

Article

Predicting LoRaWAN Behavior: How Machine Learning Can Help

Francesca Cuomo ^{1,2}, Domenico Garlisi ^{2,3,*}, Alessio Martino ¹ and Antonio Martino ¹

¹ Department of Information Engineering, Electronics and Telecommunications, University of Rome “La Sapienza”, Via Eudossiana 18, 00184 Rome, Italy; francesca.cuomo@uniroma1.it (F.C.); alessio.martino@uniroma1.it (A.M.); martino.1611005@studenti.uniroma1.it (A.M.)

² Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), 43124 Parma, Italy

³ Department of Engineering, University of Palermo, Viale delle Scienze Ed. 9, 90128 Palermo, Italy

* Correspondence: domenico.garlisi@unipa.it

Received: 14 July 2020; Accepted: 30 July 2020; Published: 31 July 2020



Abstract: Large scale deployments of Internet of Things (IoT) networks are becoming reality. From a technology perspective, a lot of information related to device parameters, channel states, network and application data are stored in databases and can be used for an extensive analysis to improve the functionality of IoT systems in terms of network performance and user services. LoRaWAN (Long Range Wide Area Network) is one of the emerging IoT technologies, with a simple protocol based on LoRa modulation. In this work, we discuss how machine learning approaches can be used to improve network performance (and if and how they can help). To this aim, we describe a methodology to process LoRaWAN packets and apply a machine learning pipeline to: (i) perform device profiling, and (ii) predict the inter-arrival of IoT packets. This latter analysis is very related to the channel and network usage and can be leveraged in the future for system performance enhancements. Our analysis mainly focuses on the use of k -means, Long Short-Term Memory Neural Networks and Decision Trees. We test these approaches on a real large-scale LoRaWAN network where the overall captured traffic is stored in a proprietary database. Our study shows how profiling techniques enable a machine learning prediction algorithm even when training is not possible because of high error rates perceived by some devices. In this challenging case, the prediction of the inter-arrival time of packets has an error of about 3.5% for 77% of real sequence cases.

Keywords: IoT; LoRa; LoRaWAN; machine learning; cluster analysis; prediction analysis; channel occupancy; network optimization

1. Introduction

Internet of Things (IoT) networks are becoming very attractive worldwide due to their simple deployment, low cost and wide range of applications that will be supported: smart homes, smart cities, utilities metering, agriculture and waste management, just to name a few. Low Power Wide Area Networks is a field in which communication technologies can make IoT a reality. This kind of network is designed to meet IoT requirements and to support large scale deployment. A promising Low Power Wide Area Network technology is LoRaWAN [1], which, by providing a robust modulation scheme and a simple network architecture, is able to connect thousands of End Devices (EDs) to gateways in an area of several kilometers.

Figure 1 sketches the LoRaWAN network architecture. The transmission by an ED is made very simple, with an adaptive tuning of the modulation parameters and without an explicit association to a specific gateway. A gateway can receive packets from different devices and the same device can

transmit to different gateways. Gateways forward packets to the Network Server, which is responsible for processing, removing duplicate packets, and forwarding their payload to IoT application servers.

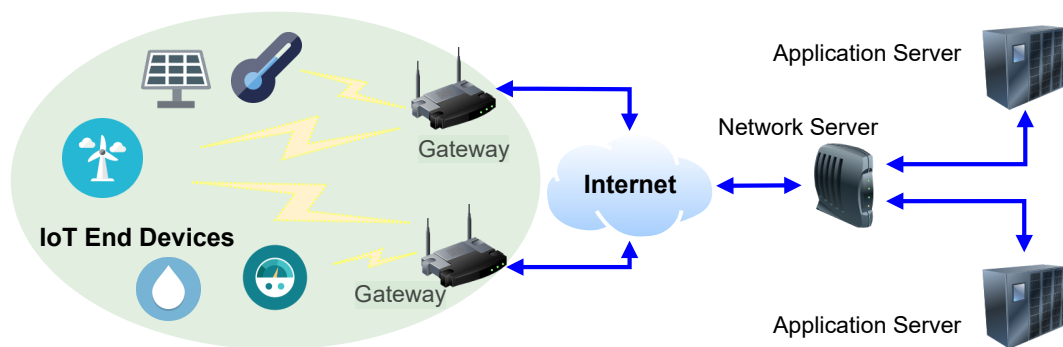


Figure 1. Long Range Wide Area Network (LoRaWAN) network architecture.

A critical aspect in LoRaWAN networks is represented by the configuration of the channels and by the allocation of data rate to different EDs. In this context, network optimization becomes a central part of the management system. In addition, context-awareness of the environment where the EDs operate, based on ED profiling and prediction schemes, is the needful tool for a suitable Spreading Factor (SF) allocation like the one proposed in [2,3].

According to the IoT Analytics forecast of 2018 [4], in the near future, the number of interacting wireless devices can be in the order of thousands or more, toward a plethora of service applications. Several different IoT scenarios can be generated, each with different capabilities and optimization constrains. For example, energy metering has challenging radio deployment with underground-indoor (e.g., basements) and indoor devices installation. For this type of service, we need to optimize the radio characteristics. Different from the energy metering, the smart street light service regards only outdoor devices, but the activity of these devices is during the moonlight. For this type of service, we need to optimize the channel access to avoid channel access collision. In this condition, is not possible to individually optimize each scenario, hence we need to find a solution that automatically selects the best network parameters in a scenario-aware fashion. Machine learning approaches allow the possibility to use the stored information to automatically learn the scenario and apply a suitable optimization strategy in a data-driven way.

This work presents an approach to use machine learning in order to create a context-aware scenario. We apply our proposed scheme to a real large-scale LoRaWAN deployment and we study the behaviour of this network through the application of two different machine learning techniques in order to perform network EDs profiling and to predict future wireless channel occupancy, respectively. The ability to predict channel occupancy in LoRaWAN is fundamental especially by noticing that a key modulation parameter on which the system can operate is the SF that is related to the Time-on-Air of transmitted packets. If thanks to these techniques we are able to find underutilized and overloaded time periods, methods like the one proposed in [3] can be used to estimate the channel occupancy and optimize the network performance, therefore tuning the number of devices that use a specific channel or SF, including also the suitable time periods. The defined algorithms can be applied in any IoT scenario with minimum human or network operator interaction.

Channel occupancy prediction is performed by prediction of Inter-arrival Time (IT) of packets, namely the time between two consecutive transmissions of the same ED. For that purpose, we combine profiling and prediction algorithms to deal with EDs that do not permit a smooth machine learning training phase. In this work, ED profiling is addressed by the k -means algorithm [5], while the prediction is performed at ED level: the network server predicts the ITs of LoRaWAN packets coming from the same ED. Specifically, our problem has been modelled as a time series prediction in which each time interval is the distance between two consecutive transmissions. To this end, our choice fell on two popular models for solving function approximation problems: Decision Tree (DT) adopting the

CART algorithm [6] and Long Short Term Memory (LSTM) network [7,8]. The choice behind these three algorithms stems from them being overall acclaimed in the machine learning community for their efficiency, effectiveness, hence widespread use for a plethora of different applications.

A prediction module needs a training phase based on ED IT time series. In LoRaWAN, in some cases, this is not possible since EDs may present high error rates, i.e., many packets may not be received due to low Received Signal Strength Indicator (RSSI) and low Signal to Noise Ratio (SNR) perceived by the gateway. In our approach, we use profiling to solve this problem. By using a subgroup of EDs, we create models that are valid for all EDs that have the same operating profile, included EDs that do not allow the training of the machine learning model due to low performance. To the best of our knowledge, none of the related works in the current literature have conducted a similar analysis as for LoRaWAN in a real commercial system. Hence, our contributions can be summarized as follows:

- propose an approach to profile LoRaWAN EDs;
- use the same machine learning model to predict behavior of EDs with different transmission patterns including also EDs that can not be used for the training model;
- we assess our approach in a real LoRaWAN scenario of a big operator.

Specifically, our main contribution is focused on the optimization of a real large scale LoRaWAN network through machine learning techniques. We combine profiling and prediction algorithms to address the follows two main problems:

1. learning process for EDs with low error rate;
2. prediction of channel access for EDs that present some anomaly behaviour.

Definitely, the optimization strategy can be applied in any LoRaWAN network scenario without operator supervision.

The rest of the paper is organized as follows. In Section 2 we present the related work while Section 3 describes the LoRaWAN large scale scenario used in our study. Section 4 presents our approach to address the different phases of clustering and prediction processes, and Section 5 presents the relevant numerical results. Finally, Section 6 concludes the paper.

2. Related Works

The interest in applying machine learning approaches to IoT systems is growing fast due to the ability of machine learning algorithms of being able to model an underlying process, usually unknown in closed-form, thanks to a finite set of data [9,10]. The data-driven peculiarity of machine learning models allows them to learn without any particular constraints and/or requirements about the application scenarios: hence, machine learning can be considered as a suitable toolbox of techniques for modelling IoT systems at different scales, provided that the data to be fed to the learning system reflects as much as possible the scenario under analysis. A detailed taxonomy of machine learning approaches for IoT is provided in [11] where a description on how machine learning algorithms can be applied to IoT smart data and what are the characteristics of IoT data in the real world are discussed. Several papers propose to use machine learning for anomaly detection or security issues (e.g., [12,13]) while others use machine learning for specific application purposes such as healthcare [14,15], traffic analysis [16,17], resource management [18] and modelling routing strategies in opportunistic networks [19]. The different application contexts and the potentialities of the use of machine learning are described in [14] where a review on the use of these techniques in different IoT application domains for both data processing and management tasks is provided.

Clustering and profiling issues are very interesting in the IoT field since they pose new challenges and pave the way to new applications. From network data it is possible to extract users' behaviours as shown in the paper by Tao et al. [16] in which users' activity features are extracted with regard to inter-arrival times and packets' traffic size and their relationships are investigated. The use of classification methods has been suggested in [17] to check thoroughly the quality of data collected by

IoT or wireless sensor networks. The authors show that classification is the most extensively adopted learning method for detecting outliers in IoT and wireless sensor networks. Clustering algorithms have been widely used to preserve security in IoT EDs, to classify EDs, to adapt the dynamic change of network topology or, in general, to analyze demands on the traffic characteristics, and a preliminary study on profiling IoT LoRaWAN EDs according to their radio behaviour can be found in [20], yet the exploitation of these profiles to improve the network performance is still unexplored. The novelty of our paper regards the concatenation of two machine learning stages to have a powerful tool for LoRaWAN operators. We adopt clustering to obtain homogeneous classes so that the traffic prediction can thereafter be applied with the purpose to plan suitable resource management mechanisms. The adoption of machine learning solutions for traffic analysis in IoT is provided in [21] where multiclass IoT flow classification task with lightweight machine learning models is applied. In this case a DT has been used, showing its potentialities in improving the performance.

As stated before, our second step is to predict channel occupancy. Channel sensing and prediction are heavily used in cognitive radio networks. In cognitive radio networks, the network needs to sense the spectrum for idle channels and establish communication on these sensed idle channels. For example, machine learning techniques can be exploited for inferring the availability of spectrum holes through channel state prediction. Cognitive radio networks can be considered as key IoT enabling technologies and their integration with future IoT architectures and services is expected to empower the IoT paradigm [22]. In IoT, cognitive radio networks cycle can automatically detect and learn from radio environment, adapt the transmission parameters in real-time, accordingly improving the spectrum efficiency. For example, in [23], channel sensing enables the cognitive radio networks cycle to coordinate coexistence between IoT and non-IoT technologies. In [24], the authors use machine learning to predict some traffic attributes. As for the use of machine learning to perform optimization tasks, the authors in [18] provide a reinforcement learning approach to derive optimal disseminating policies aiming at maximizing the accumulated average per-node throughput in a LoRaWAN system. The idea is to allow LoRaWAN gateways to optimize the transmission parameters of the EDs by deciding actions to be taken (update or not the configuration of an ED) on the basis on the policy learned from the reinforcement learning agent. This is implemented by keeping under control the transmission duty cycle, which is the percentage of time the nodes can access the shared wireless channels, often set to 1% of the time. While [18] was tested in randomly generated IoT networks, our approach is tested on a real LoRaWAN dataset. All the cited works based the prediction on the channel energy sensing. They do not consider the inter-arrival time prediction, like our proposal, which is critical concerning IoT applications to estimate the requested capacity and the free channel portions, mostly for scalability purposes. Specifically, our problem has been modelled as a time series problem in which the channel occupancy is treated as a sequence of occupied and free time periods.

Other than the two main machine learning-based tasks in this paper (profiling and prediction), machine learning has been applied in LoRaWAN networks for several applications. For example, in [25], the authors used Q-Learning for resource allocation; in [26] a deep learning-based approach has been used to exploit time difference of arrival for correct mobile station positioning; in [27] an ARIMA model has been used to predict future trends starting from air pollution sensors; in [28], the authors exploit LoRaWAN sensors and supervised learning for activity recognition; in [29,30], machine learning has been used for localization purposes and, finally, in [31] a mixture of unsupervised, supervised and decision model techniques has been used for load balancing in LoRaWAN networks.

3. LoRaWAN in a Real Large Scale Scenario

LoRaWAN operates in the industrial scientific and medical radio bands on the basis of the frequency plan provided in [32]. For the European region, the applied frequency plan is EU863-870 that provides eight channels and seven data rates. LoRaWAN, featured by the LoRaWAN Alliance [1], encompasses the networking functions and the medium access control. Furthermore, LoRaWAN also deals with application servers, security, firmware update over-the-air and network optimization.

At the physical layer, LoRaWAN uses the LoRa modulation based on the Chirp Spread Spectrum patented by Semtech [33]. In LoRa, EDs support multi-rate by exploiting six different SFs $\in \{7, \dots, 12\}$ or data rate. A detailed analysis on the impact of different configuration choices for SFs used by the EDs is presented in [34].

We consider data from a LoRaWAN infrastructure in Italy, which were provided by UNIDATA S.p.A. The deployed LoRaWAN network covers wide geographic areas in Italy and collects a huge amount of IoT data. Specifically, this network currently involves 1862 EDs and 138 gateways. In 2019, the total amount of EDs whose transmissions were received by the UNIDATA gateways were 89,528 (much higher than those officially registered with UNIDATA). Additionally this network has collected a total of 372,119,877 packets (2.25% are generated by EDs registered with UNIDATA).

The gateways are connected to the network server of the operator, where also the database is deployed. The database contains several records, each one representing a particular flow of information gathered from the network. For instance, there are data flows representing uplink and downlink packets, information exchanged between the gateway and the network server, or between EDs and the network server, or packets which have been de-duplicated because they have been received several times. The latter case happens when different gateways cover the same geographic area and the same packet reaches the network server from different gateways. Each packet represents a record index storing several fields.

To build a suitable dataset for our analysis we referred to 12 months of activity in the period ranging from January to December 2019. Furthermore, we consider a specific application, namely water metering service, in which 290 water meters are equipped with a LoRaWAN module and they transmit their measurements for about 18 times per week (the total number of packets is 115,000). The transmission pattern is unknown for these EDs, and, as shown in Section 4, it is different for each EDs. Two gateways cover the whole area where the water meters are located. Packets sent by the water meters are forwarded by the gateways to the network server.

Figure 2 plots the number of received packets, week by week, in the considered time period.

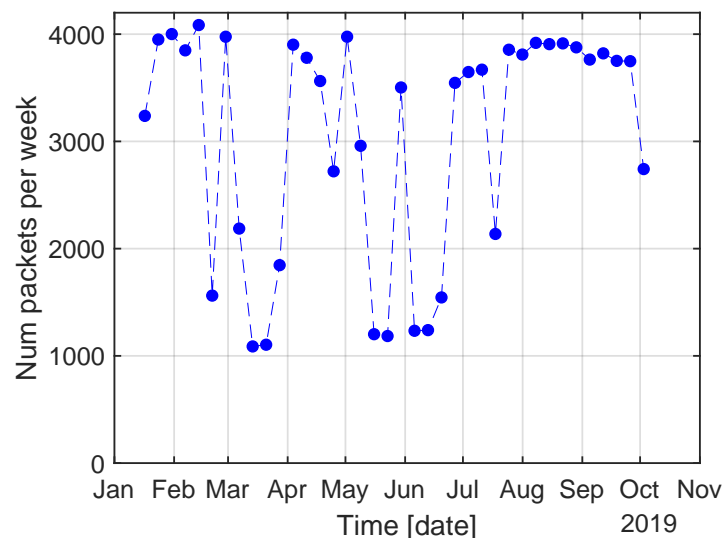


Figure 2. Number of received packets per week in a period of one year for the water meter dataset.

In this work we apply our algorithmic pipeline to both medium access control and physical layers. LoRaWAN physical information are gathered in a database from the gateway and they are related to the LoRa modulation.

Starting from the reference dataset, the following pre-processing steps have been performed to feed the machine learning algorithms:

1. packets have been grouped by transmitting ED (DevEUI field of Table 1);

- each ED has been mapped into a 6-length real-valued vector containing the following values: The SF, the average values of SNR and RSSI; the Packet Size (PktS); the Error Rate (ER) and the IT. These fields well represent the radio behavior of each ED.

Table 1 reports the adopted packet fields.

Table 1. Packet fields and parameters used in the analysis.

Field	Description
DevEUI	Device Identification Number, Unique identifier of the ED in the LoRaWAN global ecosystem
FCnt	Frame counter, Integer value used to enumerate all packets of a given ED; the pair (DevEUI, FCnt) is unique.
SF	Spreading Factor- Used to modulate the signal to transmit the packet
RSSI	Received Signal Strength Indicator (dBm) measured at the gateway
SNR	Signal to Noise Ratio (dB) measured at the gateway
PktS	Packet Size, size of the received packets
ER	Error Rate, Ratio between the number of received packets and transmitted ones (elaborated via the FCnt field)
IT	Inter Arrival Time, Time (expressed in minutes) elapsing between the received packet time instant and the time instant of its predecessor, both belonging to the same ED.

We notice that the latter two values have been computed and added in pre-processing phase.

4. Machine Learning Approach to Profile and Predict Network Activity

As introduced in Section 1, the knowledge of future ED transmission schemes could facilitate the evaluation of the optimal network operational parameters. For example, foreseeing saturation peaks in the radio channels can allow to modify in advance the adopted radio parameters, avoiding future inefficiencies and out-of-service events. On another hand, clustering algorithms can automatically group EDs according to the specific features. ED groups can be used to find EDs with same properties and force parameters settings. Nevertheless, when dealing with radio channels not all packets transmitted by EDs are received by a gateway, due to fading, shadowing and other impairments. Consequently, data-driven models can suffer due to lacks of data. In this work, we propose to use a subgroup of EDs to derive a model capable to provide forecasting for all EDs in the network. In other words, for EDs that present high error rate (corrupted IT series due to missing packets), it is not possible to train a model. We first apply a clustering phase to group EDs and afterwards we use the EDs within some clusters to train a model used to predict the EDs behaviour in different clusters. This kind of problem can be modelled as a time series one, where channel usage statistics can be converted into a time-based sequence.

Both clustering and time series prediction are well-known problems in the scientific community and multiple solutions have been provided to face both the former [5] and the latter [35]. The rationale behind choosing *k*-means for the clustering (profiling) stage not only stems from the fact that *k*-means is (possibly) the most famous clustering algorithm [36–38], but also from a previous work related to device profiling in LoRaWAN [20]. Similarly, the choice behind DT and LSTM stems from them being acclaimed techniques for time series predictions in a plethora of applications. Notable examples for DT include electricity consumption data [39], financial time series [40], medical data [41] and call transcription data [42]. LSTM networks are another effective solution for time series modelling in fields such as healthcare [43], air pollution [44] and financial time series as well [45]. Albeit these techniques have been used also in IoT contexts (see e.g., [46–48] for DT and [49,50] for LSTM), to the best of our knowledge, this is the first work in which they have been applied to a real world and large-scale LoRaWAN network.

4.1. EDs Profiling

In this subsection we describe how it is possible to apply ED profiling related to the utilization of wireless resources in a LoRaWAN network. We discuss a possible strategy that was preliminary proposed in [20] where the k -means algorithm was used to group EDs according to their radio and network behavior.

k -means, given a dataset $\mathcal{S} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ of n observations, aims at partitioning the data into k non-overlapping clusters, i.e., $\mathcal{S} = \{\mathcal{S}_1, \dots, \mathcal{S}_k\}$ such that $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ if $i \neq j$ and $\cup_{i=1}^k \mathcal{S}_i = \mathcal{S}$. k -means finds a (sub-)optimal partition of the data in such a way that the intra-cluster variance, or Within-Clusters Sum of Squares (WCSS), is minimized:

$$WCSS = \sum_{i=1}^k \sum_{\mathbf{x} \in \mathcal{S}_i} \|\mathbf{x} - \mathbf{c}^{(i)}\|^2 \quad (1)$$

where $\mathbf{c}^{(i)}$ is the centroid for cluster i , defined as the center of mass of the cluster itself. Despite its simplicity, the number of clusters k to be returned is a parameter that shall be tuned by the end-user: its value is data-dependent and is hardly known a-priori. As discussed in [20], in order to find the best k value for the data at hand, we considered several candidates $k = \{2, 3, \dots, 20\}$ and, by jointly considering the elbow plot [51], the Silhouette index [52] and then Davies-Bouldin index [53], a suitable value of $k^* = 7$ has been selected.

Figure 3 shows the seven resulting clusters for the considered dataset: the 3D plot is related to the ER, SF and SNR fields. Each cluster is represented by a different color (reported in the figure legend).

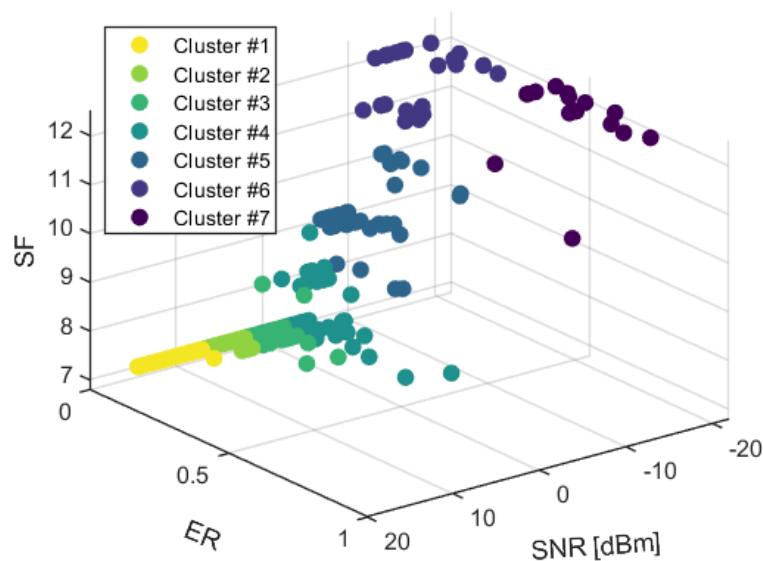


Figure 3. End Devices (EDs) representation in terms of SNR, Error Rate and SF after the k -means clustering. Different ED clusters are represented by different colours.

In a LoRaWAN network, the optimization strategy can be different for different subgroups of devices. In our approach, the goal of the profiling algorithm is indeed to automatically find the subgroups of devices that share the same behaviour, hence can rely on the same optimization strategy.

The cluster analysis was capable to capture critical behaviours inside the network. More generally, the approach can help to:

1. monitor the system behaviour and capture anomalies;
2. optimize the network planning;

- identify different spaces in which service provision can be enhanced (that is, new radio resources, more suitable parameter settings, different configurations of the IoT EDs and services).

This technique allows us to cluster EDs into “activity groups”, which consist of EDs that tend to have similar activity profiles in their use of radio resources. Further, the profiling analysis allows to identify the behaviour of EDs with low performance: if an ED is featured by low RSSI and low SNR, it is likely featured by a Higher Error Rate (we refer to those as ‘HER’ devices). When this happens, the sequence of the ITs can not be used to train a prediction model, because the received sequence is corrupted due to several missing packets. After the cluster analysis, of the 290 EDs present in the dataset, 20 have been included in HER group and they are represented by cluster #7 in Figure 3. The remaining 270 devices (represented by clusters 1–6 in Figure 3) are featured by a Low Error Rate (we refer to those as ‘LER’ devices).

4.2. Features Extraction: The Window Approach

The most interesting field for the prediction stage is indeed the IT. After a preliminary analysis of the transmission pattern of the network EDs, was deduced that the EDs send packets with a structure that is the combination of two different components: a regular part and an irregular part.

Figure 4 shows the ITs of roughly 60 packets of a single ED (the x -axis shows the FCnt of each packet). It is easy to recognize a pattern: many ITs lie at 720 min (which will be further called “part 1”), while some of them are less frequent and lie at 480 min, and others at 240 min (these latter two will be further called “part 2”). Different EDs show a similar behaviour: they share the same pattern structure and the same value for the first part, while the second part has other values that always sum up to 12 h. The combination of part 1 and part 2 represents the regular part of the transmission pattern. Figure 4 also shows a case where a packet has not been received by the network server. As a matter of fact we have in the figure a value at FCnt = 255 and another value at FCnt = 257 equal to 24 h, indicating that the packet at 12 h with FCnt = 256 has not been received.

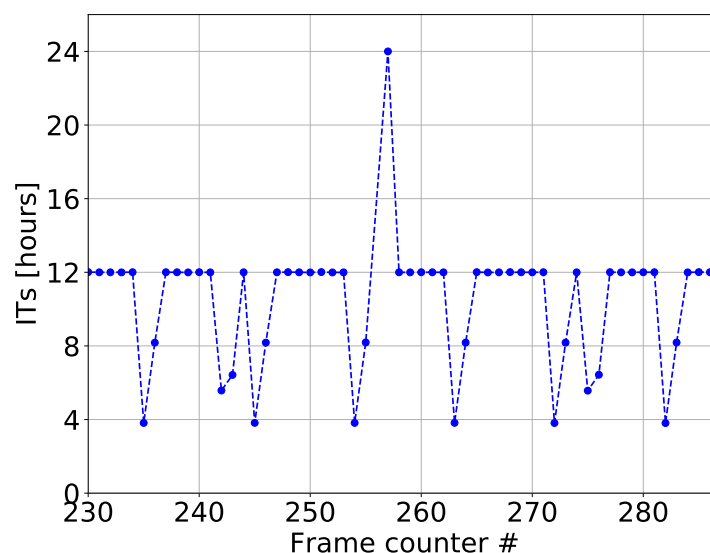


Figure 4. Pattern of the Inter-arrival Times (ITs) for the LoRaWAN water metering EDs.

The pattern also has an irregular part. The preliminary analysis allowed us to find the occasional existence of a pair of consecutive ITs that do not respect the pattern (which will be further called “part 3”). Without any specific reason, one packet of the regular part is replaced by two or more packets: their sum may differ from 12 h and it does not allow a precise forecast of the pattern. Figure 4 shows two of these situations (at (242,243) and (275,276) FCnt).

A frequency analysis of the ITs was made on the regular part only, to assure that the identified patterns comprehensively represent the behavior of the considered EDs. Figure 5 shows the frequencies

of the most frequent ITs for a subgroup of EDs (each ED is represented with a different bar). Each bar has three different colored components, whose height is equal to the IT value considered. From the Figure, we can observe that all EDs have a 12 h ITs (part 1), and two other values whose sum is 12 h (part 2), but they are different from ED to ED. For all EDs, the sum of the occurrences of these 3 ITs is the regular part and represents more than 92% of the entire set of samples (the exact value is shown above each bar). The remaining ITs, not included in the figure, represent the irregular part and they are distributed in a random position in the pattern. To evaluate how this irregular component impacts the performance of the prediction models, we used both the original dataset and a different version (named “relaxed”). In the relaxed version, all the elements of the irregular part are set to 360 min (6 h) to smooth the random components of the original transmission patterns. Every method described in the following will be tested on both these datasets.

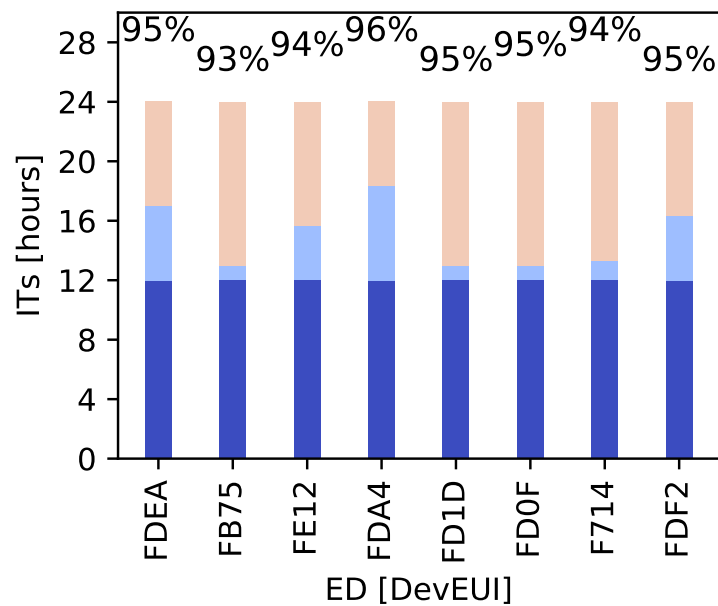


Figure 5. Frequency analysis for the ITs for a subset of water metering EDs.

Regarding the irregular part, device packets are received in unpredictable way. The focus of this paper is to predict when one of these event occurs.

The prediction problem falls under the umbrella of function approximation problems [10] where, given an input space $\mathcal{X} \subseteq \mathbb{R}^n$ and an output space $\mathcal{Y} \subseteq \mathbb{R}$ with an orientated (and usually unknown) process $f : \mathcal{X} \rightarrow \mathcal{Y}$, the goal of the learning system is to synthesize a model of f by means of a dataset, namely a finite collection of input-output pairs $D = \{\mathbf{X}, \mathbf{y}\}$ where $\mathbf{X} \in \mathbb{R}^{N \times n}$ is the instance matrix and $\mathbf{y} \in \mathbb{R}^N$ collects the output values, being n and N the number of features and patterns, respectively.

The success of any learning system relies on an accurate feature generation and a valid ground-truth output value mapping. In order to create a suitable set of n -dimensional feature vectors starting from the entire ED series, the sliding window mechanism is used [54]. In short, for the same ED, a set of n consecutive ITs is used to predict the $(n + 1)^{\text{th}}$. More in details, the goal of the learning system is to predict the IT of a subsequent packet p_{n+1} starting from the sequence of its n preceding packets p_0, \dots, p_n . Recalling that the forecasting model is based only on the IT component, the dataset is created using a sliding window with size n on the IT field. Hence, we extract only the two columns (IT, FCnt) from the dataset and for each sub-list of packets with consecutive FCnt whose size is greater than n , we take the first n values as input for the algorithm and the $(n + 1)^{\text{th}}$ value as the output. Obviously, the choice of n is critical in this phase and it will be addressed in Section 5.

An example of dataset building with the sliding window mechanism is given in Table 2, by considering:

- $n = 3$

- ITs = $[a, b, c, d, e, f, g]$
- DevEUI = #1

Table 2. Example of sliding window mechanism.

DevEUI	x_0	x_1	x_2	y
#1	a	b	c	d
#1	b	c	d	e
#1	c	d	e	f
#1	d	e	f	g

Each row reports the DevEUI of the observed ED (#1 in the example) along with the three selected features (x_0, x_1, x_2) for each of the $N = 4$ resulting patterns and the predicted value (output y), organized by rows.

4.3. Prediction Models

By using the aforementioned windowing approach, we have been able to build a feature matrix X and an output vector y , suitable for being used for training and testing machine learning algorithms. Given the nature of the data (see e.g., Figure 4), using linear models will unlikely lead to satisfactory results. To this end, our choice fell on two very popular models for solving function approximation problems: DT and LSTM.

The DT paradigm is used for both classification and regression. It incrementally builds learning models in the form of a tree, splitting up the dataset based on appropriate conditions on its features. The directed edges that connect two nodes contain values for a given source node, which is the topmost one. These trees have two types of nodes:

- Decision node: it is an internal node with two branches and it represents some features, along with a decision value. Branches are created in order to minimize the error on child nodes.
- Leaf node: It is a terminal node and it contains the target output.

In other words, the directed graph returned by the DT is created by means of rules applied on feature values in order to better differentiate the input patterns with respect to the output variable. Hence, the classification of a new pattern simply consists in traversing the graph starting from the root node and following the rules behind each branching until a leaf node is reached. Figure 6 shows an example of the DT representation resulting from our dataset: For the sake of representation, a windowing of $n = 2$ (x_0 and x_1) is considered and the tree has been trained up to a maximum depth of 3 splits (3 layers under the source node). The resulting tree starts from a single root node containing the entire dataset. Then, the second feature (x_1) is selected for splitting and two child nodes are created containing, respectively, 25.9% ($x_1 \leq 712$ min) and 74.1% ($x_1 > 712$ min) of the patterns belonging to the root node. The procedure recursively iterates until a stopping criterion is met (i.e., maximum depth). An interesting aspect of DTs lies on their rule-based structure: this makes de-facto DTs as grey-box models. Indeed, as in Figure 6, trees can be visualized and the condition behind each decision can be easily explained with simple Boolean logic.

The other approach is based on LSTM networks [7]. These are Recurrent Neural Networks (RNNs) [55] with backward feed propagation, which permits information to persist over time. One of the main problems with RNNs is that there is no possibility to select which information has to be remembered, because all the information is recorded in the same way. Hence, some notable events do not have the impact in the learning process that they should. Differently, LSTM networks add multiple operations within the RNN cell. These new operations are called ‘gates’ in the LSTM architecture.

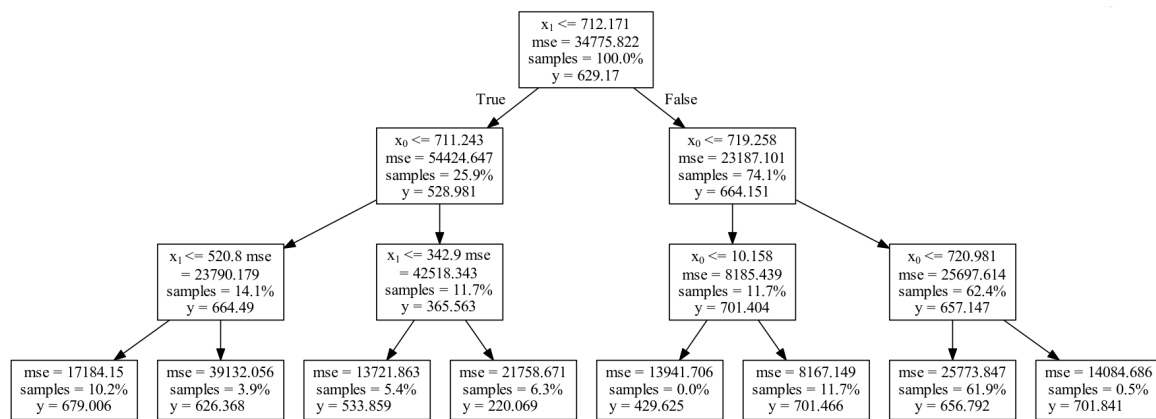


Figure 6. Example of the Decision Tree (DT) representation resulting from our dataset (for a window $n = 2$ and maximum depth of three splits).

The whole LSTM cell is shown in the Figure 7. It is important to notice that the LSTM cell has two inputs and two outputs. As input it takes both input features and the previous state, just like the feedback loop of a standard RNN. The first gate is commonly known as the ‘forget gate’, because its function is to select which of the data the network forgets at a specific time step. The second gate of the LSTM is known as the ‘input gate’ layer, which works exactly as a regular RNN. Finally, the last gate is called the ‘output gate’. This gate prepares a set of candidate values to be added to the final state of the cell. Figure 7 also shows the hyperbolic tangent and sigmoid activation functions used in the LSTM cell: the former helps in regulating the values within the network by ensuring a fixed $[-1, 1]$ range; the latter helps in preserving and forgetting data by squashing in range $[0, 1]$. Another fundamental characteristic of RNNs is that they possess the ability to act as Multiple Input Multiple Output systems. In our problem we use the “many to one” configuration, where the output is the last element returned by LSTM of the last column element in the sliding window matrix, the g value in Table 2. Differently from DTs, LSTM networks are complex models whose interpretability is still a challenge [56].

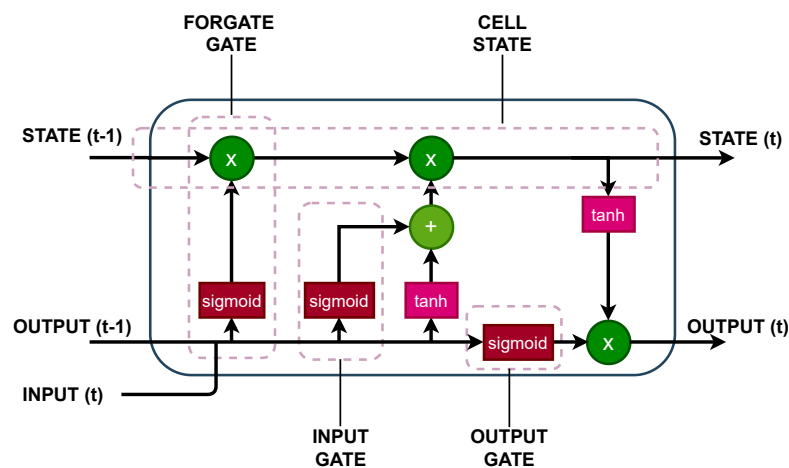


Figure 7. Long Short Term Memory (LSTM) cell architecture.

5. Computational Results

In this section we present the computational results derived by applying the proposed pipeline to our dataset. In order to quantify their performance, we used Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), both of which are negative-oriented (i.e., the lower the error, the better the model). Let m be the number of samples, y_j be the observed value for the j^{th} sample and \hat{y}_j the predicted one, then:

- MAE is defined as the arithmetic average of absolute errors:

$$MAE = \frac{1}{m} \sum_{j=1}^m |y_j - \hat{y}_j| \quad (2)$$

- RMSE is defined as the square root of the arithmetic average of the squares of the errors:

$$RMSE = \sqrt{\frac{1}{m} \sum_{j=1}^m |y_j - \hat{y}_j|^2} \quad (3)$$

The difference between these two metrics lies in the speed with which RMSE increases in presence of outliers (predicted values that are very far from the real ones.)

As discussed in Section 4.2, the choice of the window size n is critical for the learning phase. In order to find a suitable value, the dataset has been properly split in training set (80%) and test set (20%) and a 20-fold cross-validation has been applied on the training set to both predictions algorithms. Candidate values for n are $\{5, 10, 15, 20, 25, 30\}$ and we chose the one which minimizes the cross-validation MAE/RMSE.

Figure 8a shows the output of the cross-validation for DT, using $n \geq 10$, which proves to be the most promising range of values. Specifically, we chose $n = 10$ in such a way that the number of samples is maximized (63758). As for the model parameters, DT has been configured as follows: we considered the variance reduction criterion for splitting current nodes which consist in splitting the parent nodes in such a way that the ℓ_2 -norm is minimized on child nodes by considering their mean values. Furthermore, in order to construct the tree, we set the minimum number of samples required to split a node to two and the growing process continues if either a node contains more than the minimum number of samples or it is 'impure' (i.e., splitting cost greater than 10^{-7}). For LSTM, following the same method, we chose $n = 20$ and a single internal layer and 120 hidden nodes upfront candidate architectures consisting of one and two internal layers, and a number of hidden nodes in range $[50, 150]$. The network has been trained using the ADAM optimizer with learning rate of 0.0001, using MAE as loss function and for a maximum number of 150 epochs.

For a thorough investigation, we performed two different experiments:

1. in the first one, only LER EDs are considered; the training phase has been performed with 80% of EDs, the remaining EDs are used for the test part (EDs dedicated to training and test are randomly selected);
2. in the second one, training is performed on LER EDs and the resulting model is tested on HER EDs.

Figure 8b,c show the IT series of 5 sequentially EDs. We can observe that the 12 h (720 min) samples are common to all EDs, while the other two ITs are different for each ED. In the figures (as explained later) the isolated points (6 h) represent the irregular part of the pattern, which has been processed to produce the relaxed dataset. In detail, Figure 8b,c show a qualitative comparison of predicted (red star marker) and real (blue triangular marker) values for DT and LSTM, respectively. Clearly, lower error rates have been obtained with the former model.

Further, Figure 8d,e show the Cumulative Distribution Function (CDF) of the absolute relative errors in terms of minutes for the four different combinations of original and relaxed dataset, tested on 20% LER and on HER. For the DT, 90% of errors are lower than 100 min, whereas for LSTM this value is reduced to 85%. Figure 8d also shows that in case of relaxed dataset, for the DT, 90% of errors are lower than 10 min.

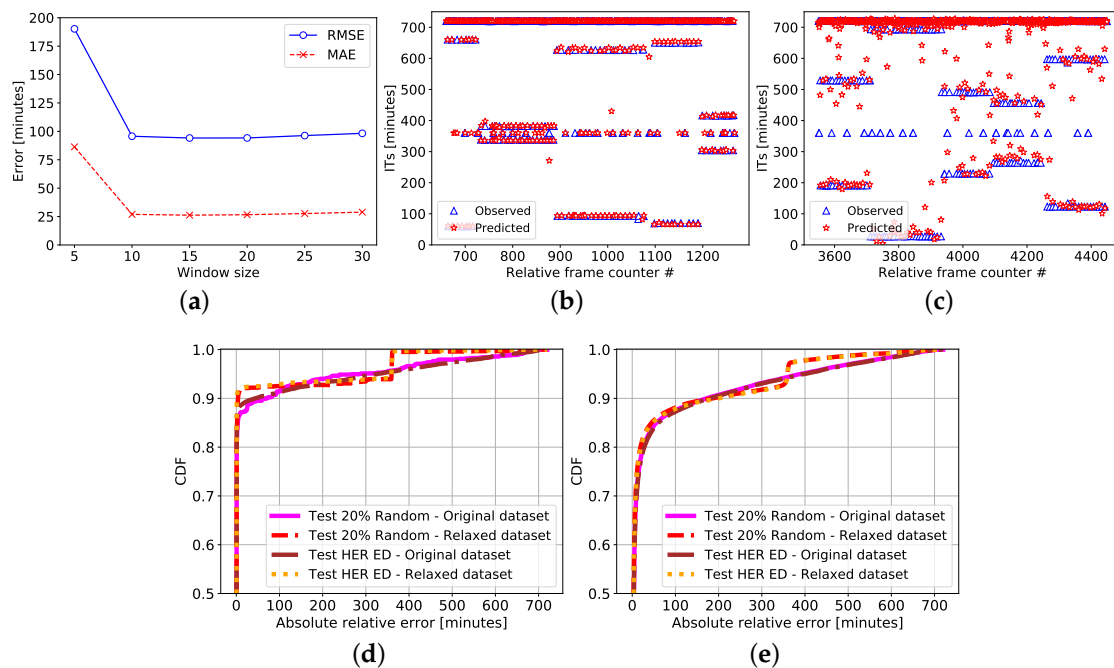


Figure 8. Twenty-fold cross validation on training set with different window sizes (a). Qualitative result of the DT prediction for the test dataset (b). Qualitative result of the LSTM prediction for the test dataset (c). CDF of the absolute prediction errors of the DT for the four datasets (d). CDF of the absolute prediction errors of the LSTM for the four datasets (e).

Finally, for both prediction algorithms, the figures confirmed our hypothesis: the shapes of the CDF resulting from the prediction on random 20% LER EDs and on 100% HER EDs are equivalent. We stress again that the training phase is performed only on LER EDs.

The MAE and RMSE values for all considered cases and predictions algorithms are reported in Table 3. We expected that the relaxation would have mainly led to score improvements for the RMSE, which is sensitive to outliers, and numerical results confirmed our idea: indeed, results in Table 3 show a significantly decreased RMSE, while the MAE is lowered by roughly 10 min.

Table 3. Performance comparison for both machine learning algorithms.

		DECISION TREE		LSTM	
Testing on ED		MAE	RMSE	MAE	RMSE
original	20% random	35.51	126.30	51.12	140.26
relaxed	20% random	24.07	91.16	46.06	125.52
original	HER	34.05	119.97	52.34	142.11
relaxed	HER	25.90	96.47	42.81	120.58

Table 3 confirms that DT exhibits better performance than LSTM and that, in the best case, the MAE is 24.07 min. If we consider that in the reference scenario the average IT per ED is almost 9 h, we are certain to predict the number of packets present in the channel hour by hour in the next 12 h.

Alongside the comparison in terms of performance, it is also worth to point out the computational burden required to train and predict the two models. The entire pipeline has been implemented in Python and executed on a 6-cores AMD Ryzen 1600X CPU@3.9 GHz and 8 GB of RAM, using the external libraries `sklearn` and `keras` for DT and LSTM, respectively. The time employed to build the windowed dataset is 12 s, with an additional 12 s to relax it. The DT fitting operation on the hardware/software available lasts less than a second (0.71 s). Conversely, the training phase for LSTM

employs 20 min. In conclusion, it is safe to say that our analyses let DT emerge as the most suitable model to predict IT for LoRaWAN network packet arrivals due to the following reasons:

- the lack of LSTM interpretability;
- the higher LSTM computational burden [57];
- the lower LSTM performances, which for certain time-window problems are worst with respect to ‘standard’ machine learning approaches: this is due to the fact that all relevant information about the next event is conveyed within the time-window itself making recurrent units useless [58].

6. Conclusions

In this paper we analyzed a LoRaWAN large-scale deployment adopting machine learning techniques. We performed two kinds of analysis. Firstly, we applied the *k*-means algorithm to profile network EDs. The clustering results are groups of EDs sharing similar behaviour. The profiling algorithm clusters EDs in Low Error Rate and High Error Rate. Secondly, we discussed a strategy based on DT and LSTM algorithms to predict EDs behaviour. Our problem has been modelled as a time series problem in which each interval of channel usage is treated as a sequence of occupied and free slots. In this way, we could apply DT and LSTM to these time series. All the presented solutions have been tested in a real LoRaWAN deployment, owned by the Italian operator UNIDATA, which in 2019 has collected 372,119,877 packets. In this challenging case, the prediction of the inter-arrival time of packets has an error of about 3.5% for 77% of real sequence cases.

Future works will be dedicated to employing the derived results for network optimization strategies to find the best parameter tuning. For instance, it is possible to use balancing algorithms that find the best balancing among the EDs on the available channels and SFs by starting from the extracted EDs information.

Author Contributions: Conceptualization, D.G. and A.M. (Antonio Martino); Methodology, D.G., A.M. (Alessio Martino) and A.M. (Antonio Martino); Software, D.G. and A.M. (Antonio Martino); Validation, D.G., A.M. (Alessio Martino) and A.M. (Antonio Martino); Formal analysis, D.G., A.M. (Alessio Martino) and A.M. (Antonio Martino); Investigation, D.G. and A.M. (Antonio Martino); Resources, D.G.; Data curation, D.G.; Writing—original draft preparation, D.G., A.M. (Alessio Martino), A.M. (Antonio Martino) and F.C.; Writing—review and editing, D.G., A.M. (Alessio Martino), A.M. (Antonio Martino) and F.C.; Visualization, D.G. and A.M. (Antonio Martino); Supervision, F.C.; Project administration, D.G. and F.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been partially funded by the Sapienza research project “IoT Networks for 5G Smart Health Applications”, 2019.

Acknowledgments: We thank UNIDATA S.p.A. who provided insight and expertise that assisted our research greatly. Part of the activities are related to the AMORE (Advanced Manufacturing e microelettronica per Ottimizzare le Reti wireless delle cose) project. KETs, POR FESR Lazio 2014–2020. We thank Marco Zanghieri for assisting us with the study of the prediction activities with LSTM algorithm.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CART	Classification And Regression Tree
CDF	Cumulative Distribution Function
ED	End Device
ER	Error Rate
DT	Decision Tree
FCnt	Frame Counter
HER	High Error Rate
IoT	Internet of Things
IT	Inter-arrival Time

LER	Low Error Rate
LoRaWAN	Long Range Wide Area Network
LSTM	Long Short Term Memory
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RSSI	Received Signal Strength Indicator
SF	Spreading Factor
SNR	Signal to Noise Ratio
WCSS	Within-Clusters Sum of Squares

References

- LoRa Alliance Technical Committee. LoRaWAN 1.1 Specification. 2017. Available online: <https://loro-alliance.org/resource-hub/lorawantm-specification-v11> (accessed on 31 July 2020).
- Cuomo, F.; Campo, M.; Caponi, A.; Bianchi, G.; Rossini, G.; Pisani, P. EXPLoRa: Extending the performance of LoRa by suitable spreading factor allocations. In Proceedings of the 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Rome, Italy, 9–11 October 2017; pp. 1–8.
- Bianchi, G.; Cuomo, F.; Garlisi, D.; Tinnirello, I. Capture Aware Sequential Waterfilling for LoraWAN Adaptive Data Rate. *arXiv* **2019**, arXiv:1907.12360v3.
- Lueth, K.L.; Scully, P.; Williams, Z.D.; Pasqua, E.; Romeo, S.; Artes, R.; Wopata, M. *State of the IoT & Short-Term Outlook*; IoT Analytics GmbH: Hamburg, Germany, 2018.
- Jain, A.K.; Murty, M.N.; Flynn, P.J. Data clustering: A review. *ACM Comput. Surv. (CSUR)* **1999**, *31*, 264–323. [[CrossRef](#)]
- Breiman, L.; Friedman, J.; Stone, C.; Olshen, R. *Classification and Regression Trees*; The Wadsworth and Brooks-Cole Statistics-Probability Series; Taylor & Francis: New York, NY, USA, 1984.
- Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. In Proceedings of the 1999 Ninth International Conference on Artificial Neural Networks ICANN 99 (Conf. Publ. No. 470), Edinburgh, UK, 7–10 September 1999; Volume 2, pp. 850–855.
- Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
- Qiu, J.; Wu, Q.; Ding, G.; Xu, Y.; Feng, S. A survey of machine learning for big data processing. *EURASIP J. Adv. Signal Process.* **2016**, *2016*, 67. [[CrossRef](#)]
- Martino, A.; Giuliani, A.; Rizzi, A. Granular Computing Techniques for Bioinformatics Pattern Recognition Problems in Non-metric Spaces. In *Computational Intelligence for Pattern Recognition*; Pedrycz, W., Chen, S.M., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 53–81.
- Mahdavinejad, M.S.; Rezvan, M.; Barekatin, M.; Adibi, P.; Barnaghi, P.; Sheth, A.P. Machine learning for internet of things data analysis: A survey. *Digit. Commun. Netw.* **2018**, *4*, 161–175. [[CrossRef](#)]
- Aceto, G.; Ciuonzo, D.; Montieri, A.; Pescapé, A. Mobile Encrypted Traffic Classification Using Deep Learning: Experimental Evaluation, Lessons Learned, and Challenges. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 445–458. [[CrossRef](#)]
- Verzegnassi, E.G.M.; Tountas, K.; Pados, D.A.; Cuomo, F. Data Conformity Evaluation: A Novel Approach for IoT Security. In Proceedings of the 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 15–18 April 2019; pp. 842–846.
- Samie, F.; Bauer, L.; Henkel, J. From Cloud Down to Things: An Overview of Machine Learning in Internet of Things. *IEEE Internet Things J.* **2019**, *6*, 4921–4934. [[CrossRef](#)]
- Ahamed, F.; Farid, F. Applying Internet of Things and Machine-Learning for Personalized Healthcare: Issues and Challenges. In Proceedings of the 2018 International Conference on Machine Learning and Data Engineering (iCMLDE), Sydney, Australia, 3–7 December 2018; pp. 19–21.

16. Tao, M.; Ming, Y.C.; Juan, C. Profiling and identifying users' activities with network traffic analysis. In Proceedings of the 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 23–25 September 2015; pp. 503–506.
17. Ghosh, N.; Maity, K.; Paul, R.; Maity, S. Outlier Detection in Sensor Data Using Machine Learning Techniques for IoT Framework and Wireless Sensor Networks: A Brief Study. In Proceedings of the 2019 International Conference on Applied Machine Learning (ICAML), Bhubaneswar, India, 25–26 May 2019; pp. 187–190.
18. Sandoval, R.M.; Garcia-Sanchez, A.; Garcia-Haro, J. Optimizing and Updating LoRa Communication Parameters: A Machine Learning Approach. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 884–895. [[CrossRef](#)]
19. Borrego, C.; Hernández-Orallo, E.; Magaia, N. General and mixed linear regressions to estimate inter-contact times and contact duration in opportunistic networks. *Ad Hoc Netw.* **2019**, *93*, 101927. [[CrossRef](#)]
20. Valtorta, J.M.; Martino, A.; Cuomo, F.; Garlisi, D. A Clustering Approach for Profiling LoRaWAN IoT Devices. In *Ambient Intelligence*; Chatzigiannakis, I., De Ruyter, B., Mavrommati, I., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 58–74.
21. Bismukhamedov, R.F.; Nadeev, A.F. Lightweight Machine Learning Classifiers of IoT Traffic Flows. In Proceedings of the 2019 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO), Yaroslavl, Russia, 1–3 July 2019; pp. 1–5.
22. Rawat, P.; Singh, K.D.; Bonnin, J.M. Cognitive radio for M2M and Internet of Things: A survey. *Comput. Commun.* **2016**, *94*, 1–29. [[CrossRef](#)]
23. Valck, P.; Moerman, I.; Croce, D.; Giuliano, F.; Tinnirello, I.; Garlisi, D.; Poorter, E.; Jooris, B. Exploiting programmable architectures for WiFi/ZigBee inter-technology cooperation. *EURASIP J. Wirel. Commun. Netw.* **2014**, *2014*, 212. [[CrossRef](#)]
24. Capela, N.; Sargento, S. Machine Learning for Resources Prediction in Multihoming Scenarios. In Proceedings of the 2015 IEEE Globecom Workshops (GC Wkshps), San Diego, CA, USA, 6–10 December 2015; pp. 1–7.
25. Aihara, N.; Adachi, K.; Takyu, O.; Ohta, M.; Fujii, T. Q-Learning Aided Resource Allocation and Environment Recognition in LoRaWAN With CSMA/CA. *IEEE Access* **2019**, *7*, 152126–152137. [[CrossRef](#)]
26. Cho, J.; Hwang, D.; Kim, K. Improving TDoA Based Positioning Accuracy Using Machine Learning in a LoRaWAN Environment. In Proceedings of the 2019 International Conference on Information Networking (ICOIN), Kuala Lumpur, Malaysia, 9–11 January 2019; pp. 469–472.
27. Thu, M.Y.; Htun, W.; Aung, Y.L.; Shwe, P.E.E.; Tun, N.M. Smart Air Quality Monitoring System with LoRaWAN. In Proceedings of the 2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS), Bali, Indonesia, 1–3 November 2018; pp. 10–15.
28. Hossain, T.; Ahad, M.A.R.; Tazin, T.; Inoue, S. Activity Recognition by Using LoRaWAN Sensor. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers; UbiComp '18*; Association for Computing Machinery: New York, NY, USA, 2018; pp. 58–61. [[CrossRef](#)]
29. Pandangan, Z.A.; Talampas, M.C.R. Hybrid LoRaWAN Localization using Ensemble Learning. In Proceedings of the 2020 Global Internet of Things Summit (GIoTS), Dublin, Ireland, 3 June 2020; pp. 1–6.
30. Janssen, T.; Berkvens, R.; Weyn, M. Comparing Machine Learning Algorithms for RSS-Based Localization in LPWAN. In *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*; Barolli, L., Hellinckx, P., Natwichai, J., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 726–735.
31. Gomez, C.A.; Shami, A.; Wang, X. Machine Learning Aided Scheme for Load Balancing in Dense IoT Networks. *Sensors* **2018**, *18*, 3779. [[CrossRef](#)] [[PubMed](#)]
32. LoRa Alliance Technical Committee Regional Parameters Workgroup. LoRaWAN[®] Regional Parameters RP002-1.0.0. 2019. Available online: <https://loro-alliance.org/resource-hub/lorawanr-regional-parameters-rp002-100> (accessed on 31 July 2020).
33. Semtech: LoRa. EP2763321 from 2013. U.S. Patent 7,791,415, 7 September 2008.
34. Li, S.; Raza, U.; Khan, A. How Agile is the Adaptive Data Rate Mechanism of LoRaWAN? In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, UAE, 9–13 December 2018; pp. 206–212.

35. Mahalakshmi, G.; Sridevi, S.; Rajaram, S. A survey on forecasting of time series data. In Proceedings of the 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16), Kovilpatti, India, 7–9 January 2016; pp. 1–8.
36. Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666. [[CrossRef](#)]
37. MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*; University of California Press: Oakland, CA, USA, 1967; Volume 1, pp. 281–297.
38. Lloyd, S. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **1982**, *28*, 129–137. [[CrossRef](#)]
39. Galicia, A.; Talavera-Llames, R.; Troncoso, A.; Koprinska, I.; Martínez-Álvarez, F. Multi-step forecasting for big data time series based on ensemble learning. *Knowl. Based Syst.* **2019**, *163*, 830–841. [[CrossRef](#)]
40. Lai, R.K.; Fan, C.Y.; Huang, W.H.; Chang, P.C. Evolving and clustering fuzzy decision tree for financial time series data forecasting. *Expert Syst. Appl.* **2009**, *36*, 3761–3773. [[CrossRef](#)]
41. Yamada, Y.; Suzuki, E.; Yokoi, H.; Takabayashi, K. Decision-tree Induction from Time-series Data Based on a Standard-example Split Test. In *Twentieth International Conference on Machine Learning*; Fawcett, T., Mishra, N., Eds.; The AAAI Press: Menlo Park, CA, USA, 2003; pp. 840–847.
42. Brunello, A.; Marzano, E.; Montanari, A.; Sciacicco, G. J48SS: A Novel Decision Tree Approach for the Handling of Sequential and Time Series Data. *Computers* **2019**, *8*, 21. [[CrossRef](#)]
43. Kaushik, S.; Choudhury, A.; Sheron, P.K.; Dasgupta, N.; Natarajan, S.; Pickett, L.A.; Dutt, V. AI in Healthcare: Time-Series Forecasting Using Statistical, Neural, and Ensemble Architectures. *Front. Big Data* **2020**, *3*, 4. [[CrossRef](#)]
44. Alhirmizy, S.; Qader, B. Multivariate Time Series Forecasting with LSTM for Madrid, Spain pollution. In Proceedings of the 2019 International Conference on Computing and Information Science and Technology and Their Applications (ICCISTA), Kirkuk, Iraq, 3–5 March 2019; pp. 1–5.
45. Zhang, X.; Liang, X.; Zhiyuli, A.; Zhang, S.; Xu, R.; Wu, B. AT-LSTM: An Attention-based LSTM Model for Financial Time Series Prediction. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *569*, 052037. [[CrossRef](#)]
46. Izal, M.; Morató, D.; Magaña, E.; García-Jiménez, S. Computation of Traffic Time Series for Large Populations of IoT Devices. *Sensors* **2019**, *19*, 78. [[CrossRef](#)] [[PubMed](#)]
47. Domb, M. An Adaptive Lightweight Security Framework Suited for IoT. In *Internet of Things*; Sen, J., Ed.; Chapter 2; IntechOpen: Rijeka, Croatia, 2018. [[CrossRef](#)]
48. Shitole, A.S.; Devare, M.H. Supervised machine learning supported time series prediction and analysis of iot enabled physical location monitoring. *Int. J. Innov. Technol. Explor. Eng.* **2019**, *8*, 3238–3245.
49. Zhang, W.; Guo, W.; Liu, X.; Liu, Y.; Zhou, J.; Li, B.; Lu, Q.; Yang, S. LSTM-Based Analysis of Industrial IoT Equipment. *IEEE Access* **2018**, *6*, 23551–23560. [[CrossRef](#)]
50. Zhang, X.; Zhao, M.; Dong, R. Time-Series Prediction of Environmental Noise for Urban IoT Based on Long Short-Term Memory Recurrent Neural Network. *Appl. Sci.* **2020**, *10*, 1144. [[CrossRef](#)]
51. Thorndike, R.L. Who belongs in the family? *Psychometrika* **1953**, *18*, 267–276. [[CrossRef](#)]
52. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [[CrossRef](#)]
53. Davies, D.L.; Bouldin, D.W. A Cluster Separation Measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *PAMI-1*, 224–227. [[CrossRef](#)]
54. Bontempi, G.; Ben Taieb, S.; Le Borgne, Y.A. Machine Learning Strategies for Time Series Forecasting. In *Business Intelligence: Second European Summer School, eBISS 2012, Brussels, Belgium, July 15–21, 2012, Tutorial Lectures*; Aufaure, M.A., Zimányi, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 62–77. [[CrossRef](#)]
55. Omlin, C.W.; Giles, C.L. Constructing Deterministic Finite-State Automata in Recurrent Neural Networks. *J. ACM* **1996**, *43*, 937–972. [[CrossRef](#)]
56. Xu, F.; Uszkoreit, H.; Du, Y.; Fan, W.; Zhao, D.; Zhu, J. Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges. In *Natural Language Processing and Chinese Computing*; Tang, J., Kan, M.Y., Zhao, D., Li, S., Zan, H., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 563–574.

57. He, T.; Xie, C.; Liu, Q.; Guan, S.; Liu, G. Evaluation and Comparison of Random Forest and A-LSTM Networks for Large-scale Winter Wheat Identification. *Remote Sens.* **2019**, *11*, 1665. [[CrossRef](#)]
58. Gers, F.A.; Eck, D.; Schmidhuber, J. Applying LSTM to time series predictable through time-window approaches. In *Neural Nets WIRN Vietri-01*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 193–200.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).