yang, Y. Shi, W. Yu, and Z. Ding, "Energy-efficient processing and robust wireless cooperative transmission for edge inference," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9456–9470, Oct. 2020.

[129] B. Hussain, Q. Du, A. Imran, and M. A. Imran, "Artificial intelligence-powered mobile edge computing-based anomaly detection in cellular networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 4986–4996, Aug. 2020.

[130] B. Yuan, J. Panneerselvam, L. Liu, N. Antonopoulos, and Y. Lu, "An inductive content-augmented network embedding model for edge artificial intelligence," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4295–4305, Jul. 2019.

[131] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation," in *Advances in Information Retrieval* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2005, pp. 345–359. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-31865-1_25

[132] I. Bedhief, L. Foschini, P. Bellavista, M. Kassar, and T. Aguili, "Toward self-adaptive software defined fog networking architecture for IIoT and industry 4.0," in *Proc. IEEE 24th Int. Workshop Comput. Aided Model. Design Commun. Links Netw. (CAMAD)*, Sep. 2019, pp. 1–5.

[133] J. Li *et al.*, "A secured framework for SDN-based edge computing in IoT-enabled healthcare system," *IEEE Access*, vol. 8, pp. 135479–135490, 2020.

[134] J. Ahn and B. M. Lee, "Enhanced smart edge broker using fog computing for smart homes," *Int. J. Artif. Intell. Appl. Smart Devices*, vol. 7, no. 1, pp. 1–6, 2019.

[135] H. Yang, A. Alphones, Z. Xiong, D. Niyato, J. Zhao, and K. Wu, "Artificial-intelligence-enabled intelligent 6G networks," *IEEE Netw.*, vol. 34, no. 6, pp. 272–280, Nov./Dec. 2020.

[136] Y.-Y. Chen, Y.-H. Lin, C.-C. Kung, M.-H. Chung, and I.-H. Yen, "Design and implementation of cloud analytics-assisted smart power meters considering advanced artificial intelligence as edge analytics in demand-side management for smart homes," *Sensors*, vol. 19, no. 9, p. 2047, May 2019.

[137] J. C. Olivares-Rojas, E. Reyes-Archundia, N. E. Rodriiguez-Maya, J. A. Gutierrez-Gnecchi, I. Molina-Moreno, and J. Cerda-Jacobo, "Machine learning model for the detection of electric energy fraud using an edge-fog computing architecture," in *Proc. IEEE Int. Conf. Eng. Veracruz (ICEV)*, Oct. 2020, pp. 1–6.

[138] R. Ke, Y. Zhuang, Z. Pu, and Y. Wang, "A smart, efficient, and reliable parking surveillance system with edge artificial intelligence on IoT devices," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 4962–4974, Aug. 2021.

[139] V. Mittal, A. Tyagi, and B. Bhushan. (2020). *Smart Surveillance Systems With Edge Intelligence: Convergence of Deep Learning and Edge Computing*. [Online]. Available: http://dx.doi.org/10.2139/ssrn.3599865

[140] T. Sultana and K. A. Wahid, "IoT-guard: Event-driven fog-based video surveillance system for real-time security management," *IEEE Access*, vol. 7, pp. 134881–134894, 2019.

[141] G. P. R. Filho *et al.*, "A fog-enabled smart home solution for decision-making using smart objects," *Future Gener. Comput. Syst.*, vol. 103, pp. 18–27, Feb. 2020.

[142] M. Bhatia, "Fog computing-inspired smart home framework for predictive veterinary healthcare," *Microprocess. Microsyst.*, vol. 78, Oct. 2020, Art. no. 103227.

[143] J. P. Queralta, T. N. Gia, H. Tenhunen, and T. Westerlund, "Edge-AI in LoRa-based health monitoring: Fall detection system with fog computing and LSTM recurrent neural networks," in *Proc. 42nd Int. Conf. Telecommun. Signal Process. (TSP)*, Jul. 2019, pp. 601–604.

[144] B. Farahani, M. Barzegari, F. S. Aliee, and K. A. Shaik, "Towards collaborative intelligent IoT eHealth: From device to fog, and cloud," *Microprocess. Microsyst.*, vol. 72, Feb. 2020, Art. no. 102938.

[145] Y. Qiu, H. Zhang, and K. Long, "Computation offloading and wireless resource management for healthcare monitoring in fog-computing based Internet of Medical Things," *IEEE Internet Things J.*, early access, Mar. 17, 2021, doi: 10.1109/JIOT.2021.3066604.

[146] Z. Yu, S. U. Amin, M. Alhussein, and Z. Lv, "Research on disease prediction based on improved DeepFM and IoMT," *IEEE Access*, vol. 9, pp. 39043–39054, 2021.

[147] J. Queiroz, P. Leitao, J. Barbosa, and E. Oliveira, "Distributing intelligence among cloud, fog and edge in industrial cyber-physical systems," in *Proc. 16th Int. Conf. Informat. Control Autom. Robot.*, 2019, pp. 1–8.

[148] X. Li, J. Wan, H.-N. Dai, M. Imran, M. Xia, and A. Celesti, "A hybrid computing solution and resource scheduling strategy for edge computing in smart manufacturing," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4225–4234, Jul. 2019.

[149] Q. Qi and F. Tao, "A smart manufacturing service system based on edge computing, fog computing, and cloud computing," *IEEE Access*, vol. 7, pp. 86769–86777, 2019.

[150] T. Kumar *et al.*, "BlockEdge: Blockchain-edge framework for industrial IoT networks," *IEEE Access*, vol. 8, pp. 154166–154185, 2020.

[151] P. Wiener, P. Zehnder, M. Heyden, P. Philipp, and D. Riemer, "Fogsy: Towards holistic industrial AI management in fog and edge environments," in *Proc. KuVS-Fachgespräch Fog Comput.*, 2020. [Online]. Available: https://www.researchgate.net/publication/340875642_Fogsy_Towards_Holistic_Industrial_AI_Management_in_Fog_and_Edge_Environments

[152] P. Moens *et al.*, "Scalable fleet monitoring and visualization for smart machine maintenance and industrial IoT applications," *Sensors*, vol. 20, no. 15, p. 4308, 2020.

[153] K. Muhammad, S. Khan, V. Palade, I. Mehmood, and V. H. C. de Albuquerque, "Edge intelligence-assisted smoke detection in foggy surveillance environments," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1067–1075, Feb. 2020.

[154] O. Debauche, S. Mahmoudi, S. A. Mahmoudi, P. Manneback, J. Bindelle, and F. Lebeau, "Edge computing and artificial intelligence for real-time poultry monitoring," *Procedia Comput. Sci.*, vol. 175, pp. 534–541, Aug. 2020.

[155] A. H. Sodhro, G. H. Sodhro, M. Guizani, S. Pirbhulal, and A. Boukerche, "AI-enabled reliable channel modeling architecture for fog computing vehicular networks," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 14–21, Apr. 2020.

[156] L. T. Tan, R. Q. Hu, and L. Hanzo, "Twin-timescale artificial intelligence aided mobility-aware edge caching and computing in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3086–3099, Apr. 2019.

[157] M. Abbasi, M. Yaghoobikia, M. Rafiee, A. Jolfaei, and M. R. Khosravi, "Energy-efficient workload allocation in fog-cloud based services of intelligent transportation systems using a learning classifier system," *IET Intell. Transp. Syst.*, vol. 14, no. 11, pp. 1484–1490, Nov. 2020.

[158] A. Seal, S. Bhattacharya, and A. Mukherjee, "Fog computing for real-time accident identification and related congestion control," in *Proc. IEEE Int. Syst. Conf. (SysCon)*, Apr. 2019, pp. 1–8.

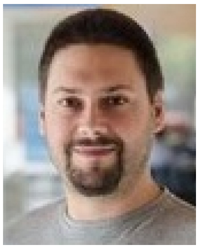
[159] D. Peak and M. Azadmanesh, "Centralization/decentralization cycles in computing: Market evidence," *Inf. Manage.*, vol. 31, no. 6, pp. 303–317, 1997.

[160] P. G. Lopez *et al.*, "Edge-centric computing," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 37–42, 2015.

[161] V. Araujo, K. Mitra, S. Saguna, and C. Åhlund, "Performance evaluation of FIWARE: A cloud-based IoT platform for smart cities," *J. Parallel Distrib. Comput.*, vol. 132, pp. 250–261, Oct. 2019.

[162] V. Bracke, M. Sebrechts, B. Moons, J. Hoebeke, F. D. Turck, and B. Volckaert, "Design and evaluation of a scalable Internet of Things backend for smart ports," *Softw. Pract. Exp.*, vol. 51, no. 7, pp. 1557–1579, Apr. 2021.

[163] D. Wang, D. Zhang, Y. Zhang, M. T. Rashid, L. Shang, and N. Wei, "Social edge intelligence: Integrating human and artificial intelligence at the edge," in *Proc. IEEE First Int. Conf. Cogn. Mach. Intell. (CogMI)*, Dec. 2019, pp. 194–201.



TOM GOETHALS received the master's degree in information engineering technology from University College Ghent, Belgium, in 2013. He is currently pursuing the Ph.D. degree with the IDLab, Ghent University, during which he has received multiple best paper awards. He worked as a Software Engineer for several years. His current research deals with scalable and reliable software systems for smart cities, working on various projects in cooperation with industry partners.

BRUNO VOLCKAERT (Member, IEEE) received the Ph.D. degree in grid resource management in 2006. He is a Professor of Advanced Programming and Software Engineering with IDLab, Ghent University and a Senior Researcher with imec. His current research deals with reliable high performance distributed software systems for smart cities, scalable cybersecurity detection, and autonomous optimization of cloud-based applications.

FILIP DE TURCK (Fellow, IEEE) leads the Network and Service Management Research Group, Ghent University, Belgium, and imec. He (co)authored over 700 peer reviewed papers and his research interests include design of efficient softwarized network and cloud systems. He is involved in several research projects with industry and academia, serves as the Chair of the IEEE Technical Committee on Network Operations and Management, and the Steering Committee Member of the IM, NOMS, CNSM, and NetSoft conferences. He serves as the Editor-in-Chief of IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT.