Journal of Network and Computer Applications 194 (2021) 103213

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

A survey on deep learning for challenged networks: Applications and trends

Kaylani Bochie [a,*], Mateus S. Gilbert [a], Luana Gantert [a], Mariana S.M. Barbosa [a], Dianne S.V. Medeiros [b], Miguel Elias M. Campista [a]

[a] Grupo de Teleinformática e Automação – GTA, PEE/COPPE-DEL/POLI, Universidade Federal do Rio de Janeiro – UFRJ, Rio de Janeiro, RJ, Brazil
[b] Universidade Federal Fluminense – UFF, PPGEET, MídiaCom, Niterói, RJ, Brazil

ARTICLE INFO

Keywords:
Challenged networks
Internet of Things
Sensor networks
Industrial networks
Wireless mobile networks
Vehicular networks
Deep learning
Machine learning

ABSTRACT

Computer networks are dealing with growing complexity, given the ever-increasing volume of data produced by all sorts of network nodes. Performance improvements are a non-stop ambition and require tuning fine-grained details of the system operation. Analyzing such data deluge, however, is not straightforward and sometimes not supported by the system. There are often problems regarding scalability and the predisposition of the involved nodes to understand and transfer the data. This issue is at least partially circumvented by knowledge acquisition from past experiences, which is a characteristic of the herein called "challenged networks". The addition of intelligence in these scenarios is fundamental to extract linear and non-linear relationships from the data collected by multiple sources. This is undoubtedly an invitation to machine learning and, more particularly, to deep learning.

This paper identifies five different challenged networks: IoT and sensor, mobile, industrial, and vehicular networks as typical scenarios that may have multiple and heterogeneous data sources and face obstacles concerning connectivity. As a consequence, deep learning solutions can contribute to system performance by adding intelligence and the ability to interpret data. We start the paper by providing an overview of deep learning, further explaining this approach's benefits over the cited scenarios. We propose a workflow based on our observations of deep learning applications over challenged networks, and based on it, we strive to survey the literature on deep-learning-based solutions at an application-oriented level using the PRISMA methodology. Afterward, we also discuss new deep learning techniques that show enormous potential for further improvements as well as transversal issues, such as security. Finally, we provide lessons learned raising trends linking all surveyed papers to deep learning approaches. We are confident that the proposed paper contributes to the state of the art and can be a piece of inspiration for beginners and also for enthusiasts on advanced networking research.

1. Introduction

The ever-increasing complexity of computational systems is associated with the vertiginous growth in the volume of data available to recent services. This phenomenon could not be any different in computer networking, where more and more applications dealing with different requirements, scenarios, data sources, etc. continuously show up (Kibria et al., 2018; Mathebula et al., 2019; Liu et al., 2020). The abundance of data, even though tempting, may impose a challenge to the design of classical networking algorithms proportional to the number of input parameters (Zhang et al., 2019). In the Internet of Things (IoT), for instance, the huge volume of heterogeneous data produced by multiple sources may impose scalability issues (Ray et al., 2016; Du et al., 2019; Makhdoom et al., 2019). Similarly, in wireless networking, the scenario dynamics introduce broadcast storms of control messages, which also have an impact on network connectivity (Khan et al., 2020; Chen et al., 2019). Such connectivity is of utmost importance to industrial scenarios, where timely responses are a critical factor to avoid accidents (Aceto et al., 2019b). Lastly, in vehicular networking, anticipating the occurrence and the duration of contacts may represent a turning point in the efficient dissemination of warning messages to security applications (Wang et al., 2019). All these traditional challenges can be summarized in five main axes that can affect or be affected by communication conditions and also by the volume of data available, as illustrated in Fig. 1. Hence, in this paper, we borrow the concept of "*challenged network*" to refer to networks that take advantage of multiple data sources to intelligently improve performance over limited

* Corresponding author.
E-mail addresses: kaylani@gta.ufrj.br (K. Bochie), gilbert@gta.ufrj.br (M.S. Gilbert), gantert@gta.ufrj.br (L. Gantert), maciel@gta.ufrj.br (M.S.M. Barbosa), diannescherly@id.uff.br (D.S.V. Medeiros), miguel@gta.ufrj.br (M.E.M. Campista).

https://doi.org/10.1016/j.jnca.2021.103213
Received 2 May 2021; Received in revised form 4 August 2021; Accepted 29 August 2021
Available online 10 September 2021
1084-8045/© 2021 Elsevier Ltd. All rights reserved.

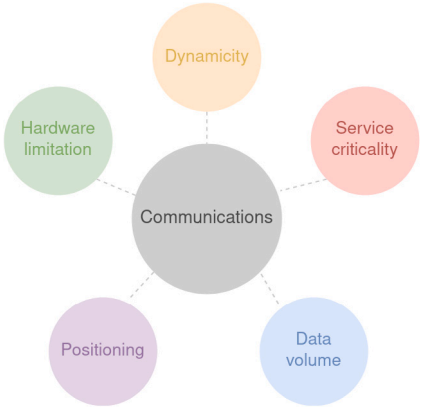

Fig. 1. Main challenges tackled by networks that could affect or be affected by communication issues. These challenges are commonly seen altogether or differently combined in particular networks, herein called *challenged networks*.

communication conditions. Note that the name challenged networks was initially employed for high delay environments and scenarios with poor or nonexistent infrastructure (Anon, 2019). In this paper, we add to this concept the notion of intelligence as an alternative to traditional challenged networks, which handle the lack of connectivity without learning from previous experiences (Cao and Sun, 2013; Silva et al., 2017; Baron et al., 2019).

Adding intelligence to challenged networks is a fertile ground for Machine Learning (ML). Even though this research area has been around for at least a few decades (Osherson et al., 1991), it was not fully deployed in the past as it requires intense computational power. This scenario, however, has changed more recently, driven by the continuous evolution of computer hardware and software (Wason, 2018). As a consequence, we are now experiencing an unprecedented hype on machine learning, which appears as an alternative to traditional rule-based algorithms. This hype is expected since machine learning techniques can "understand" complex problems, an ability that must be achieved to enable fast response times and low sensitivity to variations. To reach this point, machine learning algorithms obtain a mathematical representation capable of modeling the behavior of a function through a process called training. By training from samples usually with multiple features, these algorithms adjust their parameters to predict a new set of samples, autonomously perform a task, or extract relevant information. The data used for training may contain targets that the model must predict as the output. The same idea can be applied to networking, where nodes must collect data from the different available sources and adjust parameters to improve performance over challenging scenarios.

Traditional machine learning performance, *i.e.*, the performance of algorithms that do not leverage neural networks during their training crucially depends on feature selection (Mao et al., 2018). This issue, along with the increasing complexity of problems found in challenged networks, is currently motivating a revisit to Deep Learning (DL). Efforts in this area date back to the forties, passing through three different popularity peaks referenced by other names: "cybernetics" between the forties and sixties, "connectionism" between the eighties and the nineties, and finally, "deep learning" starting around the year 2000 (Goodfellow et al., 2016). As computational power is no longer the main problem to machine learning, we are now experiencing growth also in the utilization of deep learning techniques and algorithms. Deep learning introduces more complex mathematical models that improve feature selection and, hence, achievable results. Moreover, through multiple processing layers, deep learning becomes able to capture nonlinearities from problems with extra levels of complexity.

1.1. Existing surveys and related work

Machine learning, and more specifically, deep learning, is a trending topic in contemporaneous research. As a consequence, many surveys are tackling the application of this approach to many facets of computer networking.

Pundir and Sandhu review perceived user QoS (Quality of Service) on wireless sensor networks (Pundir and Sandhu, 2021). The authors cover a wide range of different metrics that affect user QoS, such as security, network throughput, and reliability in the context of wireless sensor networks. Hussain et al. focus on machine learning applications on IoT security, presenting solutions and current challenges (Hussain et al., 2020). The authors review multiple IoT architectures alongside a bottom-up analysis of attack vectors to the IoT stack. They also discuss the reasons behind machine learning becoming an interesting solution for problems found in IoT scenarios, pointing out the main constraints found in IoT networks. Mao et al. also discuss IoT applications but, unlike Hussain et al. they introduce a broader discussion in the context of wireless networking (Mao et al., 2018). Their work covers deep-learning-based solutions applied to different layers of the network stack, going from a general discussion regarding machine learning to more advanced topics such as the signal interference at the physical layer and enhancements to typical Medium Access Control (MAC) designs. Al-Garadi et al. also cover IoT networks by offering a detailed taxonomy analysis of the different machine learning methods applied to individual IoT scenarios (Al-Garadi et al., 2020).

Other surveys discuss the interpolation of reinforcement learning concepts aligned with deep neural networks. Lei et al. for instance, analyze the capability of Deep Reinforcement Learning (DRL) models applied to autonomous IoT (Lei et al., 2020). Finally, Tahsien et al. examine IoT networks under the light of machine-learning-based applications on network security (Tahsien et al., 2020). Luong et al. review applications regarding DRL for communication, focusing on scenarios from IoT and vehicular networks (Luong et al., 2019). Chen et al. review the recent literature to emphasize machine learning applications on wireless networks (Chen et al., 2019). They provide detailed explanations on Artificial Neural Network (ANN) applications in scenarios ranging from IoT to 5G networks. The authors also describe in great detail the foundations of machine learning, commonly used artificial neural networks, such as Recurrent Neural Networks (RNNs), and less commonly used architectures, such as Spiking Neural Networks (SNNs). We highlight that Lim et al. are the only authors that produced a survey exclusively dedicated to federated learning (Lim et al., 2020). The authors focus on the edge and, more specifically, mobile applications. They review the most popular frameworks, along with multiple applications and their concerns, such as latency, bandwidth consumption, and, more importantly, privacy. Focusing on mobile and wireless networks, Zhang et al. provide an extensive review of deep learning fundamentals with respect to network applications (Zhang et al., 2019). The authors cover extensive and recent topics, ranging from application-level and network-level mobile data analysis to network security, signal processing, and emerging applications in mobile networks. Other surveys review the literature to discuss machine learning and deep learning applications in mobile and wireless networks (Nguyen et al., 2020; Shi et al., 2020; Wang et al., 2020). Tong et al. focus on machine learning applications on vehicular networks (Tong et al., 2019). The authors review Vehicle-to-Everything (V2X) systems and machine learning applications on some domains, such as traffic estimation and network congestion control.

Challenged networks, however, are plural and have different main concerns that could be tackled by different deep learning strategies. Fig. 2 shows the main challenges from those identified in Fig. 1 regarding each type of network addressed in this paper. Note that some challenges are common to almost all networks, *e.g.*, node positioning, whereas others are particular to a single one, *e.g.*, hardware limitation. As far as we could observe, all the surveys in the literature produced

Table 1
Qualitative comparison of our main contributions and the contributions of existing surveys on machine learning and deep learning applications in challenged networks.

Survey	Year	Survey topics	Related sections in this paper	Enhancements in this paper
Pundir and Sandhu (2021)	2021	QoS on wireless sensor networks	Section 3.1	Analysis of application design and implementation in the context of sensor networks
Nguyen et al. (2020)	2020	Deep learning on future wireless networks	Section 3.2	Review of user QoE prediction techniques and a comparison between other challenged networks
Shi et al. (2020)	2020	Deep learning on edge computing	Section 3.2	Review of centralized and cloud-based solutions
Al-Garadi et al. (2020)	2020	Deep learning for IoT and sensor networks	Section 3.1	Detailed review of device-free localization, human activity recognition, and resource management
Lei et al. (2020)	2020	Deep reinforcement learning for IoT and sensor networks	Section 3.1	Survey considering different challenged networks including industrial and wireless networks
Lim et al. (2020)	2020	Mobile edge networks and federated learning	Sections 3.2 and 4.2	Broader coverage of deep learning techniques that may apply federated learning schemes
Wang et al. (2020)	2020	Machine learning on wireless networks	Section 3.2	Broader revision of recent surveys and detailed coverage of QoE prediction on mobile networks with deep learning
Hussain et al. (2020)	2020	Security on challenged networks and IoT	Sections 4.1 and 3.1	Security regarding industrial and other challenged networks
Tahsien et al. (2020)	2020	Security on IoT networks	Sections 4.1 and 3.1	Review of IoT applications outside the security aspect
Chen et al. (2019)	2019	Neural networks on wireless networks	Section 3.2	Review of CNN-based solutions
Luong et al. (2019)	2019	Deep reinforcement learning on networking	Sections 3.2, 3.1, and 3.4	Broader review grouped by network type and coverage of industrial networks
Zhang et al. (2019)	2019	Deep learning on wireless networks	Section 3.2	Analysis of network slicing using deep learning tools
Tong et al. (2019)	2019	Machine learning on vehicular networks	Section 3.4	Broader coverage of deep learning techniques applied in V2X communication and autonomous vehicles
Mao et al. (2018)	2018	Intelligent wireless networks	Section 3.2	Overview of traffic analysis and solution comparison to other challenged networks
Mohammadi et al. (2018)	2018	Deep learning for IoT	Section 3.1	Update with more recent papers, data aggregation and compression schemes

so far focus on a particular network, mainly on IoT, and therefore do not provide a broader picture of deep learning applications on different challenged networks, as we summarize in Table 1. This narrower view may lead researchers to overlook possible lessons learned from a particular network that could be also applied to similar cases from other networks. We also invite readers to take a look at other surveys that were not included in Table 1 for the sake of conciseness (Qian et al., 2020; Verbraeken et al., 2020; Waheed et al., 2020; Duc et al., 2019; Chen et al., 2020b). We also note that the challenges were "assigned" to each network based on their natural environments and node configurations. For example, service criticality is an issue for industrial networks, which are usually deployed on assembly lines and power plants (Li et al., 2018b), in contrast to wireless mobile networks which are more popular for applications such as content streaming and QoE prediction (Bhattacharyya et al., 2019).

Our goal in this paper is to provide an application-oriented analysis, which spans multiple types of computer networks and, by doing so, to highlight current trends in deep learning solutions. Thus, we are not limited by networking issues in the strict sense, such as packet forwarding and routing proposals, but on the entire environment that nodes from challenged networks participate. This paper does not propose an exhaustive revision of the literature regarding each network type. Nevertheless, it is written to be a "first stop" when researching deep-learning-based solutions for challenged networks.

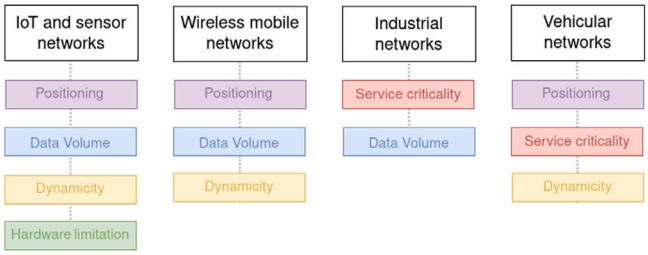

Fig. 2. Main challenges considered per challenged networks. We observe that some challenges are commonly seen on most networks, whereas others only appear on a single network. Independent of the challenge generality, at different levels, they can affect the proposed solutions.

1.2. Survey scope and contributions

The main contribution of this survey is to introduce readers to the flourishing area of deep learning as a promising approach for dealing with all complexities behind the herein called challenged networks. The application of deep learning in multiple situations became an unquestioning trend in the last few years and, consequently, an open venue for research and development. The idea of this survey is to unveil research directions, showing how and why deep learning has been employed on challenged networks. Yet, we provide enough background

Table 2
List of acronyms in alphabetical order.

Acronym	Meaning	Acronym	Meaning
AAL	Ambient Assisted Living	MIMO	Multiple-Input and Multiple-Output
AE	Autoencoder	ML	Machine Learning
AGI	Aggregated Global Information	MLP	Multilayer Perceptron
ANN	Artificial Neural Network	MRE	Mean Relative Error
AP	Access Point	MSE	Mean Squared Error
AutoML	Automated Machine Learning	NN	Neural Network
BS	Base Station	NS	Network Slicing
CASAS	Center for Advanced Studies in Advanced Systems	PCA	Principal Component Analysis
CDL	Collaborative Deep Learning	PILAE	Pseudoinverse Learning Autoencoder
CNN	Convolutional Neural Network	PR	PageRank
CPD	Canonical Polyadic Decomposition	PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analysis
CS	Compressive Sensing	PS	Parameter Server
CSI	Channel State Information	QoE	Quality of Experience
DAE	Denoising Autoencoder	QoS	Quality of Service
DBN	Deep Belief Network	RBF	Radial Basis Function
DBSCAN	Density-Based Spatial Clustering of Applications with Noise	ReLU	Rectified Linear Unit
DCGAN	Deep Convolutional Generative Adversarial Network	RNN	Recurrent Neural Network
DCT	Discrete Cosine Transform	ROC	Receiver Operating Characteristic
DFT	Discrete Fourier Transform	RR	Round-Robin
DfP	Device-free Passive	RSS	Received Signal Strength
DL	Deep Learning	RSUs	Road Station Units
DNN	Deep Neural Network	RUL	Remaining Useful Life
DOA	Direction-Of-Arrival	SAE	Stacked Autoencoder
DQN	Deep Q-Network	SBL	Sparse Bayesian Learning
DRL	Deep Reinforcement Learning	SDAE	Stacked Denoising Autoencoder
DT	Decision Tree	SGD	Stochastic Gradient Descent
E2E	End-to-End	SI	Strategy Iteration
FFT	Fast Fourier Transform	SI	Safety Index
FNN	Feedforward Neural Network	SLA	Service Level Agreement
GNN	Graph Neural Network	SLR	Service Level Requirement
GPS	Global Positioning System	SNN	Spiking Neural Network
HAR	Human Activity Recognition	SNR	Signal-to-Noise Ratio
IDS	Intrusion Detection System	SOTL	Self-Organizing Traffic Light
IIoT	Industrial Internet of Things	SPINN	Synergistic Progressive Inference of Neural Networks
IoCVs	Internet of Connected Vehicles	STVV	Spatial–Temporal Video Volume
IoT	Internet of Things	SUMO	Simulation of Urban Mobility
ITS	Intelligent Transportation System	SVM	Support Vector Machine
KNN	K-Nearest Neighbors	TT	Transmission Time
KPI	Key Performance Indicator	UAV	Unmanned Aerial Vehicle
LMR	Lost Message Ratio	V2I	Vehicle-to-Infrastructure
LSTM	Long Short-Term Memory	V2V	Vehicle-to-Vehicle
LTC	Lightweight Temporal Compression	V2X	Vehicle-to-Everything
MAC	Medium Access Control	VANET	Vehicular Adhoc NETwork
MAE	Mean Absolute Error	WAE	Weight Decaying Autoencoder
MARDDPG	Multi-Agent Recurrent Deep Deterministic Policy Gradient Algorithm	XAI	Explainable Artificial Intelligence
MDP	Markov Decision Process		

for enthusiasts so as they can become familiar with basic deep learning concepts. The key contributions of this paper are six-fold, as listed below:

- This paper provides a summarized overview of the main deep learning techniques and provides a clear and broad deep learning workflow, based on the observations made during the systematic revision of deep learning solutions;
- This paper presents an extensive survey explaining how and why deep learning has been applied to challenged networks. We separate the challenged networks into wireless mobile networks, IoT and sensor networks, industrial networks, and vehicular networks seeking to maintain their main characteristics;
- This paper provides a section to discuss recent deep-learning-based solutions that can be applied to multiple challenged networks with topics such as security, model partitioning, and federated learning;
- This paper reveals trends regarding deep learning application in challenged networks as a lesson learned, taking into account a table summarizing all surveyed papers;
- Finally, this paper stimulates the discussion about possible research topics and open issues in deep learning application in challenged networks.

1.3. Survey methodology

We apply the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) methodology to systematically select the papers used in this survey (Liberati et al., 2009). Firstly, based on previous experience, we select four main challenged networks, *i.e.*, wireless mobile networks, IoT and sensor networks, industrial networks, and vehicular networks. The systematic review based on the PRISMA methodology consists of four steps: (i) identification, (ii) screening, (iii) eligibility, and (iv) inclusion. In the first step, we identify relevant

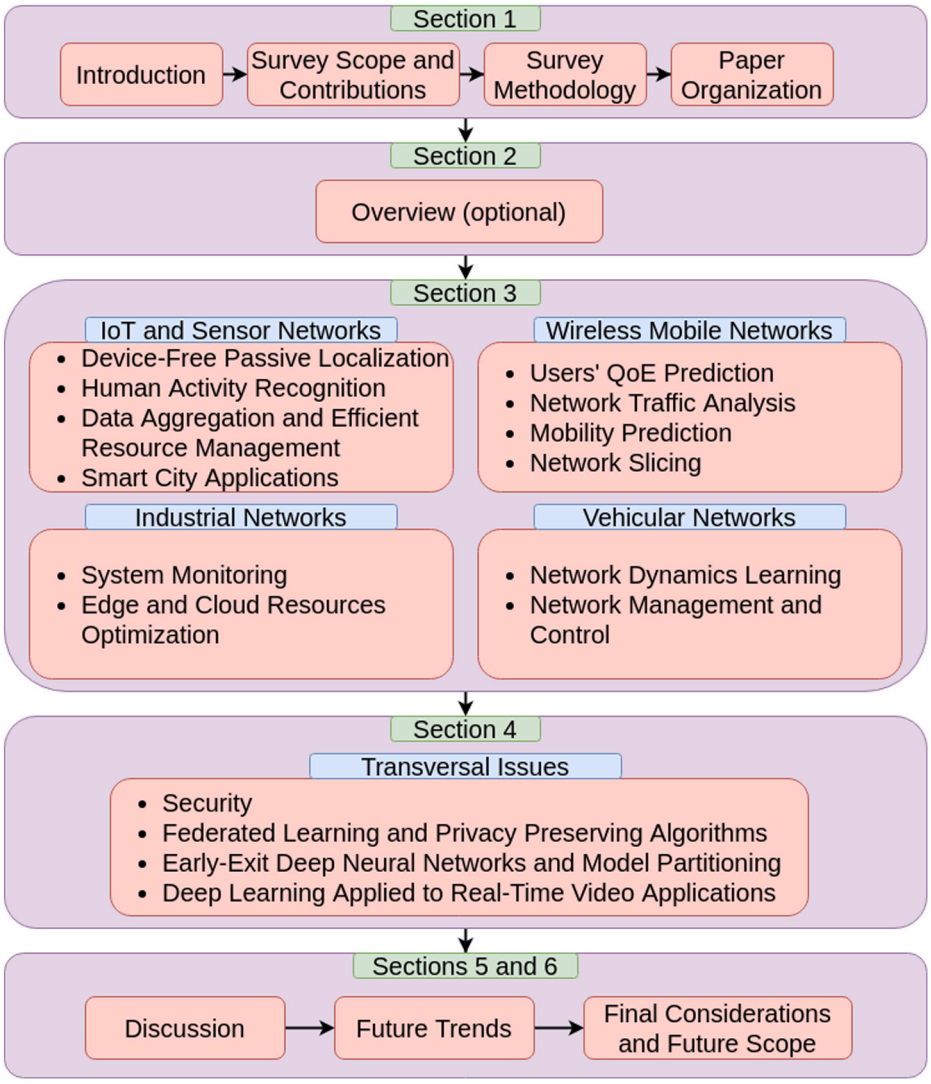

Fig. 3. Block diagram representing the flow of this survey.

documents to the scope of this paper through an initial search in a set of consolidated databases, mainly those from IEEE, ACM, and Elsevier. We conduct database searches by querying the terms "machine learning" in parallel for each challenged network, limiting the search to the last five years, except for related surveys, on which the search was limited to the last three years. This step resulted in hundreds of papers, which were then filtered in the screening step. The screening step includes the elimination of duplicates and the examination of each paper's relevance for our survey. Hence, we analyze each paper's suitability based on its title and abstract. Papers whose scope did not include "machine learning" and "computer networking" were filtered out. Another criterion for survey selection was to only review papers that are focused on at least one challenged network. In the eligibility step, we read the full text of the pre-selected papers to determine which of them are eligible for our final review. Here, we chose papers that were able to implement their proposal before going through evaluation. Nevertheless, we also include papers with simulated systems, if their results are well explored, explained, and justified. Papers that did not meet these criteria were removed. The last step consists of the inclusion of the selected papers in a database used for the qualitative analysis herein conducted.

To facilitate future reference, we provide a list of acronyms in Table 2.

1.4. Paper organization

The remainder of this paper is organized as follows: Section 2 overviews the needed background in deep learning. Section 3 goes through the main deep learning applications in challenged networks, *i.e.*, wireless mobile networks, IoT and sensor networks, industrial networks, and vehicular networks. Section 4 discusses the issues and future research directions shared by all the different challenged networks. Section 5 discusses the lessons learned regarding deep learning applications in challenged networks, whereas Section 6 concludes this paper and introduces open research issues. The diagram in Fig. 3 presents the paper organization to facilitate the visualization of each section's content and the navigation throughout the paper.

2. Deep learning overview

This section provides a summarized overview of the most common neural network architectures. However, for the sake of conciseness, we refer to previous works that explored machine learning fundamentals and deep learning concepts.

Machine learning emerges as a paradigm where algorithms can extract information from the available data without being explicitly programmed (LeCun et al., 2015; Abadi et al., 2016a; He et al., 2016; Krizhevsky et al., 2017). As a consequence, it is possible to reduce

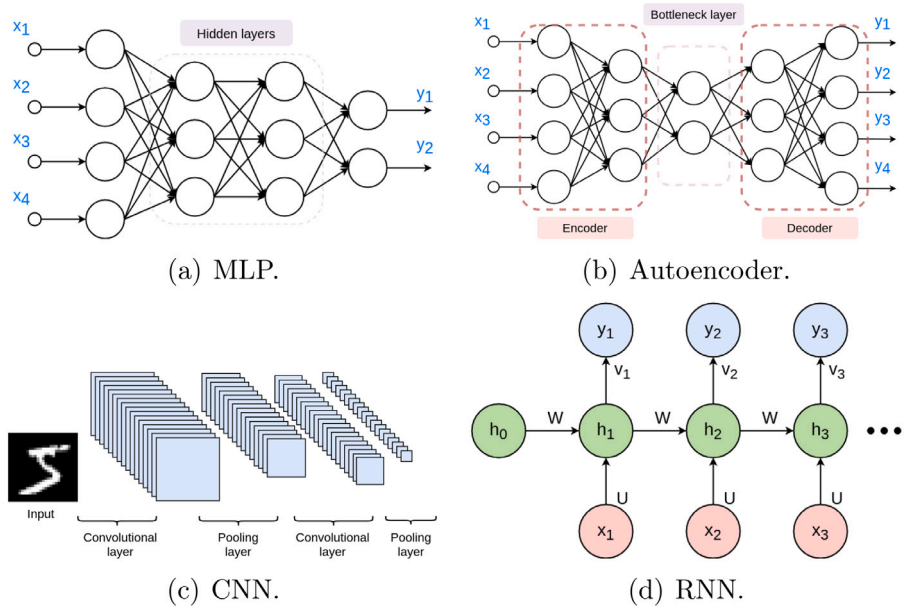

Fig. 4. (a) Example of a Feedforward Neural Network (FNN), namely an MLP with fully connected layers. (b) The autoencoder is also a type of FNN, but with a hidden bottleneck layer. (c) A CNN is a type of FNN as well, but it is characterized by its convolutional layers. The convolutional layers of a CNN are usually followed by a fully connected layer. (d) An RNN is defined by its ability to maintain hidden states, represented by the green circles.

Table 3
Overview of machine learning paradigms.

Paradigm	Main characteristics	Common applications
Supervised learning	Labeled data	Classification and regression
Unsupervised learning	Unlabeled data	Clustering and pattern identification
Reinforcement learning	Based on actions and rewards	Autonomous vehicles and general optimization tasks

Table 4
Overview of deep neural network architectures.

Architecture	Main characteristics	Common applications
MLP	Fully connected neurons	General purpose
CNN	Sparse connections	Images and time-series data
AE	Output reproduces the input	Recognition tasks and anomaly detection
RNN	Recurrent connections (memory)	Time-series data

the need for previous knowledge, which makes machine learning an appealing alternative for problems involving large volumes of data, possibly with impractical formalization (Deisenroth et al., 2019). Traditional programming, in such cases, is considered too costly or even impossible (Goodfellow et al., 2016). This is due to the need for an expert level analysis of each individual problem. Since machine learning fundamentals are consolidated in the literature, we summarize the main learning paradigms in Table 3, and the main neural network architectures in Fig. 4 and Table 4. These concepts are tackled in previous works, such as (LeCun et al., 1995, 2015; Goodfellow et al., 2016).

Although this paper focuses on deep learning applications on challenged networks, there are scenarios where deep-learning-based solutions do not fit well. While some issues impact specific DNN architectures (Stabinger et al., 2021), low data availability can hinder DNN performance. This is a critical issue, especially given the required human effort to properly configure data collection and processing mechanisms. Explainability is a common requirement for some applications and the "black box", *i.e.*, low interpretability, nature of deep learning models makes them less suitable for some applications, such as healthcare (Dave et al., 2020; Amann et al., 2020). It is worth noting that there has been an increase in efforts to enable explainable deep-learning-based applications, following the paradigm of Explainable Artificial Intelligence (XAI) (Pope et al., 2019; Angelov and Soares, 2020).

3. Deep learning applied to challenged networks

There exist a plethora of deep-learning-based applications, but they usually follow the same typical pipeline. This pipeline usually follows five steps: (i) data preparation, (ii) feature engineering, (iii) model selection, (iv) hyperparameter tuning, and (v) model selection under constraints and deployment. To facilitate the development of machine learning applications, Zhang et al. propose a framework for Automated Machine Learning (AutoML) (Zhang et al., 2019). AutoML tries to minimize the level of expertise required to implement machine learning solutions (Hutter et al., 2019; Stamoulis et al., 2020). AutoML automates every step of the pipeline needed to build a particular solution. Failure detection can also be automated. Although very important for machine learning, the workflow proposed by Zhang et al. focuses exclusively on AutoML and only covers a particular set of design principles, such as automated data preparation and model selection. Wang et al. however, propose a dedicated workflow for computer networking (Wang et al., 2018). We build upon this idea to propose a simpler, yet broader workflow, based on typical steps taken by authors

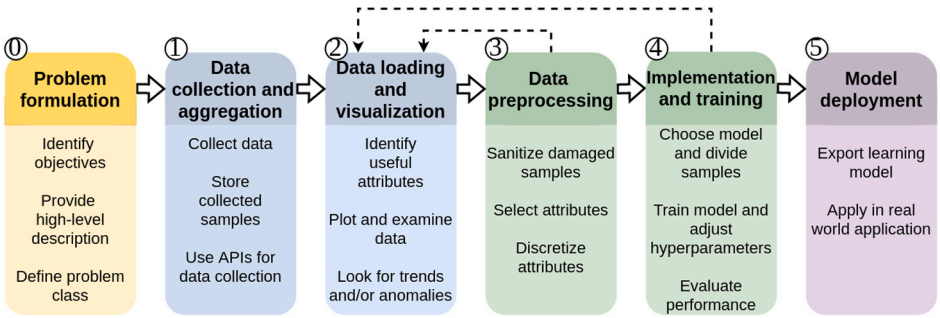

Fig. 5. Typical workflow for machine learning application in challenged networks. Step 0 is done even before choosing deep learning as a possible tool. Steps 1 and 2 fall under the umbrella of data mining and data science. Steps 3 and 4 are more intimately linked to machine learning, while Step 5 is extremely application-dependent. Step 5 can be set at cloud servers, forwarding requests to a predictor, deploying pieces of a bigger model to multiple IoT devices, or simply aggregating data and using it for a single inference task.

when solving challenged networking issues. Our proposed workflow for machine learning application, depicted in Fig. 5, is a baseline approach followed by most papers surveyed in the next section (De La Torre Parra et al., 2020; Wang et al., 2016b; Pierucci and Micheli, 2016; Bhattacharyya et al., 2019; Luo et al., 2020; Dou et al., 2020; Wu et al., 2020b; Ning et al., 2020; Al-Hawawreh et al., 2018; Teerapittayanon et al., 2016). Note that particular constraints imposed by a challenged network may incur subtle variations. These variations will be discussed in the corresponding sections to come.

0. **Problem formulation:** it is required to formally define the problem before choosing an appropriate deep learning solution, or even if one is needed in the first place. For example, when trying to implement solutions for end-to-end congestion control, many heuristic solutions may be applied before using deep learning (Zhang and Mao, 2020). Nevertheless, if the problem seems approachable by a deep learning solution, this step must define the problem's constraints and guidelines, such as the metric that must be used for performance evaluation, the type of learning paradigm that could be used, or if an algorithm must be chosen for classification, clustering, or regression.
1. **Data collection and aggregation:** data is collected from multiple sources, like IoT systems, online forms, public websites, among others. This data must be representative and preferentially without bias. In addition, the data can be collected from offline or online sources. All the collected data is put together in a dataset for later use on training, validation, and test procedures.
2. **Data loading and visualization:** tools for graphic generation and statistical analysis are used to perform a high-level analysis of the dataset. Along these lines, it is possible to identify anomalies and inconsistencies in the data, besides patterns and trends.
3. **Data preprocessing:** after identifying anomalies and inconsistencies, it is possible to parse and preprocess the dataset used as input of the machine learning algorithms. Data preprocessing is crucial for the model to achieve the required performance and, therefore, is subdivided into three further steps. The tools used in the data visualization step are paramount to data preprocessing, and these two steps are usually performed together. This is depicted by the feedback loop from step 3 to step 2 in Fig. 5.
    - **Damaged sample handling, normalization, and attribute formatting:** faulty collection processes usually lead to missing data. Values at different scales are adjusted to achieve better performance from some learning models. Furthermore, the attributes must be appropriately set for the selected algorithm. For example, categorical attributes must be converted for algorithms that require numerical values as input.
    - **Attribute selection and dimensionality reduction:** statistical tools allow identifying the most useful attributes to predict each sample's target. For example, to predict a student's score in an exam, it is much more appropriate to use the number of study hours instead of the student's height. Principal Component Analysis (PCA) is a common algorithm to perform dimensionality reduction, transforming attributes in a way that only those with less projection correlation are indicated for the following steps. It is also necessary to analyze the correlation between attributes to perform dimensionality reduction, which further reduces training times and improves the model performance. It is worth mentioning that the same data transformation must be applied to the training, validation, and test sets.
    - **Discretization (optional):** data can be discretized to make it suitable for specific machine learning algorithms. The discretization process can be performed numerically or categorically, depending almost entirely on the chosen algorithm. Common tactics include binning values using a constant interval and mapping attributes to logarithmic space beforehand.
4. **Implementation and training:** the task to be performed defines the optimal model. Choosing the model must consider, as defined in step 0, whether the problem is supervised or unsupervised, whether it is a classification or regression problem, which devices are accessible for training and deployment, etc. Although previous steps provide directions regarding possible algorithms to be used in this step, the actual implementation and algorithm selection take place in this step.
    - **Training:** the dataset is divided between training, validation, and test sets. After adjusting hyperparameters, which can be defined by common heuristics or selected by an exhaustive grid search in a hyperparameter space, the machine learning model can be trained using the training set and evaluated on the validation set. If the obtained performance is not satisfactory, the hyperparameters can be readjusted, and a new round of training can begin (Glorot and Bengio, 2010).
    - **Performance evaluation:** the test set can be presented to the model to quantitatively evaluate its performance according to a metric of interest, such as classification problems' accuracy. That is, the model is applied to unseen data to guarantee that the model can consistently achieve the desired performance.
5. **Model deployment:** in the last stage, with the trained model and its performance already evaluated, it is possible to deploy the model in a practical application, be it for load forecasting, failure rate calculation, or others. Exporting the model means

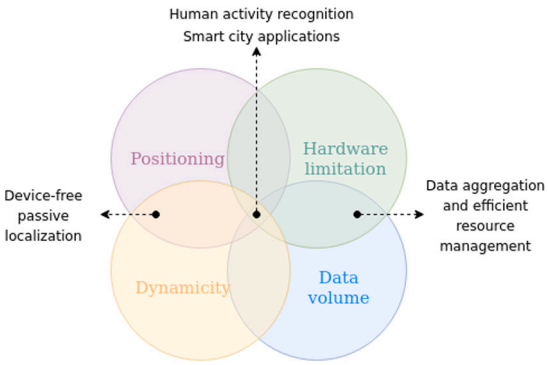

Fig. 6. Challenges in IoT and sensor networks and main deep learning research trends.

that the weights of a neural network or the coefficients of a logistical regressor, for example, are well defined, and the model can be used on unseen data in production conditions. In the event of a case study, the model can have its performance appraised in the test set for publication.

In the following sections, this survey reviews papers that apply the described workflow to challenged networks. We followed the PRISMA methodology to systematically select the surveyed papers, as described in Section 1.3.

3.1. IoT and sensor networks

A typical IoT network comprises sensors and actuators that can communicate with each other and with the Internet (Gubbi et al., 2013). Usually, such devices have low economic and energetic costs. Hence, many traditional concepts present in Wireless Sensor Networks (WSNs) are important for developing applications also in the IoT scenario. Besides, devices with embedded sensors, such as smartphones and smartwatches, are also part of typical IoT settings, offering multiple customized applications that explore the interaction with other devices (Kaur, 2018; Maskelinas et al., 2019; Sheng et al., 2015). However, the resource constraints of such devices pose an extra obstacle for systems and algorithms implementation, especially the deep-learning-based ones presented here, as they must be designed to not overload the IoT network. For instance, if one were to implement a compression scheme that was too complex, one might nullify the advantages of reducing the amount of data to be transmitted as such a system would require significant energy consumption.

A relevant characteristic of the data generated by IoT devices is heterogeneity, as it is common to deploy different sensors at the same node (Mohammadi et al., 2018). Thus, the quality of the data generated by multiple devices interconnected via an underlying network is critical for data processing. Furthermore, the network size and geographical arrangement of the participating devices suggest an existing correlation between samples from closer regions. This characteristic may be considered using data aggregation techniques to, for instance, reduce the overall number of network transmissions and consequently enhance the network lifespan.

The particular characteristics of IoT devices and the massive volume of heterogeneous data produced are fertile ground for deep learning (De La Torre Parra et al., 2020). Nevertheless, as pointed out by Ma et al. deep learning deployment in IoT applications is still at an initial stage (Ma et al., 2019). Fig. 6 shows relevant research areas that encompass many topics in IoT, namely, device-free localization, human activity recognition, smart city applications, and data aggregation and, efficient resource management. Firstly, we visit important areas for smart systems and application development, device-free localization, and human activity recognition, which are primarily concerned with users' well-being. Given that these areas are primarily concerned with real-time data analysis, approaches that handle quick changes are paramount. Moreover, sensor positioning is pivotal to effectively handle these applications. The next trend includes an overview of the smart city paradigm to demonstrate the diversity of applications that can benefit from deep learning deployment. As this is a very broad area, all the challenges faced in IoT are present here. Regarding the last trend, we present contributions regarding network lifespan extension, which aim to save energy by reducing the volume of transmitted data.

Device-free passive localization

Finding human and object positions is essential for the development of many applications in IoT systems (Zafari et al., 2019). Many traditional techniques, such as those that use Global Positioning System (GPS) data, require that the object being monitored carries a device to enable localization. Also, such techniques usually require that the monitored object actively participates in the localization process (Youssef et al., 2007). Since such conditions may be inconvenient for some tasks, either due to device size or other structural constraints, an alternative is to execute passive monitoring. In this type of monitoring, the desired task is done by measuring interactions between the object and the environment.

Along the lines of passive monitoring, one relevant research direction in IoT, inherited from WSNs, is Device-free Passive (DfP) localization. This type of localization does not require the active participation of the monitored object in the process. For instance, an IoT network can be arranged in a scenario to perform localization, exploring the fact that certain radio-frequency properties are modified due to changes to the environment (Youssef et al., 2007). Therefore, implementing a system capable of identifying such variations is an attractive alternative for many applications and systems that require localization.

Wang et al. develop a DfP system using a sparse autoencoder to identify changes in the environment and, consequently, determine the position of individuals within the monitored environment and the activity and gestures they are performing (Wang et al., 2016b). To accomplish that, one must extract relevant characteristics from the network to enable such functionalities. Fig. 7 shows a typical configuration of a DfP system in a monitored environment. The employed sensors form a network where each node is capable of communicating with the others. The autoencoder is trained to compress the signal and then extract relevant features by measuring the Received Signal Strength (RSS) between each node. After training, the decoding portion of the network is dropped, and the bottleneck layer becomes the new output layer. To perform classification, this new output layer is connected to a new layer that functions as the network output. This process is illustrated in Fig. 8. The addition of this last layer requires a new training step to tune the network parameters. The proposed approach differs from more traditional ones, as it automatically adjusts parameters and allows simultaneous identification of activities and gestures. Nevertheless, gesture classification is less successful.

Similarly to the previous work, Zhao et al. (2019) explore received signal strength. Nevertheless, the authors take an approach that combines two different neural network architectures to interpret signal strength variations. They use a convolutional autoencoder to explore the advantages of convolutional networks in image processing and the unsupervised learning capabilities of autoencoders. The proposed system collects the sensed signal variations, storing those values in a matrix that represents each state of the network links. For that, each node transmits signals to every other remaining node. These nodes, in turn, compare the received signal with the corresponding free-link measurements, *i.e.*, the RSS value without object obstruction.

Referring to Fig. 7 again, each sensor is connected to all other nodes. All sensors perform measurements and contribute to the RSS matrix. The resulting RSS matrix is then converted to an image, mapping the stored values into pixels. This transformation allows exploring

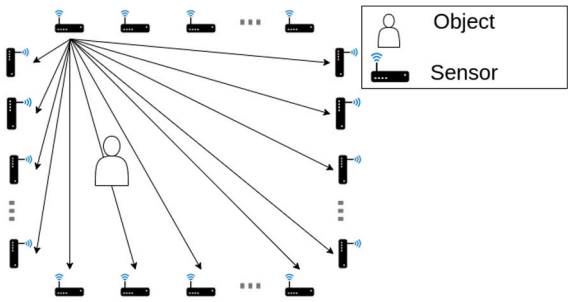

Fig. 7. An example of a typical passive localization system that uses RSS, adapted from Wang et al. (2016b), Zhao et al. (2019). Each sensor measures the RSS between itself and the other sensors. The RSS between each pair of sensors is modified depending on the object's position or its movement within a monitored scenario, for example, someone performing an activity.

characteristics of the convolutional part of the network. Then, the network is pre-trained with those images, exploring the "autoencoding" structure of the network. It is important to point out that the decoding part of the network is composed of layers that undo the convolutional and max pooling operations performed on the encoding part. After the pre-training phase, the decoding layers are dropped and replaced by a new layer, similarly to Wang et al.'s approach discussed in the previous work (Wang et al., 2016b). After the addition of this new layer, a new training phase is executed to tune the network parameters.

The convolutional autoencoder is compared with convolutional networks and autoencoders, which are already employed in DfP. For SNR values ranging from −10 dB up to 15 dB, the proposed approach achieves better accuracy in determining object localization in all studied scenarios. Another remark is the short processing time of the network when employed in a monitoring task, taking a few milliseconds. The good performance makes the approach attractive for online localization. Furthermore, sixteen sensors were required to obtain adequate accuracy when the network was employed in a scenario with 0 dB SNR.

In another work, Gochoo et al. seek to implement a non-invasive monitoring system to identify patterns indicating dementia (Gochoo et al., 2017). To this end, a deep convolutional network is used to analyze images corresponding to the movement map of monitored individuals. Each image is generated from passive infrared sensor measurements. These samples are organized into episodes which are then converted into images. This approach presents better performance than other traditional machine learning techniques, such as KNN (K-Nearest Neighbors), random forest, Naïve Bayes, among others.

Clearly, most of the recent projects lean on image processing through a transformation of the grid-like architecture of the DfP system. Such an approach implies greater adoption of convolutional networks that are more suited for DfP localization applications. Also, as sensors collect many unnecessary features, the use of autoencoders to perform dimensionality reduction can aid the localization process.

Human activity recognition

Many applications, especially of medical and security interests, focus on Human Activity Recognition (HAR). These activities can be identified through sensors that are external to the monitored individual, *i.e.*, sensors that individuals do not carry with them (Wang et al., 2016b; Gochoo et al., 2017); or, alternatively, through sensors embedded in devices carried by the monitored individual (Lara and Labrador, 2012), such as those present in smartphones and smartwatches. The IoT paradigm offers even more complex applications related to HAR, such as Smart Healthcare (Ma et al., 2019), in which patients' exams are performed without human intervention.

Ronao and Cho implement a system using a convolutional neural network that handles data generated by sensors embedded in smartphones (Ronao and Cho, 2016). The idea of using this network comes from two observations. Human activities can be broken down into few simpler actions and different individuals perform the same activity in different ways, leading to translations and other perturbations in the collected data. The system uses measurements collected from the accelerometer and gyroscope already present in all modern cellphones, which further avoids adding new sensors to users. During hyperparameter adjustment, the authors' proposed model improves the performance with the addition of at most 3 hidden layers. Another factor influencing the neural network execution is the convolutional filter size, where sizes equivalent to values within the 0.18 to 0.28 s interval improve performance. On the other hand, the authors observe that the pooling size barely impacts the overall performance. The training is conducted using data collected from volunteers, who executed a set of activities carrying smartphones. The proposed system outperforms other techniques, such as using hand-designed features combined with neural networks (Anguita et al., 2012) and Support Vector Machines (SVMs) (Noble, 2006), used to solve the same problem. The proposal benefit becomes clear for the classification of very similar activities that previous works considered hard to solve, such as distinguishing if a person is walking up or down a flight of stairs. When comparing the proposed system with SVMs, already in use in HAR problems, the former also has superior performance when classifying stationary activities.

Similar to Ronao and Cho, Ravi et al. develop a system for human activity classification from data collected from accelerometers and gyroscopes embedded on smartphones (Ravi et al., 2016a). Unlike the former, Ravi et al. focus on the development of a system able to balance performance and resource availability. Hence, the system must provide the best performance for activity classification without overwhelming the resource-constrained smartphone. In previous work, Ravi et al. observe that classification complexity is reduced when producing a spectrogram of the input data before passing it on to the neural network (Ravi et al., 2016b). Some advantages observed when applying this technique are time and sample rate invariability. Another important exploitable feature is that highly variable activities exhibit higher spectrogram values for various frequencies, whereas

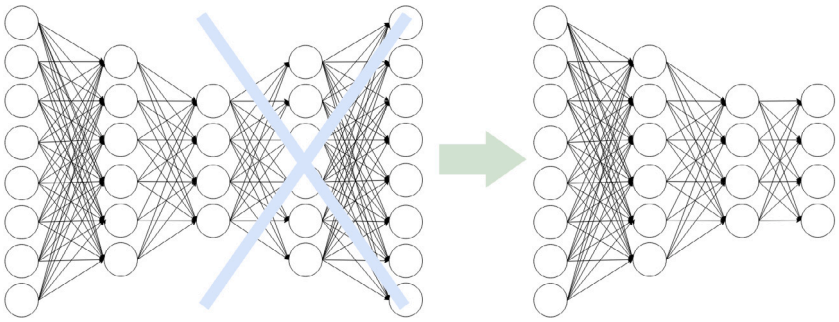

Fig. 8. Picture illustrating the training process used by Wang et al. (2016b). The decoding layer is dropped and a new layer will function as the combined network's output. Although the inserted layer has the same size as the bottleneck layer, this is not always the case.

activities with repetitive actions have only high values in specific frequencies. The authors choose a feedforward neural network with limited neuron connections, similar to CNNs. Another project decision is to limit the number of hidden layers, allowing the exploration of hierarchical constructions of data that do not increase the algorithm complexity. Training is remotely executed, and the adjusted network is downloaded into the user's smartphone so that the classification may be performed on the device. Results show that the proposed model has superior accuracy than previous work, besides outperforming other systems employed to solve the same problem (Ravi et al., 2016a; Alsheikh et al., 2015; Catal et al., 2015). Another relevant point is the short computational time of the system, which enables real-time human activity recognition.

Hammerla et al. investigate the use of different neural networks when recognizing activities using wearables or sensors fixed to the user (Hammerla et al., 2016). The goal is to gather information about pros and cons of each approach, as few projects detail how the networks and their parameters are determined. Usually, such projects provide only the performance of the best system. The networks are tested with three datasets. The first dataset consists of measurements collected from individuals performing everyday kitchen activities. The second has records of individuals carrying out predetermined activities in varied order so that the trained neural networks can classify them. Lastly, the third dataset is composed of measurements from patients with Parkinson's disease performing certain activities. These activities are known to trigger a common disease problem, where the individual has difficulties initiating certain movements. In all three scenarios, the main objective is to classify particular situations. The results show that LSTM networks outperform CNNs when identifying short-term activities, whereas CNNs have superior performance when classifying long and repetitive activities, such as running or walking. The networks present different performances according to the input parameters. It is observed, however, that feedforward neural networks show the most significant performance variation. This finding requires further investigation of network hyperparameters to produce satisfying results, especially compared to other networks, such as CNNs and RNNs. Among all evaluated neural networks, the bidirectional LSTM has a superior performance with the kitchen activities dataset by a considerable margin. Nevertheless, the number of neurons per layer influences the performance of this network.

Ambient Assisted Living (AAL) is another healthcare application offered in the IoT paradigm. AAL can benefit from HAR, as the primary goal of the application is to identify abnormal events performed by elders. In this context, Bianchi et al. focus on the development of a 24-hour patient monitoring system (Bianchi et al., 2019). The system uses sensors in wearable devices that transfer the sensed data to a neural network in a remote server so that the computationally demanding task can be done remotely. Such a server could be either a local unit or a cloud server. The latter is usually needed when new users arrive at the system, given that a new training phase is required. The authors compare CNNs and LSTM networks in three scenarios. In the first one, samples are divided randomly, splitting 60% for training and the remainder for testing. In the second scenario, the samples are partitioned by users, meaning that readings from a user present in the training set are not used in the testing phase. Lastly, in the third scenario, samples from all individuals are present in both sets. The results found show that CNNs outperform LSTM networks, especially in the third scenario, where the best results are found. Because of that, the authors suggest that a new training phase to tune the network parameters should be performed each time a new user arrives so that the network can better handle their measurements. Compared to traditionally employed methods, the CNN shows compatible results, confirming the proposed system feasibility.

Given the temporal nature of human activities, solutions based on recurrent neural networks, especially LSTM, seem to be the trend. Also, given the structured nature of some repetitive activities, the use of CNNs is observed to be beneficial due to the suitability of those networks to structured data.

Smart city applications

We discussed one of the motivations behind IoT deployments: building smart environments to improve citizens' lives. In this sense, it is natural that the smart city paradigm greatly benefits from advances in IoT, helping governments, city planners, and citizens to tackle plenty of problems such as energy management, video-based surveillance, traffic management, and pollution control (Zanella et al., 2014; Mehmood et al., 2017). For instance, Cenedese et al. handle a joint work with Padova municipality in Italy to implement a system capable of monitoring street lighting (Cenedese et al., 2014). Given the diversity of sensors, other devices that may comprise the network, and the inherent complexity of the desired applications, we observe an emerging effort under the smart city umbrella using deep learning. The idea is to enable efficient data processing to, consequently, provide responsive applications.

Kong et al. propose a short-term residential load forecasting system using LSTM networks (Kong et al., 2017). Load forecasting aims to assist power system operations by providing the electrical power needed for a given task, predicted according to users' actions. Load forecasting tends to be more challenging at residences because of the volatile nature of home power consumption. The authors show that LSTM networks are suited for this case, given the LSTM property of extracting temporal correlations in a data sequence. The results found in the paper show that LSTM networks are suited for the task. The proposal outperforms other methods, *e.g.*, empirical mean absolute percentage error minimization, based on statistical energy consumption distribution (Mousavi and Abyaneh, 2010), in most evaluated cases, especially when single-user load forecasting is studied. In particular, when compared with a conventional feedforward network, the proposed LSTM network follows peaks in consumption better than its simpler counterpart, even though the feedforward version also presents satisfactory results.

Quick response emergencies are of paramount importance for every city. In this direction, many deep-learning-based applications may help develop autonomous systems to help first response agents. Singh and Mohan develop a monitoring system to detect road accidents using a Stacked Denoising Autoencoder (SDAE) (Singh and Mohan, 2018). The SDAE is a regularly stacked autoencoder where each layer comes from an individual Denoising Autoencoder (DAE). From small units of videos called Spatial–Temporal Video Volumes (STVVs) (Li et al., 2014), three distinct SDAEs are trained to extract representations from still frames, motion from the optical flow, and the data generated with the fusion of both. For each STVV, an accident score is computed, determined from a linear combination of each reconstruction error and outlier detection value. This outlier detection value is generated from an intermediate representation in a lower dimension of the innermost hidden layer, which is fed onto a one-class SVM. The results found show that the proposed approach is viable, even though the authors faced issues when comparing their proposal with other approaches, given the scarcity of public datasets. As a secondary contribution, the authors make public the database they have assembled for the project, allowing researchers to proceed with comparative analysis.

Using a deep CNN architecture, Khan et al. develop a smoke detection system capable of identifying early potential signs of fire, even in foggy weather (Khan et al., 2019). Foggy weather is particularly challenging, as this phenomenon decreases image quality, which may lead to false positives or other more serious consequences. The authors use a VGG16 (Simonyan and Zisserman, 2015) network pre-trained on the ImageNet dataset (Deng et al., 2009), modifying it to fit a more uncertain IoT scenario. The choice for this particular CNN architecture is given by the fact that this model achieved the best results when compared to AlexNet (Krizhevsky et al., 2012) and GoogleNet (Szegedy et al., 2015). Additionally, when compared to state of the art smoke detection methods, *e.g.*, Dimitropoulos et al. (2016), the proposed method is more suited for uncertain scenarios and is also more energy-efficient.

Another source of concern in city administration is food supply. In this direction, incorporating new technologies and automated systems in agriculture is currently being researched. Mahmoud et al. propose a crop disease identification model that combines Deep Convolutional Generative Adversarial Network (DCGAN) with MLP (Mahmoud et al., 2020). In summary, the generative adversarial framework consists of training two networks together, in an adversarial manner. Details regarding DCGAN can be found in Radford et al. (2015). Additionally, the MLP is trained using a Pseudoinverse Learning Autoencoder (PILAE) (Wang et al., 2017a) algorithm. They have identified that using this approach rather than back propagation was more suited for the problem, as PILAE does not have the shortcomings of gradient vanishing or activation saturation (Mahmoud et al., 2020).

The approach proposed by Mahmoud et al. tackles two main problems. One is unbalanced datasets and the cost of building one for deep learning training, as datasets for plant classification usually have subsampled classes and require human labeling. By using the DCGAN, they were capable of balancing the dataset used for training, as the generative portion of this approach generates reliable samples in an unsupervised manner. Moreover, the DCGAN extracted relevant features that were used in the training of the PILAE, which was the portion responsible for the classification. This approach was capable of reaching high accuracy rates.

A major source of concern when developing a smart city application is latency and user privacy. Given the usual complexity of deep neural network architectures, both requirements can add complexity to the development of deep-learning-based applications, as keeping these networks in a cloud may require raw data to be transferred to a remote entity. With that in mind, White and Clark investigate how to combine the tools from deep learning with edge computing, thus moving the required computation closer to the end-user (White and Clarke, 2018, 2020). Edge computation avoids transferring all the data to a cloud server, even for training, which is desirable for private activities, where users do not feel comfortable sharing personal data. The proposed architecture, known as deep edges, is feasible thanks to recent advances in hardware, such as the Nvidia Jetson Tx2, combined with transfer learning techniques. This combination enables an existing network already trained for a given task, *e.g.*, a very deep convolutional network (Simonyan and Zisserman, 2015) trained for image classification, to be locally tuned at the network edge for a specific task. This significantly reduces the required computational power, allowing part of the training to be performed away from the cloud, for example. The first analysis shows that training using the Jetson Tx2 is not as slow as in other devices used in IoT, given its embedded GPUs, confirming the system's viability (White and Clarke, 2018). Later, in the second analysis, the use of transfer learning and data augmentation, which consists of artificially generating new training samples, further consolidates the proposed architecture viability (White and Clarke, 2020). Another approach that can achieve latency reduction is to bring neural networks to the devices, as this removes or reduces the need for forwarding data to the cloud or other nodes. Methods that reduce the burden of neural network implementation, which could be used for in-device deployment, are discussed in the next subsection.

Another approach is the use of Collaborative Deep Learning (CDL), where a group of nodes performing a similar deep-learning-enabled application can conduct the learning process collaboratively. The training phase can be done locally using the data available at the node. Then, computed gradients are sent to a centralized entity, which aggregates them into a Parameter Server (PS) and returns the resulting weights to nodes so as they can continue the learning process individually (Gupta et al., 2020). Gupta et al. highlight that such an approach may be ineffective if several nodes adopt an uncooperative strategy. Moreover, such a scenario is more likely to occur given the communication cost between nodes and the parameter server. If a node tries to do CDL alone in a scenario where no other node adopts a collaborative strategy, this single collaborative node will not get the benefits of CDL and will also adopt a selfish strategy to save energy. This generates a vicious cycle where no node cooperates, even if nodes were cooperative at the beginning. To overcome this uncooperative scenario, the authors propose a cluster-based strategy, using k-means clustering. The proposal enforces a node to adopt a collaborative strategy if another nearby node is identified. By doing so, all nodes that compose a cluster adopt a cooperative strategy, which guarantees that the desired CDL approach is employed. Despite the different name, this approach is very closely related to federated learning, discussed in more detail in Section 4.2.

Given the variety of problems that may arise in the smart cities paradigm, no clear trend on network selection, which was the case in the previous sections, is observable. Nevertheless, one relevant trend concerns the training process targeted to personal and home applications. In this scenario, training can be more challenging as fewer data may be available, or the user may have privacy concerns. To overcome this lack of data, the emergence of proposals that use federated learning and new devices capable of bringing training closer to the user are happening, such as the last approaches presented in this section.

Data aggregation and efficient resource management

The resource constraints of IoT devices raise concerns about their efficiency, especially about energy consumption during data transmission. Therefore, reducing the number of transmissions and the size of the transmitted data is vital to extend the lifespan of sensors and, as a consequence, of the entire IoT network. Data aggregation and fusion techniques explore inherent features of the data to combine them, reducing the volume of data needed to be transmitted. Furthermore, given the large volume of generated data and its high heterogeneity, it is common to combine the aggregation and fusion areas with data mining. The goal is to extract important information that may help the task being executed, enhancing the overall system performance. Nevertheless, the increase in lifespan and performance is tied to the cost–benefit of the proposed scheme used to compress or aggregate the transmitted data. As mentioned in this section's introduction, one must always bear in mind that the scheme to be implemented must satisfy the devices constraints, given that, if it were too complex, it could harm the IoT network rather than be beneficial.

Alsheikh et al. investigate the performance of three distinct autoencoders to compress measurements collected by network sensors. The authors aim to propose a computationally inexpensive system for adaptive data compression (Alsheikh et al., 2016). The first alternative is a simple undercomplete autoencoder, which uses the network structure to obtain a more compact representation of the inserted data. A variation of an autoencoder, named by the authors as Weight Decaying Autoencoder (WAE), is also studied. In this network, a regularization term is added to the cost function to penalize solutions that generate encoding and decoding matrices with high weights. Lastly, the third network is a sparse autoencoder. The work analyzes scenarios with different compression rates, exploring temporal and spatial correlations found in the data. The network is trained from historical data that is sent to a base station. This base station is responsible for computing the weight matrices for data encoding and decoding, which is respectively needed for data compression and decompression. On a purely spatial compression scenario, the proposed networks outperform conventional techniques used in sensor networks, such as Principal Components Analysis (PCA), Discrete Fourier Transform (DFT), and Fast Fourier Transform (FFT), especially when using low compression rates. In a temporal scenario, the proposed networks again outperform the Lightweight Temporal Compression (LTC) (Schoellhammer et al., 2004), a technique traditionally employed for this type of compression. The proposed networks achieve better performance particularly with low compression rates, as the case with spatial compression. When comparing the three proposed AEs, it is observable that the simple undercomplete AE performs better than the remaining two AEs when

reconstructing the compressed data. The authors attribute this finding to the neural network structure, mentioning that the addition of regularizers in the other two AEs harmed the overall reconstruction performance.

In another work employing AE networks, Ghosh and Grolinger have the objective of developing a data compressing system to reduce the volume of data being transferred in an application that requires support from the cloud (Ghosh and Grolinger, 2019). The authors studied a HAR application, which requires that the data collected in smartphones be transferred to the cloud, where this data is processed. Thus, as discussed before, data compression before transmission is important to extend the lifespan of smartphones. The authors concentrate more effort on the compression impact on the overall performance of the classification system hosted in the cloud, which is responsible for carrying out the HAR application. Undercomplete AE architectures with different depths and compression rates are analyzed. Also, the authors investigate if decompressing the data is needed. This is because the encoding portion of the AE produces a lower-dimensional representation of the data, performing dimensionality reduction, which may aid classification. The results show that the AEs are capable of reducing the volume of data to be transferred without harming classification. Another interesting observation is that the non-decoding approach, *i.e.*, which happens when classification is performed on the encoded data, slightly outperforms its decoding counterpart. Compared with PCA, the authors were unable to conclude that AEs always have a better performance, as similar results are found after classification.

Li et al. employ a denoising AE to obtain a sparse representation of the collected samples in an automated manner, and then reduce the volume of transmitted data (Li et al., 2018a). The authors point out that, usually, the transform basis to sparse out the sensed data is chosen empirically. Such an approach, besides being laborious, sometimes fails to satisfactorily handle the desired task. Another point of concern is that sensor nodes closer to the network sink tend to run out of energy more quickly because, besides transmitting their own data, they also forward the data coming from more distant nodes. Li et al. introduce a data collection algorithm based on DAE in a hybrid and cooperative manner to be run locally in each sensor. On the one hand, sensor nodes that are far away from the network sink can transmit their own produced data as well as forward the received data in an uncompressed manner. On the other hand, nodes closer to the sink only transmit compressed data. To do so, all untreated data, *e.g.*, the raw samples generated by the sensors, are compressed prior to transmission. Also, for these sensor nodes, the expected action for already compressed data is simply forwarding. The DAEs are trained offline using historical data from the task of interest. In the studied scenario, the task of interest is a surveillance task. When compared with a Discrete Cosine Transform (DCT) basis and a non-compressing scheme, the authors found that the proposed scheme achieves the best results for reconstruction accuracy and energy consumption, even achieving a short reconstruction time that further corroborates the system viability.

Yu et al. use an Unmanned Aerial Vehicle (UAV) as a processing unit of the IoT network (Yu et al., 2018), employing a denoising autoencoder for compression. Fig. 9 illustrates the system. To explore the available spatial features from the collected samples, neighboring sensors are clustered together in a set according to the k-means algorithm. Then a DAE network is deployed for each resulting set. Training is performed in the cloud, from previously collected samples, and the computed weight matrices for encoding and decoding are stored in the UAV and the cloud, respectively. The UAV flies over the monitored area, collecting sensed data from each set and compressing them before sending it to the cloud. Lastly, with the decoding matrix, the cloud recovers the collected data. The obtained results show that the proposed system outperforms methods based on Compressive Sensing (CS) to which it is compared. In particular, the difference in performance is more evident for low sample rates.

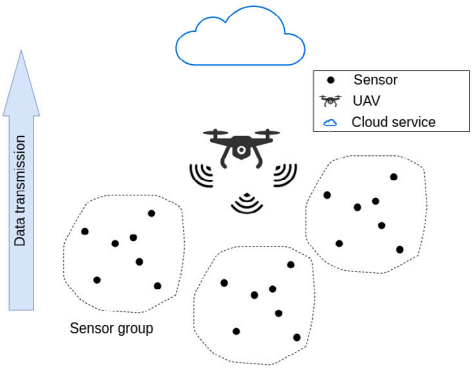

Fig. 9. Aggregating system that uses a UAV to collect data from a monitored area and compress data using DAEs with the support of a cloud service, adapted from Yu et al. (2018). The UAV flies over the sensors, collecting data. The compression and decompression processes occur in the UAV and in the cloud server, respectively.

In another work, Wang et al. adopt a convolutional neural network to fuse sensed data in order to solve a common sensing problem, known as blind drift calibration (Wang et al., 2017b). When a sensing network is working for a long period of time, it may suffer from the accumulation of small drifts in measurements, leading the sensing system to malfunction. Hence, the aim of blind drift calibration is to neutralize such drifts without a reference. This lack of reference occurs as a consequence of the nonexistence of a prior model and also the low-density deployment in certain regions, which hinders the direct use of neighboring sensors as a reference. Then, the CNN aims to extract temporal and spatial correlations in the collected data to remove these drifts. The network structure is organized so as the first layer is responsible for projecting all drifted measurements to a feature space, whilst the remaining layers are responsible for fusing the data to remove the drifts. Given that the convolutional kernel size is limited, it is desirable that correlated data be placed together in the data matrix. Wang et al. observe the need for rearranging the collected samples putting the data from neighboring sensors in adjacent regions of the matrix. The network is trained using the previously collected data, considering that these were collected from calibrated sensors, *i.e.*, from sensors with the desired ability to remove drifts. The authors adopt the strategy of pre-training the network with data including small drifts, followed by fine-tuned data with larger drifts. When compared to other calibration methods, the proposed approach shows a higher reconstruction rate, outperforming them in all analyzed scenarios. Furthermore, in the scenario where less than 15 sensors suffer from drifts, the proposed system has a reconstruction error of successful recoveries greater than a Bayesian learning-based technique, used in a previous work (Wang et al., 2016a), but has a smaller reconstruction error with failed recoveries. Wang et al. additionally evaluate different drift occurrence models. The CNN has difficulties in finding small drifts, failing to obtain the same success in larger drifts. Nevertheless, convolutional networks show robustness to overfitting the drifts, as a consequence of the methodology used for training. Altogether, it is verified that the use of CNNs is adequate for blind drift calibration.

In another project, Zhu et al. implement a transmission scheduling mechanism using deep reinforcement learning (Zhu et al., 2017). Given bandwidth and storage space limitations, it is important that IoT network nodes quickly forward their data to prevent packet losses due to full buffers. The proposed system combines a stacked autoencoder with the Q-learning algorithm. The system development consists of two parts. In the first one, the system interacts with the environment using Q-learning (Watkins and Dayan, 1992) to generate the lookup table with optimal actions. The SAE is not used in the first part. After the table is complete, the SAE is trained to extract adequate actions for each situation, mapping each state into the correct action. When

found, the action is performed, the lookup table is updated, and the process is repeated for the new state. In the end, the resulting table will serve as the state–action table from which the relay will query. The proposed mechanism is compared with the Strategy Iteration (SI) algorithm, which needs to compute all possible transmission states and also uses Q-learning, and a random selection algorithm, which forwards packets without requiring any computation. The authors verify that the proposed system is outperformed by SI only. Nevertheless, compared with this algorithm, the proposal has a much lower computational cost.

In a neural network architecture, as the number of layers increases, the number of parameters to store and the amount of computation to be done also increases. Clearly, this is troublesome when implementing deep-learning-based systems in IoT scenarios, given the restrictions of typical IoT network elements. An approach to ease these implementations is the use of network compression, which aims to reduce the number of parameters without significantly degrading the network performance. One approach, presented by Han et al. (2015), applies densely connected architectures to identify weak connections, *i.e.*, connections with low magnitude weight, removing them as they usually do not contribute much to the final result. This allows for the construction of networks with sparser layers, resulting in fewer parameters to store and fewer computations to perform. When dealing with convolutional architectures, Li et al. (2016) propose a method for removing filters and feature maps which are identified as being of less importance. This approach is more suited for CNNs than Han et al.'s approach, as sparse layers may require sparse libraries and specialized hardware that may not be suited for CNNs. Overall, both approaches in these papers are capable of reducing neural networks parameters, which is ideal for IoT applications.

As expected, given its characteristics, autoencoders are commonplace for data aggregation problems. The use of these networks is almost mandatory when deep learning is selected as a consequence of the unsupervised nature for training or data dimensionality reduction. Depending on the nature of the analyzed data, combining autoencoders with other topologies is beneficial, as their inherent features may aid the desired process. For example, the joint utilization of autoencoders and convolutional networks is interesting to explore grid-like structures like the ones found on images. Lastly, the number of parameters and computations that accompany deep neural network architectures is a source of concern when developing deep learning based applications in IoT. To mitigate this impact, different techniques that reduce the size of these networks, without significantly harming application performance, are fundamental for deep-learning-based systems in IoT.

IoT and sensor networks overview

Table 5 presents the papers reviewed in this section. We present four common areas of research that can appear in many IoT and sensor network applications. More precisely, device-free passive localization, human activity recognition, different smart city applications, and different schemes concerned with efficient resource management and increasing the network lifetime are presented. The plethora of different applications and scenarios in IoT and sensor networks results in a very heterogeneous array of deep-learning-based solutions. Even though we cannot point out a major trend in IoT and sensor networks, as we identify in other challenged networks, there are clear and typical applications for different DNN architectures. It is observed a tendency to use autoencoder networks in IoT and sensor networks. This trend may be due to the heterogeneity of this data, which may need some sort of unsupervised learning to extract relevant features or simply the need for dimensionality reduction. In both cases, autoencoders are the go-to neural networks. Additionally, a somewhat expected tendency of using CNNs with applications that deal with images or other grid-like structured data. Given the geographical proximity between sensors, the data collected from this networks carry high spatial correlation that can be exploited by CNNs. Finally, when dealing with IoT and sensors networks, privacy and concerns with energy and computational cost come up often. The first one is expected to be dominated by federated learning, which is an area of research in deep learning that flourished recently. The latter can be addressed by a variate of approaches. It is expected that different approaches that reduce the burden of implementing networks in the devices, such as neural network compression, which was presented at the end of the data aggregation and efficient resource management subsections. Additionally, bringing layers or the whole neural network closer to the devices of the IoT network can address latency problems, which is another source of concern with these types of networks.

3.2. Wireless mobile networks

Wireless communications became very popular, given the computational power increase of mobile devices and the consequent convergence of different applications and services. This convergence, combined with mobility, has culminated in a noticeable growth of mobile devices' utilization and, under the networking perspective, exponential growth in the volume of generated data. According to Cisco, the total number of global mobile subscribers will represent 71% of the population by 2023, contrasting to the 66% we had in 2018 (Cisco, 2020). Thus, applying machine learning to wireless mobile networks allows handling challenging tasks, such as network resource management, real-time analysis, and big data support, in a way that users' experience can potentially be improved (Reis et al., 2020). Among previous works regarding deep learning applied to wireless mobile networks, it is important to highlight the utilization of network indicators to predict information about the network performance, such as Quality of Experience (QoE) (Grando et al., 2019; Pierucci and Micheli, 2016), and to predict the network behavior, such as traffic classification (Aceto et al., 2019a; Nguyen and Armitage, 2008). It is important to notice how wireless networks focus on QoE as an indicator, which represents a high-level measure of users' satisfaction. This notion differs from the previously discussed QoS concept, as it relies on users' subjectivity. QoS, on the other hand, uses quantitative indicators usually collected by particular software and hardware. Moreover, mobile network indicators have intrinsic space–time comprehension of the network dynamics. For instance, graph modeling allows the computation of each node's importance to the network operation by identifying the nodes playing central roles (Medeiros et al., 2016; de Medeiros et al., 2017). As a consequence, graph modeling can capture space–time relationships between nodes and be used as input of deep neural networks to predict network performance in dynamic scenarios (Wang et al., 2018).

This section examines relevant papers that apply deep learning to wireless mobile networks, both in cellular and in local Wi-Fi scenarios. These papers focus on QoE and data traffic analysis, which seek to enhance wireless mobile network management and performance. In addition, we investigate current problems in 5G mobile networks, such as mobility prediction and network slicing. In a general picture, we observe that the data growth combined with the mobile dynamics represent the main challenges in these networks, as depicted in Fig. 10. We then select four wireless mobile networking trends that apply deep-learning-based solutions: users' QoE prediction, network traffic analysis, mobility prediction, and network slicing.

Users' QoE prediction

The increasing need for Internet services using wireless networks leads users to look for providers offering high-quality QoE. Service providers then struggle to improve users' QoE by enhancing network maintenance and operation. The idea is to enhance QoE by keeping the QoS at acceptable levels. In this direction, neural network models are used to predict users' QoE in mobile networks based on Key Performance Indicators (KPIs), which are metrics used to infer the network performance, such as jitter, latency, and handover success rate (Pierucci

Table 5
Summary of applications and deep learning trends on IoT and sensor networks.

Paper and authors	High-level description	Deep learning method	Employed dataset
Wang et al. (2016b)	DfP Activity recognition Real-time analysis	Autoencoder	Proprietary
Zhao et al. (2019)	DfP Real-time analysis	Convolutional autoencoder	Proprietary
Gochoo et al. (2017)	Real-time analysis DfP Non-invasive monitoring	CNN	Provided by CASAS
Ronao and Cho (2016)	HAR with smartphones Real-time analysis	1D-CNN	Proprietary
Ravi et al. (2016a)	HAR with wearables Real-time analysis Edge computing	1D-CNN	ActiveMiles, WISDM, Shoka, and Daphnet
Hammerla et al. (2016)	HAR with wearables Real-time analysis	MLP, CNN, and LSTM	Opportunity (Opp), PAMAP2, and Daphnet Gait (DG)
Kong et al. (2017)	Load forecasting Real-time analysis	LSTM	Smart-Grid Smart City (SGSC)
Bianchi et al. (2019)	AAL Real-time analysis Remote server	1D-CNN and LSTM	Proprietary
Singh and Mohan (2018)	Traffic monitoring Real-time analysis Image processing	SAE	Proprietary, collected from CCTV footage
Khan et al. (2019)	Smoke detection Image processing	CNN	Mivia Fire Detection Dataset and proprietary
Mahmoud et al. (2020)	Smart agriculture Plant disease identification	MLP and CNN	PlantVillage, Swedish Leaf Dataset and Leafsnap
White and Clarke (White and Clarke, 2018, 2020)	Augmented reality DL with edge computing Transfer learning	CNN	Dogs vs. cats
Gupta et al. (2020)	Collective training Privacy Federated learning	Unspecified	ARAS human activity dataset
Alsheikh et al. (2016)	Data compression Data transfer Energy-saving	Autoencoder	Grand St Bernard
Ghosh and Grolinger (2019)	Data compression Data transfer Energy-saving Edge computing	Autoencoder	Human activity recognition using smartphones
Li et al. (2018a)	Data compression Data transfer Energy-saving	DAE	Intel Berkeley Research Lab WSN
Yu et al. (2018)	Data compression with UAVs Energy-saving	DAE	Unspecified
Wang et al. (2016a)	Blind drift calibration Real-time analysis and Auto-tuning	CNN	Proprietary
Zhu et al. (2017)	Transmission scheduling Energy-saving Data transmission	Deep Q-learning and SAE	Unspecified
Han et al. (2015)	Network compression	MLP and CNN	MNIST and ImageNet
Li et al. (2016)	Network compression	CNN	CIFAR-10 and ImageNet

and Micheli, 2016). Once QoS and QoE metrics are obtained, we can use them to guide the development of new network control and management mechanisms based on, for instance, deep reinforcement learning (Bhattacharyya et al., 2019). Users' QoE is subjectively measured through a group of voluntary participants. Nevertheless, some QoS characteristics measured from production networks have a strong relationship with users' QoE. These characteristics can then be used as additional parameters to users' QoE measurements.

The complexity of evaluating the effect of each QoS metric on the QoE is a consequence of the inability to process large volumes of data received from service providers. Thus, deep learning models become a valuable tool to estimate multiple QoS metrics, which can later be used to estimate users' QoE. In this context, Pierucci and Micheli analyze the database of a service provider in Italy to find out which QoS metrics have more impact on users' QoE (Pierucci and Micheli, 2016). The general idea is to use these metrics as the input of an MLP to improve users' QoE. Initially, the data volume and the data throughput

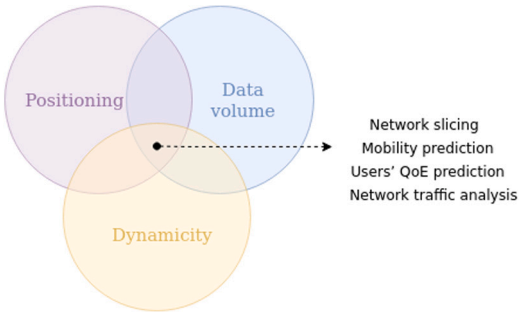

Fig. 10. Wireless mobile network challenges and main deep learning research trends.

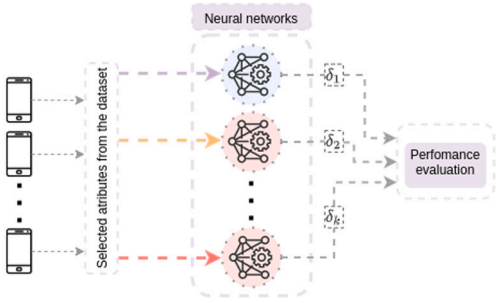

Fig. 11. Architecture for adjusting and comparing learning techniques, adapted from Aceto et al. (2019a). Each data extraction block selects a traffic feature and transforms it into input data for the deep learning model. This architecture implements multiple distinct techniques which are evaluated by predicting the network traffic generated by mobile device applications.

obtained by users are divided into four regions. Each one of these regions indicates a different QoE, which can be classified from bad to excellent. The authors use an MLP with two hidden layers. The network output indicates in which region each user's QoE resides. Their results show that the proposed model is capable of classifying users' QoE with high enough precision. By using a different approach, Bhattacharyya et al. focus on QoE and QoS metrics to improve video transmission for users (Bhattacharyya et al., 2019). To achieve that, the authors develop the QFlow platform, based on deep reinforcement learning, which injects packets into different priority queues of Access Points (APs). The authors consider an access point under saturated conditions, relaying data packets from a Youtube video. The YouTube application is considered because of its QoE requirements and its current popularity.

By using a control agent, QFlow sends new instructions to assign packets to AP priority queues. The QoE information is obtained from users' applications, whereas QoS information is directly obtained from the AP. The reinforcement learning agent uses the QoE as the basis to formulate its reward function. Also, the agent is designed using an MLP containing two hidden layers with 64 and 34 neurons, respectively. The authors show that the proposed approach can achieve higher QoE levels compared to different priority-queue-assignment techniques, such as vanilla and Round-Robin (RR) approaches (Balogh et al., 2010).

Network traffic analysis

The rapid growth of wireless network traffic leads to significant bottlenecks in the underlying communication medium. Hence, network traffic prediction helps resource allocation and, as a consequence, it becomes vital to achieve high network performance (Wang et al., 2018). Network traffic classification can play an essential role for some services such as intrusion detection, resource allocation, and network resource identification used by clients, just to cite a few examples (Nguyen and Armitage, 2008). Along with network traffic prediction, there is a great challenge in network resource management and allocation, which also affects users' QoE (Tang et al., 2017). In this context, deep learning arises as a solution to analyze large volumes of network traffic because of its capability to deal with large volumes of data. Moreover, traditional traffic analysis techniques present some difficulty with possible changes to the packet format, as well as to the protocol operation. These changes are mainly a consequence of new updates often released to commonly used mobile applications. Deep learning approaches introduce adaptive techniques that can circumvent such changes.

Aceto et al. evaluate different deep learning techniques, such as CNNs and LSTM networks, in traffic classification with different types of input data. The general idea is to enable future deep learning employment for traffic classification in mobile devices (Aceto et al., 2019a). The data collection is composed of mobile user's data traffic collected from multiple Android and iOS applications, such as Google Maps, EFood, Google Hangouts, and Crackle. Fig. 11 shows the architecture used to adjust and compare the selected learning techniques. After data collection and visualization, the input data is extracted through data pre-processing following different techniques. The output is then applied to many deep learning techniques for implementation and training, in which hyperparameters are adjusted. The system performance is improved by excluding classifications that fail to achieve minimum threshold values. The performance is computed after analyzing the technique validity for each type of input data. The first and the second characteristics extracted from the data are the first bytes from the packet payload and the first bytes of the entire packet. These characteristics serve as the input of a Stacked Autoencoder (SA) with five layers, one CNN with a single dimension (1D-CNN), a CNN with two dimensions (2D-CNN), and an LSTM network. The third extracted characteristic is based on header fields from the first packets in a packet group. These fields contain unencrypted payload and information such as source and destination ports and the TCP window size. This information is used as input data for a 2D-CNN, an LSTM network, and a hybrid model. The hybrid alternative models the output of a 2D-CNN as a matrix, then the matrix serves as the input of an LSTM network. The results show that the traffic classification performance for Android and iOS improves when the initial bytes from the payload are considered. The best performing architectures are the 1D-CNN and 2D-CNN. It is worth noting that the other techniques also present satisfying results and, therefore, that all techniques can perform traffic classification. Furthermore, the study shows that, despite the progress, additional effort is still needed to identify an architecture capable of serving all datasets with the same performance. As such, hybrid networks can be a promising solution.

Along the lines of traffic classification, the diversity of applications and users' behavior pose additional challenges, depicted in Fig. 10, to traffic prediction in wireless mobile networks. Users' mobility and social behavior also influence network traffic requirements (Wang et al., 2018). The mobility shows spatial dependence, whereas the social behavior predominantly shows temporal dependence, *e.g.*, users may have different network behavior on weekdays and weekends. Despite this dynamic nature and time variation, Graph Neural Networks (GNNs) are capable of predicting expected traffic in cellular networks (Defferrard et al., 2016; Khalil et al., 2017; Battaglia et al., 2018). In a GNN, the data preprocessing step models the dataset as a graph which is used as the input of a deep neural network. Wang et al. apply a GNN to a dataset composed of information captured by cellular towers from a city in China. This dataset contains important information about each data transmission from the cellular network, like the detection of new data transmissions, user and cellular tower identification related to that transmission, the application, and the type of device used (Wang et al., 2018). The users' identities are anonymized to ensure privacy. The information regarding data transmission captures the temporal aspect, while the information regarding tower identification captures the spatial aspect. The traffic between mobile devices and the nearest

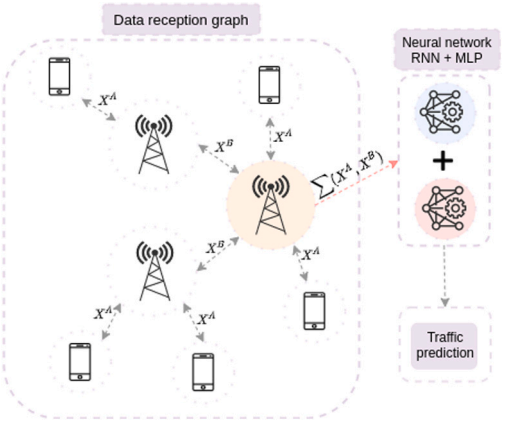

Fig. 12. Traffic prediction model for cellular base stations, adapted from Wang et al. (2018). The network traffic is modeled as a graph, which is composed of data traffic between the mobile device and the tower (X^A), and between towers (X^B). The traffic is used as the input of a GNN, which is composed by associating an RNN and an MLP.

tower is identified as traffic internal to the tower (X^A), while the traffic between towers is identified as traffic external to the tower (X^B). The traffic is modeled as a vector computed at every time interval t that corresponds to a downlink of thirty minutes. Fig. 12 illustrates the graph representation of the data reception in towers and the neural network implemented. Each tower contains the traffic generated from the devices directly connected to it. For the highlighted tower, the input data is X^A and X^B at each instant t. The neural network model adopted by Wang et al. is composed of an RNN and an MLP composed of two hidden layers each.

Wang et al. train the proposed model and verify its performance by measuring the Mean Absolute Error (MAE). The results show that the proposal achieves better performance when compared with different deep neural network architectures, such as an LSTM network. The impact of spatial dependence is verified by classifying towers using PageRank (PR) (Page et al., 1999), a centrality metric that considers both the number and the quality of a node's links and the number and the quality of its neighbors' links. The top three positions classified through PR are towers located in highly populated areas, like shopping centers and universities. Finally, the authors analyze the MAE throughout a day to capture the temporal dependence. The results show that the proposal achieves better performance when compared to other neural networks. The analysis also shows the importance of capturing spatial and temporal information related to the traffic to perform predictions. The use of GNNs standout as a trend in mobile networks for traffic classification, this is due to the aforementioned spatial and temporal information present in the data, which can also be explored with LSTM networks.

Mobility prediction

Mobility traffic analysis and prediction are valuable for network providers. By inspecting traffic flow in a network, network operators can perform dynamic decisions to optimize network performance and the user-perceived QoE (Xu et al., 2017; Akbari Torkestani, 2012; Zhang and Dai, 2019). Since mobile networks generate large volumes of dynamic data, traditional data processing techniques are ineffective and deep learning can be employed to help network providers in the decision-making process (Anagnostopoulos et al., 2011).

Deep learning helps mobile networks maintain nonstop service to their users, which is usually achieved by tracking the users' mobility and providing consistent connection without compromising the users' QoE. Orzturk et al. propose different cost models for both predictive and non-predictive handover management (Ozturk et al., 2019). The authors use an MLP and a deep-stacked LSTM network, with the latter outperforming the former, which is attributed to LSTM ability to explore and learn time-series data. The training and evaluating step is framed as a classification task, in which the deep learning model must determine the most probable next user position, given a timestamp and previously visited locations determined by the base stations. It is known that recurrent neural networks are best suited for time-series applications, such as mobility prediction (Zhang et al., 2020). Nevertheless, despite the better performance achieved by LSTM networks, it is notable that feedforward neural networks were outperformed by at most 5% accuracy in less-performing scenarios, and an MLP achieved comparable performance to LSTM networks in some scenarios. These results reinforce the power of finely-tuned feedforward neural networks when compared to hastily-designed and more complex models.

Network Slicing

Network Slicing (NS) has emerged to provide different sets of services to computer networks in which multiple users have different Service Level Requirements (SLRs), especially in the context of 5G networks (Zhang et al., 2017; Addad et al., 2020). This is achieved by three main components: (i) network functions, which are the building blocks for network slices; (ii) virtualization, which enables the abstraction of physical network functions through software; and (iii) orchestration, which is responsible for connecting and managing each network function in a network slice. Since 5G networks produce large volumes of data, deep learning can be used to deploy and maintain these virtual networks in shared physical infrastructure, a task that is even more challenging because of node roaming (Kafle et al., 2018; Toscano et al., 2019; Sun et al., 2020).

Assuming network slice availability, DeepSlice is an approach that aims to select the appropriate slice for a new device entering the network, analyze traffic to predict possible resource changes, and adapt the chosen slices in case of network failure (Thantharate et al., 2019). The authors perform these tasks by leveraging a CNN to consume commonly used KPIs and maximize these indicators. The system performance is verified by using multiple datasets, mainly from 5G and IoT networks. Three key scenarios are evaluated: slice selection for unknown devices, load balancing, and network slice failure. New and unknown devices are a common nuisance in 5G networks, and common heuristics for specific devices cannot be applied in this scenario. Load balancing is an implementation requirement when constructing and orchestrating virtual networks, which is the case of network slicing in 5G networks. Finally, the impact of network slice failure must be minimized. The approach chosen by the authors reflects what is most commonly seen in the literature, which is traffic redirection to a master slice. As a possible research subject, RNNs and their strong prediction power on time-series data can be explored to predict network slice failure. This prediction could be used to redirect network traffic in advance since ongoing data transmissions to or from network slices can be lost during reconfiguration.

Wang et al. leverage the deep reinforcement learning paradigm to optimize resource utilization in End-to-End (E2E) communication in 5G networks (Wang et al., 2019). The authors mention users' privacy as an issue. Since users' data can be used to profile individual behavior and that this tool cannot be used to maintain users' privacy, a DRL model can be implemented to interact with the network and dynamically adjust resources allocated to network slices. The goal is to maximize user's QoS and QoE. The DRL-based scheduler is used to achieve both high performance and fairness. DRL algorithms require a state representation and an action space. The state representation relies on the aggregation of information, such as resource requirements and each slice KPIs. In turn, the action space is obtained by constraining modifications to a percentage of the current state. The authors compare the proposal to greedy, random, and heuristic-based approaches. The system can minimize resource utilization and maximize performance while operating under different Service Level Agreements (SLAs).

Table 6
Summary of applications and deep learning trends on wireless mobile networks.

Paper and authors	High-level description	Deep learning method	Employed dataset
Pierucci and Micheli (2016)	QoE prediction	MLP and RBF network	Proprietary, provided by the Measurement Simtel Open Platform (SMOP)
Aceto et al. (2019a)	Traffic classification	MLP, SAE, 1D-CNN, 2D-CNN, and LSTM	Proprietary
Wang et al. (2018)	Cellular traffic prediction	GNN e LSTM	Proprietary, collected by a cellular carrier
Bhattacharyya et al. (2019)	QoE prediction of video streaming	Deep Q-learning and MLP	Proprietary
Tang et al. (2017)	Wireless traffic control	2D-CNN	Proprietary
Ozturk et al. (2019)	Handover prediction	MLP and LSTM	Based on social studies, developed by MIT Human Dynamic Lab
Zhang et al. (2020)	Traffic and mobility prediction	LSTM	Unspecified
Toscano et al. (2019)	Network slicing	LSTM	Proprietary, generated with NS-3
Thantharate et al. (2019)	Network slicing	CNN	DeepSlice dataset
Wang et al. (2019)	Network slicing	CNN and DRL	Proprietary

Wireless mobile networks overview

Table 6 presents the papers reviewed in this section. We present four common applications in wireless mobile networks, namely, users' QoE prediction, network traffic analysis, mobility prediction, and network slicing. Among the corresponding papers, we can identify the preferable use of recurrent neural networks, specifically LSTM networks. Therefore, the RNN designed for processing sequential data is able to cope well with the dynamic nature of the data due to users' mobility. Furthermore, the use of 1D-CNNs is a promising architecture because of its potential to explore sequential data, even though this type of data is currently more explored with RNNs.

3.3. Industrial networks

The fourth industrial revolution, or Industry 4.0, is the ongoing modernization process of manufacturing technologies. This modernization allows optimal and dynamic configuration of manufacturing processes to efficiently fit the global market (Zhong et al., 2017). To accomplish this, the fourth industrial generation strongly relies on cyber–physical systems, big data analytics, and new technologies, such as cloud and fog computing (Aceto et al., 2019b). All these technologies together are also the basis of the Industrial Internet of Things (IIoT), which can be defined as a particular case of IoT connecting industrial elements, such as machines and control systems to information systems and business processes. Thus, it is possible to monitor all the factory elements and make decisions by diagnostics or predictions based on large volumes of data permanently collected. All this industrial progress comes with challenges mainly correlated to plant monitoring. These challenges relate to the requirement of the network being always available, which is known as service criticality, and to the data volume generated by the information exchanged between network nodes to control and monitor industrial systems. Fig. 13 shows the main challenges in industrial networks, originated from the service criticality, represented by the right red circle, and the high volume of generated data, represented by the left blue circle.

Two research directions using deep learning arise from the requirements shown in Fig. 13: system monitoring, and edge and cloud resource optimization. The system monitoring research trend has the objective of obtaining current or future status from different industrial

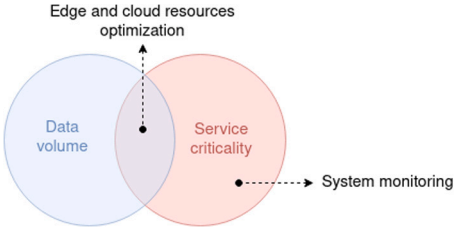

Fig. 13. Challenges in industrial networks and main deep learning research trends.

settings (Gantert et al., 2021). The idea is to eliminate or reduce unexpected interruptions. For instance, normal and abnormal conditions can be classified by DNNs, such as in Luo et al.'s approach (Luo et al., 2020). Li et al. in an alternative effort, adopt images provided by cameras from the production line to detect product defects (Li et al., 2018b). Also, LSTM networks enable Cheng et al. to check bearing status by vibration signals (Cheng et al., 2018), and Chen et al. to estimate machine's Remaining Useful Life (RUL), enabling scheduled maintenance (Chen et al., 2020a). The second trend includes works that deal with the efficient exchange of large volumes of data to and from edge and cloud nodes. This scenario requires resource optimization, which turns out to be a real problem that can be solved with deep-learning-based approaches. To determine how edge and cloud resources can be better allocated regarding latency requirements, Zeng. et al. use network partitioning (Zeng et al., 2019). Zhang et al. evaluate and predict cloud workload by industrial machines to guarantee optimal allocation (Zhang et al., 2018). Liang et al. in turn, prove that DNNs can mitigate congestion and decrease network traffic using edge nodes for training instead of the cloud (Liang et al., 2020). In addition, Dou et al. select key frames in video transmission and reduce resolution when the available network bandwidth is scarce (Dou et al., 2020). All these research directions are detailed next.

System monitoring

Systems for fault diagnosis, RUL estimation, and process monitoring allow optimal maintenance and avoid interruptions on industrial

plants. Luo et al. propose a DNN to detect normal and abnormal conditions during methanol distillation. The method achieves better performance on F1-score and AUC compared with CNNs, RNNs, and LSTMs (Luo et al., 2020).

Machine health supervising is possible by combining anomaly detection with methods of fault prognosis. Cheng et al. propose a method to check bearing status that starts with features in time-domain, frequency-domain, and time–frequency extracted from signal vibrations selected by Euclidean distance (Cheng et al., 2018). Then, a clustering algorithm is adopted for anomaly detection. An LSTM network is employed for failure prognostics by using the point where the anomaly was previously detected as its first input.

Prediction of machine RUL enables planned maintenance, reducing the cost compared with immediate maintenance (de Jonge et al., 2017). LSTM networks can be used for predicting machine RUL, but the importance of a feature and time steps are not learned. To solve this issue, Chen et al. propose an approach with two types of features: extracted from raw data and handcrafted (Chen et al., 2020a). Features extracted directly from raw data are used as inputs of an LSTM network, an attention layer, and a merge layer. Handcrafted features are used as inputs of a fully connected layer. The complete features are formed by the concatenation of both. Thus, a regression algorithm is applied for RUL prediction. Experimental results demonstrate better performance than MLP, SVR, Relevance Vector Machine (RVM), feedforward neural networks, and RF.

Focusing on monitoring the production line Li et al. develop a system capable of intelligently identifying and classifying defects by using information extracted from multiple cameras. The system seeks to operate in real-time, despite the large volume of data generated by the cameras. To achieve that, the authors use fog and cloud nodes to decrease the inference time. If the nodes at the fog perform the classification, it is not needed to send data back and forth to the cloud (Li et al., 2018b). Nevertheless, on the one hand, the computational power required is an obstacle to nodes at the fog with possibly limited resources. This can have a negative impact on high accuracy classification in all scenarios. On the other hand, leaving the entire task to cloud nodes can increase response time. To deal with this trade-off, the proposed system adopts offline results always using a pre-determined accuracy threshold. If the threshold is not achieved, results from the cloud are used.

The system proposed by Li et al. identifies product defects and measures their severity levels, classifying them as conforming or not to the factory policies. Fig. 14 shows the proposed system. Initially, the images captured on the production line are sent to local computers, *i.e.*, the fog nodes. Afterward, the data goes through two convolutional layers located at the local nodes as an attempt to identify and classify the product malfunction. The resulting data from the analysis performed at the local nodes are called intermediary results. In parallel, the results are sent to cloud servers, where they go through two additional convolutional layers and two fully connected layers, where it is possible to perform classification and estimate the severity level with a regressor. The results from the fog node are considered if the local approach is able to correctly analyze the camera footage. If this happens, the intermediate results are not sent to the cloud and the process can be considered finished. The system determines if the local approach is able to analyze the data by using a pre-determined threshold acting as a maximum for the cost function. This final result obtained without using cloud servers is called “early exit” since the system is quicker to perform defect evaluation. It is possible to adopt multiple early exits in the local approach, by sending the data in parallel not only to the cloud servers, but also to other convolutional layers.

One of the challenges faced by Li et al. is the development of a cost function that allows efficient and simultaneous training of the regressor and classifier. The proposed online cost function is the sum of three other functions and their respective weights with distinct goals: identify the defect through a softmax-based cost function, measure the defect

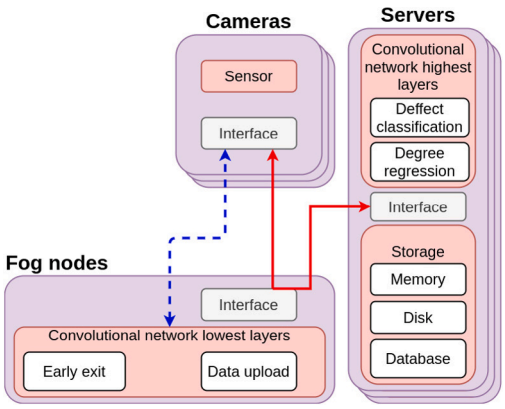

Fig. 14. System used to evaluate product defects at production lines.
Source: Adapted from Li et al. (2018b).

level, and reduce overfitting. The proposed local function is the sum of all the cost functions on each early exit, and these functions are the same as the ones used by the remote proposal.

The manually captured dataset is composed of ten distinct defect categories. The diagnosing ability of the system is compared with two other methods used in defect detection: the contour detection approach and the pixel-based method using the Receiver Operating Characteristic (ROC) curve. In at least nine situations, the proposed system achieves better performance while maintaining lower execution times when compared to local computing.

Edge and cloud resources optimization

Cloud computing is widely employed in industrial networks given its elasticity property, in addition to low-cost data processing and storage (Al-Dhuraibi et al., 2017). Nevertheless, cloud-based solutions in IIoT may cause network congestion as a consequence of the massive data traffic generated. Then, edge-based and hybrid alternatives emerge as an option to implement deep learning without compromising the network. Zeng et al. propose a framework to find the optimal partition and exit point in the classical AlexNet model (Krizhevsky et al., 2017) with five exit points corresponding to a branch covering 12, 16, 19, 20, and 22 layers (Zeng et al., 2019). In their work, the authors run a DNN in both edge servers and devices to get information such as execution time and accuracy on each layer. The main idea is to determine the exit-point and partition that satisfies latency requirements. The edge server executes the DNN until the partition point and then it moves the intermediate results to the devices to finish inference. The optimal point can be determined by an iterative algorithm or a DRL approach. On DRL, the reward function is composed of total latency and accuracy. Device and edge-only are not able to satisfy the latency requirement in most cases. Even under the strict latency requirement, the DRL alternative achieves the same accuracy as the best scenarios.

To reduce network traffic and avoid congestion, Liang et al. use edge nodes to train a CNN and send well-trained models to the cloud (Liang et al., 2020). The number of layers is selected by experiments demonstrating that the accuracy does not have a relevant increase with more than four layers. A dataset to simulate industrial component classification is used to evaluate the proposal in the industrial environment. The training time decreases when compared with the LeNet5 (LeCun et al., 1989) and VGG16 (Simonyan and Zisserman, 2015) DNN architectures while achieving comparable accuracy.

Besides empirical approaches, optimal resource allocation is possible by evaluating and predicting the cloud workload imposed by industrial machines. The workload prediction in this scenario is challenging since the machines generate dynamic workloads. The prediction allows QoS guarantees and optimal usage of industrial network

resources. Zhang et al. indicate that training a deep learning model is a time-consuming task due to the large number of parameters to be considered. They propose a cloud workload prediction model by adopting the Canonical Polyadic Decomposition (CPD) to compress the parameters (Zhang et al., 2018). The model's goal is to predict CPU usage of the virtual machine with the highest workload for a given day, and the usage of multiple virtual machines in four future intervals.

The evaluation is performed using four metrics: approximation error, decrease in classification accuracy, parameter reduction, and increase in training speed. The approximation error and decrease in accuracy are caused by parameter conversion. Parameter reduction is the proportion between the number of original and compressed parameters. The increase in training speed refers to the rate between the execution time of a traditional model using SAEs and the proposed model. The proposed model is compared with a traditional approach and another model that performs parameter compression by leveraging another method, known as Tucker decomposition (Malik and Becker, 2018). To evaluate the prediction accuracy, the authors compare the model performance with traditional neural networks techniques and Deep Belief Networks (DBNs), achieving superior training speed with negligible accuracy reduction and performing the CPU usage prediction with higher accuracy.

Lastly, not disrupting video transmission even when the available network bandwidth is low in an industrial environment is also a challenge. Video interruption for vigilance systems needs to be avoided. Dou et al. conduct streaming optimization with DNN to support keyframes selection. The most important frames are transmitted with reduced resolution until the quality of the video can be increased (Dou et al., 2020). An edge server runs an object detection algorithm based on deep learning and reduces frame resolution when the users' network quality is low. When it changes, the video can be streamed on the original quality. YOLOv3 and SSD are well-known object detection DNNs used in this work (Redmon and Farhadi, 2018; Liu et al., 2016).

CNNs are the most common deep neural network architecture used to solve challenges in industrial networks. Besides video monitoring, the data from sensors to control factory assets can be converted into images. Thus, this justifies CNNs' popularity in industrial networks. The approaches based on video or images generate big data, becoming resource allocation another research trend. Then, the network availability is guaranteed based on DNNs' partition.

Industrial networks overview

Table 7 presents the papers reviewed in this section. We present two common applications in industrial networks, namely, system monitoring and edge and cloud resources optimization. Industrial networks also favor the use of CNNs, this is mostly a consequence of applications that rely on image and video data to perform classification and detection. Nevertheless, instead of simply offloading large DNNs to the cloud, the solutions attempt to reduce communication overhead and delay by using different techniques, such as early-exit approaches and model partitioning. We would like to highlight that these techniques are still unexplored in other challenged networks. Additionally, LSTM networks are also popular in industrial networks, given the natural time-dependent features present in data collected by sensors in the industrial machinery. As mentioned before, 1D-CNNs are also capable of leveraging time dependencies in data but are still somewhat unexplored and are less favored than multiple RNN architectures.

3.4. Vehicular networks

In vehicular networks, the communication typically happens between vehicles, referred to as Vehicle-to-Vehicle (V2V), or between vehicles and infrastructure, referred to as Vehicle-to-Infrastructure (V2I). As a consequence of V2V and V2I, new possibilities are emerging to include communications between vehicles and pedestrians, vehicles, and cloud servers, and also vehicles and cellular networks. These possibilities culminate on the Vehicle-to-Everything (V2X) paradigm, which can be generalized to all communications between vehicles and any device able to interact and support applications under the umbrella of Intelligent Transportation Systems (ITSs) and smart cities (Liang et al., 2019; Zhu et al., 2019). Besides the communication power, ITS applications can leverage the ever-increasing number of sensors embedded into vehicles and other mobile devices for improving driving safety, comfort, and traffic efficiency.

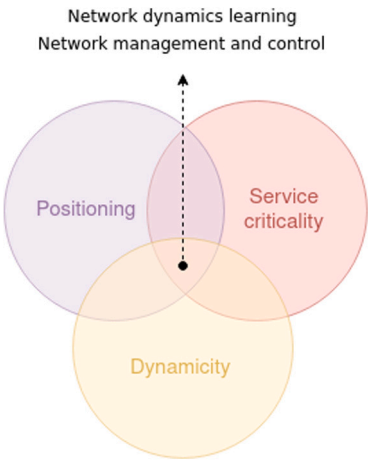

Fig. 15. Challenges in vehicular networks and main deep learning research trends.

Concerning safety, the quick and reliable exchange of situational messages is one of the main requirements of V2X to prevent accidents (MacHardy et al., 2018). To accomplish that, node positioning, network dynamics, and service criticality are challenges to be handled in ITSs, and consequently the target of solutions based on artificial intelligence. More specifically, deep neural networks have proven today to be a preferred choice. Fig. 15 illustrates all the challenging characteristics shared by two research directions using deep learning: network dynamics learning, and network management and control. Firstly, the comprehension of the network dynamics allows the prediction of future traffic behavior as well as optimal or near-optimal decisions, which depends on learning from a large volume of data. Hence, in the first research direction, we discuss important trends to improve drivers and pedestrians' safety and comfort, such as vehicle localization (Wan et al., 2020), traffic prediction (Lv et al., 2014), intelligent traffic light control (Wu et al., 2020b; Wei et al., 2018; Van der Pol and Oliehoek, 2016), safety analysis improvements (Peng et al., 2018; Lago et al., 2021), and motion planning for autonomous vehicles (Aradi, 2020; Isele et al., 2018). In the second research direction, we introduce works that deal with network management and control, which aim to improve network resource allocation and reduce the volume of control data exchanged (Ning et al., 2020; Ye et al., 2019; Wang et al., 2020).

Network dynamics learning

Intelligent transportation systems aim to enhance the safety and the comfort of trips by leveraging data, such as driving safety status, vehicle traffic flow on highways, and road visibility conditions (Tong et al., 2019). Therefore, precise vehicle positioning is a relevant issue to ensure safety in ITSs. With that in mind, Wan et al. develop a method based on Direction-Of-Arrival (DOA) estimation to accurately estimate vehicle position in real-time (Wan et al., 2020). DOA estimation is based on the SBLNet that contains several layers that perform Sparse Bayesian Learning (SBL). In this approach, base stations contain a large number of antennas that receive the vehicles' sensor information. Then,

Table 7
Summary of applications and deep learning trends on industrial networks.

Paper and authors	High-level description	Deep learning method	Employed dataset
Zeng et al. (2019)	Network partitioning Resource allocation	CNN	CIFAR-10
Liang et al. (2020)	Model training at the edge Resource allocation	CNN	T-Less
Zhang et al. (2018)	Workload prediction Resource allocation	SAE	Proprietary, generated with PlaneLab and CloudSim
Dou et al. (2020)	Network partitioning Real-time video	YOLOv3 (Redmon and Farhadi, 2018), SSD (Liu et al., 2016) and R-CNNs	COCO
Luo et al. (2020)	Fault diagnosis Process monitoring	Unspecified	Simulations for the Tennessee Eastman Process
Cheng et al. (2018)	Failure prognostics Machine monitoring	LSTM	Vibration signals by the NSF I/UCR Center
Chen et al. (2020a)	RUL prediction Machine monitoring	LSTM	C-MAPSS PHM 2008
Li et al. (2018b)	Product defect detection Factory monitoring	CNN	Proprietary, collected in a tile production

the information of this massive Multiple-Input and Multiple-Output (MIMO) is used as input data of the SBLNet, whereas the output is the DOA of the autonomous vehicle.

One important research direction is traffic flow prediction. Lv et al. propose a deep learning model that leverages stacked autoencoders to learn generic traffic flow characteristics and predict future behavior (Lv et al., 2014). The proposed model predicts traffic flows for the next time interval considering four different sizes: 15, 30, 45, and 60 minutes. Each time interval uses different numbers of neurons per layer, determined through trial and error. Fig. 16 shows the proposed architecture, which employs a stacked autoencoder followed by an output predictor. Initially, some characteristics are extracted by the stacked autoencoder from the input data before going through the predictor. Just as our workflow illustrated in Fig. 5, the data is collected by multiple detectors distributed by road systems in California. Furthermore, the data collected from the vehicular traffic is aggregated to generate the mean traffic flow on the roads with multiple detectors. Thus, the data collected by each detector is aggregated using longer time intervals. For the implementation, the traffic data is submitted to the stacked autoencoder. During training, the layers are trained from the input towards the output. The goal is to minimize the cost function before training the next model's layer. The last output layer is the input of the prediction layer. The predictor initializes its parameters at random or through supervised learning. After training each layer, a new training step is performed for the complete network. In this step, all parameters are learned through standard backpropagation. The proposal's performance is assessed using three indicators: MAE, MRE, and MSE. Experiments are conducted comparing the proposed model with other commonly used traffic prediction approaches, such as Radial Basis Function (RBF) networks. The proposal achieves a higher mean accuracy when compared with the other models and the best performance is the result of proper hyperparameter tuning, such as the number of hidden layers and the number of neurons per layer. Thus, the experiment shows that proper hyperparameter tuning is an important factor to achieve satisfactory results.

Another relevant research direction, intelligent traffic light control, is an emerging topic to optimize the efficiency of transportation systems. Traffic lights are typically controlled by pre-defined fixed-time or vehicle-actuated control methods, neglecting the current traffic conditions. The central point of this problem is the inability to deal with high randomness in traffic. Traffic control can impact factors such as waiting time for vehicles and the number of vehicles passing through intersections. In this direction, Wei et al. develop an approach based on deep Q-learning with offline and online steps called Intellilight (Wei et al., 2018). The data is collected by using a fixed timetable for the lights during the offline phase and used by the model for training. Afterward, in the online phase, the agent performs the action based on state observations. The comparison analysis is between the Intellilight and three types of methods: fixed-time control method, the Self-Organizing Traffic Light (SOTL) control (Cools et al., 2013), and a proposal based on Deep Q-Networks (DQNs), a neural network assisted by Q-learning, proposed by Van der Pol and Oliehoek (2016). The evaluation uses the average length of the queue of cars, delay, the waiting time, and the average travel duration. The results demonstrate the capability of the proposal to achieve smaller or the same values of all metrics compared with all the other three approaches used in the analysis.

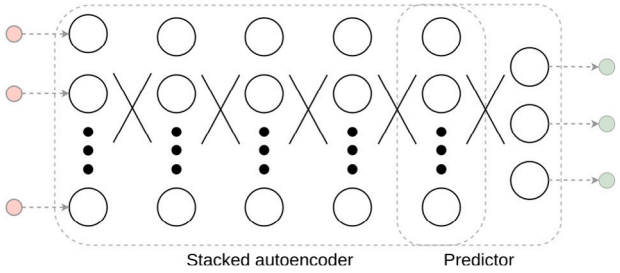

Fig. 16. Traffic flow prediction architecture.
Source: Adapted from Lv et al. (2014).

Tong et al. also tackle urban traffic light control, proposing a multi-agent deep reinforcement learning method to minimize congestion and improve urban environments for drivers and pedestrians (Wu et al., 2020b). The DRL model is based on the Multi-Agent Recurrent Deep Deterministic Policy Gradient Algorithm (MARDDPG). The work also considers waiting time for pedestrians to cross the road. Furthermore, the buses have a higher priority to move. Therefore, the states have the following characteristics: the cars' queue length in the road, the vehicles' speed, the number of pedestrians crossing the road, and the traffic light phase. Each state transition is defined as a Markov Decision Process (MDP). The reward function considers the length of the car queue, waiting time for each vehicle, and the total waiting time of pedestrians. The agents are placed at each road intersection to adjust the time of the traffic light phase. The model uses an LSTM network on the critic network and the actor-network for each agent. During the offline training, each agent receives the environment observations and the other agents' actions. Therefore, the agent adjusts the local

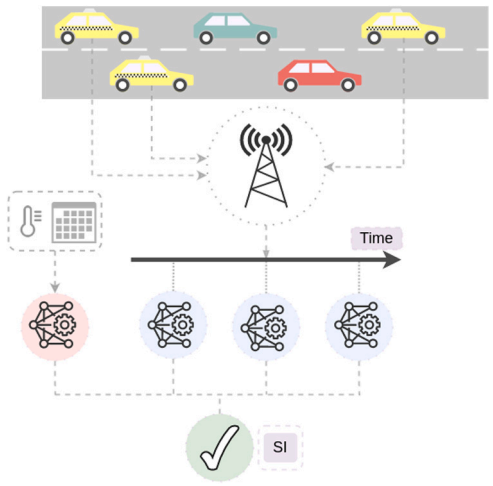

Fig. 17. DeepRSI is a framework for evaluating road safety by a Safety Index (SI), adapted from (Peng et al., 2018). The vehicle data and external data are, respectively, the input of the convolutional and the fully connected networks.

policy according to the other agents' estimated policies to improve the convergence process. Nevertheless, each agent makes the decision based only on local information in the online test. This learning process is centralized but with distributed execution. The proposal's evaluation is conducted in the Simulation of Urban Mobility (SUMO) simulator. The authors compare their method with a baseline approach, achieving better results for medium-scale traffic networks.

Peng et al. focus on improving the safety analysis through a hybrid neural network (Peng et al., 2018). Research on vehicular safety usually follows two distinct approaches. Driver and pedestrian safety approaches use information about driver behavior and vehicular information, such as speed, camera pictures, and vehicle built-in sensors. The road safety approach can use mathematical models or picture analysis. Nevertheless, Peng et al. argue that external-factors-based approaches use mathematical models that may not be applicable to every urban setting due to empirical assumptions. The authors propose a deep learning framework called DeepRSI that considers two approaches to road safety. The DeepRSI is a hybrid neural network that improves the road safety analysis precision by capturing sensor information from cars and external environmental factors, such as the weather (Peng et al., 2018). The input data consists of weather and event data collected by 13,000 taxis that contribute to GPS routes and vehicular sensors' information.

Fig. 17 illustrates the DeepRSI framework, which is composed of three convolutional networks and a fully connected network for each city region. Space–time learning is assigned to the convolutional layers. In this sense, the dataset is divided into three temporal segments: recent, near, and distant. One distinct convolutional neural network receives one temporal segment, which performs spatial learning from the dense convolutional layers. Then, the output aggregation is through a method based on parametric matrix fusion. The fully connected layer receives the external environment factors as inputs. Its output is added to the output of the convolutional layer. The framework uses the Rectified Linear Unit (ReLu) activation function to speed up convergence, while the softmax (Bishop, 2006) function is used to perform classification. The Safety Index (SI) is the model output for each region. Therefore, the SI evaluates road safety considering all important aspects. In other words, the SI considers both information related to vehicles and the environment. The evaluation of the DeepRSI framework is done through the analysis of precision and recall. The results show that DeepRSI presents better performance when compared with other approaches such as Decision Trees (DTs) and Support Vector Machine (SVM). The authors argue that the proposed framework is generic enough to be applied to any scenario because its learning is not limited to local images, such as other approaches related to road safety. The important factor of DeepRSI is that data aggregation is used for road safety analysis, which contrasts with its usual application concerning drivers and pedestrians.

Motion planning for autonomous vehicles is also considered a relevant research direction as a consequence of the typical scenario dynamics. In this sense, Aradi discusses DRL employment in motion planning with approaches in different scenarios, considering single or multiple agents (Aradi, 2020). Isele et al. propose motion planning using DRL for unsignaled intersections (Isele et al., 2018). In their work, the DRL model is a DQN. The DQN receives from the environment the heading angle, velocity, and an indicator term. The indicator term informs whether the road grid is occupied. The approach considers two possible actions: time-to-go and sequential actions. The time-to-go represents the actions wait-to-go, where each decision to wait allows the next action to be wait or go. Nevertheless, the agent does not allow a wait decision after a go action. The sequential mode represents the movement actions, which have a more complex behavior, where the agent can slow down and wait for oncoming traffic to conclude. The authors train one DQN for each action, *i.e.*, one DQN for sequential actions and another for time-to-go. The sequential network has 12 outputs that correspond to three actions: accelerate, decelerate, and maintain velocity; whereas the time-to-go network has 5 outputs, one corresponding to go and all the others to wait. Isele et al. conduct experiments using SUMO simulator with data collected from an autonomous vehicle using six Lidar sensors. The vehicular traffic model is the Intelligent Driver Model (IDM). The simulation considers five scenarios among turn left, turn right, and move forward on the road. The results show that DQNs are able to learn about the scenarios, although in some tests a collision occurs.

Network management and control

Vehicular communications are essential to many ITS applications, such as safety. Nevertheless, the communication environment is challenging and imposes obstacles to vehicular networking. One approach to deal with these obstacles is devoted to improvements of wireless resource allocation. In this context, recent works propose deep learning to promote efficient resource allocation. These approaches can be centralized or decentralized. In the former, the central controller is responsible for resources management (Ning et al., 2020); whereas in the latter, each node, such as a vehicle or a base station, can manage its resources (Ye et al., 2019).

Using a centralized approach, a network traffic control system based on deep reinforcement learning is employed to deal with user requests and resource allocation on the Internet of Connected Vehicles (IoCVs) (Ning et al., 2020). Traffic control systems allow better network resource allocation and cache content, consequently maximizing operators' profit. The IoCV architecture is hierarchical and composed of one cellular Base Station (BS) and several Road Side Units (RSUs), which provide communication infrastructure and storage services to vehicles. RSUs have limited processing and storage capacity at the network edge, whereas cellular base stations have large processing and storage capacity. The DRL model is positioned at the BS and, therefore, its agent is in charge of managing traffic allocation and storage tasks for all devices in the network, *i.e.*, vehicle, RSU, and BS. The BS receives all environment information, such as vehicle speed, and processing and storage capacity of each RSU.

The speed is used to estimate the communication time between the vehicle and the RSU and, consequently, the BS defines whether the RSU can receive and process the data from the vehicle. This environmental information is used as the input of the CNN, which is the DNN architecture of the DRL model, whereas the output is a set of actions. These actions are the execution of the computing tasks and requested contents for each RSU. Furthermore, the output

also defines the corresponding resource allocation to each vehicle. The results suggest that the proposal achieves satisfactory performance in IoCVs.

The centralized approach achieves high performance for resource allocation mainly because of the complete view of the network. Nevertheless, the growth in the number of vehicles can increase the network signaling overhead and the communication time. Therefore, Ye et al. propose a decentralized system based on deep reinforcement learning to enable efficient communication in Vehicular Adhoc NETworks (VANETs) (Ye et al., 2019). The authors implement an autonomous agent capable of making decisions and determining the sub-band and the transmission power that optimize communications with only local information. The proposal uses a reward function to ensure the latency requirements for V2V links, while it ensures that V2V communications do not interfere with V2I links. The proposed model employs deep Q-Learning to allocate network resources and is designed for unicast and broadcast scenarios. Besides unicast scenarios, broadcast scenarios are essential to safety applications given the high network dynamics. Applying the proposed model to the unicast scenario at each instant t, the agent can observe the communication status, and consequently, select the best sub-band and transmission power according to the learning policy. In the broadcast scenario, the agent can also relay the previously received broadcast message. The proposed model is trained and tested in a simulated environment. Vehicles are randomly placed on a road, and it is assumed that each vehicle needs to establish communications with three other vehicles. The number of vehicles is between 20 and 160 in the experiments. The model architecture is a fully connected neural network with three hidden layers. The proposed model is compared with others in both unicast and broadcast scenarios, showing more efficient use of communication links. The proposal, however, imposes more computational time, contrasting to the initial claim of decentralized approaches. Aware of that, Ye et al. plan to reduce the time needed to compute the deep neural network.

In another work, Wang et al. compress the Channel State Information (CSI) and allocate resources in V2X networks (Wang et al., 2020). The authors analyze two approaches for allocating resources for each V2V link based on DRL, one centralized, called C-Decision, and another decentralized, called D-Decision. The authors refer to a V2V link as an abstraction regarding the pair of nodes connected by a link. Hence, the state of each V2V link has the following characteristics: transmission power, the current channel power gains, and the interference power of all channels, V2V and V2I. This local information is compressed through a DNN at the vehicle to reduce the network signaling overhead. After that, the compressed information becomes the input of the DRL model.

In the C-Decision scheme, the DRL model is at the BS, which is the agent that receives the compressed information to employ in a DQN. The goal of the reward is to guarantee the V2V link transmission with a minimal impact over V2I links. The BS action determines the channel allocation vector for each V2V link. Finally, the BS sends the results to all vehicles. Conversely, in the D-Decision scheme, the same DRL architecture is employed at each V2V link. Furthermore, a DNN at the BS aggregates the feedback from all vehicles. The output is the Aggregated Global Information (AGI), which becomes the input of the deep Q-learning model along with local observations. Then, according to the reward policy, the DRL model determines the action to choose the channel allocation. In the experiments, the model architecture is a fully connected neural network, with three hidden layers and a different number of neurons per layer in each neural network. Both approaches have a near-optimal performance. When they compare the centralized with the decentralized approach, the C-decision scheme needs fewer training episodes. We highlight that, in the context of reinforcement learning, an episode is a sequence of states, actions, and rewards, ending on a terminal state. Furthermore, the D-Decision has a small performance degradation, which still allows its implementation. In network management and control, we observe that the centralized approach has better results with less overall computational time. Nevertheless, the decentralized approach has greater decision autonomy with less traffic control data.

An important remark on vehicular networking is the application of deep reinforcement learning in several research directions especially when information about the environment is required.

Vehicular networks overview

Table 8 presents the papers reviewed in this section. We present two common applications in vehicular networks, namely, network dynamics learning and network management and control. In vehicular networks, DRL is a common trend. The dynamics, coupled with exact SLRs, fosters an environment where the time consumed by transferring data to cloud servers paves the way to deep learning on the network edge. Reinforcement learning is a powerful paradigm to perform actions in real-time, as long as the agent can be placed at the network edge. The most prevailing deep reinforcement learning model is based on Q-learning. DNNs, especially CNNs, enable DRL for VANETS.

4. Transversal issues

Challenged networks share many issues and research directions. For instance, security is a general concern and, consequently, can be considered transversal to mobile, IoT, industrial, and vehicular scenarios. Similarly, data privacy, which is fundamental for deep-learning-based solutions, can also affect all challenged networks. From the performance viewpoint, solutions to enhance deep-learning-based proposals are also of common interest. Early-exit deep neural networks and model partitioning have been investigated as a way of guaranteeing satisfactory inference results, whilst working under latency constraints.

This section reviews solutions that are not being developed for a specific challenged network in mind, or solutions that can be applied to multiple types of challenged networks. We group solutions by subject instead of network type and highlight to which challenged network the corresponding solution could be applied.

4.1. Security

The large number of devices with communication capabilities and Internet access in challenged networks significantly increases network vulnerability. If efficient security mechanisms are missing, these devices become entry points to malicious users. In industrial networks, for example, online security systems are primordial to keep the information regarding new attack strategies to critical structures always updated. Likewise, in challenged networks, intrusions are a serious concern. Intrusion Detection Systems (IDSs) are fundamental and represent a transversal issue for challenged networks. IDSs often deal with large volumes of data usually collected from network traffic, bringing deep learning approaches into scene. The advantages of deep-learning-based IDSs over those using classical machine learning techniques are lower false positive rates and more ease in identifying new attack types (Al-Hawawreh et al., 2018). This section discusses the key points for the development of secure challenged networks. We emphasize that most solutions are general enough to be replicated to another type of network, even though it is proposed to a given challenged network type.

Al-Hawawreh et al. propose an anomaly detection system with an architecture based on autoencoders and feedforward neural networks. The dataset used for system rating is composed of data collected from TCP/IP traffic, which are then divided into three subsets. While subset A contains only samples from normal network behavior, subsets B and C contain regular and attack samples (Al-Hawawreh et al., 2018). During data pre-processing, non-numerical values are mapped into numerical values before applying the Z-score normalization, which maps the values around the set mean. Training is then performed in two

Table 8
Summary of applications and deep learning trends on vehicular networks.

Paper and authors	High-level description	Deep learning method	Employed dataset
Wan et al. (2020)	Precise location estimation Autonomous vehicles	Deep Sparse Bayesian Learning (SBL)	Proprietary
Lv et al. (2014)	Vehicle traffic flow prediction	MLP and SAE	Collected from Caltrans Performance Measurement System (PeMS)
Wu et al. (2020b)	Urban traffic light control Real-time analysis	Multi-agent DRL, and LSTM	Proprietary, simulated with Simulation of Urban Mobility (SUMO)
Wei et al. (2018)	Urban traffic light control Real-time analysis	Deep Q-learning and CNN	Proprietary, simulated with SUMO
Peng et al. (2018)	Safety analysis Temporal analysis	MLP and CNN	Proprietary, collected from an online survey
Isele et al. (2018)	Motion planning Autonomous vehicles Unsignaled intersections	Deep Q-learning	Proprietary, simulated with SUMO
Ning et al. (2020)	Resource allocation Edge computing Real-time analysis	DRL and CNN	Proprietary
Ye et al. (2019)	Resource allocation Real-time analysis	Deep Q-learning and MLP	Proprietary
Wang et al. (2020)	Resource allocation Data compression Real-time analysis	Deep Q-learning and MLP	Proprietary

steps. The reason for dividing the training is to make sure that the algorithm responsible for classification is initialized with optimal weights and biases, since random initialization can incur slower convergence. The first training phase is performed with autoencoder networks using subset A. In the second phase, the accuracy of the method is tested with the samples from subset B, which includes attack samples. The mean squared error is chosen as the cost function to be minimized during training, using Stochastic Gradient Descent (SGD). At the end of this phase, the prediction model is ready to be tested. Lastly, subset C is used for testing. The authors evaluate the system using the NSL-KDD and UNSW-NB15 datasets. The results show that the proposed system achieves a higher performance on the NSL-KDD scenario with high accuracy and detection rate, but with a small false positive rate. On the UNSW-NB15 dataset, the proposal also achieves a satisfactory performance. However, there are slightly reduced accuracy and detection rate, and a higher false positive rate. A similar application is explored by Ferrag et al. which also consists of performing intrusion detection, but on datasets tailored to agriculture 4.0 scenarios (Ferrag et al., 2021). In addition to MLP and CNN evaluation, the authors also use an LSTM network to explore the long-short term dependencies between samples. Their results show that, with 1D-CNNs, one can also achieve equivalent or superior results compared with LSTM networks.

Unlike the previous work, Sharafaldin et al. evaluate the performance of deep learning techniques compared with traditional machine learning models applied to anomaly detection (Sharafaldin et al., 2018). The authors use the random forest regressor model to extract the best attributes from the 78 present in the original dataset and use an MLP to perform classification. As such, it is possible to infer which attributes have high predictive power in relation to the target, as illustrated in Fig. 5. The chosen attributes vary from the type of attack triggered against the network. It is worth noting that the preprocessing step indicates the most effective attributes for identifying and classifying the attacks, even before applying the deep learning model. After the preprocessing step, the authors use six traditional machine learning models and an MLP as the deep learning model. The models are evaluated using precision, recall, and F1-score as metrics. The results show that the proposed deep learning model does not achieve satisfactory performance, confirming that developing deep learning applications is still a challenge.

Similarly to the previous work, Panwar et al. use traditional machine learning models, showing the need for further investigation on deep learning models and their performance to IDS development (Panwar et al., 2019). Nevertheless, even though Sharafaldin et al. and Panwar et al. show that anomaly detection models based on traditional machine learning techniques achieve better performance than simple deep learning models, Vinayakumar et al. present a detailed analysis showing the potential of deep neural networks for attack classification in different computer network types (Vinayakumar et al., 2019). In addition to the extensive analysis using the NSL-KDD, UNSW-NB15, and CICIDS datasets, the authors also perform an analysis on hyperparameter tuning for the deep learning model. The authors achieve over 96% accuracy both in multiclass and binary classification on the CICIDS dataset with deep regularized neural networks with 3 layers and 1 layer, respectively. The best performance achieved with classical machine learning techniques is obtained using a random forest model, with around 95% accuracy both in multiclass and binary classification and 94.0% accuracy in binary classification on the same dataset. A similar approach proposed by Wu et al. can be seen in Wu et al. (2020a).

As a particular challenge of deep-learning-based solutions for security applications, we have noticed a lack of properly constructed datasets. As a consequence, papers are usually evaluated on a limited number of datasets, which is an obstacle to model generalization. Since new network attacks are constantly created by malicious users, it is of utmost importance that deep learning models become general enough to, ideally, detect previously unseen attacks.

4.2. Federated learning and privacy-preserving algorithms

The ever-increasing interest in digital privacy originates a large number of solutions proposed to protect users' data during neural network training. Differential privacy achieves this goal by leveraging the concept of adjacent databases (Dwork, 2011; Dwork and Roth, 2014; Ho et al., 2021). Abadi et al. have demonstrated that it is possible to train DNNs within an adjustable privacy budget (Abadi et al., 2016b). Shokri and Shmatikov implemented a collaborative training system for DNNs, which enables multiple nodes to learn a deep learning model without sharing their inputs (Shokri and Shmatikov,

2015). Their implementation of a distributed selective stochastic gradient descent algorithm can be viewed as an early implementation of decentralized federated learning. Even though the final system does not strictly specify the way a central server selects and controls the participating nodes, as it is common in most current implementations of federated learning, their implementation is able to achieve comparable performance to centralized SGD while maintaining a per-parameter privacy budget. Following this trend, federated learning has emerged as an alternative to large deep neural networks training, while maintaining privacy (Lim et al., 2020; Li et al., 2020; McMahan et al., 2017). Users' privacy is achieved by leveraging a distributed training process, where each user involved is responsible for training a machine learning model and reporting its updated parameters to a central server. The server is responsible for aggregating the newly received parameters and reporting the global model to the nodes, which can then be used to perform local inference and improve the devices' usability. While none of the papers reviewed in this section are dedicated to a particular challenged network, they generalize well and could be applied to scenarios where multiple underutilized nodes are available. Although we could not find many research papers in federated learning, we strongly believe that the approach is promising. There are already experiments using federated learning on vehicular and industrial networks, where privacy and latency are crucial (Tan et al., 2020; Konečný et al., 2016; Cioffi et al., 2020). In vehicular networks, federated learning tackles privacy requirements alongside the already present mobility challenge, whilst in industrial networks, federated learning must be used without affecting the service criticality.

4.3. Early-exit deep neural networks and model partitioning

As neural networks grow deeper, many applications aim to employ shallower networks for local and fast inference, while they maintain a deeper model on a cloud server for more challenging inference tasks (Laskaridis et al., 2020; Teerapittayanon et al., 2016; Scardapane et al., 2020; Pacheco and Couto, 2020; Pacheco et al., 2021). Smaller neural networks are less capable of generalization when compared to their deeper counterparts. This drawback, however, can be circumvented if the task at hand is not very difficult, such as identifying a vehicle instead of classifying the vehicle type. This even allows smart caching systems to save smaller learning models in the device to reduce inference latency (Drolia et al., 2017).

As described in Section 3.3, Li et al. have applied early-exit neural networks to industrial networks (Li et al., 2018b). Laskaridis et al. build an entire system based on this concept (Laskaridis et al., 2020). Their system, called SPINN (Synergistic Progressive Inference of Neural Networks) over Device and Cloud, is capable of leveraging multiple early-exit points in a DNN through the use of a progressive inference method, where the system is capable of evaluating predictions and their respective confidence levels at multiple exit points in the neural network. The authors evaluate the system and suggest its usage in high-mobility applications, such as mobile and vehicular networks.

The deployment of early-exit neural networks incurs higher development costs. Nevertheless, it has great potential when a particular network has to handle variable performance using highly dynamic data. This scenario is particularly interesting in IoT, mobile, and vehicular networks, where node positioning can affect the data distribution. For example, in the case of a node that does not change its position for long periods of time, a shallower neural network can handle the inference, whilst a more dynamic node can use the power of a deeper neural network hosted on a cloud server.

4.4. Deep learning applied to real-time video applications

From surveillance footage in industrial networks to video calls in mobile networks, video streaming contributes to a large volume of traffic on the Internet (Cisco, 2020). Depending on the task, different deep learning techniques have been applied to improve video streaming, be it by maximizing user QoE (Mao et al., 2017) or by decreasing latency in vehicle detection (Du et al., 2020).

Du et al. have implemented a server-driven video streaming system capable of reducing network latency while maintaining comparable performance to baseline approaches (Du et al., 2020). This is achieved by using deeper neural network models on the server side to control the edge devices and how they should prioritize pixels in a video stream. The edge camera sends a low-quality video stream to the server and the server requests only more "interesting" parts of the video to be sent in higher quality. Kang et al. have also developed a system to reduce the overall latency. Nevertheless, they tackle the problem of neural network querying at scale, *i.e.*, they aim to efficiently find the neural network architecture more suited for a given inference task (Kang et al., 2017). The NoScope system evaluates to which extent simpler models are faster to run and can be capable of performing less generic tasks. NoScope has a "calibration" step, where it uses a larger neural network model to train a smaller and more specific model, which is capable of performing well in a particular scenario. In this sense, NoScope uses a larger DNN to label a video, which will be used to train a smaller and faster learning model. Then, a cost-based optimizer is used to search for the best model. Both papers' proposals are evaluated on videos from traffic cameras, and both techniques are promising for applications in vehicular networks, where network latency is critical.

Transversal issues overview

Table 9 presents the papers reviewed in this section. We present four common applications that are transversal to all challenged networks, namely, security, federated learning and privacy-preserving algorithms, early-exit deep neural networks and model partitioning, and deep learning applied to real-time video applications. It is possible to identify a heavy interest in security and privacy applications, which seems to permeate all challenged networks. For applications that implement intrusion detection systems, which usually rely on package inspection, standard approaches such as MLP are used. However, for federated learning applications, we can observe CNNs as the most popular choice. This is because mobile applications typically consist of computer vision tasks and federated learning is often applied to mobile scenarios.

5. Discussion and future trends

In this section, we group deep learning methods and discuss the most adopted applications considering all challenged networks. Afterward, we highlight some possible future trends in deep learning that can bring improvements for applications in challenged networks. Fig. 18 summarizes the most used deep learning methods and applications for the challenged networks previously addressed. The methods are eventually combined to solve a problem and, consequently, they can share the same application. Due to the popularity of CNN in image and video applications, this method concentrates the works using early-exit approaches. Thus, the model partitioning enables minimizing inference latency. AEs, on the other hand, are usually employed for research on energy saving and data compression by the encoder and decoder layers. We also observe that CNNs and AEs can be combined for resource allocation and image processing. MLPs concentrate works for QoE prediction. There are efforts on real-time analysis and network traffic classification adopting all the methods, being able to combine more than one simultaneously. In addition, LSTMs and CNNs are adopted for Industrial assets monitoring and network slicing. CNNs, LSTMs, and MLPs share works for intrusion detection systems and privacy-sensitive applications. Lastly, CNNs and MLPs can be adopted for network compression, *i.e.*, reduce the number of parameters compromising network performance as little as possible. The empty spaces of Fig. 18 demonstrate, in this work, that no papers were found matching the corresponding combination.

Table 9
Summary of applications and deep learning trends on transversal issues.

Paper and authors	High-level description	Deep learning method	Employed dataset
Al-Hawawreh et al. (2018)	IDS on IIoT	MLP, DBN, and RNN	NSL-KDD and UNSW-NB15
Ferrag et al. (2021)	IDS on agriculture 4.0	MLP, CNN, and LSTM	CIC-DDoS2019 and TON_IoT
Sharafaldin et al. (2018)	IDS dataset construction	MLP	CICIDS
Vinayakumar et al. (2019)	Intelligent IDS	MLP	CICIDS
Abadi et al. (2016b)	Privacy	CNN	MNIST and CIFAR-10
Shokri and Shmatikov (2015)	Privacy	MLP and CNN	MNIST and SVHN
McMahan et al. (2017)	Federated learning	MLP, CNN, and LSTM	MNIST and CIFAR-10
Laskaridis et al. (2020)	Progressive inference Cloud offloading	CNN, ResNet-50/-56 (He et al., 2016), VGG16 (Simonyan and Zisserman, 2015), MobileNet (Howard et al., 2017), and Inception-v3 (Tang, 2018)	CIFAR-100 and ImageNet
Teerapittayanon et al. (2016)	Early-exit neural networks	CNN, LeNet (LeCun et al., 1989), AlexNet (Krizhevsky et al., 2017), and ResNet (He et al., 2016)	MNIST and CIFAR10
Du et al. (2020)	Video streaming	CNN, FasterResNet-101, and FCN-ResNet101	Proprietary and public
Kang et al. (2017)	Scaling neural network querying	CNN and YOLOv2 (Redmon and Farhadi, 2016)	Proprietary

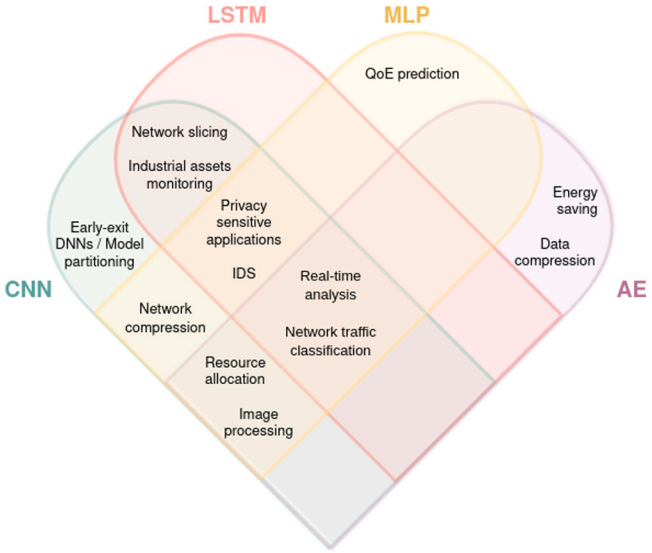

Fig. 18. The main deep learning methods and the corresponding applications in challenged networks.

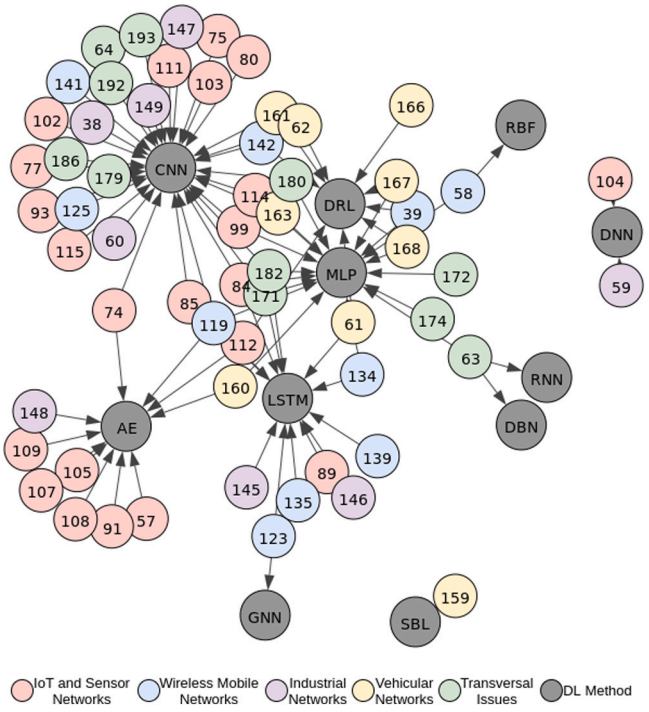

Fig. 19. The graph shows the relationship between the deep learning techniques and the works surveyed in this paper. The nodes' colors relate to the section in which the work appears in this survey. An edge links the work to the deep learning method used by the authors. The CNNs are the most used architecture.

While Fig. 18 generalizes and presents only the most common methods observed, the graph in Fig. 19 illustrates the relationship between papers and deep learning methods. The gray nodes represent the deep learning method and the colored nodes, the papers. Different colors are assigned to papers according to their respective challenged network. We have grouped multiple networks, such as AlexNet, VGG16, and YOLOv2 by their common architectures, which are CNNs. We have also grouped DQNs into the broader DRL paradigm. We note that CNNs and AE are the most commonly used methods, considering the surveyed papers. Also, even though most papers use only one deep learning approach, we can observe a few examples that combine two or three approaches. Furthermore, the use of autoencoders in IoT networks becomes apparent. As mentioned before, this is because this type of DNN can achieve good performance on simple problems while using fewer parameters than other architectures. Additionally, almost all implementations of recurrent neural networks are made with LSTM networks, proving their ability to better handle sequential data compared with traditional RNNs. Furthermore, Fig. 19 shows research trends for the surveyed challenged networks. As previously mentioned, the use of autoencoders and CNNs is a trend in IoT and sensor networks due to data characteristics. Although LSTM can also be employed by some papers, MLP is considered a trend for prediction applications in wireless mobile networks. Furthermore, CNN utilization is also observed for other approaches in wireless mobile networks, such as network slicing and traffic classification. In industrial networks, CNNs are well suited since the data's features are usually based on images or videos. Image processing is also the reason behind CNNs being considered a trend in vehicular networks. In addition, MLP is commonly used for applications that do not rely on image analysis. Finally, we can advocate that the key deep learning methods are CNN and MLP for the transversal issues, if we consider the same rationale used for vehicular networks.

Looking at broader solutions, it is a common trend to evaluate new solutions using famous neural network architectures, such as YOLOv3 and MobileNet, which may be initialized with already trained parameters. This method saves valuable time that would be spent training and tuning a large DNN. It is also very usual to compare results by testing on common datasets, such as CIFAR-100 or MNIST, called benchmark

datasets. Nevertheless, unlike problems in computer vision, newer research areas, such as federated learning, still lack available benchmark datasets. This poses an additional challenge when comparing different proposals. However, some recent papers (Ghosh et al., 2020; Marfoq et al., 2020) have started using datasets tailored for federated scenarios, such as the FEMNIST and the Shakespeare datasets (Caldas et al., 2019).

The previous discussions in this paper highlight some issues and approaches that can be addressed by the research community in the near future. The directions pointed out in our work are just some of the several trends in challenged networks. It is worth mentioning though that one of the growing trends is the increasing adoption of reinforcement learning in scenarios with high dynamics between network nodes, such as in vehicular networking (Wu et al., 2020b; Wei et al., 2018; Isele et al., 2018).

Among possible future trends that were not explored in this paper, cognitive computing techniques emerge to support obtaining solutions by simulating human thinking to answer complex questions, which is a trend to deal with big data (Gupta et al., 2018). In this way, cognitive computing techniques can be used to deal with the large volume of data generated in challenged networks. The application of explainable artificial intelligence with deep learning methods to make the decisions by AI-based systems more transparent for humans is also in the spotlight of future trends. The mobile traffic classification proposed by Nascita et al. is an example of an XAI application with deep learning to solve a problem in a challenged network (Nascita et al., 2021).

Furthermore, the open issues observed for the different challenged networks provide additional clues about future trends. Although multiple papers evaluate their solutions on public datasets, reproducibility is still an issue. The works surveyed demonstrate that there is not a consensus on which dataset is the most suitable for each application. In this direction, some recent works have tried to use both public and specific datasets (Du et al., 2020). Public datasets can be used to compare the performance with other proposals whereas private datasets are used to evaluate the proposed solution for a specific use case. Generative adversarial networks can produce artificial samples to deal with the lack of data for deep learning models training (Yi and Mak, 2020). Additionally, data quality can also be a problem for performance evaluation. As a consequence, machine learning can also be applied for data quality improvement (Okafor et al., 2020).

6. Final considerations and future scope

Machine learning and, moreover, deep learning, has become of paramount importance for research in the last few years. This paper provides a comprehensive survey on deep-learning-based solutions when applied to challenged networks. These networks are typically composed of multiple and heterogeneous data sources that deal with intermittent connectivity. In our paper, we extend this definition to also include networks that can benefit from intelligence acquisition from multiple sources. Then, we divide the challenged networks into four main categories with particular characteristics and constraints: IoT and sensor, mobile, industrial, and vehicular networks. We have presented a quick overview of key machine learning concepts and the main existing neural networks. Then, we have proposed a simple, yet broader workflow in contrast to others in the literature, based on typical steps taken by authors when solving challenged networking issues. This workflow is the result of a methodological review of the corresponding state of the art. We have discussed recent deep-learning-based solutions and techniques applied to each challenged network showing promising and concrete results. Even though the focus was application oriented, since solutions do not necessarily tackle networking issues, it was noticeable that the environment affects and can also be affected by communications.

We have additionally covered transversal solutions, which were, or can be, applied to multiple types of networks. We summarized the approaches to each challenged network and transversal solutions, which allowed the observation of trends in application development. This organization yielded the binding of neural network architectures with particular challenges. For instance, we observed the popularity of recurrent neural networks in mobile networks, the frequent use of autoencoders in IoT, the utilization of convolutional neural networks in industrial settings, and the preference for deep reinforcement learning in vehicular problems. Even though we do not have the ambition to claim that these are absolute trends, depending on the challenge, it can be at least pointed out as a preferred choice today. Finally, we highlight some potential approaches that could be seen as future trends for applying deep learning to challenged networks.

As future research, we identified that the advent of federated learning in the last few years has created a fertile ground for new research, which must be properly surveyed. Also, we plan to extend our research on recent topics such as cognitive computing and explainable AI. Finally, as a parallel effort, we would like to investigate the availability of public datasets and the possibility of reproducing previously obtained results.

CRediT authorship contribution statement

Kaylani Bochie: Conceptualization, Validation, Investigation, Writing – original draft, Writing – review & editing. **Mateus S. Gilbert:** Conceptualization, Investigation, Resources, Writing – original draft, Writing – review & editing. **Luana Gantert:** Validation, Resources, Writing – original draft, Writing – review & editing. **Mariana S.M. Barbosa:** Investigation, Resources, Writing – original draft, Writing – review & editing. **Dianne S.V. Medeiros:** Methodology, Writing – original draft, Writing – review & editing, Supervision. **Miguel Elias M. Campista:** Methodology, Writing – original draft, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. It was also supported by CNPq, Brazil, FAPERJ, Brazil Grants E-26/211.144/2019 and E-26/202.689/2018, and FAPESP, Brazil Grant 15/24494-8. The authors would like to thank Professors Heraldo Luís Silveira de Almeida, José Gabriel Rodríguez Carneiro Gomes, and Rodrigo de Souza Couto for their help and kind attention. We also thank the anonymous reviewers for their feedback, which helped us on improving this survey.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., 2016a. Tensorflow: A system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). USENIX Association, Savannah, GA, pp. 265–283.

Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L., 2016b. Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. CCS '16, Association for Computing Machinery, New York, NY, USA, pp. 308–318. http://dx.doi.org/10.1145/2976749.2978318.

Aceto, G., Ciuonzo, D., Montieri, A., Pescapé, A., 2019a. Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges. IEEE Trans. Netw. Serv. Manag. 16 (2), 445–458.

Aceto, G., Persico, V., Pescapé, A., 2019b. A survey on information and communication technologies for industry 4.0: State-of-the-art, taxonomies, perspectives, and challenges. IEEE Commun. Surv. Tuts. 21 (4), 3467–3501.

Addad, R.A., Taleb, T., Flinck, H., Bagaa, M., Dutra, D., 2020. Network slice mobility in next generation mobile systems: Challenges and potential solutions. IEEE Netw. 34 (1), 84–93. http://dx.doi.org/10.1109/MNET.2019.1800268.

Akbari Torkestani, J., 2012. Mobility prediction in mobile wireless networks. J. Netw. Comput. Appl. 35 (5), 1633–1645, service Delivery Management in Broadband Networks.

Al-Dhuraibi, Y., Paraiso, F., Djarallah, N., Merle, P., 2017. Elasticity in cloud computing: state of the art and research challenges. IEEE Trans. Serv. Comput. 11 (2), 430–447.

Al-Garadi, M.A., Mohamed, A., Al-Ali, A., Du, X., Ali, I., Guizani, M., 2020. A survey of machine and deep learning methods for internet of things (IoT) security. IEEE Commun. Surv. Tuts. 22 (3), 1646–1685.

Al-Hawawreh, M., Moustafa, N., Sitnikova, E., 2018. Identification of malicious activities in industrial internet of things based on deep learning models. J. Inf. Secur. Appl. 41, 1–11. http://dx.doi.org/10.1016/j.jisa.2018.05.002.

Alsheikh, M.A., Lin, S., Niyato, D., Tan, H.-P., 2016. Rate–distortion balanced data compression for wireless sensor networks. IEEE Sensors J. 16 (12), 5072–5083.

Alsheikh, M.A., Selim, A., Niyato, D., Doyle, L., Lin, S., Tan, H.-P., 2015. Deep activity recognition models with triaxial accelerometers. arXiv preprint arXiv:1511.04664.

Amann, J., Blasimme, A., Vayena, E., Frey, D., Madai, V.I., 2020. Explainability for artificial intelligence in healthcare: a multidisciplinary perspective. BMC Med. Inform. Decis. Making 20 (1), 310. http://dx.doi.org/10.1186/s12911-020-01332-6.

Anagnostopoulos, T., Anagnostopoulos, C., Hadjiefthymiades, S., 2011. Mobility prediction based on machine learning. In: 2011 IEEE 12th International Conference on Mobile Data Management, Vol. 2. pp. 27–30. http://dx.doi.org/10.1109/MDM.2011.60.

Angelov, P., Soares, E., 2020. Towards explainable deep neural networks (xDNN). Neural Netw. 130, 185–194. http://dx.doi.org/10.1016/j.neunet.2020.07.010, https://www.sciencedirect.com/science/article/pii/S0893608020302513.

Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J.L., 2012. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In: International Workshop on Ambient Assisted Living. Springer, pp. 216–223.

Anon, 2019. CHANTS'19: Proceedings of the 14th Workshop on Challenged Networks. Association for Computing Machinery, New York, NY, USA.

Aradi, S., 2020. Survey of deep reinforcement learning for motion planning of autonomous vehicles. IEEE Intell. Transp. Syst. Mag..

Balogh, T., Luknárová, D., Medvecký, M., 2010. Performance of round robin-based queue scheduling algorithms. In: 2010 Third International Conference on Communication Theory, Reliability, and Quality of Service. pp. 156–161. http://dx.doi.org/10.1109/CTRQ.2010.34.

Baron, B., Spathis, P., Dias de Amorim, M., Viniotis, Y., Ammar, M.H., 2019. Mobility as an alternative communication channel: A survey. IEEE Commun. Surv. Tuts. 21 (1), 289–314.

Battaglia, P., Hamrick, J.B.C., Bapst, V., Sanchez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G.E., Vaswani, A., Allen, K., Nash, C., Langston, V.J., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M., Vinyals, O., Li, Y., Pascanu, R., 2018. Relational inductive biases, deep learning, and graph networks. arXiv.

Bhattacharyya, R., Bura, A., Rengarajan, D., Rumuly, M., Shakkottai, S., Kalathil, D., Mok, R.K., Dhamdhere, A., 2019. QFlow: A reinforcement learning approach to high QoE video streaming over wireless networks. In: Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 251–260.

Bianchi, V., Bassoli, M., Lombardo, G., Fornacciari, P., Mordonini, M., De Munari, I., 2019. IoT wearable sensor and deep learning: An integrated approach for personalized human activity recognition in a smart home environment. IEEE Internet Things J. 6 (5), 8553–8562.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, Berlin, Heidelberg.

Caldas, S., Wu, P., Li, T., Konečný, J., McMahan, H.B., Smith, V., Talwalkar, A., 2019. LEAF: A benchmark for federated settings. In: Workshop on Federated Learning for Data Privacy and Confidentiality (NeurIPS 2019).

Cao, Y., Sun, Z., 2013. Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. IEEE Commun. Surv. Tuts. 15 (2), 654–677.

Catal, C., Tufekci, S., Pirmit, E., Kocabag, G., 2015. On the use of ensemble of classifiers for accelerometer-based activity recognition. Appl. Soft Comput. 37, 1018–1022.

Cenedese, A., Zanella, A., Vangelista, L., Zorzi, M., 2014. Padova smart city: An urban internet of things experimentation. In: Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014. IEEE, pp. 1–6.

Chen, M., Challita, U., Saad, W., Yin, C., Debbah, M., 2019. Artificial neural networks-based machine learning for wireless networks: A tutorial. IEEE Commun. Surv. Tuts. 21 (4), 3039–3071.

Chen, Z., Wu, M., Zhao, R., Guretno, F., Yan, R., Li, X., 2020a. Machine remaining useful life prediction via an attention-based deep learning approach. IEEE Trans. Ind. Electron. 68 (3), 2521–2531.

Chen, Y., Zheng, B., Zhang, Z., Wang, Q., Shen, C., Zhang, Q., 2020b. Deep learning on mobile and embedded devices: State-of-the-art, challenges, and future directions. ACM Comput. Surv. 53 (4), http://dx.doi.org/10.1145/3398209.

Cheng, Y., Zhu, H., Wu, J., Shao, X., 2018. Machine health monitoring using adaptive kernel spectral clustering and deep long short-term memory recurrent neural networks. IEEE Trans. Ind. Inform. 15 (2), 987–997.

Cioffi, R., Travaglioni, M., Piscitelli, G., Petrillo, A., De Felice, F., 2020. Artificial intelligence and machine learning applications in smart production: Progress, trends, and directions. Sustainability 12 (2), http://dx.doi.org/10.3390/su12020492.

Cisco, 2020. Cisco annual internet report (2018–2023) white paper. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

Cools, S.-B., Gershenson, C., D'Hooghe, B., 2013. Self-organizing traffic lights: A realistic simulation. In: Advances in Applied Self-Organizing Systems. Springer, pp. 45–55.

Dave, D., Naik, H., Singhal, S., Patel, P., 2020. Explainable AI meets healthcare: A study on heart disease dataset. CoRR abs/2011.03195.

de Jonge, B., Teunter, R., Tinga, T., 2017. The influence of practical factors on the benefits of condition-based maintenance over time-based maintenance. Reliab. Eng. Syst. Saf. 158, 21–30.

De La Torre Parra, G., Rad, P., Choo, K.-K.R., Beebe, N., 2020. Detecting internet of things attacks using distributed deep learning. J. Netw. Comput. Appl. 163, 102662. http://dx.doi.org/10.1016/j.jnca.2020.102662.

de Medeiros, D.S., Campista, M.E.M., Mitton, N., de Amorim, M.D., Pujolle, G., 2017. The power of quasi-shortest paths: ρ-geodesic betweenness centrality. IEEE Trans. Netw. Sci. Eng. 4 (3), 187–200.

Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16. Curran Associates Inc., Red Hook, NY, USA, pp. 3844–3852.

Deisenroth, M.P., Faisal, A.A., Ong, C.S., 2019. Mathematics for Machine Learning. Cambridge University Press, Cambridge.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 248–255.

Dimitropoulos, K., Barmpoutis, P., Grammalidis, N., 2016. Higher order linear dynamical systems for smoke detection in video surveillance applications. IEEE Trans. Circuits Syst. Video Technol. 27 (5), 1143–1154.

Dou, W., Zhao, X., Yin, X., Wang, H., Luo, Y., Qi, L., 2020. Edge Computing-Enabled Deep Learning for Real-Time Video Optimization in IIoT. IEEE Trans. Ind. Inform..

Drolia, U., Guo, K., Tan, J., Gandhi, R., Narasimhan, P., 2017. Cachier: Edge-caching for recognition applications. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). pp. 276–286. http://dx.doi.org/10.1109/ICDCS.2017.94.

Du, K., Pervaiz, A., Yuan, X., Chowdhery, A., Zhang, Q., Hoffmann, H., Jiang, J., 2020. Server-driven video streaming for deep learning inference. In: Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication, SIGCOMM '20. Association for Computing Machinery, New York, NY, USA, pp. 557–570. http://dx.doi.org/10.1145/3387514.3405887.

Du, R., Santi, P., Xiao, M., Vasilakos, A.V., Fischione, C., 2019. The sensable city: A survey on the deployment and management for smart city monitoring. IEEE Commun. Surv. Tuts. 21 (2), 1533–1560.

Duc, T.L., Leiva, R.G., Casari, P., Östberg, P.-O., 2019. Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey. ACM Comput. Surv. 52 (5), http://dx.doi.org/10.1145/3341145.

Dwork, C., 2011. A firm foundation for private data analysis. Commun. ACM 54 (1), 86–95. http://dx.doi.org/10.1145/1866739.1866758.

Dwork, C., Roth, A., 2014. The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. 9 (3–4), 211–407. http://dx.doi.org/10.1561/0400000042.

Ferrag, M.A., Shu, L., Djallel, H., Choo, K.-K.R., 2021. Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0. Electronics 10 (11), http://dx.doi.org/10.3390/electronics10111257, https://www.mdpi.com/2079-9292/10/11/1257.

Gantert, L., Sammarco, M., Detyniecki, M., Campista, M.E.M., 2021. A supervised approach for corrective maintenance using spectral features from industrial sounds. In: IEEE 7th World Forum on Internet of Things (WF-IoT).

Ghosh, A., Chung, J., Yin, D., Ramchandran, K., 2020. An efficient framework for clustered federated learning. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (Eds.), Advances in Neural Information Processing Systems, Vol. 33. pp. 19586–19597, Curran Associates, Inc..

Ghosh, A.M., Grolinger, K., 2019. Deep learning: Edge-cloud data analytics for IoT. In: 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE). IEEE, pp. 1–7.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, pp. 249–256.

Gochoo, M., Tan, T.-H., Velusamy, V., Liu, S.-H., Bayanduuren, D., Huang, S.-C., 2017. Device-free non-privacy invasive classification of elderly travel patterns in a smart house using pir sensors and dcnn. IEEE Sensors J. 18 (1), 390–400.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

Grando, F., Granville, L.Z., Lamb, L.C., 2019. Machine learning in network centrality measures: Tutorial and outlook. ACM Comput. Surv. 51 (5), 102:1–102:32. http://dx.doi.org/10.1145/3237192.

Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M., 2013. Internet of things (IoT): A vision, architectural elements, and future directions. Future Gener. Comput. Syst. 29 (7), 1645–1660.

Gupta, S., Kar, A.K., Baabdullah, A., Al-Khowaiter, W.A., 2018. Big data with cognitive computing: A review for the future. Int. J. Inf. Manage. 42, 78–89.

Gupta, D., Kayode, O., Bhatt, S., Gupta, M., Tosun, A.S., 2020. Learner's dilemma: IoT devices training strategies in collaborative deep learning. In: 2020 IEEE 6th World Forum on Internet of Things (WF-IoT). IEEE, pp. 1–6.

Hammerla, N.Y., Halloran, S., Plötz, T., 2016. Deep, convolutional, and recurrent models for human activity recognition using wearables. arXiv preprint arXiv:1604.08880.

Han, S., Pool, J., Tran, J., Dally, W.J., 2015. Learning both weights and connections for efficient neural networks. arXiv preprint arXiv:1506.02626.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778. http://dx.doi.org/10.1109/CVPR.2016.90.

Ho, S., Qu, Y., Gu, B., Gao, L., Li, J., Xiang, Y., 2021. DP-GAN: Differentially private consecutive data publishing using generative adversarial nets. J. Netw. Comput. Appl. 185, 103066. http://dx.doi.org/10.1016/j.jnca.2021.103066.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. http://arxiv.org/abs/1704.04861.

Hussain, F., Hussain, R., Hassan, S.A., Hossain, E., 2020. Machine learning in IoT security: Current solutions and future challenges. IEEE Commun. Surv. Tuts. http://dx.doi.org/10.1109/COMST.2020.2986444, http://arxiv.org/abs/1904.05735.

Hutter, F., Kotthoff, L., Vanschoren, J., 2019. Automated machine learning, first ed. In: The Springer Series on Challenges in Machine Learning, Springer International Publishing, http://dx.doi.org/10.1007/978-3-030-05318-5.

Isele, D., Rahimi, R., Cosgun, A., Subramanian, K., Fujimura, K., 2018. Navigating occluded intersections with autonomous vehicles using deep reinforcement learning. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 2034–2039.

Kafle, V., Fukushima, Y., Martinez-Julia, P., Miyazawa, T., 2018. Consideration on automation of 5G network slicing with machine learning. In: 2018 ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K). pp. 1–8. http://dx.doi.org/10.23919/ITU-WT.2018.8597639.

Kang, D., Emmons, J., Abuzaid, F., Bailis, P., Zaharia, M., 2017. Noscope: Optimizing neural network queries over video at scale. Proc. VLDB Endow. 10 (11), 1586–1597. http://dx.doi.org/10.14778/3137628.3137664.

Kaur, K., 2018. A survey on internet of things – architecture, applications, and future trends. In: 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC). pp. 581–583. http://dx.doi.org/10.1109/ICSCCC.2018.8703341.

Khalil, E., Dai, H., Zhang, Y., Dilkina, B., Song, L., 2017. Learning combinatorial optimization algorithms over graphs. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Vol. 30. Curran Associates, Inc, pp. 6348–6358.

Khan, R., Kumar, P., Jayakody, D.N.K., Liyanage, M., 2020. A survey on security and privacy of 5G technologies: Potential solutions, recent advancements, and future directions. IEEE Commun. Surv. Tuts. 22 (1), 196–248.

Khan, S., Muhammad, K., Mumtaz, S., Baik, S.W., de Albuquerque, V.H.C., 2019. Energy-efficient deep CNN for smoke detection in foggy IoT environment. IEEE Internet Things J. 6 (6), 9237–9245.

Kibria, M.G., Nguyen, K., Villardi, G.P., Zhao, O., Ishizu, K., Kojima, F., 2018. Big data analytics machine learning and artificial intelligence in next-generation wireless networks. IEEE Access 6, 32328–32338.

Konečný, J., McMahan, H., Ramage, D., Richtárik, P., 2016. Federated optimization: Distributed machine learning for on-device intelligence.

Kong, W., Dong, Z.Y., Jia, Y., Hill, D.J., Xu, Y., Zhang, Y., 2017. Short-term residential load forecasting based on LSTM recurrent neural network. IEEE Trans. Smart Grid 10 (1), 841–851.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 25, 1097–1105.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. Imagenet classification with deep convolutional neural networks. Commun. ACM 60 (6), 84–90. http://dx.doi.org/10.1145/3065386.

Lago, T.K., González, E.R., Campista, M.E.M., 2021. Towards a real-time system based on regression model to evaluate driver's attention. In: 2021 7th IEEE International Smart Cities Conference (ISC2), pp. 1–7.

Lara, O.D., Labrador, M.A., 2012. A survey on human activity recognition using wearable sensors. IEEE Commun. Surv. Tuts. 15 (3), 1192–1209.

Laskaridis, S., Venieris, S.I., Almeida, M., Leontiadis, I., Lane, N.D., 2020. SPINN: Synergistic progressive inference of neural networks over device and cloud. In: Proceedings of the 26th Annual International Conference on Mobile Computing and Networking. Association for Computing Machinery, New York, NY, USA, pp. 1–15.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.

LeCun, Y., Bengio, Y., et al., 1995. Convolutional networks for images, speech, and time series. In: The HandBook of Brain Theory and Neural Networks, Vol. 3361, (10), p. 1995.

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D., 1989. Backpropagation applied to handwritten zip code recognition. Neural Comput. 1 (4), 541–551. http://dx.doi.org/10.1162/neco.1989.1.4.541.

Lei, L., Tan, Y., Zheng, K., Liu, S., Zhang, K., Shen, X., 2020. Deep reinforcement learning for autonomous internet of things: Model, applications and challenges. IEEE Commun. Surv. Tuts. 22 (3), 1722–1760.

Li, N., Guo, H., Xu, D., Wu, X., 2014. Multi-scale analysis of contextual information within spatio-temporal video volumes for anomaly detection. In: 2014 IEEE International Conference on Image Processing (ICIP). IEEE, pp. 2363–2367.

Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P., 2016. Pruning filters for efficient convnets. arXiv preprint arXiv:1608.08710.

Li, G., Peng, S., Wang, C., Niu, J., Yuan, Y., 2018a. An energy-efficient data collection scheme using denoising autoencoder in wireless sensor networks. Tsinghua Sci. Technol. 24 (1), 86–96.

Li, T., Sahu, A., Talwalkar, A., Smith, V., 2020. Federated learning: Challenges, methods, and future directions. IEEE Signal Process. Mag. 37, 50–60. http://dx.doi.org/10.1109/MSP.2020.2975749.

Li, et al., 2018b. Deep learning for smart industry: Efficient manufacture inspection system with fog computing. IEEE Trans. Ind. Inform. 14 (10), 4665–4673. http://dx.doi.org/10.1109/TII.2018.2842821.

Liang, L., Ye, H., Li, G.Y., 2019. Toward intelligent vehicular networks: A machine learning framework. IEEE Internet Things J. 6 (1), 124–135. http://dx.doi.org/10.1109/JIOT.2018.2872122, http://arxiv.org/abs/1804.00338.

Liang, F., Yu, W., Liu, X., Griffith, D., Golmie, N., 2020. Toward edge-based deep learning in industrial internet of things. IEEE Internet Things J. 7 (5), 4329–4341.

Liberati, A., Altman, D., Tetzlaff, J., Mulrow, C., Gøtzsche, P., Ioannidis, J., Clarke, M., Devereaux, P.J., Kleijnen, J., Moher, D., 2009. The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: Explanation and elaboration. J. Clin. Epidemiol. 62 (10), e1–e34.

Lim, W.Y.B., Luong, N.C., Hoang, D.T., Jiao, Y., Liang, Y.C., Yang, Q., Niyato, D., Miao, C., 2020. Federated learning in mobile edge networks: A comprehensive survey. IEEE Commun. Surv. Tuts. 22 (3), 2031–2063. http://dx.doi.org/10.1109/COMST.2020.2986024.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. SSD: Single shot multibox detector. In: European Conference on Computer Vision. Springer, pp. 21–37.

Liu, Y., Bi, S., Shi, Z., Hanzo, L., 2020. When machine learning meets big data: A wireless communication perspective. IEEE Veh. Technol. Mag. 15 (1), 63–72.

Luo, L., Xie, L., Su, H., 2020. Deep learning with tensor factorization layers for sequential fault diagnosis and industrial process monitoring. IEEE Access 8, 105494–105506.

Luong, N.C., Hoang, D.T., Gong, S., Niyato, D., Wang, P., Liang, Y., Kim, D.I., 2019. Applications of deep reinforcement learning in communications and networking: A survey. IEEE Commun. Surv. Tuts. 21 (4), 3133–3174.

Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.-Y., 2014. Traffic flow prediction with big data: a deep learning approach. IEEE Intell. Transp. Syst. Mag. 16 (2), 865–873.

Ma, X., Yao, T., Hu, M., Dong, Y., Liu, W., Wang, F., Liu, J., 2019. A survey on deep learning empowered IoT applications. IEEE Access 7, 181721–181732.

MacHardy, Z., Khan, A., Obana, K., Iwashina, S., 2018. V2X access technologies: Regulation, research, and remaining challenges. IEEE Commun. Surv. Tuts. 20 (3), 1858–1877.

Mahmoud, M.A., Guo, P., Wang, K., 2020. Pseudoinverse learning autoencoder with DCGAN for plant diseases classification. Multimedia Tools Appl. 79 (35), 26245–26263.

Makhdoom, I., Abolhasan, M., Lipman, J., Liu, R.P., Ni, W., 2019. Anatomy of threats to the internet of things. IEEE Commun. Surv. Tuts. 21 (2), 1636–1675.

Malik, O.A., Becker, S., 2018. Low-rank tucker decomposition of large tensors using TensorSketch. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18. Curran Associates Inc., Red Hook, NY, USA, pp. 10117–10127.

Mao, Q., Hu, F., Hao, Q., 2018. Deep learning for intelligent wireless networks: A comprehensive survey. IEEE Commun. Surv. Tuts. 20 (4), 2595–2621. http://dx.doi.org/10.1109/COMST.2018.2846401.

Mao, H., Netravali, R., Alizadeh, M., 2017. Neural adaptive video streaming with pensieve. In: Proceedings of the Conference of the ACM Special Interest Group on Data Communication, SIGCOMM '17. Association for Computing Machinery, New York, NY, USA, pp. 197–210. http://dx.doi.org/10.1145/3098822.3098843.

Marfoq, O., Xu, C., Neglia, G., Vidal, R., 2020. Throughput-optimal topology design for cross-silo federated learning. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (Eds.), Advances in Neural Information Processing Systems, Vol. 33. Curran Associates, Inc, pp. 19478–19487, https://proceedings.neurips.cc/paper/2020/file/e29b722e35040b88678e25a1ec032a21-Paper.pdf.

Maskelinas, R., Damaševičius, R., Segal, S., 2019. A review of internet of things technologies for ambient assisted living environments. Future Internet 11 (12), http://dx.doi.org/10.3390/fi11120259.

Mathebula, I., Isong, B., Gasela, N., Abu-Mahfouz, A.M., 2019. Analysis of SDN-based security challenges and solution approaches for SDWSN usage. In: 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE), pp. 1288–1293.

McMahan, H.B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A., 2017. Communication-efficient learning of deep networks from decentralized data. http://arxiv.org/abs/1602.05629.

Medeiros, D.S.V., Campista, M.E.M., Mitton, N., Dias de Amorim, M., Pujolle, G., 2016. Weighted betweenness for multipath networks. In: Proc. of the Global Information Infrastructure and Networking Symposium (GIIS '16), pp. 1–6.

Mehmood, Y., Ahmad, F., Yaqoob, I., Adnane, A., Imran, M., Guizani, S., 2017. Internet-of-things-based smart cities: Recent advances and challenges. IEEE Commun. Mag. 55 (9), 16–24.

Mohammadi, M., Al-Fuqaha, A., Sorour, S., Guizani, M., 2018. Deep learning for IoT big data and streaming analytics: A survey. IEEE Commun. Surv. Tuts. 20 (4), 2923–2960.

Mousavi, S.M., Abyaneh, H.A., 2010. Effect of load models on probabilistic characterization of aggregated load patterns. IEEE Trans. Power Syst. 26 (2), 811–819.

Nascita, A., Montieri, A., Aceto, G., Ciuonzo, D., Persico, V., Pescapé, A., 2021. XAI meets mobile traffic classification: Understanding and improving multimodal deep learning architectures. IEEE Trans. Netw. Serv. Manag..

Nguyen, T.T., Armitage, G., 2008. A survey of techniques for internet traffic classification using machine learning. IEEE Commun. Surv. Tuts. 10 (4), 56–76.

Nguyen, D.C., Cheng, P., Ding, M., Lopez-Perez, D., Pathirana, P.N., Li, J., Seneviratne, A., Li, Y., Poor, H.V., 2020. Enabling AI in future wireless networks: A data life cycle perspective. IEEE Commun. Surv. Tuts. Early Access http://dx.doi.org/10.1109/COMST.2020.3024783.

Ning, Z., Zhang, K., Wang, X., Obaidat, M.S., Guo, L., Hu, X., Hu, B., Guo, Y., Sadoun, B., Kwok, R.Y., 2020. Joint computing and caching in 5G-envisioned internet of vehicles: A deep reinforcement learning-based traffic control system. IEEE Intell. Transp. Syst. Mag..

Noble, W.S., 2006. What is a support vector machine? Nature Biotechnol. 24 (12), 1565–1567.

Okafor, N.U., Alghorani, Y., Delaney, D.T., 2020. Improving data quality of low-cost IoT sensors in environmental monitoring networks using data fusion and machine learning approach. ICT Express 6 (3), 220–228.

Osherson, D., Stob, M., Weinstein, S., 1991. A universal inductive inference machine. J. Symbolic Logic 56 (2), 661–672. http://dx.doi.org/10.2307/2274708.

Ozturk, M., Gogate, M., Onireti, O., Adeel, A., Hussain, A., Imran, M.A., 2019. A novel deep learning driven low-cost mobility prediction approach for 5G cellular networks: The case of the control/data separation architecture (CDSA). Neurocomputing 358, 479–489.

Pacheco, R.G., Couto, R.S., 2020. Inference time optimization using BranchyNet partitioning. In: 2020 IEEE Symposium on Computers and Communications (ISCC). http://dx.doi.org/10.1109/iscc50000.2020.9219647.

Pacheco, R.G., Couto, R.S., Simeone, O., 2021. Calibration-aided edge inference offloading via adaptive model partitioning of deep neural networks. In: 2021 IEEE Symposium on Computers and Communications (ISCC).

Page, L., Brin, S., Motwani, R., Winograd, T., 1999. The PageRank citation ranking: Bringing order to the web. Tech. rep., Stanford InfoLab.

Panwar, P., Lokesh, Shailesh, 2019. Implementation of machine learning algorithms on CICIDS-2017 dataset for intrusion detection using WEKA. Int. J. Recent Technol. Eng. 8, 2195–2207. http://dx.doi.org/10.35940/ijrte.C4587.098319.

Peng, Z., Gao, S., Li, Z., Xiao, B., Qian, Y., 2018. Vehicle safety improvement through deep learning and mobile sensing. IEEE Netw. 32 (4), 28–33.

Pierucci, L., Micheli, D., 2016. A neural network for quality of experience estimation in mobile communications. IEEE Multimedia 23 (4), 42–49.

Pope, P.E., Kolouri, S., Rostami, M., Martin, C.E., Hoffmann, H., 2019. Explainability methods for graph convolutional neural networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10764–10773. http://dx.doi.org/10.1109/CVPR.2019.01103.

Pundir, M., Sandhu, J.K., 2021. A systematic review of quality of service in wireless sensor networks using machine learning: Recent trend and future vision. J. Netw. Comput. Appl. 103084. http://dx.doi.org/10.1016/j.jnca.2021.103084.

Qian, B., Su, J., Wen, Z., Jha, D.N., Li, Y., Guan, Y., Puthal, D., James, P., Yang, R., Zomaya, A.Y., Rana, O., Wang, L., Koutny, M., Ranjan, R., 2020. Orchestrating the development lifecycle of machine learning-based IoT applications: A taxonomy and survey. ACM Comput. Surv. 53 (4), http://dx.doi.org/10.1145/3398020.

Radford, A., Metz, L., Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

Ravi, D., Wong, C., Lo, B., Yang, G.-Z., 2016a. A deep learning approach to on-node sensor data analytics for mobile or wearable devices. IEEE J. Biomed. Health Inform. 21 (1), 56–64.

Ravi, D., Wong, C., Lo, B., Yang, G.-Z., 2016b. Deep learning for human activity recognition: A resource efficient implementation on low-power devices. In: 2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN). IEEE, pp. 71–76.

Ray, S., Jin, Y., Raychowdhury, A., 2016. The changing computing paradigm with internet of things: A tutorial introduction. IEEE Des. Test. 33 (2), 76–96.

Redmon, J., Farhadi, A., 2016. Yolo9000: Better, faster, stronger. http://arxiv.org/abs/1612.08242.

Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. arXiv.

Reis, L.H.A., Magalhães, L.C.S., de Medeiros, D.S.V., Mattos, D.M.F., 2020. An unsupervised approach to infer quality of service for large-scale wireless networking. J. Netw. Syst. Manage..

Ronao, C.A., Cho, S.-B., 2016. Human activity recognition with smartphone sensors using deep learning neural networks. Expert Syst. Appl. 59, 235–244.

Scardapane, S., Scarpiniti, M., Baccarelli, E., Uncini, A., 2020. Why should we add early exits to neural networks? Cogn. Comput. 12, http://dx.doi.org/10.1007/s12559-020-09734-4.

Schoellhammer, T., Greenstein, B., Osterweil, E., Wimbrow, M., Estrin, D., 2004. Lightweight temporal compression of microclimate datasets [wireless sensor networks]. In: 29th Annual IEEE International Conference on Local Computer Networks, pp. 516–524.

Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A., 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: International Conference on Information Systems Security and Privacy (ICISSP). pp. 108–116. http://dx.doi.org/10.5220/0006639801080116.

Sheng, Z., Mahapatra, C., Zhu, C., Leung, V.C.M., 2015. Recent advances in industrial wireless sensor networks toward efficient management in IoT. IEEE Access 3, 622–637. http://dx.doi.org/10.1109/ACCESS.2015.2435000.

Shi, Y., Yang, K., Jiang, T., Zhang, J., Letaief, K.B., 2020. Communication-efficient edge AI: Algorithms and systems. IEEE Commun. Surv. Tuts. 22 (4), 2167–2191. http://dx.doi.org/10.1109/COMST.2020.3007787.

Shokri, R., Shmatikov, V., 2015. Privacy-preserving deep learning. In: 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton). pp. 909–910. http://dx.doi.org/10.1109/ALLERTON.2015.7447103.

Silva, B.M.C., Rodrigues, J.J.P.C, Kumar, N., Han, G., 2017. Cooperative strategies for challenged networks and applications: A survey. IEEE Syst. J. 11 (4), 2749–2760.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May (2015) 7-9, Conference Track Proceedings, pp. 1–14.

Singh, D., Mohan, C.K., 2018. Deep spatio-temporal representation for detection of road accidents using stacked autoencoder. IEEE Trans. Intell. Transp. Syst. 20 (3), 879–887.

Stabinger, S., Peer, D., Rodríguez-Sánchez, A., 2021. Arguments for the unsuitability of convolutional neural networks for non-local tasks. Neural Netw. 142, 171–179. http://dx.doi.org/10.1016/j.neunet.2021.05.001, https://www.sciencedirect.com/science/article/pii/S0893608021001775.

Stamoulis, D., Ding, R., Wang, D., Lymberopoulos, D., Priyantha, B., Liu, J., Marculescu, D., 2020. Single-path mobile AutoML: Efficient ConvNet design and NAS hyperparameter optimization. IEEE J. Sel. Topics Signal Process. 14 (4), 609–622. http://dx.doi.org/10.1109/JSTSP.2020.2971421.

Sun, G., Xiong, K., Boateng, G.O., Liu, G., Jiang, W., 2020. Resource slicing and customization in RAN with dueling deep Q-network. J. Netw. Comput. Appl. 157, 102573. http://dx.doi.org/10.1016/j.jnca.2020.102573.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9.

Tahsien, S.M., Karimipour, H., Spachos, P., 2020. Machine learning based solutions for security of internet of things (IoT): A survey. J. Netw. Comput. Appl. 161, 102630. http://dx.doi.org/10.1016/j.jnca.2020.102630.

Tan, K., Bremner, D., Kernec, J.L., Imran, M., 2020. Federated machine learning in vehicular networks: A summary of recent applications. In: 2020 International Conference on UK-China Emerging Technologies (UCET). pp. 1–4. http://dx.doi.org/10.1109/UCET51115.2020.9205482.

Tang, J., 2018. Intelligent Mobile Projects with TensorFlow. Packt Publishing.

Tang, F., Mao, B., Fadlullah, Z.M., Kato, N., Akashi, O., Inoue, T., Mizutani, K., 2017. On removing routing protocol from future wireless networks: A real-time deep learning approach for intelligent traffic control. IEEE Wirel. Commun. 25 (1), 154–160.

Teerapittayanon, S., McDanel, B., Kung, H., 2016. Branchynet: Fast inference via early exiting from deep neural networks. In: 2016 23rd International Conference on Pattern Recognition (ICPR). pp. 2464–2469. http://dx.doi.org/10.1109/ICPR.2016.7900006.

Thantharate, A., Paropkari, R., Walunj, V., Beard, C., 2019. Deepslice: A deep learning approach towards an efficient and reliable network slicing in 5G networks. In: 2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON). pp. 0762–0767. http://dx.doi.org/10.1109/UEMCON47517.2019.8993066.

Tong, W., Hussain, A., Bo, W.X., Maharjan, S., 2019. Artificial intelligence for vehicle-to-everything: A survey. IEEE Access 7, 10823–10843.

Toscano, M., Grunwald, F., Richart, M., Baliosian, J., Grampín, E., Castro, A., 2019. Machine learning aided network slicing. In: 2019 21st International Conference on Transparent Optical Networks (ICTON). pp. 1–4. http://dx.doi.org/10.1109/ICTON.2019.8840141.

Van der Pol, E., Oliehoek, F.A., 2016. Coordinated deep reinforcement learners for traffic light control. In: Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016).

Verbraeken, J., Wolting, M., Katzy, J., Kloppenburg, J., Verbelen, T., Rellermeyer, J.S., 2020. A survey on distributed machine learning. ACM Comput. Surv. 53 (2), http://dx.doi.org/10.1145/3377454.

Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A., Venkatraman, S., 2019. Deep learning approach for intelligent intrusion detection system. IEEE Access 7, 41525–41550. http://dx.doi.org/10.1109/ACCESS.2019.2895334.

Waheed, N., He, X., Ikram, M., Usman, M., Hashmi, S.S., Usman, M., 2020. Security and privacy in IoT using machine learning and blockchain: Threats and countermeasures. ACM Comput. Surv. 53 (6), http://dx.doi.org/10.1145/3417987.

Wan, L., Sun, Y., Sun, L., Ning, Z., Rodrigues, J.J., 2020. Deep learning based autonomous vehicle super resolution DOA estimation for safety driving. IEEE Intell. Transp. Syst. Mag..

Wang, M., Cui, Y., Wang, X., Xiao, S., Jiang, J., 2018. Machine learning for networking: Workflow, advances and opportunities. IEEE Netw. 32 (2), 92–99.

Wang, K., Guo, P., Xin, X., Ye, Z., 2017a. Autoencoder, low rank approximation and pseudoinverse learning algorithm. In: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, pp. 948–953.

Wang, J., Jiang, C., Zhang, H., Ren, Y., Chen, K.C., Hanzo, L., 2020. Thirty years of machine learning: The road to pareto-optimal wireless networks. IEEE Commun. Surv. Tuts. 22 (3), 1472–1514. http://dx.doi.org/10.1109/COMST.2020.2965856.

Wang, J., Liu, J., Kato, N., 2019. Networking and communications in autonomous driving: A survey. IEEE Commun. Surv. Tuts. 21 (2), 1243–1274.

Wang, H., Wu, Y., Min, G., Xu, J., Tang, P., 2019. Data-driven dynamic resource scheduling for network slicing: A deep reinforcement learning approach. Inform. Sci. 498, 106–116. http://dx.doi.org/10.1016/j.ins.2019.05.012.

Wang, Y., Yang, A., Chen, X., Wang, P., Wang, Y., Yang, H., 2017b. A deep learning approach for blind drift calibration of sensor networks. IEEE Sensors J. 17 (13), 4158–4171.

Wang, Y., Yang, A., Li, Z., Chen, X., Wang, P., Yang, H., 2016a. Blind drift calibration of sensor networks using sparse bayesian learning. IEEE Sensors J. 16 (16), 6249–6260.

Wang, L., Ye, H., Liang, L., Li, G.Y., 2020. Learn to compress CSI and allocate resources in vehicular networks. IEEE Commun. Mag. 68 (6), 3640–3653.

Wang, J., Zhang, X., Gao, Q., Yue, H., Wang, H., 2016b. Device-free wireless localization and activity recognition: A deep learning approach. IEEE Veh. Technol. Mag. 66 (7), 6258–6267.

Wang, X., Zhou, Z., Xiao, F., Xing, K., Yang, Z., Liu, Y., Peng, C., 2018. Spatio-temporal analysis and prediction of cellular traffic in metropolis. IEEE Trans. Mobile Comput. 18 (9), 2190–2202.

Wason, R., 2018. Deep learning: Evolution and expansion. Cogn. Syst. Res. 52, 701–708.

Watkins, C.J., Dayan, P., 1992. Q-learning. Mach. Learn. 8 (3–4), 279–292.

Wei, H., Zheng, G., Yao, H., Li, Z., 2018. Intellilight: A reinforcement learning approach for intelligent traffic light control. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2496–2505.

White, G., Clarke, S., 2018. Smart cities with deep edges. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, pp. 53–64.

White, G., Clarke, S., 2020. Urban intelligence with deep edges. IEEE Access 8, 7518–7530.

Wu, Z., Wang, J., Hu, L., Zhang, Z., Wu, H., 2020a. A network intrusion detection method based on semantic re-encoding and deep learning. J. Netw. Comput. Appl. 164 (1), 102688.

Wu, T., Zhou, P., Liu, K., Yuan, Y., Wang, X., Huang, H., Wu, D.O., 2020b. Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks. IEEE Veh. Technol. Mag. 69 (8), 8243–8256.

Xu, F., Li, Y., Wang, H., Zhang, P., Jin, D., 2017. Understanding mobile traffic patterns of large scale cellular towers in urban environment. IEEE/ACM Trans. Netw. 25 (2), 1147–1161. http://dx.doi.org/10.1109/TNET.2016.2623950.

Ye, H., Li, G.Y., Juang, B.-H.F., 2019. Deep reinforcement learning based resource allocation for V2V communications. IEEE Trans. Veh. Technol. 68 (4), 3163–3173.

Yi, L., Mak, M.-W., 2020. Improving speech emotion recognition with adversarial data augmentation network. IEEE Trans. Neural Netw. Learn. Syst..

Youssef, M., Mah, M., Agrawala, A., 2007. Challenges: device-free passive localization for wireless environments. In: Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking, pp. 222–229.

Yu, T., Wang, X., Shami, A., 2018. UAV-enabled spatial data sampling in large-scale IoT systems using denoising autoencoder neural network. IEEE Internet Things J. 6 (2), 1856–1865.

Zafari, F., Gkelias, A., Leung, K.K., 2019. A survey of indoor localization systems and technologies. IEEE Commun. Surv. Tuts. 21 (3), 2568–2599.

Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M., 2014. Internet of things for smart cities. IEEE Internet Things J. 1 (1), 22–32.

Zeng, L., Li, E., Zhou, Z., Chen, X., 2019. Boomerang: On-demand cooperative deep neural network inference for edge intelligence on the industrial internet of things. IEEE Netw. 33 (5), 96–103.

Zhang, H., Dai, L., 2019. Mobility prediction: A survey on state-of-the-art schemes and future applications. IEEE Access 7, 802–822. http://dx.doi.org/10.1109/ACCESS.2018.2885821.

Zhang, H., Hua, Y., Wang, C., Li, R., Zhao, Z., 2020. Deep Learning Based Traffic and Mobility Prediction. John Wiley & Sons, Ltd, pp. 119–136, Ch. 7.

Zhang, H., Liu, N., Chu, X., Long, K., Aghvami, A., Leung, V.C.M., 2017. Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges. IEEE Commun. Mag. 55 (8), 138–145. http://dx.doi.org/10.1109/MCOM.2017.1600940.

Zhang, T., Mao, S., 2020. Machine learning for end-to-end congestion control. IEEE Commun. Mag. 58 (6), 52–57. http://dx.doi.org/10.1109/MCOM.001.1900509.

Zhang, C., Patras, P., Haddadi, H., 2019. Deep learning in mobile and wireless networking: A survey. IEEE Commun. Surv. Tuts. 21 (3), 2224–2287. http://dx.doi.org/10.1109/comst.2019.2904897.

Zhang, Q., Yang, L.T., Yan, Z., Chen, Z., Li, P., 2018. An efficient deep learning model to predict cloud workload for industry informatics. IEEE Trans. Ind. Inform. 14 (7), 3170–3178.

Zhang, W., Zhang, Z., Chao, H., Guizani, M., 2019. Toward intelligent network optimization in wireless networking: An auto-learning framework. IEEE Wirel. Commun. 26 (3), 76–82.

Zhao, L., Huang, H., Li, X., Ding, S., Zhao, H., Han, Z., 2019. An accurate and robust approach of device-free localization with convolutional autoencoder. IEEE Internet Things J. 6 (3), 5825–5840.

Zhong, R.Y., Xu, X., Klotz, E., Newman, S.T., 2017. Intelligent manufacturing in the context of industry 4.0: a review. Engineering 3 (5), 616–630.

Zhu, J., Song, Y., Jiang, D., Song, H., 2017. A new deep-Q-learning-based transmission scheduling mechanism for the cognitive internet of things. IEEE Internet Things J. 5 (4), 2375–2385.

Zhu, L., Yu, F.R., Wang, Y., Ning, B., Tang, T., 2019. Big data analytics in intelligent transportation systems: A survey. IEEE Intell. Transp. Syst. Mag. 20 (1), 383–398. http://dx.doi.org/10.1109/TITS.2018.2815678.

Kaylani Bochie is an undergraduate student at Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil and an undergraduate research student at Grupo de Teleinformática e Automação (GTA/UFRJ). His main research interests are mobile networks, computer network security, and federated learning.

Mateus S. Gilbert is an undergraduate student at Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil and an undergraduate research student at Grupo de Teleinformática e Automação (GTA/UFRJ). His research interests are machine learning, data aggregation and wireless sensor networks.

Luana Gantert received her B.S. degree in Control and Automation Engineering from Federal Center for Technological Education Celso Suckow da Fonseca (CEFET-RJ) in 2017. She is currently pursuing a Master's degree at the Universidade Federal do Rio de Janeiro (UFRJ) in Electrical Engineering. Her research interests are industrial networks, internet of things, and machine learning.

Mariana S.M. Barbosa is a Ph.D. student at Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil. Mariana received her Master's degree on Electric Engineering from the Universidade Federal do Rio de Janeiro in 2019, and received her bachelor on Telecommunications Engineering from Universidade Federal Fluminense (UFF), Rio de Janeiro, Brazil, in 2010. Her main research interests are wireless and vehicular networks, complex networks and internet of things.

Dianne Scherly Varela de Medeiros is a professor at the Universidade Federal Fluminense (UFF). Dianne received her Master's degree on Telecommunications Engineering from UFF in 2013, and her D.Sc. degree on Electric Engineering from the Universidade Federal do Rio de Janeiro in 2017. She was in exchange from 2015 to 2016 to work on her thesis at the Laboratoire d'Informatique de Paris 6 (LIP6), at Sorbonne Université, Paris, France. Her main research interests are on wireless communications, complex networks and internet of things.

Miguel Elias M. Campista is an associate professor with the Electronic and Computer Engineering Department of Poli/UFRJ and a full professor with the Electrical Engineering Program (PEE) of COPPE/UFRJ. He received the Fellowship PQ2 from CNPq, was awarded Young Scientist of Rio de Janeiro State from FAPERJ, elected Affiliate Member of the Brazilian Academy of Science, and is currently on the board of directors of LARC. Miguel is an Associate Editor of the Annals of Telecommunications journal and an IEEE senior member. His research interests are on computer networking, data and network science.