FIREMAN

WP4 Deliverable 4.1
Initial Results on Heterogeneous Big Data Aggregation

Project Title:	Framework for the Identification of Rare Events via MAchine learning and IoT Networks

Title of Deliverable:	Initial Results on Heterogeneous Big Data Aggregation
Status-Version:	Final-0.5
Delivery Date:	30/10/2020

Contributors:	Jean Michel de Souza Sant'Ana, Nelson Mayedo Rodriguez, Eslam Eldeeb, Hirley Alves, Mehar Ullah, Pedro Juliano Nardelli, Charalampos Kalalas, Merim Dzaferagic
Reviewers:	Francisco Vázquez, Samuel Montejo-Sanchez
Approved by:	All Partners

Document Revision History

Version	Date	Description
v0.1	02 September 2020	Table of Contents
v0.2	02 October 2020	Main contents added
v0.3	09 October 2020	Additional contents added
v0.4	12 October 2020	First version delivered for external review
v0.5	29 October 2020	Approved form of Deliverable D4.1

Table of Contents

1 Introduction

1.1 Objective of the document

The objective of Deliverable 4.1 is to present some initial outputs carried out in **Task 4.1: Aggregation of heterogeneous big data**. The goal of this task is "*to show how a large number of sensors and its data could be accommodated and aggregated at small cost for reliably detecting rare events*". Thus, this task represents the link between WP3, where we focus mainly on the communication aspects, transforming the multiple streams of data into something more appropriate to apply the data reduction algorithms from Task 4.2. Note that Task 4.2 may present data reduction algorithms that take place on different levels of the FIREMAN architecture, and thus, they are tied not only as an output from Task 4.1, but also as an input. To do that, the partners have been working into data aggregation scenarios: How the traffic behaves? Which performance indicator to use? How to efficiently aggregate the data? Moreover, it is important to understand the factors to take into account when choosing a platform to store the aggregated data. Finally, the document presents two new datasets used by the FIREMAN consortium. The first is an extension of the Teeneesse Eastman Process, that will enable the evaluation of some big data techniques. The second dataset represents two real use cases from SEAT, the industrial partner of the consortium.

1.2 Structure of the document

The remainder of this document is structured as follows: Section 2 presents some insights on data aggregation techniques for machine-type communications. On Section 3 we show a data aggregation clustering signalling method and evaluate it in terms of cluster average delay and resource efficiency. Section 4 presents a discussion on how and where to store the aggregated data, altogether with a discussion on the dataset used by FIREMAN. On Section 5 we summarize our conclusions.

2 Heterogeneous Data Aggregation in Machine-type Communications

2.1 Meta distribution of the SIR in Massive MTC Networks with Scheduling and Data Aggregation

A promising way to enable a massive number of simultaneously connected devices relies on the concept of data aggregation. This structure [1]: $i)$ shortens the distance in the communication, while diminishing the power consumption of the machine type devices (MTDs); $ii)$ reduces the number of connections to the core, thus decreasing the congestion; and $iii)$ extends network coverage.

Recently, several articles have investigated and exploited the advantages of data aggregation and clustering techniques in massive MTC (mMTC). Authors in [2] present a possible application of this concept for different phases of buildings life-cycle, especially the construction and exploitation phases. The goal is to design concrete equipped with embedded sensors that can regularly monitor temperature, corrosion and cracks, to ensure the safety of the building. To this end, the performance of three different data aggregation methods—cluster-based, tree-based and chain-based configurations—is evaluated. Arriving to the conclusion that cluster-based approach provides a robust network and low delay; thus, it may be a good choice for the exploitation phase. In [3], a novel data aggregation scheme is designed based on clustering of the nodes and an extended extreme learning machine algorithm, which efficiently reduces redundant data and energy consumption. Moreover, Kalman filter is employed to filter the data (minimize the variance) at each sensor node before sending it to the cluster head; and cosine similarity is used to cluster the nodes based on data similarity and density. In [4], it is proposed a combined clustering based data aggregation mechanism that can apply multiple clustering approaches simultaneously—static clustering for sensed targets located close from the sink, or dynamic clustering techniques when sensing activity is performed far from the sink—in a single network, depending on the network environment (number of hop count to the sink node, number of targets to sense and the target velocity). This mechanism can increase the data aggregation efficiency as well as improve energy efficiency. In [5], the authors proposed a cross-layer energy based clustering technique with dynamic data aggregation for heterogeneous networks, to form clusters of sensor nodes in hexagonal shape. In order to increase energy efficiency, every member node selects a cluster head on the basis of deduced optimum distance (proportional to the optimum consumption of energy) and the remaining energy in the nodes having a value greater than a threshold value (average residual energy). In order to make a balance between the consumption of energy and the network traffic, the rotation of cluster heads is performed dynamically. Finally, the authors of [6] presented a Distributed Clustering approach—the nodes are completely autonomous and decide by themselves if they become a cluster head or not—for data aggregation, guided by the base station at specific instants with only three transmissions (setup information) during the network lifetime. The transmissions help the nodes know about the status of the network. The nodes implement a novel Type-2 model for the fuzzy system to adapt the selection of cluster head to the changing characteristics of the network, which is more appropriate in real-time applications.

As first result of Task 4.1 ("**Aggregation of heterogeneous big data**"), we use the meta distribution (MD) of the signal-to-interference ratio (SIR) concept to fully characterize the performance of the uplink traffic in a Poisson network with data aggregation. We adopt the random resource scheduling (RRS) and channel-aware resource scheduling (CRS) scheduling schemes recently proposed in [7] to deal with the limited spectrum resources, but in contrast, we provide a more fine-grained performance characterization of a typical link. The results presented herein constitute a middle point between mathematical tractability and practical implementation feasibility, that can be used as a benchmark to analyze more challenging industrial scenarios (more sophisticated scheduling schemes, channel access and clustering mechanisms).

2.1.1 System Model

We study the uplink transmissions of a large-scale single-tier cellular network where aggregator nodes are spatially distributed according to an independent homogeneous Poisson point process (HPPP) [8], represented by Φ_p, with intensity λ_p. Since Φ_p is a stationary process, the distribution of the points is invariant with respect to translation of the origin; therefore, the SIR analysis does not depend on the particular location of each aggregator. Thus, according to Slivnyak's theorem [9] and without loss of generality, we consider a "typical" aggregator located at the origin which is subject to interference produced by the other non-intended transmitters in the network.

At any instant, the MTDs across the entire network transmit information to their serving aggregators through the same set $\mathcal{N} = 1, \ldots, N$ orthogonal channels; and each aggregator can accommodate only one MTD per channel, out of K requesting service within its coverage area—K is a Poisson distributed random variable with mean m, $K \sim \mathrm{Poiss}(m)$. Thus, the only contribution to the interference comes from the MTDs in the serving zones of other aggregators using the same channel (inter-cluster interference)[1]. Notice that each MTD transmits whenever it has new information to send and its corresponding aggregator has allocated resources for transmission—conforming to two scheduling schemes: RRS and CRS[2] described in the following sections. Assuming that the MTDs have low mobility, we can model their location as a Matérn cluster process (MCP)[3], where the aggregators form the parent point process [7].

The MCP can be defined as

$$\Phi \triangleq \bigcup_{\mathbf{v} \in \Phi_p} \mathbf{v} + \mathcal{B}^{\mathbf{v}}, \tag{1}$$

where $\mathcal{B}^{\mathbf{v}}$ denotes the offspring point process; and each point $\mathbf{s} \in \mathcal{B}^{\mathbf{v}}$ is independent and

[1] The probability that any MTD within any cluster generates interference does not depend on its position in the area respect the typical link—the channel occupation probability is $P_0 = K/N$ when $N > K$ and 1 otherwise. Thus, based on the independent thinning property, we can model the interference field observed from the typical link as a HPPP with density $P_0\lambda_p$.

[2] The authors in [7] show that CRS and RRS schemes achieve similar performance as long as the available resources in the aggregator are not very limited, while CRS outperforms RRS when the number of MTDs requesting service exceeds the amount of available channels. These scheduling mechanisms were extended in [10], [11] under non-orthogonal multiple access (NOMA) and imperfect successive interference cancellation.

[3] This is a doubly Poisson cluster process that reflects the properties of the scenario treated in this work, compared to other point processes belonging to the same group such as the Thomas process [9].

identical distributed around the cluster center $\mathbf{v} \in \Phi_p$ with distance distribution $f(r_d) = \frac{2r_d}{R_d^2}$, where R_d is the radius of the clusters formed by the aggregators and its corresponding MTDs [12]. Notice that the definition of the MCP implies that each MTD is associated with a single aggregator even though it might be the case that a particular MTD is located within the coverage areas of several aggregators.

We adopt a channel model that consists of the commonly used power-law path-loss as the large-scale propagation effect, where the signal power decreases at a rate of $r^{-\alpha}$ with the propagation distance r, and $\alpha \geq 2$ is the path loss exponent. Also, quasi-static fading is considered as the small-scale effect, which means that the channel is constant during a transmission block, and changes independently from block to block. Additionally, Rayleigh multi-path fading environment is assumed, where intended and interfering channel power gains (h and g, respectively) are exponentially distributed with unit mean. This allows us to examine the worst case scenario, without line of sight.

All MTDs use full inversion power control. This is, each device controls its transmit power in such a way that the average signal power received at the serving aggregator is equal to a predefined constant value ρ. This guarantees a uniform user experience while saving an important amount of energy [10]. Due to the high density of MTDs and aggregators, we consider an interference-limited scenario (i.e., performance of all links is limited by the co-channel interference, and the thermal noise at the receiver side can be neglected); consequently, the received SIR determines the network performance and the value of ρ is irrelevant.
The system model presented in this section could be used as a benchmark to analyze more complex industrial scenarios with more sophisticated scheduling policies.

2.1.2 The MD of the SIR

Since MD is a relatively new concept introduced in Wireless Communications—the formal definition was firstly given in [13]—we find useful to provide some interpretations related to the MD. The link success probability given a SIR threshold θ, $p_s(\theta) = \mathbb{P}(\mathrm{SIR} > \theta)$, is a performance metric of interest in large-scale interference-limited networks because it allows designers to know the fraction of MTDs that succeed in transmitting. The computation of p_s requires spatial averaging over the point process, thus, it does not reveal how concentrated the link success probabilities are; therefore, it is not possible to properly distribute the resources across the network. For this reason, it is important to measure the fluctuation of the link reliability around p_s to fully characterize the performance of the network in terms of connectivity, end-to-end delay and QoS. Thus, we center our attention in random variables of the form

$$P_s(\theta) \triangleq \mathbb{P}(\mathrm{SIR} > \theta | \Phi), \tag{2}$$

where the conditional probability is taken over the fading and the channel access scheme and given the position of the nodes for a particular realization of the network. Following this notation, the standard success probability would be $p_s(\theta) = \mathbb{E}[P_s(\theta)]$. The intention is then to find the two-parameter complementary cumulative distribution function (CCDF) of $P_s(\theta)$, defined as [13]

$$\bar{F}(\theta, x) \triangleq \mathbb{P}(P_s(\theta) > x), \tag{3}$$

where $x \in [0, 1]$ refers to the target reliability level. Due to the ergodicity of the point process, one can understand $\bar{F}(\theta, x)$ as the fraction of links or users that achieve an SIR threshold

θ with probability at least x [14]. Closed-form expressions or approximations to this metric different methodologies were proposed in [13], [15], [16].

2.1.3 Random Resource Scheduling (RRS)

Under RRS, each aggregator randomly assigns the channels in $\mathcal{N}$ to the MTDs. Notice that this mechanism does not need channel state information (CSI). Supose the typical MTD is transmitting to the typical aggregator. Since the MTDs use inversion power control, the SIR experienced by the typical user is $\mathrm{SIR} = \frac{h}{I}$, where $I = \sum_{i\in\Phi\setminus\{0\}} g_i r_{d_i}^{\alpha} y_i^{-\alpha}$ is the aggregated interference from MTDs in other clusters transmitting over the same channel, $\{y_i\}$ denotes the distance of the interfering MTDs respect the typical user, h and $\{g_i\}$ are the fading power gains on the desired and interfering links, respectively, and $\{r_{d_i}\}$ is the distance between the MTDs and their serving aggregators. For an arbitrary but fixed realization of Φ, the MD can be obtained from (2) as

$$\bar{F}(\theta, x) = \mathbb{P}\left[\mathbb{P}\left(\frac{h}{\sum\limits_{i\in\Phi\setminus\{0\}} g_i r_{d_i}^{\alpha} y_i^{-\alpha}} \geq \theta \middle| \Phi\right) > x\right] = b a^{-\frac{1}{2}} \Gamma\left(\frac{1}{2}, -\frac{\theta}{\ln x}\right), \tag{4}$$

where $a = \frac{t^2}{4}$, $b = \frac{c}{2\sqrt{\pi}}$ and $c = \frac{1}{2} P_0 \lambda_p \pi R_d^2 \Gamma\left(1 - \frac{2}{\alpha}\right)$. Note that $P_0\lambda_p$ is the density of the served MTDs and $P_0 = \mathbb{E}_K\left[\frac{min(K,N)}{N}\right]$ is the average channel occupation probability, which is independent of the scheduling scheme.

2.1.4 Channel-aware Resource Scheduling (CRS)

Contrary to RRS scheme, the aggregators that implement CRS allocate the available channel resources to the MTDs with better SIR (equivalently, better channel gains). Herein, each aggregator is assumed with perfect CSI knowledge of its associated MTDs. The SIR MD is then

$$\bar{F}(\theta, x) = \mathbb{P}\left(1 - \sum_{l=q}^{K}\sum_{r=0}^{l} \binom{K}{l}\binom{l}{r} (-1)^r \prod_{i\in\Phi_I} \frac{1}{1+\theta_{l,r}(\frac{r_{d_i}}{y_i})^{\alpha}} > x\right), \tag{5}$$

where $\theta_{l,r} = \theta(K-l+r)$, $q = K - \nu + 1$ and $\nu = 1, ..., N$ corresponds to the selected MTDs ordered according their SIR as $h(1) > \ldots > h(\nu) > \ldots > h(N)$. Herein, we analyze the worst case performance, which is when $\nu = N$. Notice that (5) allows a semi-analytical computation of the MD that depends only on the position of the interfering nodes inside the clusters and with respect to the typical link. Therefore, it is not necessary to model neither the channel fading nor the scheduling process, which reduces significantly the computation time.

2.1.5 Numerical Results

The numerical results and insights provided in this deliverable constitute part of the paper [17]. The aggregators are deployed in a disk of radius 3 km with density $\lambda_p = 3 \times 10^{-6}$ aggregators/m^2, which guarantees 100 aggregators deployed on average in the area, while

eliminating the impact of border effect in the simulation. It was selected $N = 20$, $m = 60$, and $R_d = 40$ m to produce visualization of errors in the order of 10^{-2}. Monte Carlo based results are obtained with 10^5 samples and are included in the figures with markers to validate our analytical (for RRS) and semi-analytical (for CRS) expressions, represented with lines.

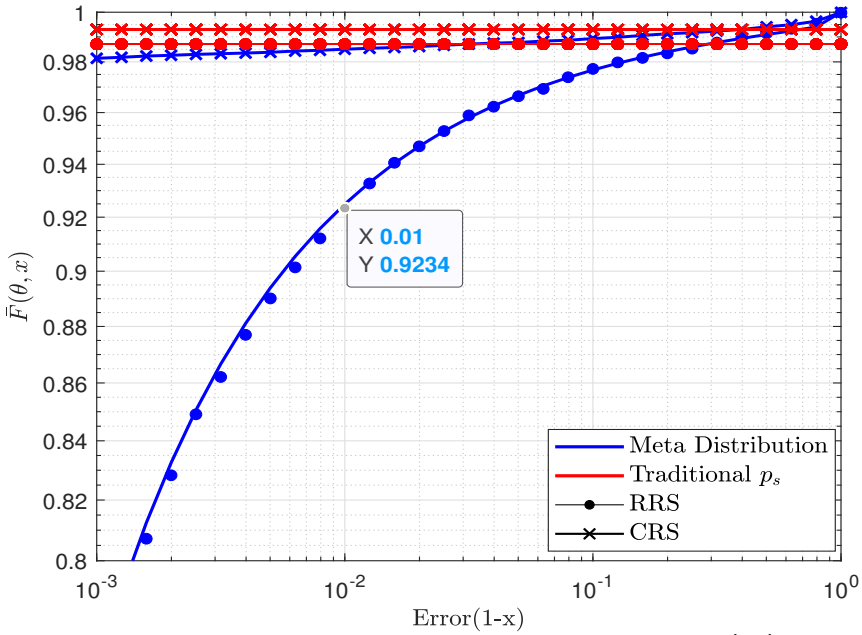


Figure 1: Meta distribution and the traditional success probability (p_s) for $\alpha = 4$ and $\theta = 0$ dB. The marked point can be interpreted as approximately 92% of the users can communicate with probability of error less than or equal to 0.01..

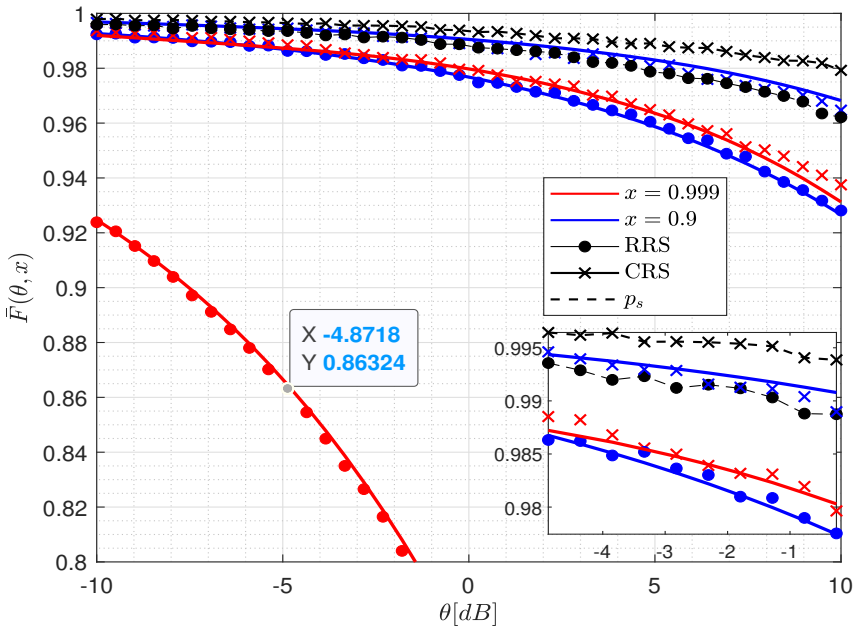


Figure 2: Meta distribution and traditional success probability (p_s) as a function of θ for different values of reliability (x).

Fig. 1 shows the efficacy of the SIR MD for describing the system per-link performance. One can realize that the traditional p_s does not guarantee QoS for any node in the network, which is even more remarkable when RRS is enabled in the aggregation phase. This is because the channels are assigned to links that communicate with high error probability. In contrast, having the MD in hand allows for a more effective distribution of those resources as the exact fraction of links communicating with a target reliability is known in advance. Moreover, aggregators implementing CRS admit a higher percentage of links achieving a target reliability in the resources-constrained communication system.

Fig. 2 permits a more rigorous analysis of the fraction of links that achieve certain reliability given the SIR threshold θ. For example, the marked point shows that the transmission rate should be set no greater than $\log_2(1 + 10^{-5/10}) = 0.396$ bits per channel use (bpcu) to guarantee nearly 86% of the links achieving at least 99.9% success probability. One may notice that nearly the same fraction of devices can transmit with the same rate under both scheduling schemes, but with reliability improved from 0.9 when using RRS to 0.999 under CRS. If simplicity is desired in the network, RRS can be a solution, but with only a small percentage of devices achieving high reliability. However, if a larger number of devices need to communicate, CRS seems to be the best option to provide them the required reliability.

For future deliverable, we aim to explore analytical approaches to characterize the meta distribution under CRS and implement an algorithm that ensures efficient power and rate control.

2.2 Coupled Markov Modulated Poisson Process Traffic Model

Traffic models are one of the fundamental parts of any communication system [18]. Every communication system has its own requirements and targets, which need to be achieved using different models and algorithms. Designing an efficient and robust model, that meets communication system targets, is not a straightforward task, but, it needs a deep understanding of all network parts that are related and may affect this model. Traffic modeling means to capture the behaviour of the physical quantities using a probabilistic model that can be implemented easily using computer software [19]. Understanding the traffic of the network is a key step while implementing new models to enhance the network performance. Modelling the traffic behavior of the network in terms of how that traffic is arriving, distributed in time and space and the relation between the traffic of different devices within the same network will help us implementing an efficient aggregator. In this section, we will discuss different current traffic models and their limitations, then, we will design an efficient traffic model that capture the traffic efficiently and could be used as an input for data aggregation schemes.

2.2.1 Source Semi-Markov Model (SSMM)

Firstly, we should define the types of traffic of machine type communications. According to the SSMM model presented in [20], MTC traffic is classified into 3 different classes:

1. Periodic Update (PU)

   It describes a periodic traffic, which is established every period of time. For example, a temperature sensor that senses the temperature and transmit a packet every 2 minutes.

It is characterized by their short packets, small number of packets. If we imagine the status of the sensor as a state, data to be transmitted can be found in two states only: Transmission State and Silent State. In addition, it is easy to schedule these devices with resources to transmit their data and perform aggregation algorithms.

2. Event-Driven (ED)

   It describes a non-periodic traffic, which is established due to a certain random trigger within unknown instants. For example, a temperature sensor that senses the temperature and only sends packets when the temperature exceeds a certain threshold. It is characterized by their large packets, where alarm packets are known to be longer than normal data packets. It can be found in two states as well: Transmission state and Silent state. In addition, some devices are characterized by the two traffic patterns PU and ED. For example, the temperature sensor senses the temperature and reports their data to a gNB every two minutes with short data packets, then if the temperature exceeds a certain threshold, it reports to the gNB with long alarm packets. These devices can be found in three states: Data Transmission State, Alarm Transmission State and Silent State. These kinds of packets describe most of real-time traffic patterns of real-time devices.

3. Payload Exchange (PE)

   It describes a traffic that usually comes after one of the above two traffic kinds PU or ED. It is characterized by the bursty traffic. For example, after transmitting PU / ED traffic, the data is supported by a picture or a summary report which has very large packet size.

2.2.2 Traffic Models Types

Traffic models are classified into source traffic models and aggregated traffic models [21]. Source traffic models treat every single MTD as a single entity, where it tries to capture the traffic of each individual device. Source traffic models are very accurate, however, it becomes extremely complex when we deal with a network that has large number of devices. In such cases, source traffic model will not be a good candidate. Massive MTC (mMTC) applications, for example, are not recommended to be modeled using source traffic modeling approach. On the other hand, aggregated traffic models treat all the MTDs within the network as one entity. It simply accumulates all the traffic to have one stream. Aggregated traffic models are suitable in mMTC applications, where heterogeneous traffic can be aggregated as a single stream. Reader may think that aggregated traffic models are the one that should be used, especially, that aggregation is on of our goals, however, MTD has become more complex than before, making aggregation of traffic less efficient. Aggregated traffic models are less complex than source traffic modeling, but less accurate as well. We need to design an efficient traffic model, that can capture the traffic behaviour efficiently and simply. Next, we will discuss CMMPP traffic model based on the work in [22].

2.2.3 CMMPP Traffic Model

Markov processes and Poisson processes are very common in traffic models studies and queuing problems [23], [24]. Markov Modulated Poisson Processes (MMPP) are one of the recent theories that has been developed mainly for traffic modelling scenarios. Using such models in capturing the traffic of MTC devices seems to be very promising. It suggests that each device is simulated with a Markov Chain and a Poisson Process. Markov Chains $s_n[t]$ determine the transition rate $\lambda_i[t]$, which modulates the Poisson Processes. Transition probabilities are formed for each device to define the transition between states for each device. Let us consider a 2-state MMPP, where the first state is the data state, which describes regular packets transmission and the second state is alarm state, which describes longer packet transmission. Each device is transiting from data state to alarm state according to state probabilities, which is MMPP-based. State transition matrix P and state probability vector π are formed as follows [22]:

$$P = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,j} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ p_{i,1} & p_{i,2} & \cdots & p_{i,j} \end{pmatrix} \tag{6}$$

$$\pi = \begin{pmatrix} \pi_1 \\ \pi_2 \\ \vdots \end{pmatrix}. \tag{7}$$

Furthermore, this source traffic model based MMPP can be simpler if we considered one background process acting as a modulator for all MTC devices. This background process influences all the MTC devices, but with different strengths according to its position from the epicenter (position of the background process). There are 2 main key points, which we should refer to while explaining this model: 1. the correlated behaviour of the MTC devices, which helps understanding how the background process is affecting them, 2. This background process describes some sudden event such as fire or high temperature, which causes some devices to transit from state 1 (data) to state 2 (alarm). This background process is acting as a master node which controls the transition of devices and allows coupling multiple MMPP. Coupling Markov Chains means that multiple chains mutually influence their transition probability matrices. This background process correlates the devices in both time and space. Let us consider the state probability matrix be composed of 2 matrices, coordinated matrix P_C and uncoordinated matrix P_U, which are valid for all devices. The main assumption for this model is that all devices tend to trigger data packets all the time. For the coordinated matrix, which describes the behaviour of the devices near to the epicenter, its main characteristic is to issue an alarm at a time (alarm state), then go back to the data state again. On the other hand, for the uncoordinated matrix, which describes the behaviour of the devices away from the background process, its main characteristic is to remain at the data state and never switch to the alarm state. The master background process issues samples $\theta[t]$, which are function of

time and space for all devices. Transition matrix can be calculated as follows:

$$P_n[t] = \theta_n[t].P_C + (1 - \theta_n[t]).P_U, \tag{8}$$

$$\theta_n[t] = \delta_n.\theta[t], \tag{9}$$

where P_C is the transition matrix for coordinated devices, and P_U is the transition matrix for uncoordinated devices. In addition, $\theta[t]$ consists of samples, which describe how the devices are affected by the background process in time. It is uniformly distributed over time and $\theta[t] \in [0, 1]$. $\delta_n \in [0, 1]$ consists of samples, which describe how the devices are affected by the background process in space. Devices near to the epicenter of the background process are highly affected by the process and transit to state 2 (Alarm) than devices far away from the epicenter. It is normally distributed over space, its mean is the epicenter of the background process and the variance describes how strong is the background process (higher variance leads to stronger process which means further devices affected). Multiplying $\theta[t]$ and δ_n results in $\theta_n[t]$, which describes how the devices are affected by the background process in jointly space and time, that is why $\theta_n[t]$ samples are considered to be generated by the background process. The same idea is repeated with the state probability vector $\pi_n[t]$, which will be composed of π_C and π_U.

2.2.4 Simulation and Results

Let us consider a 2-state model, state 1 (Data) and state 2 (Alarm). Let us also assume that the Poisson arrival rate of the data state is very small, while the Poisson arrival rate of the alarm state is much larger than the data state. Moreover, define the transition matrices:

$$P_C = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{10}$$

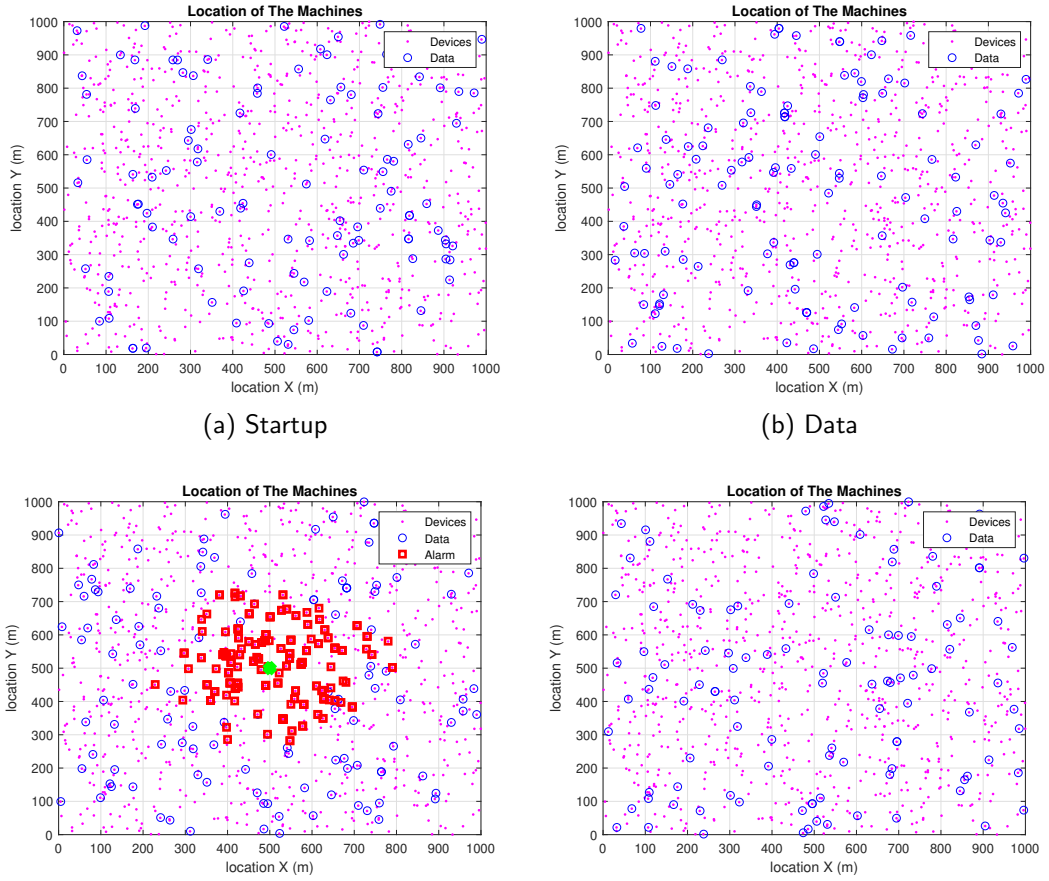
$$P_U = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}. \tag{11}$$

Next, we can extend this 2 state model to 4 state model named as startup state and silent state. Both states, can be extended from the original 2 states model, without changing the transition matrices or the arrival rates. The startup state is assumed to be the state that describes the starting behaviour of the devices while transmitting data randomly. In addition, devices at alarm state go directly to the silent state after transmitting the alarm packets. We call it silent state. Using the distributions of $\theta[t]$ and δ_n, the coordinated matrix in (10), and the uncoordinated matrix in (11) along with the numerical values in Table 1, we will simulate the CMMPP traffic model in matlab.

We can see in Fig. 3 the 4 states previously mentioned. First, in Fig. 3 (a) Startup state, all the devices tend to transmit data. Then, in Fig. 3 (b) Data state, some devices transmit data at different instants and different packet lengths. In Fig. 3 (c) Alarm state, devices near to the epicenter transmit an alarm in a correlated behaviour with large packet lengths. Finally, in Fig. 3 (d) Silent state, the devices that transmitted alarm become silent.

Table 1: Values used in CMMPP Simulation

Parameter	**Value**
Run Time	60 s
Time step (Instants for each second)	$\frac{1}{60}$
ΔT	1
Total Time Instants	1:3600
Number of Machines	1000
Number of States	2
λ_{Data}	$\frac{1}{3600}$
λ_{Alarm}	$\frac{1}{\delta T} = 1$
Time Interval for Background Process	1700:1900
Epicenter of the Background Process	500
Variance of the Background Process	100
Intervals for the 4 States	[0 400; 1000 1400; 1600 2000; 2400 2800]



(a) Startup

(b) Data

(c) Alarm

(d) Silent

Figure 3: CMMPP Results: 1000 MTD, runtime = 60 s, $\lambda_{Data} = \frac{1}{3600}$, and $\lambda_{Alarm} = 1$

3 Signature-based Cluster Formation

3.1 Background

One of the key objectives of Task 4.1 is the efficient aggregation of the heterogeneous and highly dimensional data generated by a large number of sensors in industrial environments. In this context, the exploitation of localized cluster formulation techniques holds the promise of reducing the necessary signalling exchanges for the establishment of sensors' connectivity with the aggregator nodes [25, 26]. Besides, local sensor clustering aims to exploit the pairwise correlations in the ambient measurement space yielding a partition of the set of available sensors into subsets of highly space-temporal correlated sensors. In turn, timely identification of highly correlated streams can be further exploited at the aggregator nodes as an indication to verify the existence of a detected rare event [27].

Traditional clustering mechanisms need to be enhanced to accommodate the signalling overhead associated with cluster formation (e.g. due to cluster header selection and assignment, or device discovery), taking into account the communication capabilities (e.g., limited communication infrastructure or provisionally deployed access points) and network topology characteristics in the industrial plant. At the same time, cluster formation time, i.e., the required latency for discovering all potential sensor-aggregator links that formulate the clusters, should be kept at minimum, especially for the time-sensitive industrial applications with stringent delay budgets.

Another important aspect in cluster formulation refers to the radio resource utilization, i.e., the amount of resources required for the sensors to establish connectivity links with their cluster head. Clustering algorithms that employ periodic beacon-based peer-discovery methods typically suffer from uncontrolled collisions in highly-dense scenarios which often lead to underutilization of the scarce radio resources. The uncontrolled collisions in the transmission of beacons may result in severe performance degradation and failures in connectivity establishment, especially in use cases where cluster formation is event-triggered as a result of an incident.

3.2 Proposed Method

Motivated by the aforementioned limitations, we introduce a signature-based cluster formation scheme that aims to minimize the signalling overhead based on a flexible structure of the involved peer-discovery signatures [28]. Each signature constitutes a binary representation of each sensor/aggregator ID and offers a compact probabilistic way to represent the identity of the sensor. Assuming a cellular network deployment, the scheme relies on a discovery entity, i.e., residing at the base station (gNB), which gathers information relevant to the proximity of the sensors, to formulate the clusters. In particular, we consider a single-cell network composed by a gNB and multiple sensors, some of them with increased computational capabilities capable of acting as aggregator nodes, i.e., data sinks. Sensors are able to broadcast discovery messages with information that could be of interest to the aggregator nodes in the cluster. In turn, aggregators listen for certain information of interest, transmitted by neighbouring sensors. We further assume that each sensor transmission in a cluster is dimensioned with certain power such that potential aggregators within a discovery distance D will detect it with high

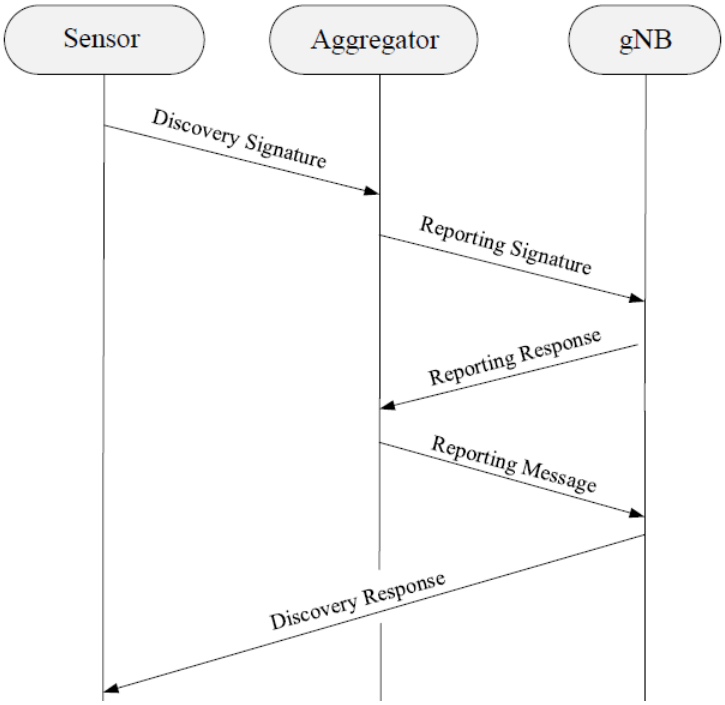


Figure 4: Proposed signal exchange for cluster formation among sensors, aggregators and the gNB.

probability.

As illustrated in Figure 4, the peer-discovery and cluster formulation scheme consists of three sequential phases of message exchange among a transmitting sensor, a receiving aggregator residing within D, and the discovery entity at the gNB. Cluster formation is initiated by a sensor which broadcasts a discovery signature to its nearby aggregators in the cluster. Each discovery signature constitutes a probabilistic data structure which contains the identification (ID) of the sensor. In the case when more than one sensors transmit the same signature, the resulting intra-cluster interference may hinder the successful decoding of the discovery signatures at the aggregator.

In the second phase, each potential aggregator within distance D becomes activated upon the reception of at least one discovery signature in its cluster. The aggregator node then transmits a reporting signature to the gNB which contains its ID. In turn, the gNB decodes the received reporting signatures and replies with a reporting response message to the detected aggregator nodes. If more than one aggregators transmit the same reporting signature, a decoding failure at the gNB may occur. The reporting response message contains an uplink resource grant allocated to each successfully-detected activated aggregator node for the transmission of the subsequent reporting message. Each reporting message contains information on the sensors' IDs received by the activated aggregator during the first phase.

At the end of the second phase of the cluster formation mechanism, the gNB has received the reporting messages from all successful aggregator nodes and acquires full knowledge of the sensor IDs in each cluster. Thus, by comparing the reported IDs, the gNB is able to identify the proximity relations of the sensor-aggregator pairs for each cluster. In the third phase, the

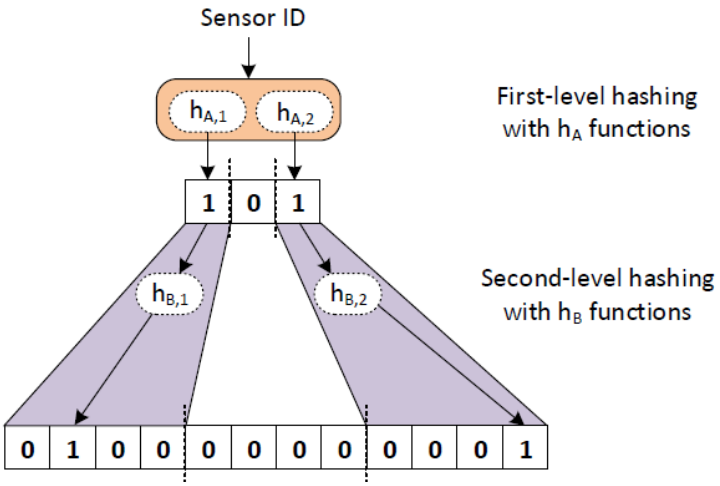


Figure 5: Illustration of a two-tiered signature generation with K=2, L=3 and M=4.

gNB transmits a discovery response message to all sensor IDs which initially sent a discovery signature. The rationale of the proposed mechanism therefore ensures that the resources are allocated in response to the number of discovered sensors within a cluster, preventing the underutilization of the available spectrum [28].

3.2.1 Signature Generation

A key part of the proposed cluster formation mechanism refers to the signature construction process. Recall that each signature constitutes the binary representation of each sensor/aggregator ID. As illustrated in Figure 5, each signature is a randomized data structure generated following the Bloom filter principles [29] and consists of a sequence of L resource slots in the time domain.

In particular, the signature generation process follows two sequential steps where the sensors' features are mapped into indices in a two-dimensional binary array using K modulo-operation hash functions. First, K index positions are activated in a frame of length L by using K independent hash functions h_A. At a second stage, by using another set of independent functions, h_B (one for each activated slot), a contending sensor/aggregator randomly selects and transmits one of the M available resource preambles [30]. The signatures are first initialised as empty Bloom filters with all the resource slots set to 0. Then, the aforementioned double hashing technique is applied as a space-efficient way of vectorizing the IDs of the sensors/aggregators and successively mapping them in the discovery/reporting signatures, respectively.

The inherent flexible structure of the signature, i.e., combination of several resource slots, allows the minimization of the collision probability of the transmitted signatures. The preamble collision probability can be also improved by a proper spatial allocation of the available orthogonal preambles along the different discovery clusters [30]. However, the mitigation of signature collisions comes at the cost of introducing false positives, an intrinsic characteristic of probabilistic data structures. In particular, while hashing collisions can be avoided in the signature generation, false positives may occur, i.e., signatures that were not originally transmitted by active sensors/aggregators but are detected as such at the receiving entity,

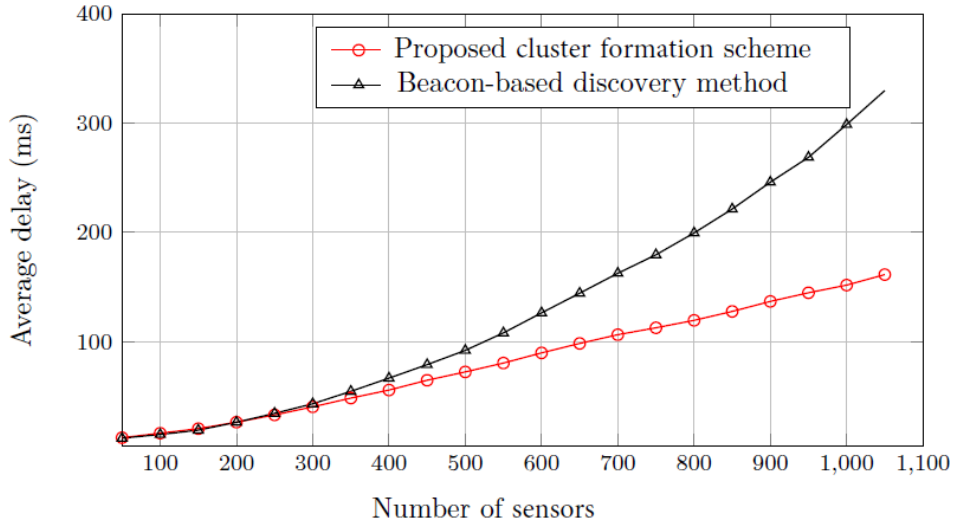


Figure 6: Average delay for cluster formation with increasing network load.

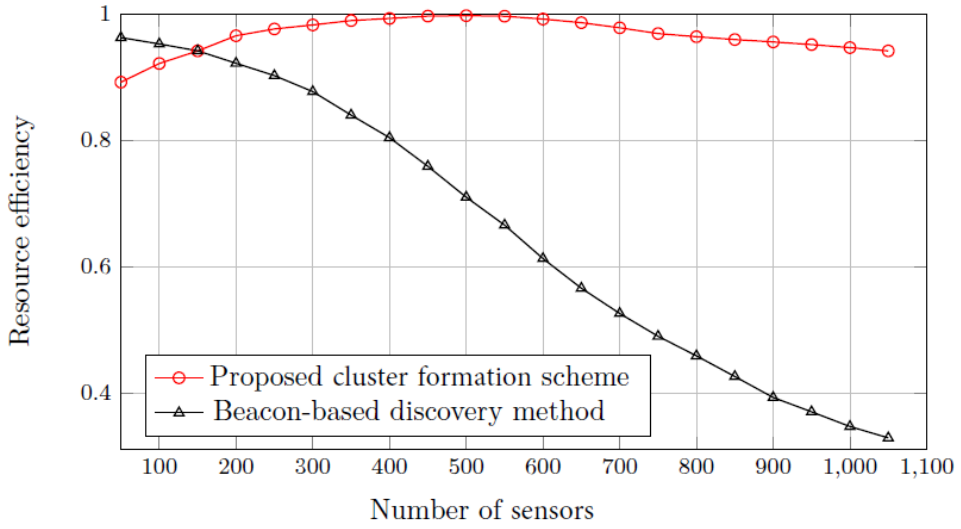


Figure 7: Resource efficiency with increasing network load.

i.e., aggregator/gNB respectively. To optimize the cluster formation scheme, the signature properties, i.e., K, L, M, can be dynamically tuned based on the estimated network load to minimize the false-positive probability.

3.3 Indicative Results

For performance analysis, stochastic geometry tools have been used to derive analytical expressions of fundamental metrics, such as collision probability (at all stages of the cluster formation) and link discovery probability. A preliminary assessment of the proposed cluster formation scheme against beacon-based schemes in terms of average delay for cluster formation and resource utilization is shown in Figures 6-7, respectively. The numerical results demon-

strate the superiority of the proposed method especially in the high traffic load regime where it substantially outperforms the benchmark scheme which suffers from uncontrolled collisions. In addition, allocation of peer-discovery resources is performed in response to the number of discovered sensor-aggregator pairs, thus preventing the underutilization of the scarce radio resources due to beacon collisions and false-positive signatures.

4 Databases and Use cases

4.1 Internet of Things and its Platform

The advancement in technology has increased opportunities for companies to expand their business area and process. Internet-of-Things (IoT) is one of the fastest growing technologies that is gaining momentum in the various domains like transportation, healthcare, industrial automation, education sector etc. The main idea of IoT is to connect the physical world with the digital world [31]. The foundation technology for IoT is Radio-Frequency IDentification (RFID) technology, which is used to identify, track and monitor any object with RFID tags and allow microchips to transmit the identification information to a reader through wireless communication [32]. Nowadays, IoT applications have already moved further away than just simple RFIDs, incorporating different sources of data collection from sensors. This data stream needs to be moved somewhere where this (big) data can be processed using, for example, machine learning techniques.

For companies to run their specific IoT applications, an IoT platform is then needed. The IoT platform provides important services and features to applications: endpoint management, connectivity and network management, analysis and processing, data management, application development, security, event processing, monitoring, access control and interfacing [33].

The technological change creates many challenges for businesses, governments and companies, which have little experience about the infrastructure of IoT and IoT platforms.There are hundreds of IoT platforms in the market, most with similar functionality with differences related to their implementation and underlying technologies [34]. Selecting a suitable IoT platform among all existing options is a tricky task since this decision needs to incorporate not only the current needs but also the potential future ones [34].

The main theme of this task is on how to store a huge amount of aggregated data. Thus, first we want to highlight the key building blocks of IoT for the understanding of functionality and significance of IoT, identify and verify the key factors of an IoT platform using experts opinion. With the key factors in hand, we could then propose an objective and general methodology to compare the different service providers. In this case, our study develops a theoretical framework that will support companies in selecting a suitable IoT platform for their business needs. we have followed three steps: (i) data collection, (ii) data verification and characterization, and (iii) application of the proposed framework. These are the questions that were used to guide our research: (1) What is IoT as well as its building blocks? (2) What are the important factors of an IoT platform? (3) What factors should be considered for selecting an appropriate IoT platform for specific organizations?

Our goal is to provide an objective while general methodology that different organizations can apply when selecting the most suitable platform based on their particular needs.

4.1.1 Building blocks of IoT

To understand the functionality and significance of IoT, it is essential to understand its building blocks; they are the components of IoT, which work together to deliver its functionality. There are six IoT building blocks that work together and provide functionality [31]. In the following, we will explain each of them in more details.

Identification block: The identification method is used to identify devices in the network. Devices are identified with the Object ID, which is the name of the device, and the object address, which provides the address of the device in the communication network [35]. The main addressing methods of IoT objects are IPv6 and IPv4 [31].

Sensing block: Sensors are used for collecting the data of objects/environment in the communication network and sending the collected data to the destination database or to the cloud. The data collected is analyzed in the cloud. Actuators, i.e. hardware mechanical devices such as switches, are also used in IoT platforms and operate in the opposite way to a sensor [31], [36], [37].

Communication block: It contains many heterogeneous objects that exchange data and various services with each other and with the platform. The communication block contains IoT communication protocols like MQTT and CoAP that are used to connect different objects to IoT and to send data from those connected objects to the management system. The sensors and other devices are connected to the Internet by communication technologies like ZigBee, NFC, UWB, Wi-Fi, SigFox, and BLE [38], [31].

Computation block: The computation block consists of two parts, hardware and software. Many hardware platforms have been built to run IoT applications, for example, Intel Galileo, Raspberry PI, Gadgeteer, UDOO, and Arduino. Similarly, there are many software platforms that are used to perform the functionalities of IoT. The main software platform is the operating system that runs throughout almost the whole activation time of the device. The cloud platform is also a computational component of the IoT; it enables small objects to send data to the cloud, it facilitates big data processing in real time and helps the end user to obtain knowledge extracted from the big data [38], [31].

Services block: IoT services aid IoT application developers by providing a starting point for development. When developers know the services available, they mainly focus on building the application rather than designing the service and architecture for supporting the IoT application. IoT services are divided into four categories. i) Identity related services, which are divided in active and passive. Services that broadcast information and have a constant power or take power from the battery are active. Passive identity related services have no power source and need some external device or mechanism to transmit its identity; ii) Information aggregation services refer to the actions of collecting data from sensors, processing that data, and transferring it to the IoT application for processing; iii) Collaborative aware services use the data provided by the information aggregation services to make decisions and react accordingly; iv) Ubiquitous services provide collaborative aware services anytime to anyone who needs it anywhere [36], [39], [40].

Semantic Block: IoT provides different services, for which it needs knowledge, and in order to get that knowledge in an effective way, IoT uses different machines. Knowledge extraction can include finding and using resources, modeling information, and recognizing and analyzing data to reach some decision and provide the correct service. So, it can be claimed that the semantic block is the brain of the IoT [38], [31], [36].

4.1.2 Key factors of an IoT platform

An IoT platform is the main part of an IoT solution. There are hundreds of IoT platform vendors in the market, and finding and selecting a suitable IoT platform that is reliable and

Table 2: Questions used in survey, during the Delphi method

Q#	Survey question
Q1	What is your opinion about the importance of stability of IoT platform?
Q2	What is your opinion about the importance of Scalability of the enterprise of IoT platform?
Q3	Do you think that IoT platform should be flexible with the advancement of technologies?
Q4	Do you think it is important to know about the pricing models before selecting IoT platform?
Q5	Do you think IoT platform should provide security at both the ends, software and hardware?
Q6	Do you think IoT platform can reduce Time to market for the business?
Q7	Do you think IoT platform should support the basic descriptive, predictive and perspective analytics?
Q8	Do you think it is important to know who will own the data collected by IoT platform?
Q9	Is it important to know the application environment of IoT platform?
Q10	Do you think it is important to know the Ownership of cloud infrastructure?
Q11	Do you think extend of legacy architecture in IoT platform is important?
Q12	Do you think Edge intelligence is important for IoT platform?
Q13	Do you think IoT platform needs high bandwidth networking?
Q14	Do you think it is important for IoT platform to support new Protocols and its updated versions?
Q15	Do you think the IoT platform venders should implement some steps to keep System performance high?
Q16	Do you think the IoT platform providers should have some dedicated infrastructure to handle customer data if there is some problem in IT infra?
Q17	Do you think Hybrid cloud is important for IoT platforms?
Q18	Do you think IoT platform providers should provide facilities to customers for any possible migration to other IoT platform in future?
Q19	Do you think IoT platform Interoperability will enable the organization to get higher productivity?
Q20	Is it necessary to check the previous experience of IoT platform, before selection?
Q21	Is it necessary that user interface of the IoT Platform should be simple and attractive?

scalable is difficult. However, consideration of some key factors prior to making a platform selection decision can enable companies to find and select an appropriate IoT platform for their business. The platform requirements are context-specific and it is not necessary that a platform include all the factors discussed below, but can have a maximum. These factors were identified from literature by studying various IoT platforms [41–45], articles [38, 46, 47] and websites, such as [48–50].

We employed here a two-round Delphi study. During the first round of the Delphi study fifteen experts from three different universities were selected based on their experience in IoT field. A questionnaire was designed based on twenty-one questions related to the key factors of IoT platforms as show in Table 2. A 5-point Likert rating scale was used: (1) totally disagree, (2) disagree, (3) neutral, (4) agree, and (5) totally agree. The questionnaire was sent to the experts by email to be answered within two weeks. Fourteen experts replied and the response percentage was 93%. In the first round, agreed percentage is 80%, disagree percentage is 6% and neutral percentage is 14%.

There was little conflict between the opinions of the experts about the first round questions. The second round questionnaire was designed based on the experts' opinion of the first round. The original questions were the same as before only the summary of the experts opinion of the first round was subsequently sent to the same experts. In the second round, fourteen experts replied and the response percentage was 93%. In the second round, some of the experts have changed their opinion based on the summary of opinions of first round. The result shows that the agreed percentage was then 81%, disagree percentage is 4% and neutral percentage is 15%. The results of both the rounds are shown in Table 3. Note that for simplicity, we have merged "totally agree" and "agree" to "Agree", and "totally disagree" and "disagree" values to "Disagree".

4.1.3 Framework for the selection of an IoT platform

To show how our general framework can be applied to assessing and choosing an IoT platforms, in this study we have selected the top five IoT platforms based on market share. We have compared these IoT platforms according to the twenty-one key IoT platform factors that we have identified from the literature and verified using Delphi study. We have compared these twenty-one key factors with the features provided by those selected five IoT platforms as shown in Table 4.

More specifically, the entries of Table 4 have the following meaning related to the specific feature to be considered: 'yes' means the feature is available, 'high' indicates strong, 'bad' shows weak, 'good' indicates that the feature is very good, '-' shows that the feature is unknown and 'no' indicates that the feature is not available in the platform. In order to identify and fill the features of the selected five IoT platforms, different articles [38, 51–56] have been studied from many databases. Some websites [41–45, 49] have been used, especially the websites of those selected IoT platforms. A few white papers [57] have also been studied.

The framework for selection of an IoT platform is illustrated in Fig. 8 as a schematic of the selection procedure. The whole process consists of five stages. In the first stage the company finalize their business requirements. In the second stage the company requirements are applied to prioritising which factors are required (R), important (I) and not required (-) for this business context. In the third stage the R and I factors are compared with the features

Table 3: Results of Delphi study both rounds. The mean and median are taken from the agreed values.

		Survey round 1					Survey round 2				
F	**Factor**	Mean	Median	Disagree %	Neutral %	Agree %	Mean	Median	Disagree %	Neutral %	Agree %
F1	Scalability	4	4	14%	7%	79%	4	4	0%	7%	93%
F2	Flexibility	4	4.5	0%	7%	93%	4	4.5	0%	7%	93%
F3	Data anlytics	4	5	7%	0%	93%	5	5	0%	7%	93%
F4	Disaster recovery	4	4	0%	7%	93%	4	4	0%	7%	93%
F5	Stability	5	5	0%	7%	93%	5	5	0%	7%	93%
F6	Security	5	5	0%	7%	93%	5	5	0%	7%	93%
F7	Data ownership	5	5	0%	7%	93%	5	5	0%	7%	93%
F8	Protocol support	5	5	7%	0%	93%	5	5	0%	7%	93%
F9	System performance	5	5	0%	7%	93%	5	5	0%	7%	93%
F10	Time to market	4	4	7%	7%	86%	4	4	7%	7%	86%
F11	Legacy architecture	4	4	14%	7%	79%	4	4	0%	14%	86%
F12	Attractive interface	5	5	0%	14%	86%	5	5	0%	14%	86%
F13	Pricing model	4	4.5	0%	21%	79%	4	4.5	0%	21%	79%
F14	Cloud ownership	4	4	7%	14%	79%	4	4	0%	21%	79%
F15	Interoperability	4	4	14%	7%	79%	4	4	7%	14%	79%
F16	App. environment	4	4	0%	29%	71%	4	4	0%	29%	71%
F17	Hybrid cloud	4	4	7%	29%	64%	4	4	0%	36%	64%
F18	Platform migration	4	4	7%	29%	64%	4	4	7%	29%	64%
F19	Previous experience	4	4	7%	29%	64%	4	4	7%	29%	64%
F20	Edge intelligence	4	4	14%	29%	57%	4	4	14%	29%	57%
F21	Bandwidth	4	4	21%	21%	57%	4	4	14%	29%	57%
–	**Percentage**	–	–	**6%**	**14%**	**80%**	–	–	**4%**	**15%**	**81%**

provided by the five selected IoT platforms. The IoT platform/s that provide a maximum of the features as compared to the requirements are selected and shifted to the stage four. In stage four there might be one or many IoT platforms that match the required and important factors. Stage five is the decision, which is explained next.

If there is one IoT platform that provides the most required and important features then the same IoT platform can be selected for the business application. But, if there are multiple IoT platforms providing these features then the company may choose an IoT platform based on the comparison of their match to I factors like pricing, time to market etc. and select a suitable IoT platform for their business needs. There might also be chances that none of the platforms provide all of the required features; this might indicate that new platforms should be selected and evaluated accordingly.

After evaluating the possible services available, we should investigate how our datasets behave as we can use them to emulate the characteristics of a real data traffic through an IoT network.

4.2 Extended Tennessee Dataset

Due to the large amount of data that is generated and the variety of outages that might occur in an industrial environment, the data processing techniques have to be designed in a way to support the wide spectrum of possible scenarios. Once the techniques are chosen, the

Table 4: Reflecting the twenty-one key IoT platform features in the five main IoT platforms

Factors	AWS	Azure	Google cloud	IBM Watson	Oracle IoT
Scalability	yes	yes	yes	yes	yes
Flexibility	yes	-	yes	-	yes
Data analytic	yes	yes	yes	yes	yes
Disaster recovery	yes	yes	no	no	no
Stability	yes	yes	yes	-	-
Security	high	high	high	high	high
Data ownership	-	yes	-	-	-
Protocol support	yes	yes	-	yes	yes
System performance	yes	-	yes	yes	-
Time to market	yes	yes	-	-	yes
legacy architecture	yes	-	-	-	yes
Attractive interface	yes	yes	-	no	-
Pricing model	bad	bad	good	-	-
Cloud ownership	yes	yes	yes	-	yes
Interoperability	yes	-	-	-	yes
App. environment	yes	yes	yes	yes	yes
Hybrid cloud	yes	yes	-	-	-
Platform migration	yes	yes	-	-	-
Previous experience	yes	yes	-	-	-
Edge intelligence	yes	yes	yes	-	yes
Bandwidth	-	-	good	-	-

challenges lie in gathering such a dataset to train the models and test the performance. A good example of such a dataset that can be used to develop and fine-tune our algorithms is the Tennessee Eastman (TE) process dataset. This is a labeled dataset that was collected from simulations of the TE process. Out of the total 53 measured variables, 22 variables are continuous process measurements, 19 variables are composition measurements and the remaining 12 are manipulated variables. One of the variables is constant throughout the simulation, and therefore this variable is excluded from the analysis.

The process contains 21 process faults related to different variations of the related variables. The dataset contains two sets of generated data - training and testing datasets. Both sets contain normal process data as well as fault data. The original training TE process dataset was generated by performing 22 runs (one normal and 21 fault runs). Each run simulates 25 hours of operation with a sampling interval of 3 min. The faults are introduced after one hour of operating time for the training dataset. The total number of observations in each training run is 500. The testing dataset consists of 22 runs and simulates 48 operating hours. In the testing dataset the faults are introduced after 8 hours of operation.

Considering the small size of the original TE process dataset, training of different machine learning models was very difficult. Additionally, a small dataset does not reassemble a realistic industrial use-case. Therefore, an additional dataset based on the Tennessee Eastman process

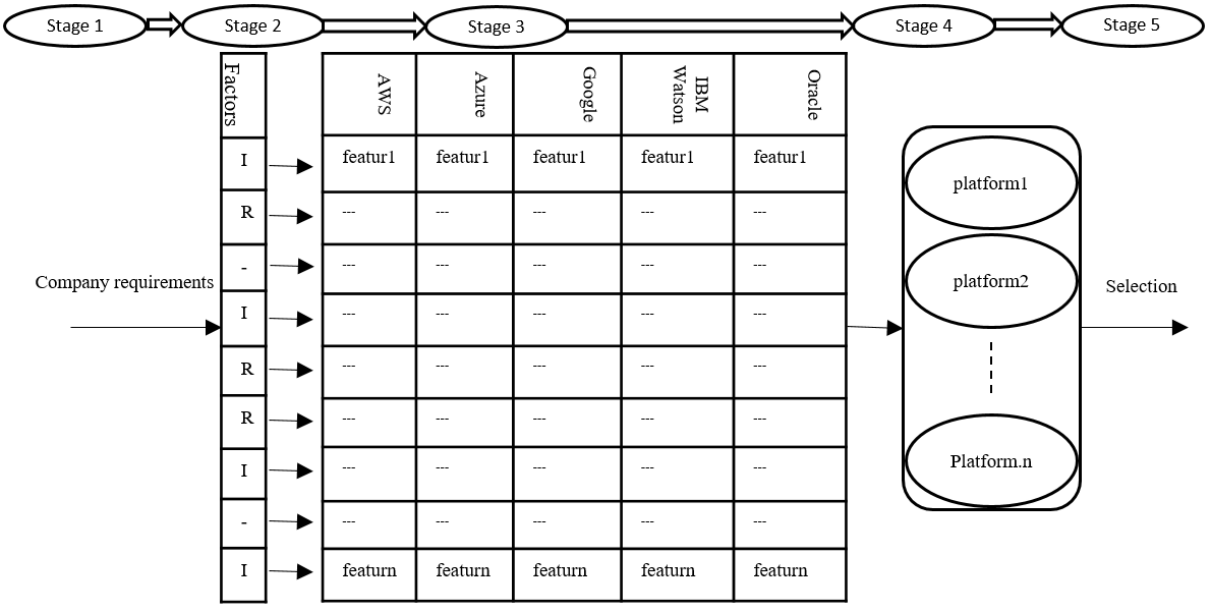


Figure 8: Comparing Key factors with the features offered by the IoT platform.

was generated. The additional dataset follows the same rules as the original one (sampling interval 3 min, separation of the dataset into training and testing, faults introduced after 1 hour for the training dataset and after 8 hours for the testing dataset). However, it was generated by simulating 500 independent realizations of each scenario (each fault), extending the dataset to a size that is useful when training machine learning models and large enough to reassemble a realistic industrial use-case.

4.3 SEAT Dataset Cases

SEAT provides two use cases focused on predictive maintenance from its main factories SEAT Martorell, SEAT Barcelona, and SEAT Components.

The first one consists in early detection of failures in mechanical components (bearing) of the drive chain in Paint shop, causing axial displacement that results in significant production stops and corresponding loss of bodies. The main goals in this use case are to reduce the mean time to repair, increase the mean time between failures and achieve zero production loss. Thus, predictive maintenance models are needed to detect mechanical components degradation by using significant data to create an early warning system that identifies problems before an axial displacement occurs. Available datasets integrate miniaturized data (1440 samples of data per day and variable), stored minute by minute about engine group binary signals, tensor, security, position sensors (axial monitoring), etc. The datasets include identified failures for previous years.

The second one concentrates on prediction of failures in spindles of CNC (Computer Numerical Control) machines of the machining centre, generating defects in components of the gearboxes produced in the SEAT Components plant. The spindle is an element (kind of actuator) of a CNC machine that ensures lineal movement over a surface. The spindle misbehaviour

is detected after quality losses or failure. The main objectives in this use case are to reduce quality losses, improve the operational equipment efficiency, and avoid unplanned stops and production of NOK parts. Therefore, predictive maintenance models are required to detect the spindle failure and avoid it. Available datasets are composed of over 50 variables, such as axis positions, engine energy consumption, pressure, temperature, fluids levels, preventive control points, failures, quality reports, etc. The datasets provide no identified failures for previous years.

5 Conclusions

This document presented some initial insights of FIREMAN outputs from Task 4.1. Two main areas of research have risen within the document. The first focusing on the interface between WP3 and WP4, investigating different techniques to model the aggregation of heterogeneous data, as presented on Sections 2 and 3. The later is more interested on the data itself, where partners are looking for new datasets and how to proper store this data. This part will be of great importance for the continuation of WP4 in Task 4.2, where this data will be processed and compressed. Finally, this document serve as a initial discussion on the development of Task 4.1, which still active at the publication date of this report. and will have its finals results published on Deliverable 4.3.

References

[1] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, "Toward Massive Machine Type Cellular Communications," *IEEE Wirel. Commun.*, vol. 24, no. 1, pp. 120–128, February 2017.

[2] W. Hang, D. Michael, and D. William, "Defining the communication architecture for data aggregation in wireless sensor networks: application to communicating concrete design," in *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE, 2019, pp. 102–108.

[3] I. Ullah and H. Y. Youn, "Efficient data aggregation with node clustering and extreme learning machine for wsn," *The Journal of Supercomputing*, pp. 1–27, 2020.

[4] W.-S. Jung, K.-W. Lim, Y.-B. Ko, and S.-J. Park, "Efficient clustering-based data aggregation techniques for wireless sensor networks," *Wireless Networks*, vol. 17, no. 5, pp. 1387–1400, 2011.

[5] S. Randhawa and S. Jain, "Cross-layer energy based clustering technique for heterogeneous wireless sensor networks," *WIRELESS PERSONAL COMMUNICATIONS*, 2020.

[6] A.-J. Yuste-Delgado, J.-C. Cuevas-Martinez, and A. Triviño-Cabrera, "A distributed clustering algorithm guided by the base station to extend the lifetime of wireless sensor networks," *Sensors*, vol. 20, no. 8, p. 2312, 2020.

[7] J. Guo, S. Durrani, X. Zhou, and H. Yanikomeroglu, "Massive Machine Type Communication with data aggregation and resource scheduling," *IEEE Trans. Commun.*, vol. 65, no. 9, pp. 4012–4026, 2017.

[8] X. Yu, C. Li, J. Zhang, and K. B. Letaief, *Stochastic Geometry Analysis of Multi-Antenna Wireless Networks*. Springer, 2019.

[9] M. Haenggi, *Stochastic Geometry for Wireless Networks*. Cambridge University Press, 2012.

[10] O. L. A. López, H. Alves, P. H. J. Nardelli, and M. Latva-aho, "Aggregation and resource scheduling in machine-type communication networks: A stochastic geometry approach," *IEEE Trans. Wireless Commun.*, vol. 17, no. 7, pp. 4750–4765, 2018.

[11] O. L. A. López, H. Alves, P. H. Nardelli, and M. Latva-aho, "Hybrid resource scheduling for aggregation in massive machine-type communication networks," *Ad Hoc Netw.*, vol. 94, p. 101932, 2019.

[12] C. Saha, M. Afshang, and H. S. Dhillon, "Poisson cluster process: Bridging the gap between PPP and 3GPP HetNet models," in *ITA*. IEEE, 2017, pp. 1–9.

[13] M. Haenggi, "The Meta Distribution of the SIR in Poisson Bipolar and Cellular Networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2577–2589, 2015.

[14] X. Tang, X. Xu, and M. Haenggi, "Meta distribution of the SIR in moving networks," *IEEE Trans. Commun.*, 2020.

[15] S. Guruacharya and E. Hossain, "Approximation of meta distribution and its moments for Poisson cellular networks," *IEEE Wireless Commun. Lett.*, vol. 7, no. 6, pp. 1074–1077, Dec 2018.

[16] M. Haenggi, "Efficient calculation of Meta Distributions and the performance of user percentiles," *IEEE Wireless Commun. Lett.*, vol. 7, no. 6, pp. 982–985, Dec 2018.

[17] N. M. Rodriguez, O. L. A. Lopez, H. Alves, and M. Latva-aho, "On the SIR meta distribution in massive MTC networks with scheduling and data aggregation," *submitted to VTC2021*, 2020.

[18] B. Chandrasekaran, "Survey of network traffic models," 2006.

[19] A. Adas, "Traffic models in broadband networks," *IEEE Communications Magazine*, vol. 35, no. 7, pp. 82–89, 1997.

[20] N. Nikaein, M. Laner, K. Zhou, P. Svoboda, D. Drajic, M. Popovic, and S. Krco, "Simple traffic modeling framework for machine type communication," in *ISWCS 2013; The Tenth International Symposium on Wireless Communication Systems*, 2013, pp. 1–5.

[21] M. Centenaro and L. Vangelista, "A study on m2m traffic and its impact on cellular networks," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015, pp. 154–159.

[22] M. Laner, P. Svoboda, N. Nikaein, and M. Rupp, "Traffic models for machine type communications," in *ISWCS 2013; The Tenth International Symposium on Wireless Communication Systems*, 2013, pp. 1–5.

[23] H. Heffes and D. Lucantoni, "A markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance," *IEEE Journal on Selected Areas in Communications*, vol. 4, no. 6, pp. 856–868, 1986.

[24] S. L. Miller and D. Childers, "Chapter 9 - markov processes," in *Probability and Random Processes (Second Edition)*, second edition ed., S. L. Miller and D. Childers, Eds. Boston: Academic Press, 2012, pp. 383 – 428. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780123869814500126

[25] S. R. Pokhrel, S. Verma, S. Garg, A. K. Sharma, and J. Choi, "An Efficient Clustering Framework for Massive Sensor Networking in Industrial IoT," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020.

[26] J. Diaz-Rozo, C. Bielza, and P. Larrañaga, "Clustering of Data Streams With Dynamic Gaussian Mixture Models: An IoT Application in Industrial Processes," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3533–3547, 2018.

[27] C. Kalalas and J. Alonso-Zarate, "Sensor data reconstruction in industrial environments with cellular connectivity," in *2020 IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE PIMRC '20)*, August 2020.

[28] C. Kalalas and J. Alonso-Zarate, "Massive connectivity in 5g and beyond: Technical enablers for the energy and automotive verticals," in *2020 2nd 6G Wireless Summit (6G SUMMIT)*, 2020, pp. 1–5.

[29] B. H. Bloom, "Space/Time Trade-offs in Hash Coding with Allowable Errors," *Communications of the ACM*, vol. 13, pp. 422–426, 1970.

[30] C. Kalalas and J. Alonso-Zarate, "Efficient Cell Planning for Reliable Support of Event-Driven Machine-Type Traffic in LTE," in *IEEE Global Communications Conference (GLOBECOM)*, December 2017, pp. 1–7.

[31] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[32] X. Jia, Q. Feng, T. Fan, and Q. Lei, "RFID technology and its applications in Internet of Things (IoT)," *2012 2nd International Conference on Consumer Electronics, Communications and Networks, CECNet 2012 - Proceedings*, pp. 1282–1285, 2012.

[33] J. Mineraud, O. Mazhelis, X. Su, and S. Tarkoma, "A gap analysis of Internet-of-Things platforms," *Computer Communications*, vol. 89-90, pp. 5–16, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2016.03.015

[34] J. Guth, U. Breitenbucher, M. Falkenthal, F. Leymann, and L. Reinfurt, "Comparison of IoT platform architectures: A field study based on a reference architecture," *2016 Cloudification of the Internet of Things, CIoT 2016*, 2017.

[35] N. Koshizuka and K. Sakamura, "Ubiquitous ID: Standards for ubiquitous computing and the internet of things," *IEEE Pervasive Computing*, vol. 9, no. 4, pp. 98–101, 2010.

[36] Y. Song, "Security in Internet of Things," *Trita-Ict-Ex Nv - 2013:196*, vol. Independen, no. 41, p. 28, 2013. [Online]. Available: http://kth.diva-portal.org/smash/get/diva2:702223/FULLTEXT01.pdf{%}0Ahttp://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-142451

[37] Sukanya Mandal, "Internet of Things (IoT) - Part 2 (Building Blocks & Architecture)," 2016. [Online]. Available: https://www.c-sharpcorner.com/UploadFile/f88748/internet-of-things-part-2/

[38] H. Hejazi, H. Rajab, T. Cinkler, and L. Lengyel, "Survey of platforms for massive IoT," *2018 IEEE International Conference on Future IoT Technologies, Future IoT 2018*, vol. 2018-Janua, pp. 1–8, 2018.

[39] M. Gigli and S. Koo, "Internet of Things: Services and Applications Categorization," *Advances in Internet of Things*, vol. 01, no. 02, pp. 27–31, 2011. [Online]. Available: http://www.scirp.org/journal/doi.aspx?DOI=10.4236/ait.2011.12004

[40] F. H. Mohammed and R. Esmail, "Survey on IoT Services: Classifications and Applications," *International Journal of Science and Research*, vol. 4, no. 1, pp. 2124–2127, 2015.

[41] Amazon Web Services, "IoT Applications & Solutions — What is the Internet of Things (IoT)? — AWS," 2019. [Online]. Available: https://aws.amazon.com/iot/

[42] Microsoft Azure IoT, "Microsoft Azure," 2019. [Online]. Available: https://azure.microsoft.com/en-in/pricing/details/iot-hub/

[43] Internet of Things, "Internet of Things (IoT) — Oracle," 2019. [Online]. Available: https://www.oracle.com/internet-of-things/

[44] Google Cloud, "Google Cloud IoT - Fully managed IoT services — Google Cloud," 2019. [Online]. Available: https://cloud.google.com/solutions/iot/

[45] IBM Watson IoT Platform, "IBM Watson IoT Platform - Overview - Finland," 2019. [Online]. Available: https://www.ibm.com/fi-en/marketplace/internet-of-things-cloud

[46] S. Narula, A. Jain, and Prachi, "Cloud Computing Security: Amazon Web Service," *2015 Fifth International Conference on Advanced Computing & Communication Technologies*, pp. 501–505, 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7079135/

[47] V. Gazis, M. Gortz, M. Huber, A. Leonardi, K. Mathioudakis, A. Wiesmaier, F. Zeiger, and E. Vasilomanolakis, "A survey of technologies for the internet of things," *IWCMC 2015 - 11th International Wireless Communications and Mobile Computing Conference*, pp. 1090–1095, 2015.

[48] J. Lee, "How to Choose the Right IoT Platform: The Ultimate Checklist."

[49] IOTIFY, "Top 10 criteria to choose the best IoT cloud platform." [Online]. Available: https://iotify.io/top-10-selection-criteria-for-your-iot-cloud-platform/

[50] Jeffrey Lee, "The 6 Complexities of Building a Managed IoT Platform," 2018. [Online]. Available: https://hackernoon.com/the-6-complexities-of-hosting-a-managed-iot-service-b7696eea52ba

[51] P. P. Ray, "A survey of IoT cloud platforms," *Future Computing and Informatics Journal*, vol. 1, no. 1-2, pp. 35–46, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.fcij.2017.02.001

[52] P. Agarwal and M. Alam, "Investigating IoT Middleware Platforms for Smart Application Development," pp. 1–14, 2018. [Online]. Available: http://arxiv.org/abs/1810.12292

[53] M. Ammar and B. Crispo, "slimIoT : Scalable Lightweight Attestation Protocol For the Internet of Things."

[54] S. Challita, F. Zalila, C. Gourdin, and P. Merle, "A precise model for Google cloud platform," *Proceedings - 2018 IEEE International Conference on Cloud Engineering, IC2E 2018*, pp. 177–183, 2018.

[55] M. Ullah and K. Smolander, "Highlighting the Key Factors of an IoT Platform," *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 901–906, 2019.

[56] B. Nakhuva and T. Champaneria, "Study of Various Internet of Things Platforms," *International Journal of Computer Science & Engineering Survey*, vol. 6, no. 6, pp. 61–74, 2015.

[57] G. Cloud, I. Security, D. Overview, G. Cloud, G. Cloud, A. Layer, and T. Security, "Google Cloud Security Whitepapers," no. March, pp. 1–97, 2018.