Fog Computing
2025/26 - Autumn 2025 - UiO

Machine Learning

Prof. Paulo Ferreira
paulofe@ifi.uio.no
UiO/IFI/PT

Distributed Machine Learning for IoT Applications in the Fog. Rocha Neto, A.F., Delicato, F.C., Batista, T.V. and Pires, P.F. (2020). In Fog Computing (eds A. Zomaya, A. Abbas and S. Khan). https://doi.org/10.1002/9781119551713.ch12

Introduction (1/4)

It is not about IoT but about data

- Given the number of **physical objects on Earth**, if only a **tiny fraction of them are in fact sensors/actuators**:
  - IoT will certainly have the potential to be the largest **source of Big Data** on the planet and as such,
  - brings numerous research **challenges** to be solved to enjoy all the possible benefits of this paradigm
- Although the **IoT paradigm** is focused on the connection of objects:
  - its real potential lies not in the objects themselves, but in **the ability to generate valuable knowledge from the data extracted from these objects**
- It can be said that IoT actually is **not about things but about data**:
  - thus, the process of transforming the raw data collected through the perception of the physical environment in value-added information and, ultimately, in high-level knowledge capable of optimizing decision making, is **crucial to obtaining the real benefit of IoT**
- At the **heart of this transformation process is computational intelligence (CI)**:
  - in particular, **machine learning (ML) techniques are promising** to process the data generated in order to transform them into **information, knowledge, to predict trends, produce valuable insights, and guide automated decision-making processes**
  - however, **the use of machine learning (ML) techniques in IoT brings up several challenges**, especially regarding the computational requirements demanded by them

Introduction (2/4)

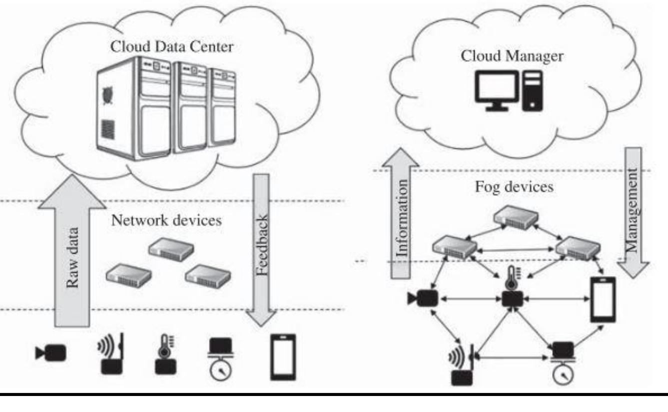
IoT devices can not perform complex processing

- The **IoT devices themselves may be part of this data transformation process:**
  - however, **with their limited resources, they generally cannot perform very complex processing**
- With their vast computing capabilities, **cloud platforms are the natural candidates as backend of IoT systems** and perform computationally intensive analysis and long-term data storage:
  - however, with the increasing growth in data generated by sensors and IoT devices, **their discriminated transmission to the cloud began to generate a set of problems**
- Among them:
  - the **excessive use of network bandwidth** has brought congestions and **poor performance** in the communication infrastructure
  - several IoT applications have **critical response time requirements**, and the **high and unpredictable latency of the Internet** for accessing remote data centers in the cloud is not acceptable in this context

Introduction (3/4)

minimizing network latency and improving computational performance

- The challenge of **minimizing network latency** and **improving computational performance** in the context of IoT environments has been recognized among the **main concerns of the IoT research community**
- In this context, the concept of **fog computing** was suggested by the industry and further extended by academic researchers



Introduction (4/4)

Where processing can happen

- It enables that **computations be executed on small datacenters close to the edge device**
- In this way, some **advanced data analytics and processing can be performed near the devices**, thus **reducing system response time and network traffic**
- The data produced by the IoT devices can then be **processed**:
  1. **in the data source itself** (also called the Things tier/layer),
  2. **in cloud platforms**, and
  3. **in the intermediate layer of fog/edge**
- Depending on the **application requirements** (such as response time) and the **complexity of the required processing technique**:
  - **each tier will be more suitable for performing the computation**

Big Data, Big Data Stream, and Data Stream Processing

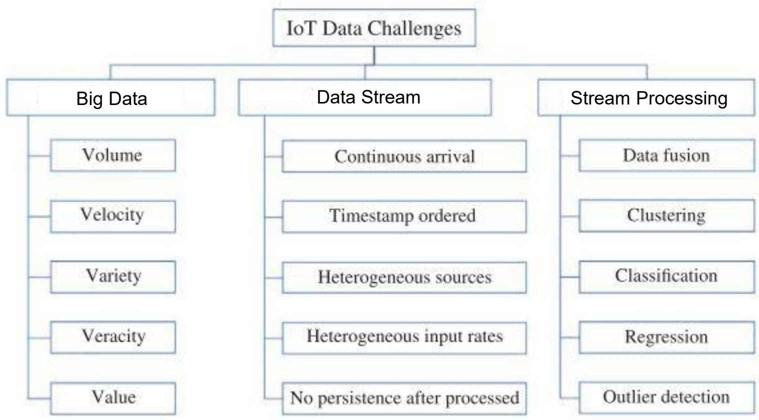
Challenges in Data Processing for IoT

- The **number of IoT devices/sensors has increased exponentially**
- A **plethora of applications can emerge**, exploiting the diversity of real-time monitored data, and combining them in different ways to provide value-added information to users
- **Examples** of advanced and intelligent applications are **smart home**, **smart building**, and **smart city**
- We discuss the challenges introduced by IoT systems in three **different dimensions**:
  - **generation**, **transmission**, and **processing of data**
- Such dimensions are typically related to and addressed by three well-known research fields:
  - **Big Data**,
  - **Big Data Stream**, and
  - **Data Stream Processing**

Big Data in IoT (1/5)

Big Data

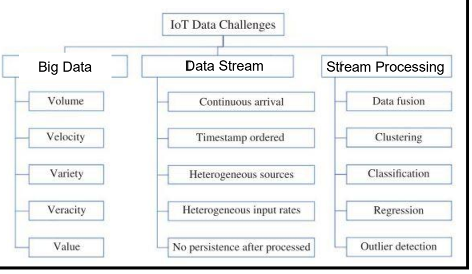
- **Big Data refers to voluminous and complex data sets** that require **advanced data processing** application software:
  - it involves **capturing, storing, sharing, searching, transferring, analyzing, visualizing, updating, and protecting** data



Big Data in IoT (2/5)

Big Data volume, velocity, …

- Data scientists have defined six **features** to address all the issues related to Big Data:
  1. **Volume** - key aspect to consider a data set to be "big," as the **size of the data** is measured as volume
  2. **Velocity** - related to the rate of data generation and transmission; the **huge amount and speed of IoT Big Data** coming from several sources demands the support for big data in real-time
  3. **Variety** - with increasing volume and velocity comes increasing variety, which describes the **huge diversity of data types** any IoT system can produce
  4. **Variability** - **different rates of data flow**, which may vary from time to time or place to place; e.g., in an IoT application for traffic management using sensors, the speed at which data is loaded in a database can have expressive variations depending on the heavy traffic hours
  5. **Veracity** - how accurate and how truthful is the data set; it is more than reliability of the data source, it also involves the **degree of confidence in the analysis of the data**; this feature is quite important in IoT system that uses crowd-sensing data
  6. **Value -** data must have value, i.e., all the infrastructure required to collect and interpret data on an IoT system must be justified by the **insights you can gather** to improve the user experience and services



Big Data in IoT (3/5)

Big Data freshness

- Another important aspect for the value of data is its **temporariness**:
  - the potential value of data depends on **data freshness** that needs to be processed
  - otherwise, **the processing results and actions become less valuable or even worthless**

- Three **categories for IoT data** (regarding different characteristics):
  - **data generation**,
  - **data quality**, and
  - **data interoperability**

- **Data generation**:
  - related to the Big Data features aforementioned such as **velocity, scalability, dynamics, and heterogeneity**

Big Data in IoT (4/5)

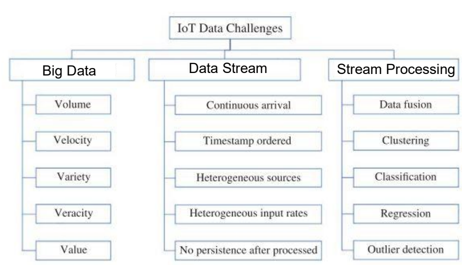
- **Data quality**:
  - **Uncertainty** - in IoT data, uncertainty can refer to **variance from the expected states of the data**, due to several reasons such as transmission errors or sensing precision
  - **Redundancy** - **errors in sensor reading logic can produce multiple data that can interfere in the semantic of sensor data**; e.g., in radio-frequency identification (RFID) data, the same tag can be read multiple times at the same location when there are multiple RFID readers at the same spot to improve the sensing
  - **Ambiguity** - refers to **different ways of interpreting data** from diverse things to distinct data consumers; this misunderstanding may interfere with the overall inference about the sensed environment
  - **Inconsistency** - **such as ambiguity, inconsistency may occur** in the interpretation of data that may produce misunderstandings; it is common to detect inconsistency in sensing data in case of several sensors monitoring an environment

Big Data in IoT (5/5)

- **Data interoperability**:
  - **Incompleteness** - in many IoT applications that detect and react to events in real time, the **combination of data from different sources is important to build a complete picture of the environment**; however, in this cooperation of mobile and distributed things who are generating the data, some problems related to the data quality from any source may negatively interfere in the whole system
  - **Semantics** - semantic technologies have been used to endow machines with the ability of **processing and interpreting IoT data**
- **Despite presenting all the aforementioned characteristics**:
  - the **data produced by IoT devices have additional features, not always observed in Big Data**
- **Typically, data produced by IoT sensors consist of**:
  - **time-series values**, which are **sampled over a defined period** and then **transmitted to a gateway for further processing**
- **Compared with Big Data produced by other sources, such as social data**, sensor data streams have additional features that call for a **shifting from the Big Data paradigm to the Big Data Stream paradigm**

Big Data Stream (1/3)

Data items arrive continuously and sequentially



- **Time-series data** are:
  - **not processed in the same way as the typical Big Data**,
  - **nor as easily interpretable such as, for instance, a document, video, or other data available on the Internet**
- **Data items**:
  - **arrive continuously and sequentially** as a stream, and
  - **usually ordered by a timestamp value** besides including other additional attribute values about the data item
- **Differently from Big Data** that is typically produced in controlled and owned data warehouses:
  - **data streams are usually generated by heterogeneous and scattered sources**
- In general, **IoT systems do not have direct access or control over such data sources**
- Moreover, **the input characteristics of a data stream are usually not controllable and typically unpredictable**

Big Data Stream (2/3)

Data items arrive in different rates

- Data streams are provided in **different rates**:
  - e.g., **from small number of bytes per second to several gigabits**
- The inherent nature of the input **does not allow one to easily make multiple passes over a data stream while processing** (and still retaining the usefulness of the data)
- **Big Data stream–oriented systems** need to:
  - **react effectively to changes**,
  - and **provide smart, semi-autonomous behavior to allocate resources for data processing**,
  - thus **implementing scalable and cost-effective services**
- The inherent **inertia of Big Data approaches, that commonly rely on batch-based processing**:
  - are **not proper for data processing** in IoT contexts with dynamism and real-time requirements
- The **Big Data stream paradigm**:
  - **enables ad-hoc and real-time processing** to connect streams of data and consumers,
  - benefiting **scalability, dynamic configuration, and management of heterogeneous data formats**

Big Data Stream (3/3)

Data items arrive are heterogeneous

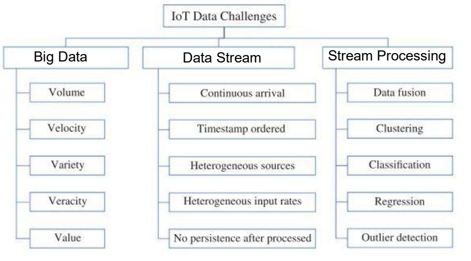
Data items are not kept

- Although **Big Data and Big Data stream cope with massive numerous heterogeneous data**:
  - **Big Data centers on the batch analysis of data**,
  - **Big Data stream deals with the management of data flows and real-time data analysis**
- **This feature has an impact** also on the **data that are considered relevant to consumer** applications:
  - **while for Big Data applications it is important to keep all sensed data in order to be able to perform any required computation**,
  - **Big Data stream applications might decide to perform real-time data processing on the raw data produced by IoT devices** to reduce the **latency** in transmitting the results to consumers, with **no need to persist such raw data**
- **Translating the huge amount of sensor generated data streams from their raw state into higher-level representations** and making **them accessible and understandable for humans or interpretable by machines and decision-making systems**:
  - one **promising approach to tackle this issue is by applying techniques of CI**
  - **CI refers to the ability of a computer system to learn a specific task from data or experimental observation**

Data Stream Processing (1/2)

Requires some processing applied to each data item

- **Data stream processing is a paradigm that deals with**:
  - **a sequence of data** (a stream),
  - and **a series of operations being applied to each element in the stream**
- **Integration with several data sources is particularly relevant for time-sensitive IoT applications**:
  - deal with **data provided by multiple data sources**, and
  - a **timely fusion of data is needed** to bring all pieces of data together
- **Information fusion and sharing** play a critical role for **fast analysis** and consequently providing **reliable and accurate actionable insights**



Data Stream Processing (2/2)

Information fusion in data stream processing

- **Information Fusion** deals with **three levels of data abstraction**:
  - **measurement**,
  - **feature**, and
  - **decision**,
- It can be **classified into these categories**:
  - **low-level fusion** - also referred to as signal (measurement) level fusion; **inputs are raw data that are joined with a new piece of more accurate data** (with reduced noise) than the individual inputs
  - **medium-level fusion** - attributes or features of an entity (e.g., shape, texture, position) are **fused to obtain a feature map** that may be used for other tasks (e.g., segmentation or detection of an object); this type of fusion is also known as feature/attribute level fusion
  - **high-level fusion** - also known as symbol or decision level fusion; **inputs are decisions or symbolic representations that are joined to produce a more confident and/or a global decision**
  - **multilevel fusion** - this fusion **involves data of different abstraction levels**, i.e., multilevel fusion occurs when both **input and output of fusion can be of any level** (e.g., a measurement is fused with a feature to provide a decision)

ML in Fog Computing

Computational Intelligence and Fog Computing

- To **minimize the communication latency and improve the computational performance of IoT applications**:
  - the **emerging paradigms of fog and edge computing** have been advocated as potential and promising solutions
  - the **compute nodes of the edge or fog tier** will perform all or part of the various data stream processing steps
- To **tackle the issues related to performance, scalability, and real-time responsiveness**, and considering the **massive amount of data produced by large IoT systems**:
  - researchers have investigated the **deployment of ML techniques in the fog computing paradigm**

ML is the creation and use of models learned from data

Machine Learning (1/4)

- **ML is a research field of CI**:
  - it has been **widely implemented in a number of domains** that depend on complex and massive data processing,
  - e.g., **medicine, biology, and engineering**,
  - providing **solutions to gather the information hidden in the data**
- The **central goal of CI** is to:
  - **provide a set of algorithms and techniques that can be used to solve problems that humans perform intuitively and near automatically but are otherwise very challenging for computers**
- ML is the **creation and use of models** that are **learned from data**:
  - **a model is a specification of a mathematical (or probabilistic) relationship that exists between variables**
- Typically, in **IoT scenarios the aim of adopting ML techniques is using historical and stream data to develop models** able to predict various outcomes for new data or to provide insights

ML examples and subdomains

Machine Learning (2/4)

- **Examples**:
  - **predicting the increase rate of energy consumption** based on the current temperature and power status of an IoT system
  - **predicting** whether a set of events represents a dangerous situation in some environment (for instance a smart building)
  - **predicting** an intersection congestion when the number of vehicles increases in a smart road application
- ML techniques can be divided into **three subdomains**:
  - **supervised**,
  - **unsupervised learning**, and
  - **reinforcement learning**

ML subdomains

Machine Learning (3/4)

- **Supervised** learning:
  - the **input data can be labeled** with the desired outputs of the algorithm
- **Unsupervised** learning:
  - you can interpret data based only on **input data** trying to find some hidden pattern or intrinsic structure
- **Reinforcement** learning:
  - differs from normal ML because it does not use training data set
  - it uses interactions with the **external environment** to constantly adapt and learn on given points as a kind of feedback

Learning types	Data processing tasks	Methodology	Learning algorithms	Example of IoT applications
Supervised learning	Classification/ Regression/ Estimation	Statistical classifiers	K-nearest neighbors Naïve Bayes Hidden Markov model Bayesian networks	Fruit identification through classification of data patterns
		Computational classifiers	Decision trees Support vector machine	
		Connectionist classifiers	Neural networks	
Unsupervised learning	Clustering/ Prediction	Parametric	K-means Gaussian mixture model	Anomaly detection in IoT healthcare system
		Nonparametric	Dirichlet pr. mix model X-means	
Reinforcement learning	Decision-making	Model-free	Q-learning R-learning	Urban traffic prediction
		Model-based	TD learning Sarsa learning	

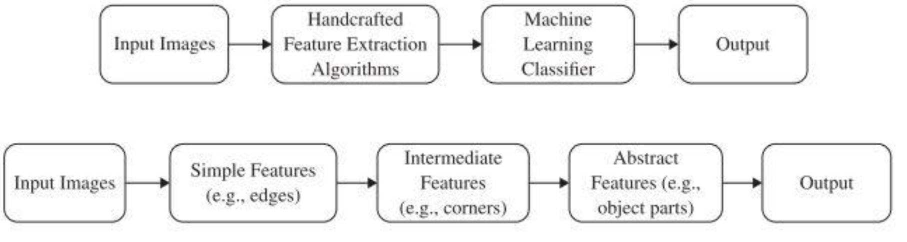
ML supervised, unsupervised, and reinforcement learning

Machine Learning (4/4)

- **Supervised and unsupervised learning** is suitable for **data analysis** while **reinforcement learning** is adequate for **decision-making** problems
- **For example**:
  - to **find unusual data points and anomalies in smart data**, the support vector machine (SVM) algorithm is suggested
  - to **predict the categories (labels) of data**, neural networks use approximation functions to obtain the desired output
  - to **find hidden patterns or intrinsic structure in data**, unsupervised learning applies the clustering technique
- **K-means is the most widely used clustering algorithm for exploratory data analysis**
- **K-means is one of the most naive ML algorithms being used for classification and regression problems**
- **Decision tree algorithms are used to develop classification models in the form of a tree structure**

Deep Learning (1/5)

- In **recent years, with the increasing computational capacities**:
  - **new generation IoT devices have been able to apply more advanced neural networks** to capture and understand their environments
  - e.g., in recent years, with the increasing computational capacities, new generation IoT devices have been able to apply more advanced **neural networks** to capture and understand their environments
- **Using a cascade of nonlinear processing units (layers) for feature extraction and transformation**:
  - a **deep learning (DL)** method can extract from an image a hierarchy of concepts that correspond to different levels of abstraction

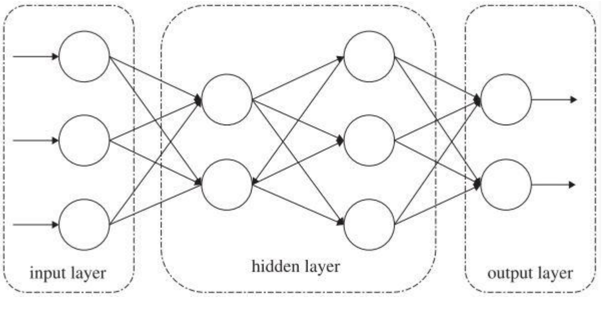


Deep Learning (2/5)

- DL architectures such as **Deep Neural Network (DNN)** and **Convolutional Neural Network (CNN)** are classes of **ANN (artificial neural network):**
  - have been **applied to many fields including computer vision, audio recognition, speech transcription, and natural language processing**, where they have produced results comparable to human experts
- A **DNN is basically a feed-forward neural network with several hidden layers of interconnected neurons** (or processing units):
  - collectively, the **neurons of the hidden layers** are responsible for transforming the data of the input layer into the desired output data (inference classes) of the output layer
  - **each hidden layer neuron has a vector of weights** with a size equal to the number of inputs as well as a bias
  - **these weights are optimized** during the training process
  - the **hidden layer neuron** receives the inputs from the previous layer, calculates a weighted summation of these inputs, and calls an **activation function** to produce the output, passing the resulting sum as a parameter
  - **these weights are what make each neuron unique**
  - **they are fixed during the test of the network**, but during the training these weights are changed in order to teach the network how to produce the expected output for the respective data inputs

Deep Learning (3/5)

- An **artificial neural network (ANN), or neural network for short, is a technique inspired by the way the brain operates**:
  - a **collection of neurons** wired together
  - the **neurons** that represent processing units are **organized in layers** – input layer, one or more hidden layers and output layer
  - **each neuron in a layer** receives the outputs of the previous layer neurons, does a calculation with an activation function, and then forwards the results to the next layer neurons
- The **algorithms used in neural networks form a framework for many ML techniques capable of dealing with complex data inputs**

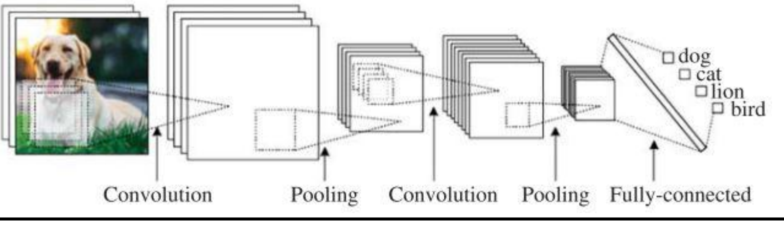


Deep Learning (4/5)

- A **CNN is a class of ANN and an alternative to DNN** when applied to:
  - **analyzing visual imagery due to scalability issues** (e.g., high resolution images imply thousands of neurons), and
- **DNN might not learn the features that undergo minor changes in some images**, like a rotation of the face:
  - **CNN has solved these problems**
- **As any ANN, a CNN also consists of**:
  - **input layer, multiple hidden layers, and an output layer**
  - **the hidden layers of a CNN typically consist of one or more convolutional and pooling layers**, and **finally a fully connected layer equivalent to those used in DNNs**
- **Convolutional layers** apply convolution operations to the input, **called filters**, extracting the features from data, reducing the number of parameters and passing the result to the next layer

Deep Learning (5/5)

- In the training process, **a CNN automatically learns the values for these filters**, which represent hierarchical feature maps (e.g., edges, corners and object parts) of the input images
- The **pooling layers** operate on the feature maps produced by the filters combining the outputs of neuron clusters at one layer into a single neuron in the next layer
- **Max and average pooling** uses the maximum and average value, respectively, from output of a cluster of neurons at the prior layer causing dimensionality reduction and the representations to be invariant to translations
- The **last layers (i.e., a DNN fully connected) are applied to complete classification**



Challenges for Running Machine Learning on Fog Devices (1/7)

- **Traditionally, ML algorithms run on powerful computers** hosted in cloud data centers to accommodate their high demand for computation, memory, and power resources
- **Data streams are generated (and eventually processed)** by devices located at the edge of the network that have limited power and hardware resources
- **Smart applications that use large sets of sensing data**, such as audio and images, aggravate the memory demand and bandwidth consumption
- Therefore:
  - **applying modern data analytics processing using ML techniques without imposing communication overheads is a challenge**, and
  - **researchers have recently investigated how to deploy CI on fog devices in an efficient way**
- **Challenges for running any resource demanding program on fog devices are related to**:
  - **processing time**,
  - **memory demand**,
  - **bandwidth**, and
  - **power consumption**

 32

32

Challenges for Running Machine Learning on Fog Devices (2/7)

- **Processing time and bandwidth consumption** are directly related to the performance and the energy required to run the application:
  - so, **any reduction in these two factors** is important to save energy and increase the overall efficiency of the system
- Studies have shown that:
  - **running on a low-capacity computer is still possible**, but
  - the **response time and energy consumption** will probably make the solution infeasible for most Smart IoT applications
- **One way to overcome this problem is to use GPUs**:
  - **CNN algorithms require a huge number of mathematical operations** using **large matrices of floating-point numbers** that are intensive time-consuming if performed in pipeline mode by the CPU
  - however, **the hundreds or thousands of processing cores of GPUs** can easily speed up the execution of a huge set of mathematical operations such as the ones required by CNN algorithms

 33

33

Challenges for Running Machine Learning on Fog Devices (3/7)

- **Hardware** resources found and analyzing capacities for running ML algorithms
- Proposed **classes of devices** with the respective suitable algorithms
- Devices class 1 are the **smallest and least computational powerful** of them, and its role is only to produce data:
  - **class 1 devices are known as the IoT sensors** by the research community, with some basic computational resources

Class	Hardware capacity	Power consumption	Suitable algorithms	Main application
1	No storage Low CPU and memory	≤1 W	Basic computation	Data generation
2	Storage ≤4 GB Memory ≤512 MB CPU single-core	≤2 W	Basic statistic	Measurement level fusion
3	Storage ≤8 GB Memory ≤2 GB CPU quad-core	≤4 W	Classification/ Regression / Estimation	Feature level fusion
4	Storage ≥16 GB Memory ≥4 GB CPU and GPU	≤8 W	Prediction/ Decision-making	Decision level fusion
5	Very large	Very high	Any	Autonomous system

Challenges for Running Machine Learning on Fog Devices (4/7)

- **At the other end**:
  - **class 5 devices are the highest powerful computational equipment** and can be considered as the unlimited resources available in cloud data centers
- **Devices of class 2, 3, and 4** showed that they have capacity to perform the three levels of information fusion:
  - **measurement**,
  - **feature**, and
  - **decision**

Class	Hardware capacity	Power consumption	Suitable algorithms	Main application
1	No storage Low CPU and memory	≤1 W	Basic computation	Data generation
2	Storage ≤4 GB Memory ≤512 MB CPU single-core	≤2 W	Basic statistic	Measurement level fusion
3	Storage ≤8 GB Memory ≤2 GB CPU quad-core	≤4 W	Classification/ Regression / Estimation	Feature level fusion
4	Storage ≥16 GB Memory ≥4 GB CPU and GPU	≤8 W	Prediction/ Decision-making	Decision level fusion
5	Very large	Very high	Any	Autonomous system

Challenges for Running Machine Learning on Fog Devices (5/7)

- **Unlike class 1 devices that have microcontrollers**:
  - **devices of class 2, 3 and 4** have true CPU, RAM memory and a multitask operating system
- **Devices class 3 and 4 are the ones really used for running ML algorithms on the fog**:
  - the **main difference between them** is that class 4 devices have a **powerful GPU** for parallel processing,
  - which is **very important for the execution of algorithms for training DNNs**, for instance

Class	Hardware capacity	Power consumption	Suitable algorithms	Main application
1	No storage Low CPU and memory	≤1 W	Basic computation	Data generation
2	Storage ≤4 GB Memory ≤512 MB CPU single-core	≤2 W	Basic statistic	Measurement level fusion
3	Storage ≤8 GB Memory ≤2 GB CPU quad-core	≤4 W	Classification/ Regression / Estimation	Feature level fusion
4	Storage ≥16 GB Memory ≥4 GB CPU and GPU	≤8 W	Prediction/ Decision-making	Decision level fusion
5	Very large	Very high	Any	Autonomous system

Challenges for Running Machine Learning on Fog Devices (6/7)

- **Class 3 device** can use a trained neural network to make the mid-level data fusion and extract the relevant features of the data, like a detection of an object
- **Class 4** device can go further and taking decisions using a set of higher-level data fusion

Class	Hardware capacity	Power consumption	Suitable algorithms	Main application
1	No storage Low CPU and memory	≤1 W	Basic computation	Data generation
2	Storage ≤4 GB Memory ≤512 MB CPU single-core	≤2 W	Basic statistic	Measurement level fusion
3	Storage ≤8 GB Memory ≤2 GB CPU quad-core	≤4 W	Classification/ Regression / Estimation	Feature level fusion
4	Storage ≥16 GB Memory ≥4 GB CPU and GPU	≤8 W	Prediction/ Decision-making	Decision level fusion
5	Very large	Very high	Any	Autonomous system

Challenges for Running Machine Learning on Fog Devices (7/7)

- On the one hand:
  - **IoT devices are to be deployed in non-refrigerated environments and with limited power sources**, less computation requires less power and yields less heat
- On the other hand:
  - **devices with high computational power are more expensive**, require a lot of energy to work and yield more heat, so that they should be used with care in IoT projects

Class	Hardware capacity	Power consumption	Suitable algorithms	Main application
1	No storage Low CPU and memory	≤1 W	Basic computation	Data generation
2	Storage ≤4 GB Memory ≤512 MB CPU single-core	≤2 W	Basic statistic	Measurement level fusion
3	Storage ≤8 GB Memory ≤2 GB CPU quad-core	≤4 W	Classification/ Regression / Estimation	Feature level fusion
4	Storage ≥16 GB Memory ≥4 GB CPU and GPU	≤8 W	Prediction/ Decision-making	Decision level fusion
5	Very large	Very high	Any	Autonomous system

Conclusion (1/2)

- There is a **consensus that IoT is the leading technology to enable advanced, ubiquitous and intelligent systems**
- Using all kinds of data, provided by a **growing number of heterogeneous devices** (from wearables to cameras), a **huge diversity of information** can be extracted and interpreted to enhance the user experience
- The **gathering and analysis of this Big Data** is one of the main challenges faced by the community nowadays
- As we have seen, **ML techniques have been applied** to perform advanced data analysis in a very efficient way in order to allow for valuable inferences even in resource-constrained devices
- Looking at the opportunities of a new market, the **industry has developed new hardware** to overcome the limitations of running heavy ML algorithms on edge devices

Conclusion (2/2)

- The **limited hardware of edge/fog device** is only part of the problem, and there are further issues that have been addressed by researchers developing IoT intelligent systems:
  - **accuracy vs. energy consumption**
  - **lack of complete training data set for each domain**
  - **security and privacy**
- We can conclude with a final thought that the **ML algorithms will evolve along with IoT devices**, pushing forward the development of new hardware, software, and network protocols to address all the challenges presented