Dethroning GPS: Low-Power Accurate 5G Positioning Systems using Machine Learning

Joao Gante, *Student Member, IEEE,* Leonel Sousa, *Senior Member, IEEE,* and Gabriel Falcao, *Senior Member, IEEE*

***Abstract*—Over the last years positioning systems have become increasingly pervasive, covering most of the planet's surface. Although they are accurate enough for a large number of uses, their precision, power consumption and hardware requirements establish the limits for their adoption in mobile devices.**

In this paper, the energy consumption of a proposed deep learning-based millimeter wave positioning method is assessed, being subsequently compared to the state-of-the-art on accurate outdoor positioning systems. Requiring as low as 0.4 mJ per position fix, when compared to the most recent assisted-GPS implementations the proposed method has energy efficiency gains of $47\times$ and $85\times$ for continuous and sporadic position fixes (respectively), while also having slightly lower estimation errors. Therefore, the proposed method significantly reduces the energy required for precise positioning in the presence of millimeter wave networks, enabling the design of more efficient and accurate positioning-enabled mobile devices.

***Index Terms*—5G, Beamforming, Deep Learning, Low-Power, mmWaves, Outdoor Positioning**

I. Introduction

NOWADAYS, an increasing number of tasks rely on Global Navigation Satellite Systems (GNSSs)' precise localization capabilities for their operating success. Since positioning capabilities are required by mobile devices, the energy efficiency of the used method have a major influence on their operation, being one of the most significant causes of battery drain on smartphones [1]. For small and light devices, such as location trackers or Internet of Things (IoT) devices, it can even dictate their design and capabilities, due to severe battery limitations.

5G brought back to attention Millimeter Wave (mmWave) communication systems [2], resulting in new proposals for positioning systems [3]. The accuracy attainable in controlled conditions is very high, reaching sub-meter precision in indoor [4] and ultra-dense Line-of-Sight (LOS) outdoor scenarios [5]. Nevertheless, in order to be broadly applicable to outdoor localization, a mmWave positioning system must be able to accurately locate devices in non-Line-of-Sight (NLOS) locations, while employing a limited number of Base Stations (BSs). These requirements, combined with multiple, often overlapping non-linear propagation phenomena such as reflection and scattering, pose serious challenges to the traditional geometry-based positioning methods. In fact, the recent mmWave experimental work in [6] shows that geometry-based methods cannot be applied to locate NLOS targets, and thus different approaches are required. As it was demonstrated in our previous work [7], the properties of mmWave transmissions can be leveraged to create an information-rich fingerprint, coined as Beamformed Fingerprint (BFF). With the availability of the BFF, Deep Learning (DL) methods [7] and hierarchy techniques [8] are then suggested to infer accurate position estimates, obtaining state-of-the-art results for single-point estimates (3.3 m), in a scenario containing mostly NLOS positions and employing a single BS.

The goal of any positioning system is to estimate the position of a target, which is a direct consequence of its movement. The movement of a user, in its turn, is limited by physical restrictions, such as velocity and acceleration, as well as human-made constraints, such as traffic rules. As a consequence, it is possible to leverage additional sources of information if sequences are considered, as opposed to single-point estimates. When sequences of BFFs are available, the use of sequence-based DL architectures was proposed to effectively enable the system to track a mobile device [9]. The result was a state-of-the-art average estimation error as low as 1.78m, even in the presence of heterogeneous movement types and NLOS locations, using BFFs from a single BS.

The advent of mmWaves did brought a new set of techniques that can potentially displace GNSS-based predictions, especially in urban environments. However, to have a chance of doing so, namely in mobile and autonomous devices, a critical question remains unanswered: are they energy-efficient? Therefore, the key contribution of this paper is the demonstration that mmWave positioning methods can not only provide accurate estimates, but also do so with high energy efficiency.

Since, from the proposed method, each beamformed transmission only lasts for a couple of microseconds [7], the complete BFF will only require a few milliseconds to obtain. As result of such short listening time, the majority of the device-side energy costs for each position fix will be result of either the uplink transmission of the received data, for further processing at the BS, or of the position estimate computation at the device. For the former, the received data patterns are sparse, which means that the data can be heavily compressed before transmission to the BS. As for the later, the advent of highly efficient embedded systems with small form factor tailored for DL systems [10][11] also enable a highly efficient solution. Regardless of where the BFF position estimate is computed, the results section in this paper will show that it is more energy efficient that the existing positioning approaches.

Therefore, this paper makes the following **contributions**:

- It closes the loop on a previously proposed BFF positioning method for mmWaves, by assessing its energy efficiency over multiple DL architectures. It has been shown that this method achieves unmatched accuracy levels in the presence of NLOS (one order of magnitude better than the previous state-of-the-art);
- To the best of our knowledge, this is the first paper that addresses comparative energy consumption studies for positioning and tracking systems working with mmWaves. A significant number of conclusions are transferable to other fingerprint positioning methods, since they can share the same DL architectures;
- We have developed and tested a system that shows superior results, as compared to GNSS-based systems, simultaneously in terms of accuracy, energy efficiency, and, if the device also uses mmWaves for communications, hardware requirements.

The remaining of this paper is organized as follows. In Section II, the non-mmWave positioning methods are revised, as well as their energy consumption, while Section III provides an overview of the state-of-the-art for mmWave positioning. Section IV fully describes and discusses the herein proposed positioning method, including the theoretical background required to estimate its energy consumption, and Section V summarizes the possible DL architectures that can be used to solve the problem. Section VI lays out the full simulation apparatus, while Section VII exposes our experimental results, focusing on the energy efficiency. Finally, in Section VIII, the conclusions will be drawn.

II. Positioning Systems

Nowadays, an increasing number of tasks rely on GNSSs' precise localization capabilities for their operating success. The Global Positioning System (GPS), a GNSS, was launched in the 1970s, based on a fleet of satellites broadcasting data frames at the very low rate of 50 bps to receivers on the Earth's surface. The signal is encoded using a pseudo-random sequence, unique to each satellite, transmitted at 1.023 million pulses (*chips*) per second, and each frame consists of 5 sub-frames of 300 bits each [12], as depicted below:

- Sub-frame 1: accurate timing information generated by the atomic clock embedded into the satellite itself;
- Sub-frames 2 and 3: its precise orbital information used to compute its location, which remains valid for up to 4 hours (the *ephemeris*);
- Sub-frames 4 and 5: ionospheric conditions and the operating status of the whole system, typically updated every 24 hours (the *almanac*).

Depending on the existing information on the mobile device, the Time to First Fix (TTFF) of a stand-alone GNSS localization system can vary significantly. When the device lacks a valid almanac, then it must receive the full signal, consisting of 25 frames (12.5 minutes), also known as *cold start*. If the receiver was recently active, then it can perform a *warm start* by obtaining the ephemeris data, which takes up to 30 seconds. In optimal conditions, the receiver can acquire the GPS signal right away, returning a position estimate within a couple of seconds. This is known as *hot start*.

When the GPS was conceived, the system was designed for long periods of continuous navigation with relatively short TTFF. However, with the advent of heterogeneous mobile services, a low TTFF became vital for the user experience.

Since a mobile network BS can view the same satellites as a nearby mobile device, it has access to the desired satellites' time and orbital information, as well as to the device's coarse location. Thus, with Assisted Global Positioning System (A-GPS), the BS is capable of providing assistance to the device, minimizing its TTFF [12]. Depending on whether the position estimate is computed on the device or offloaded to the BS, A-GPS technologies can be classified as:

- Mobile Station Based (MSB) - the mobile device receives the ephemeris, almanac, time, and coarse location from the BS, enabling a hot start regardless of the starting conditions;
- Mobile Station Assisted (MSA) - the mobile device acquires raw satellites' signals and sends them to the BS, which computes and then returns the device location.

With MSB, the cold start TTFF is often below 10 seconds, with typical latency values akin to a standard hot start [13]. Its main energy consumption is driven by the GNSS signal processing, although acquiring the assistance data from the BS has significant costs [13]. State-of-the-art low-power implementations claim requiring about 18 mJ per position fix when continuously tracking [14][15], with significant penalties when sporadic tracking is desired (e.g. [14] requires 504 mJ per fix when tracking the device once per minute).

MSA was proposed to avoid the sporadic tracking penalty, where the mobile device has to perform the costly signal synchronization with each visible satellite before an estimate [16]. With MSA, the mobile device just needs to capture and send a snapshot of the received GNSS signal. The duration of the captured signal must be a multiple of approximately 1 ms (*i.e.* the period of the pseudo-random sequence signal sent by the satellites), to ensure a coherent integration time. Considering the minimum sampling frequency of 2.046 MHz, each position fix requires transmitting at least 2046 bits. However, to obtain the desired GNSS positioning accuracy, most practical implementations capture the signal at more than 16 MHz, with durations exceeding 10 ms [17], requiring a transmission of hundreds of kilobits per position fix. This leads to significant energy costs, and thus MSB A-GPS approaches are often preferred over MSA.

Although A-GPS can solve the majority of problems associated to the start up latency, it still requires dedicated GNSS hardware and has a high peak power consumption when downloading the assistance data from the BS [13]. To provide an alternative, multiple network-operated localization systems were considered in the past decade [3]. With Release 9 of Long Term Evolution (LTE) networks, the Observed-Time-Difference-of-Arrival (OTDoA) was introduced, possessing a theoretical achievable error similar to GNSS devices [18]. However, to achieve that error level, OTDoA has to operate under optimal conditions and has to employ expensive detection mechanisms, unfit for low-power devices, as discussed in

[19] (and further addressed in LTE Release 14). In practical scenarios, the average error fairly exceeds 20 m [20], and thus cannot be considered a high-accuracy outdoor positioning system. More recently, the works in [21] and [22] proposed enhancements to OTDoA through additional opportunistic measurements and Compressive Sensing (CS), respectively, obtaining near GNSS accuracy at the cost of expensive signal processing on the mobile device.

In addition to methods described above, which are deployed, other outdoor positioning methods for conventional frequencies ($<$ 6 GHz) have been proposed, such as the works in [23], [24], or [25]. In [23], an unsupervised DL-based system is designed to perform a lower-dimensionality mapping for the Channel State Information (CSI), which can be expanded into a semi-supervised siamese network to learn the conversion of that mapping into a user's position. The work in [24] also leverages the CSI of the received signal that, combined with the Received Signal Strength Indicator (RSSI), can be used to train a Deep Neural Network (DNN) for positioning. These two methods achieve good average accuracies (below 5 m and 6.45 m, respectively), but still cannot displace the A-GPS. Finally, the work in [25] explores intelligent surface-based positioning, where the antennas are embedded within building walls, with promising performance bounds (sub-meter precision).

III. Millimeter Wave Positioning Systems

The works developed in [26], [27], [28], [29], [30] aim to locate devices in both LOS and NLOS outdoor locations. The work in [26] uses multiple access points to build a fingerprint database of received powers and Angle-of-Arrival (AoA), while in [27] CS is applied on information gathered from static listeners. In [28], multiple Beamforming (BF) transmissions are used together with an iterative algorithm to estimate the position and orientation of the device. The same parameters are achieved in [29], through the estimation of the AoA, Time-of-Arrival (ToA), and Angle-of-Departure (AoD), making concurrent use of LOS and NLOS transmissions. However, the methods described so far do not comply with typical outdoor positioning requirements: [27] and [26] assume that each mobile device is always in range of multiple static transceivers, while the other two methods struggle with NLOS positions, requiring transmission paths from at least three scattering points [28] or preferring to not expose the results for those locations [29].

The work in [30] overcomes the restrictions discussed above by creating a fingerprint database of uplink pilots transmitted to a single massive Multiple-Input Multiple-Output (MIMO) BS that contains multiple antennas distributed over a confined area. Using a Gaussian process regression to predict the position, this work achieves a Root-Mean-Square-Error (RMSE) of 34 m. For sake of comparison, let us consider the network-enabled OTDoA and the ubiquitous GNSS. The former has a theoretical average error of approximately 10 m [18], assuming optimal conditions and complex detection systems. On the other hand, state-of-the-art GNSS receivers are capable of obtaining superior accuracy, averaging 3 m in continuous measurement scenarios [15], with significant penalties for sporadic measurements due to the extensive use of Kalman filters [31]. Thus, in the presence of NLOS positions, there is a significant precision gap between state-of-the-art mmWave systems and the existing outdoor positioning solutions.

In a typical deployment, BSs are placed in elevated positions and, in urban scenarios, the majority of the surrounding obstacles found with mmWave transmissions to ground users are buildings. As a consequence, we can expect most of obstacles to be static. Therefore, consecutive measurements of the received Power Delay Profile (PDP) at a given location are expected to remain comparable until a significant change in the surrounding space occurs. If a BS transmits a sequence of directive beamformed signals, so as to cover all transmission angles (maximizing the covered space), then the receiver is able to measure numerous distinct PDPs, one for each beamformed signal. Due to the non-linear propagation phenomena found in mmWaves, that set of PDPs is expected to have noticeable discontinuities throughout the target localization space, which provide important spatial information. In [7], the use of the set of PDPs to produce the aforementioned BFF was proposed as a foundation for an accurate mmWave outdoor positioning method. The BFF positioning method has an additional attractive aspect: contrarily to most accurate positioning methods (including the method suggested in [30], GNSS, or OTDoA), it only requires a single-anchor [3][32]. In other words, the system should be able to provide accurate estimates whenever there is mmWave coverage.

The information held in a BFF is a result of non-linear interactions and, consequently, it requires a method that is able digest non-linear relationships in order to extract any meaningful conclusion. Given the requirements of the problem and the recent state-of-the-art results in datasets containing non-linear relationships, DL techniques become a solid candidate to untangle the BFF. In [7], use of Convolutional Neural Networks (CNNs) [33] was proposed to exploit the data structure within a BFF. The previous system was improved in [8] with a hierarchical structure, taking advantage of the BFFs' expected similarity along adjacent positions, at the cost of additional processing power. Finally, in [9], sequence-based DL techniques were added to the BFFs position estimates, enabling the system to track a mobile device. However, there is no implementation of such a system, which allows an experimental evaluation of its capacity, namely in what concerns power and energy consumption.

IV. Beamformed Fingerprint Positioning

The transmitted mmWave radiation, suffering from phenomena such as reflection and scattering, is shaped by the encountered obstacles. As result, a transmitted signal can have more than one propagation path between the BS and the receiver, each with a unique power attenuation and delay. From an information theory point of view, each new path carries additional spatial information, and thus enhances the predictive power of the system. Based on this principle, the BFF can transport enough information to locate a listening mobile device.

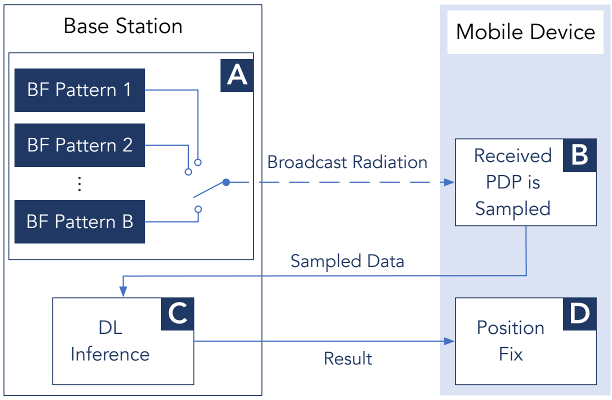

Fig. 1. Diagram with a summary of the beamformed fingerprint positioning system [7]. The mobile device samples the received PDPs from beamformed transmissions, resulting in a beamformed fingerprint that can then be translated into its position.

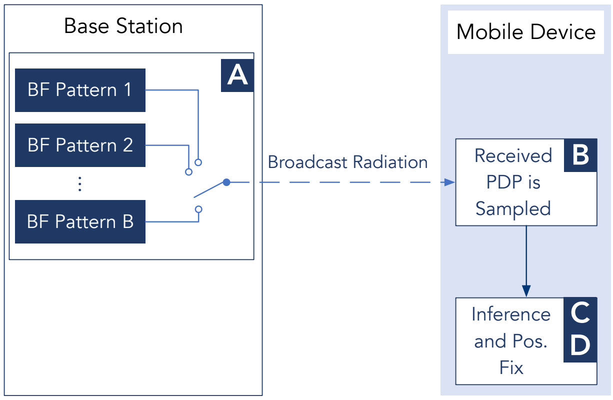

Fig. 2. Alternative mode of operation of the system depicted in Fig. 1, where the DL inference is performed in the mobile device. Although it does not require the upload of the sampled data, this mode of operation has additional storage and processing requirements.

A. System Model

A critical component of any learnable dataset is its consistency, as it then allows the system to extract helpful information from a trained mathematical model. In other words, the distribution of the data that is used at test time must be comparable to the distribution of the data seen at train time. To ensure so, the input data generation must be constrained to a constant set of rules, especially in terms of the transmission and receiving procedures of the signal, in order to build an appropriate fingerprinting system. To comply with such requirements, the system depicted in Fig. 1 was originally suggested in [7]. It operates in four distinct phases, as labeled in the diagram, whose role is further described below. In phase A, a BS broadcasts radiation using a fixed BF codebook, while phase B focuses on measuring the resulting PDP at the target device. After all the required measurements are performed and transmitted back to the BS, phase C infers the device's position, which is communicated back to it in phase D.

One of the critical aspects that define the resolution of the information held in the BFF is the directivity of the beamformed BS transmissions, which are defined in phase A. The directivity determines how narrow the beam of transmitted radiation is and, as such, increasing the directivity of a given transmission results in a PDP containing information with higher specificity. In other words, that transmitted beam is now more focused in a particular sub-set of possible propagation paths. Additionally, given that the radiation is more focused, there is a higher fraction of transmission paths whose energy can be found above the receiving antennas' detection threshold. Unfortunately, there is no free lunch: to cover all possible angles of transmission, higher BF directivities correspond to a higher number of BS transmissions, which results in a higher number of PDP measurements required per position fix. Throughout this paper, the exact mechanism to capture a PDP is abstracted, knowing that it can be done through various real implementations, as mentioned in [9].

Let us consider a fixed codebook $\mathbf{C}_{Tx}$ containing B_{Tx} BF patterns. To generate enough data for a position fix, the BS must transmit a sequence of signals employing the B_{Tx} BF patterns. Assuming a BS with N_S antennas, the frequency-domain received signal for the i-th transmitter BF at a mobile device with N_R antennas, $r \in \mathbb{C}$, can be written as

$$r = \mathbf{w}^T \mathbf{H} \mathbf{f}_i s + \mathbf{w}^T \mathbf{z}, \qquad (1)$$

where the superscript T denotes a matrix transpose, $\mathbf{w} \in \mathbb{C}^{N_R \times 1}$ corresponds to the BF at the receiver, $\mathbf{H} \in \mathbb{C}^{N_R \times N_S}$ is the channel matrix, $\mathbf{f}_i \in \mathbb{C}^{N_S \times 1}$ denotes the currently selected transmitter BF, $s \in \mathbb{C}$ is the signal to be detected, and $\mathbf{z} \in \mathbb{C}^{N_R \times 1}$ represents noise. Since the transmitter BF is codebook-based, it is important to state that $\mathbf{f}_i \in \mathbf{C}_{Tx}$ ($\mathbf{C}_{Tx} = \{\mathbf{f}_1, \ldots, \mathbf{f}_{B_{Tx}}\}$).

As the BS transmits the sequence of beamformed signals, it is important to avoid losing information due to interference. To safeguard the correct measurement of each PDP, a small time interval (T_{guard}) should be considered between successive transmissions. This value should be designed so as to account for the longest paths.

The process of obtaining the BFF from the BS transmissions must result in similar data regardless of the listening device. To ensure so, the second key information resolution dictating aspect, the PDP sampling rate in phase B, must be the same for all devices. To understand how close the sampling rate is related to the resolution of the embedded information, consider a single propagation path between the BS and the mobile receiver. As discussed in [34], the maximum theoretical spatial resolution for a single time-based measurement is given by

$$d_{th} = T \times c, \qquad (2)$$

where d_{th} is the theoretical resolution of the distance in meters, T is the sampling period in seconds, and c is the speed of light in meters per second. As we can observe, the maximum resolution of the hidden information provided by the measured delay of each path is inversely proportional to the selected sampling rate. However, the sampling rate has associated trade-offs: using a higher sampling rate requires the allocation of additional radio spectrum resources, raises the energy requirements for the detection of each path due to thermal noise, and also places tougher hardware requirements for the mobile devices. In summary, it places harder practical constraints.

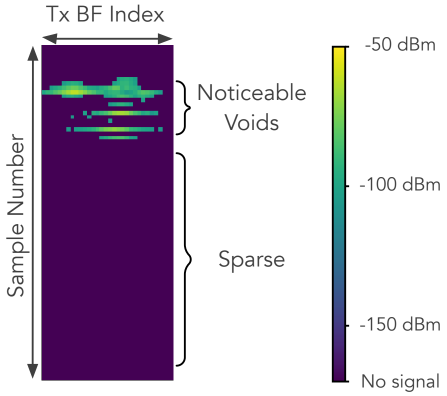

Fig. 3. Standard example of a noiseless beamformed fingerprint from the experimental simulations, containing the PDP (sampled at 20 MHz) for each beamformed transmission on the vertical axis.

A fixed BF gain should also be established for all receivers, to guarantee that the BFFs are device-invariant (and thus removing an undesirable degree of freedom from the system). In that case, the receivers would have to define their own BF codebook, $\mathbf{C}_{Rx}$, containing B_{Rx} elements ($\mathbf{C}_{Rx} = \{\mathbf{w}_1, \dots, \mathbf{w}_{B_{Rx}}\}$). The codebooks would have to be designed so as to search over all AoAs with similar gain, so as to avoid a scenario akin to the *orientation unaware* situation described in [32], where the device orientation becomes an extra variable (*i.e.*, to make the system as rotation-invariant as possible). With BF at the receiver, the device would have to sample each transmitter BF B_{Rx} times, storing the maximum measured value for each sample within a PDP. The acquired data from the i-th transmitter BF, $\mathbf{x}_i$, can thus be written as

$$\mathbf{x}_i[n] = \max_{j=1,\dots,B_{Rx}} r_j(nT), \quad n = 0, 1, \dots, N-1, \qquad (3)$$

where r_j is the time-domain sampled signal using the receiver beamforming $\mathbf{w}_j$, and N is the number of samples considered per PDP. Since eq. (3) considers the maximum value among all receiver BFs, the obtained fingerprint data ($\mathbf{X}$) has a negligible dependency on the mobile device orientation if the codebook was designed as described above.

After the BFF is obtained, a trained DL method will infer the device position in phase C (Fig. 1). With a DL method, the system learns to interpret the non-linearities introduced by reflections and other propagation artifacts. Interestingly, the work in [6], released shortly after the original proposal of the BFFs [7], pointed out machine learning methods as a possible solution to cope with the non-linearities, which were the flagged cause for their lack of positive NLOS experimental measurements. It should be noted that each BS will generate their own dataset and, therefore, will need their own model. Nevertheless, just like with image-processing DL systems, the BFFs relative to different BSs are still a result of the same physical world, and thus it should be possible to generate a pre-trained model to speed up downstream training procedures.

During phase D, the BS sends the position estimate to the mobile device. Alternatively, phase C could be performed at the mobile device, avoiding the data upload to the BS, as depicted in Fig. 2. However, the device would have to download a DLmodel for each BS, placing a significant memory and energy constrain on the device, and thus herein the predictions are suggested to be computed at the BS (as depicted in Fig. 1). Nevertheless, it is an option to be considered, as it will be analysed throughout this paper.

B. Beamformed Fingerprint Data Analysis

In the previous subsection, two key drivers of the amount of information found within a BFF were flagged: the directivity of the beamformed transmissions and the PDP sampling rate. This subsection will expand on those topics, in reverse order.

Let us consider a mobile device with a given sampling period, T. Besides the theoretical limit described by eq. (3), there is an additional detail that we should consider when defining T, in order to obtain high-quality data. If the sampling frequency exceeds 10 MHz (*i.e.*, $T < 100$ ns), the radiation arriving from the multiple propagation paths is detected in clusters, as described in [35]. Consequently, if such sampling frequencies are applied in the BFF positioning system, the PDPs will contain voids large enough to be detected. The ability to distinguish these voids provides a meaningful shape to the resulting data, enhancing the learning capabilities of the system (as further analysed in [9]).

The existence of strong reflections in the transmitted mmWave radiation suggests the existence of long propagation paths that can stay confined to limited areas. As such, the mobile device should gather a substantial number of samples per transmitter BF (N), so as to account for those infrequent, yet powerful sources of information. A consequence of this approach is the sparseness of the data, as it can be observed in the example plotted in Fig. 3. In fact, due to this sparseness, the relative position of the acquired non-zero samples in the data contains the majority of the extractable information, and we can use that to compress the BFF signal (as will be seen in the next subsection). In fact, it was shown in [7] that using binary-sampled PDPs has a marginal impact on the final accuracy, while greatly reducing the requirements for the mobile device.

When examining a BFF, it is interesting to notice a visual pattern that arises when the sequence of transmitted BF indices correspond to a continuous sweep over the azimuth (as in the simulation that resulted in Fig. 3). That visual pattern consists in distinct lines along the BF axis, which means that adjacent BF patterns will likely end up having similar clusters when measured from the same location, carrying partially redundant information. This characteristic indicates that there are clear diminishing returns if additional beamformed transmissions are sent without increasing their directivity.

C. Beamformed Fingerprint Power Requirements

Any positioning system will be mostly used by mobile devices, and thus it is of paramount importance to assess their energy requirements. For the BFF positioning method, it is observable from Fig. 1 that the energy requirements can be broken down to:

- *a)* Sampling the received radiation, so as to extract the desired data;

- *b)* Sending that data back to the BS and receiving the position fix; OR
- *c)* Performing inference on the device.

The energy necessary to sample the PDP consists mostly on the energy required by whole mmWave Radio Frequency (RF) front-end during the listening time (E_{Rx}). Considering the listening time needed by each of the $B_{Tx} \times B_{Rx}$ transmitted pulses required in order to obtain a BFF, we can estimate E_{Rx} as

$$E_{Rx} = P_{Rx}\big((T \times N) + T_{guard}\big)\big(B_{Tx} \times B_{Rx}\big), \qquad (4)$$

where P_{Rx} is the average power required by the RF front-end, and $((T \times N) + T_{guard})$ is the time per pulse. In [36], an assessment of the state-of-the-art for mmWave RF components concluded that a device's receiver front-end should require about 125 mW – the mmWave systems are under exhaustive study, and thus these figures should improve throughout the following years.

Let's assume now that inference is to be performed at the BS. In that case, the obtained data must now be transmitted to the BS. Considering that each of the $N \times B_{Tx}$ data samples contains k bits, if the system has an energy efficiency of E_{ef} Joules per transmitted bit, the required transmit energy (E_{Tx}) can be written as

$$E_{Tx} = (k \times N \times B_{Tx})\, E_{ef}. \qquad (5)$$

Since the device will have mmWave antennas installed, it should be able to use a 5G-enabled mmWave connection. With the data from the study performed in [37], it is conservative to assume an uplink energy consumption of 0.2 μJ per transmitted bit (or 5 Mbits per Joule).

As mentioned in the previous sub-section, when a small sampling period is used, the resulting data is sparse, and thus it can be efficiently compressed. For instance, if the transmitted data contains exclusively pairs of non-zero values (the received power) and their respective positions in the data sample, the required transmit energy can be rewritten as

$$E_{Tx} = \Big(V \times \big(\lceil log_2\,(N \times B_{Tx})\rceil + k\big)\Big) E_{ef}, \qquad (6)$$

where V is the number of valid entries (*i.e.*, non-zero entries), $\lceil log_2\,(N \times B_{Tx})\rceil$ is the number of bits required to encode all possible positions, and k represents the actual data for each valid entry. Furthermore, when the data is obtained through the simpler binary detection, the data in k (received power) becomes redundant, and thus

$$E_{Tx} = \big(V \times \lceil log_2\,(N \times B_{Tx})\rceil\big) E_{ef}. \qquad (7)$$

Considering that the power required to receive the final result is negligible, the total required energy per position fix can be approximated by adding (4) to either (6) or (7) (for non-binary and binary data, respectively).

Let's now consider the other alternative for the position inference – the mobile device performs its own machine learning computations. Although DL architectures are often considered computationally demanding, in practice, one can build a complex model with a very limited set of computational operations. This spurred the development of energy-efficient computing architectures specifically tailored to them, often based on Graphics Processing Unit (GPU) or Field-Programmable Gate Array (FPGA) architectures [10][11] . In fact, for general purpose mobile devices such as the smartphone, a large number of them are currently sold with dedicated hardware for DL computations. As such, to assess the energy consumption of the BFF position inference when it is computed in the mobile device, one must measure the energy consumption at the used dedicated hardware.

V. Learning the Beamformed Fingerprints

A. Positioning

The BFF positioning problem can be seen as the supervised learning of the training set $\mathcal{T}$, with samples obtained from a fixed distribution $\mathcal{D}_{\mathcal{X}\times\mathcal{Y}}$. The input space $\mathcal{X} = \mathbb{R}^{(N \times B_{Tx})}$ corresponds to the set of possible BFFs, whereas the target space $\mathcal{Y} = \mathbb{R}^d$ represents the set of all possible positions, and d is the number of dimensions of the positioning space. The objective of the BFF positioning system is to train a mapping function $f : \mathcal{X} \mapsto \mathcal{Y}$ using $\mathcal{T}$, so that it can generalize to new, unseen samples.

The simplest DL architecture applicable to BFF positioning, typically called a DNN, contains a sequence of fully connected layers with multiple neurons each. This layer type is ubiquitous and is used in most DL architectures. The vector containing the output of the i-th layer of neurons $\mathbf{n}_i$ can be described as

$$\mathbf{n}_i = a\left(\mathbf{U}_i\ \mathbf{n}_{i-1} + \mathbf{b}_i\right), \qquad (8)$$

where $\mathbf{U}_i$ represents the weight matrix, $\mathbf{b}_i$ is the bias, and a depicts an activation function, a non-linear subdifferentiable function. The input data $\mathbf{X}$, a BFF in the context of this section, feeds the first layer ($\mathbf{n}_0$), which is also known as input layer.

The presence of a non-linear activation function is absolutely critical, as it enables learning non-linear relationships. To map the input BFF data to the target label, the network is trained using a gradient-based algorithm [33]. This supervised training is guided by a loss function that represents a measurement of the average similarity between the true labels and the model's predictions. For the proposed system, all DL architectures are trained to perform a regression in the output layer, minimizing the Mean-Square-Error (MSE) to the labeled position $\mathbf{y}$, *i.e.*,

$$\hat{\mathbf{y}}^* = \arg\min_{\hat{\mathbf{y}}}\ \mathbf{E}\Big\{\big(\hat{\mathbf{y}} - \mathbf{y}\big)^{\mathbf{T}}\big(\hat{\mathbf{y}} - \mathbf{y}\big)\Big\}, \qquad (9)$$

where $\hat{\mathbf{y}}^*$, the output of the last layer of the neural network, denotes the position estimate given the input data $\mathbf{X}$.

Let us now consider the two indexing dimensions of the BFF, the PDP sample number and the transmitter BF index (see Fig. 3). As described in Section IV-B, if the sequence of BF indices represents to a continuous sweep over the azimuth, it also becomes possible to extract coherent information from the sequence of data points along the dimensions. Even though the different dimensions have completely different meanings (as opposed to images), the nature of the problem at hand makes CNNs a suitable candidate, as illustrated in Fig. 4.

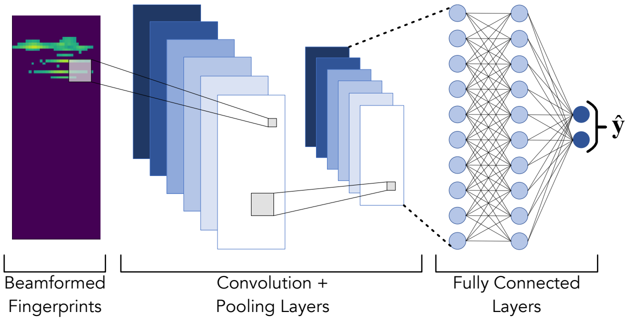

Fig. 4. Even though the two dimensions in a BFF have distinct meanings, as opposed to an image, the data sequences along both dimensions carry significant information. For that reason, the system can efficiently tap an additional source of information by using a CNN [7]

CNNs introduce the convolutional layer, where the model can learn the most effective set of short bi-dimensional filters to apply to the received data. These allow the extraction of information from the neighborhood of each data point within a sample, in addition to the information contained in the data point itself, building a more efficient representation. For the i-th convolutional layer of neurons $\mathbf{N}$, which is now a matrix, the output of the f-th filter can be described as

$$\mathbf{N}_i^f = a\left(\sum_{\tilde{f}=1}^{\tilde{F}} \left(\mathbf{U}_i^{f,\tilde{f}} \, \mathbf{N}_{i-1}^{\tilde{f}} \right) + \mathbf{1} \times b_i^f \right), \tag{10}$$

where $\tilde{F}$ is the number of filters in the previous layer, $\mathbf{1}$ is a bi-dimensional matrix of ones, bias b_l^f becomes a single scalar, and each $\mathbf{U}_i^{f,\tilde{f}}$, now denoting a bi-dimensional filter, is a double block circulant matrix (a special case of a Toeplitz matrix). Due to the new structure, if $\mathbf{U}_i^{f,\tilde{f}}$ is built from a $L1$ by $L2$ bi-dimensional filter, it only contains $L1 \times L2$ parameters. The number of learnable parameters in a convolutional layer is then much smaller than a fully connected layer's, for neural networks with similar performance [33]. Another major benefit that arises from the use of convolutional layers is the enhanced generalization capability, as the model will use the same filters in different parts of the input data.

The outdoor positioning problem maps a set of input data to a continuous space $\mathcal{Y}$, the position. Due to physical laws that determine electromagnetic propagation, a given transmitted signal is expected to be highly correlated when measured in adjacent positions. In fact, if it was not for the non-linear phenomena introduced with mmWave frequencies, the received BFFs would show smooth changes throughout the considered space. The non-linear phenomena introduces discontinuities to the BFF data, if assessed throughout a continuous route, segmenting the output space into multiple potential sub-regions, each with specific patterns in the input data. In [8], a hierarchical system that makes use of this abstract sub-region concept was proposed, to further refine the single BFF learning mechanism.

Let us consider a given BS's covered space, which can be seen as a set of K sub-regions with arbitrary boundaries[1] $\mathbf{S}$ ($\mathbf{S} = \{s_1, \ldots, s_K\}$, $\bigcup_{k=1}^{K} s_k = \mathcal{Y}$). If a dedicated CNN is assigned to each sub-region, these K CNNs will become specialists in their own data partition. Adjacent positions are very likely to be highly correlated and, as a consequence, they are expected to contain similar data patterns. This implies that each dedicated CNN regressor has to learn fewer patterns, facilitating its learning process. Moreover, we can train a CNN classifier to predict $\hat{s}_k$, the sub-region that is most likely to hold the present input, to select the dedicated CNN regressor that should be used to estimate the device location. Such an architecture is further referred to as Hierarchical Convolutional Neural Network (HCNN), where the positioning problem is split in a hierarchical manner: sub-region selection followed by the position estimation.

To train the aforementioned CNN classifier ,which attempts to select the correct sub-region, cross-entropy between prediction and ground truth is minimized, such as

$$p(\hat{\mathbf{s}}) = \underset{p(\hat{s}_k),\ k=1,\ldots,K}{\arg\min} \ \mathbf{E}\Big\{ -\sum_k p(s_k|\mathbf{X}) \log(p(\hat{s}_k)) \Big\}, \tag{11}$$

where $p(\hat{\mathbf{s}})$ denotes the output vector of the classifier neural network, containing the predicted probabilities $p(\hat{s} = s_k)$ for an input data $\mathbf{X}$. The above formulation allows $\mathbf{S}$ to contain overlapped subregions, as $p(s_k|\mathbf{X})$, the true probability of being in s_k given the input data $\mathbf{X}$, can be 1 for multiple k. Once the classifier's output is obtained, the most suitable dedicated CNN regressor $\hat{s}$ is selected by determining

$$\hat{s} = \underset{k=1,\ldots,K}{\arg\max} \ p(\hat{s} = s_k), \tag{12}$$

which in turn provides the final position estimate $\hat{\mathbf{y}}$.

B. Tracking

The previous subsection concerned DL architectures that can convert a single BFF into a position estimate. Nevertheless, many systems continuously request localization services during a significant amount of time, and their movement is a strong information source. Not only there are physical constraints, such as the velocity of the mobile device carrier, but also there are human-imposed restrictions, such as traffic rules. Therefore, by accessing information regarding recent positions, expected trajectory, and other devices' movement history, the system can infer the range plausible positions, and thus significantly narrow down its final estimate.

In this subsection, sequence-based DL architectures for the BFF positioning system are proposed. This new set of architectures aims to learn the mapping function $f : \mathcal{X}^M \mapsto \mathcal{Y}$, where M is the input sequence length (or the system's memory size). Consequently, the training set $\mathcal{T}$ is now obtained from the fixed distribution $\mathcal{D}_{\mathcal{X}^M \times \mathcal{Y}}$, where $\mathcal{X}^M$ is now the set of possible BFF sequences.

The historical default DL architecture to deal with sequences is the Recurrent Neural Network (RNN). In recent years, multiple variants of RNNs were proposed, namely Long Short-Term Memorys (LSTMs) [38], which were developed to handle the vanishing and exploding gradient problems that often plagued vanilla RNNs' training. LSTMs are known for

[1] Please note that the methods that can be used to define the sub-regions $\mathbf{S}$, briefly discussed in [8], are outside the scope of this paper.

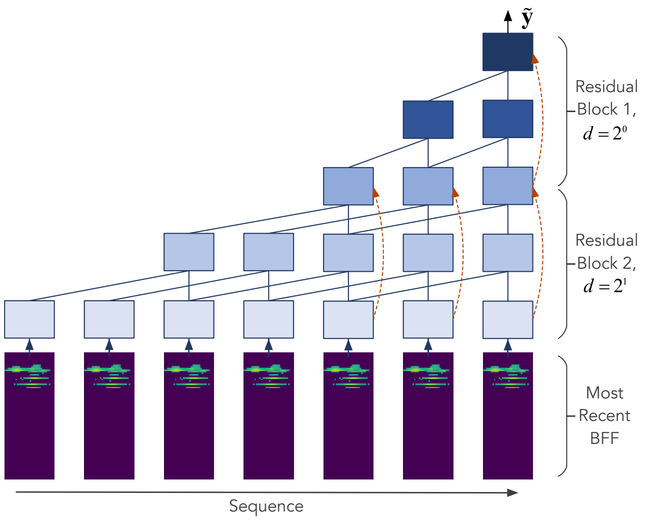

Fig. 5. Core of a TCN model, excluding the output layer after the last residual block's output ($\tilde{\mathbf{y}}$) [9]. With each subsequent residual block, the receptive field increases exponentially, due to the dilation factor d. The dashed lines depict the residual connections.

their good results in multiple sequence-based tasks, including indoor tracking using WiFi fingerprints [39].

Unlike DNNs, RNN-based architectures hold an internal state that retains information as a sequence is processed. This mechanism allows a model to digest sequences of arbitrary length, while keeping an understanding of the chain of events. It also shares the model's trained weights as it traverses the sequence (the same operations are applied to each BFF), which, as mentioned in the previous subsection, results in better generalization capabilities.

Each step of the sequential model can be abstracted within a LSTM module. The output of the m-th LSTM module is described as

$$\mathbf{h}_m = \mathbf{o}_m \odot \tanh\left(\mathbf{C}_m\right), \tag{13}$$

where $\mathbf{C}_m$ is the *cell state*, $\mathbf{o}_m$ is the *output gate*, $\odot$ denotes the Hadamard product, and $\tanh(\cdot)$ represents the hyperbolic tangent function. The output gate, containing a mixture of the current input sample being assessed and the previous module's output, selects which parts of the cell state's information are to be passed to the output module. More specifically, the output gate is

$$\mathbf{o}_m = \sigma\left(\mathbf{U}_o\left[\mathbf{h}_{m-1};\mathbf{x}_m\right] + \mathbf{b}_o\right), \tag{14}$$

with $\sigma(\cdot)$ denoting the sigmoid function. Consistently with the previous sections' notation, $\mathbf{U}$, $\mathbf{b}$ and $\mathbf{x}$ represent weights, bias and BFF data (as a vector), respectively.

The cell state can be defined as

$$\mathbf{C}_m = \mathbf{f}_m \odot \mathbf{C}_{m-1} + \mathbf{i}_m \odot \tilde{\mathbf{C}}_m, \tag{15}$$

where $\mathbf{f}_m$ represents the *forget gate*, and $\mathbf{i}_m$ the *input gate*. The *forget gate* controls the information to be discarded by the cell state, relative to its own past state, while the *input gate* filters the information contained in $\tilde{\mathbf{C}}_m$, which is then added to the cell state. These two expressions are given by

$$\mathbf{f}_m = \sigma\big(\mathbf{U}_f\left[\mathbf{h}_{m-1};\mathbf{x}_m\right] + \mathbf{b}_f\big) \quad \text{and} \tag{16}$$

$$\mathbf{i}_m = \sigma\big(\mathbf{U}_i\left[\mathbf{h}_{m-1};\mathbf{x}_m\right] + \mathbf{b}_i\big), \tag{17}$$

and the candidate values to be added to the cell state, $\tilde{\mathbf{C}}_m$, are given as

$$\tilde{\mathbf{C}}_m = \text{tahn}\big(\mathbf{U}_c\left[\mathbf{h}_{m-1};\mathbf{x}_m\right] + \mathbf{b}_c\big). \tag{18}$$

Equations (13)-(18) imply two different activation functions: the sigmoid and the hyperbolic tangent. The former, whose output ranges from 0 to 1, is used as an information filter (*gates*), while the later, ranging from -1 to 1, adds critical non-linearities and limits the output range of the data passed between LSTM modules. Frequently, fully connected layers are usually placed after the last LSTM module, mapping the output vector $\mathbf{h}_M$ to the desired output format ($\hat{\mathbf{y}}$).

Although LSTMs are a proven tool to learn from sequences, they are known to be difficult to train [40]. Also, there are multiple sequence-based problems for which CNN provide the best solution [41]. To harness the potential of the convolution operation, well known to the signal-processing community and designed to handle sequences, while being able to process long sequences, TCNs were proposed in [41].

Compared to typical CNNs, TCNs present three main differences. First and foremost, any non-sequence-dimension (*feature*) size mismatch between two subsequent layers is processed through a 1D convolution [42]. This guarantees that for each step in the input sequence, there is a single corresponding step in each hidden layer (as observable in Fig. 5).

If the convolution operation is applied directly over the sequence dimension, its size grows linearly with the expected sequence size, which is undesirable. For instance, RNN-based architectures in theory do not need to scale with the sequence length. Consequently, the second feature of a TCN lies on the introduction of dilated convolutions, which enable an exponentially large receptive field. The dilated convolution operation F on element m of the sequence $\mathbf{x}$, using a filter f, is defined as

$$F[m] = \big(\mathbf{x} *_d f\big)[m] = \sum_{l=1}^{L} f[l] \cdot \mathbf{x}[m - d \cdot l], \tag{19}$$

where L is the length of the dilated convolution and d the dilation factor. Since d is set to grow exponentially with the depth of the network, each subsequent layer can be interpreted as a *zoom out* in the sequence data, enabling the network to perceive larger sequences with a limited set of learnable parameters. If the TCN's receptive field is larger than the input sequence, the input sequence can be zero-padded.

Finally, the last key element of a TCN consists on the use of the residual block [43]. In it, the network accesses the original input data every two dilated convolution layers, which is critical to stabilize large networks. More formally, if $\mathbf{x}$ is the input of a given residual block, its output $\tilde{\mathbf{y}}$ is defined as

$$\tilde{\mathbf{y}} = a\big(\mathcal{F}(\mathbf{x}) + \mathbf{x}\big), \tag{20}$$

where a represents an activation function, and $\mathcal{F}$ a series of transformations corresponding to the two dilated convolutions within the residual block (and 1D convolutions being performed to match $\mathbf{x}$ to $\mathcal{F}(\mathbf{x})$, if needed). The stack of all

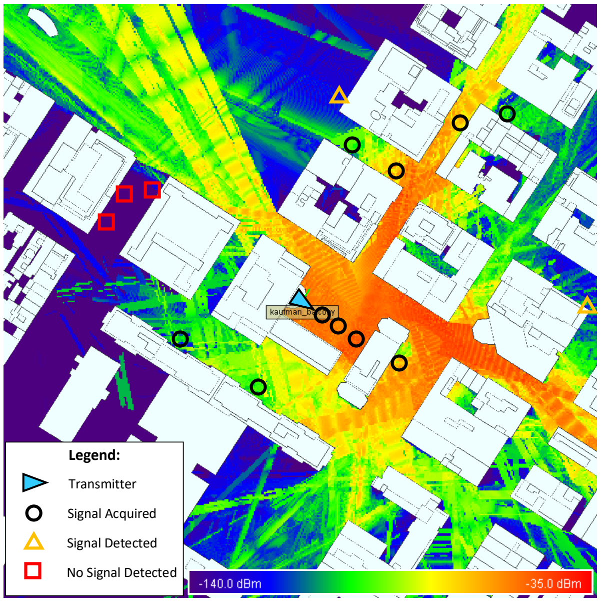

Fig. 6. Ray-tracing simulation in the NYU area, using the parameters in Table I. The results shown correspond to the *maximum received power for all possible transmit BFs*. In [44], it was shown that this simulation matched the experimental measurements in [35], depicted by the symbols.

TABLE I
RAY-TRACING SIMULATION PARAMETERS

Parameter Name	Value
Carrier Frequency	28 GHz
Transmit Power	45 dBm
Tx. Antenna Gain	24.5 dBi (horn antenna)
HPBW	10.9°
Transmitter Downtilt	10°
Codebook Size	32 (155° arc with 5° between entries)
Receiver Grid Size	160801 (400 × 400 m, 1 m between Rx, 1 m above the ground)
Samples per Tx. BF	82 (4.1 μs @ 20 MHz)
Assumed Rx. Gain	10 dBi (as in [46])
Detection Threshold	-100 dBm
Added Noise	σ = [2, 10] dB (Log-Normal)

TABLE II
SYNTHETIC PATH GENERATION PARAMETERS

Parameter Name	Pedestrian-like	Vehicle-like
Default Speed (m/s)	1.4	8.3
Maximum Speed (m/s)	2.0	13.9
Maximum Acceleration (m/s^2)	0.3	3
Maximum Direction Change ($^\circ/s$)	10.0	5.0
p(No Movement Change)	0.8	0.8
p(Full Stop)	0.1	0.02
p(Speed Change)	0.05	0.05
p(Direction Change)	0.05	0.13

these residual blocks builds a TCN. The output of the last residual block, $\tilde{\mathbf{y}}$, must then go through the output layer, so as to produce the desired prediction ($\hat{\mathbf{y}}$).

VI. SIMULATION APPARATUS

To evaluate the proposed system's accuracy and its power consumption, a ray tracing-based dataset is used. That dataset was built from ray tracing simulations in the NYU area, containing BFF data from 160801 different positions in a 400×400 m area. The propagation specifications in Table I were inherited from the experimental measurements in [35] and, in [44], it was shown that these ray tracing simulations (briefly presented in Fig. 6) matched those experimental measurements. Although the acquired dataset has some limitations that negatively impact the accuracy results, such as a minimum distance of 1 m between samples, the results shown in this paper do not consider some adverse scenarios, such as weather effects (which are known to have a significant effect on mmWave systems).

The ray tracing software used, Wireless InSite 3.0.0.1 [45], was unable to control BF patterns. A physically rotating horn antenna was used instead, producing similar directive radiation patterns. For each of the 32 elements in $\mathbf{C}_{Tx}$, the corresponding PDP was sampled at 20 MHz over a spawn of 4.1 μs, which contained over 99% of the path data. Regarding BF at the receiver, a 10 dBi gain was considered (as in the mobile device in [46], which contains a codebook with 8 entries). In the forthcoming simulations, noise is added to the obtained ray tracing data following a log-normal distribution (*slow fading*). The noise was introduced before applying a detection threshold of -100 dBm, which was selected due to the thermal noise at room temperature for the considered bandwidth (-101 dBm). In all simulations, the data is binarized after applying the detection threshold.

Data resulting from the experiments was labeled with the corresponding bi-dimensional position, in a 400×400 m^2 area centered at the BS. Regarding the HCNN, the sub-regions correspond to subsequent bisections of the output space (e.g. when 64 partitions are considered, each dimension is bisected 8 times, resulting in partitions with 50×50 m^2).

The sequences of BFFs generated for the LSTMs and the TCNs simulations consider three distinct types of synthetic user paths: static, pedestrian-like, and vehicle-like. While static users remain in the same position for the duration of the sequence, users following the other two path types move according to the specifications depicted in Table II. The pedestrian-like paths were generated with the typical human preferred walking speed (5 km/h), but that can quickly stop or change their direction. On the other hand, vehicle-like paths were generated with higher default speed (30 km/h) and maximum acceleration, but with a restricted steering angle. The probabilities depicted in the second half of Table II are applied once per second, where a *full stop* stops a user for a second, before restarting its movement in a random direction (uniformly sampled) with the default speed, and the *speed* and *direction changes* modify the existing speed or direction by a value uniformly sampled between the specified maximum and its negation (e.g. a vehicle-like path can accelerate or decelerate by an amount ranging from -3 to 3 m/s^2).

In an attempt to emulate the behavior or typical civilian GNSS receivers, the sequences of BFFs are created by drawing a noisy BFF sample once per second (*i.e.* sampled at 1 Hz). To be representative of a real-life scenario, where most users are moving, there is a ratio of $8:1$ moving to static paths (the moving paths are uniormly distributed between pedestrian-

TABLE III
POSITIONING HYPERPARAMETERS

Parameter Name	Value
Convolutional Layers	1 layer (8 features with 3×3 filters)
Pooling Layers	2×1 max-pooling
Hidden Layers	12 (256 neurons each)
Regression Output	Linear with 2 Neurons (2D position)
Classification Output (HCNN's 1st model)	Softmax with $K = 64$ classes
Epochs	Up to 1000 (early stopping [48] after 50 non-improving epochs)
Batch Size	64
Optimizer	ADAM[49]
Learning Rate	10^{-4}
Learning Rate Decay	0.995
Dropout	0.01

TABLE IV
TRACKING HYPERPARAMETERS

Parameter Name	LSTMs	TCNs
LSTM Units	512	—
TCN Blocks	—	2
TCN Filter Length	—	3
TCN Features	—	512
MLP Layers	2 (512 neurons each)	0
Regression Output	Linear with 2 Neurons (2D position)	
Total # of Sequences	720918	
Sequence Length (M)	7	
Epochs	Up to 100 (early stopping [48] after 5 non-improving epochs)	
Batch Size	64	
Optimizer	ADAM	
Learning Rate	5×10^{-5}	5×10^{-4}
Learning Rate Decay	0.995	

and vehicle-like paths). The test paths, corresponding to 20% of the generated paths, are hidden while training, to avoid a simple memorization of possible paths. The total training time required for the presented results was also considered – training any of the assessed architectures with the defined parameters takes less than 10 hours on an Nvidia GTX 780 Ti GPU, using Google's TensorFlow framework [47]. In a practical setting, this train time can be further reduced by employing better hardware, through pre-trained models, or by cherry-picking a dataset that contains more samples in regions hard to locate. Finally, we believe that reproducibility is fundamental to validate proposals and to push research forward, and thus we made the simulation code and used dataset publicly available[2].

The next section addresses the experiments that assess the energy efficiency of the positioning method. When the position inference is performed in the BS, the equations from Section IV-C are used. However, to assess the energy consumption when the computations are to be performed in the mobile device, a low-power, mobile-friendly GPU is used (Nvidia Jetson TX2 [10]). This particular device contains the possibility of probing its own internal power consumption counters through software, from which we can derive the energy consumed per position fix. The repository linked above contains simulation code and includes all the instructions required to reproduce the energy consumption measurements for the present family of devices used.

VII. EXPERIMENTAL RESULTS

In this section, the proposed system is simulated and evaluated by applying the data and the parameters discussed in the previous section. The results are obtained using a 32-element codebook dataset with binary detection, $K = 64$ for HCNN and $M = 7$ for sequence-based approaches. For the impact of different codebook sizes and different sampling frequencies, the use of non-binary BFF samples, and different values for K and M, please refer to the accuracy results in [7] and [9].

In Fig. 7, the achievable accuracy for the DL architectures presented throughout Section V is shown. It can be seen that the accuracy improves in the same order as the DL architectures were presented, with a noticeable performance increase when sequence-based architectures are used. Furthermore, the use of sequence-based architectures is particularly effective at containing the 95^{th} percentile error, due to the fact that they use multiple BFFs, and thus can handle noise spikes in an individual sample. The best accuracy results are obtained when TCNs are used, with an average and 95^{th} percentile errors of 2.03 and 5.81 meters, respectively, when $\sigma = 6$ dB. These results indicate that BFF positioning methods can have a better accuracy than commercial GNSS positioning devices. Moreover, in [9], it was shown that these results were obtained with a single anchor in a heavy presence of NLOS positions, and that they are nearly $10\times$ more accurate than the previous state-of-the-art [30] for mmWave positioning with NLOS.

As mentioned throughout the paper, the proposed system has two modes of operation: either the mobile device sends the BFFs to the BS, delegating the inference process, or the mobile device computes the position estimate itself. The first mode of operation, where the mobile device can be oblivious of the methods used to estimate its position, has its energy requirements independent of the DL architecture used. The equations shown in Section IV-C are evaluated with the parameters from Table I. Considering $P_{Rx} = 125$ mW [36] and $T_{guard} = 2.9$ μs, so that a PDP is collected every 7 μs, the system would require 0.224 mJ to obtain the data regarding the received radiation. The previous value considers $B_{Tx} = 32$ and $B_{Rx} = 8$, as discussed in the previous apparatus section. From the simulations performed, the average number of non-zero entries per sample (V) ranged from 63.38 to 68.62, for $\sigma = 10$ dB and $\sigma = 0$ dB, respectively. Therefore, assuming a network energy efficiency of 0.2 μJ per transmitted bit [37], the system would need between 0.152 mJ ($\sigma = 10$ dB) and 0.165 mJ ($\sigma = 0$ dB) to upload the gathered information to the BS, on average. Combining both, the mobile device would need to spend between 0.376 mJ and 0.389 mJ per position fix.

For the mode of operation that makes computations happen at the mobile device, the energy consumption greatly depends on the DL architecture used. The DL architectures mentioned throughout Section V were implemented in a mobile GPU (Nvidia Jetson TX2 [10]), with the results shown in Table V. In this Table, it is shown the power consumption of the

[2] https://github.com/gante/mmWave-localization-learning

TABLE V
ENERGY CONSUMPTION OF INFERENCE FOR THE TESTED DL ARCHITECTURES ON A MOBILE GPU (NVIDIA JETSON TX2).

	Samples/s	GPU Power (mW)	Memory Power (mW)	Total Power (mW)	mJ/sample
CNN	3194.4	780	1141	3820	1.196
HCNN	1577.3	773	1210	4010	2.542
TCN	1367.1	4284	2271	9259	6.773
LSTM	1833.4	3747	2273	8682	4.735

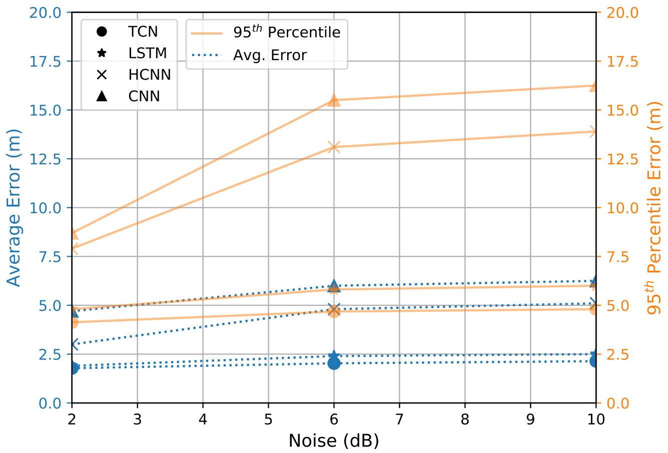

Fig. 7. A summary of the accuracy for the discussed DL architectures. As it can be seen, sequence-based positioning can achieve far greater precision, particularly with respect to the 95^{th} percentile error in the presence of noise.

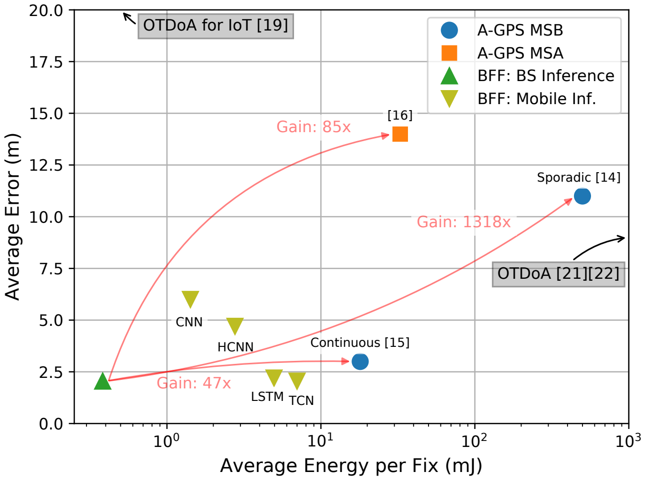

Fig. 8. Average error *vs* average energy required per position fix for the positioning technologies discussed in this paper. The proposed system is plotted for its two operation modes, depending on where the DL inference is computed. It is assumed that BS inference mode of operation uses the most accurate DL architecture available (TCN, in our experiments) As it is observable, the proposed system has an accuracy comparable to A-GPS, while achieving energy efficiency gains exceeding 47× per position fix.

Jetson TX2's components that are DL-related, as well as the total power consumption of the device (which includes, among other things, the system-on-a-chip). Together with the system's inference throughput, we can infer the energy cost of each position estimate. To the values shown, depicting the cost of the inference for each position estimate, the energy required to obtain the BFF must be added (0.224 mJ). Although this mode of operation seems to be less energy-efficient, the used GPU is still over-dimensioned for the problem (only a few samples per second are required), and there are known techniques to reduce the energy consumption at the inference stage, as elaborated in Section VIII.

In order to contextualize these values, let's recap the energy consumption methods for other positioning methods (to the best of our knowledge, the energy consumption for other mmWave positioning methods has yet to be studied). For the MSB A-GPS systems, the used data was taken directly from [14] and [15], which correspond to two state-of-the-art low-power A-GPS chips. Since the periodicity of the position fixes has a great impact on the energy consumption and accuracy of the MSB A-GPS method, two data points were considered: one for continuous measurements of one fix per second, resulting in full A-GPS accuracy and an average energy consumption of 18 mJ [15], and another for sporadic measurements (once per minute), with decreased accuracy and an average energy consumption of 504 mJ [14]. When evaluating MSA A-GPS systems, the majority of the energy consumption goes to the uplink transmission, which depends on the network used. Therefore, to enable a fair comparison, it is assumed that the MSA A-GPS system also has access to energy efficient mmWave networks. In order to obtain a position fix, the system in [17] takes binary samples of the GNSS signal at 16.368 MHz during 10 ms. Considering a network energy efficiency of 0.2 μJ per uploaded bit [37], that system would require 32.736 mJ to obtain a position fix with an average error of 14 m.

Fig. 8 plots the energy consumption versus the accuracy for the aforementioned methods. For better visualization, all data points for BFF positioning use the same noise value, $\sigma = 6$ dB, and for the case where the inference is done at the BS, the most accurate method is used. When compared to the assessed A-GPS implementations, the BFF positioning system with the inference made at the BS shows energy efficiency gains of 47× for continuous measurements (*vs* A-GPS MSB), and 85× for sporadic position fixes (*vs* A-GPS MSA using a mmWave network), while keeping slightly better accuracy levels. Furthermore, the proposed method is available whenever there is mmWave coverage, while requiring no additional hardware at devices with mmWave capabilities. As such, in this thesis we can conclude that BFF-based positioning methods can dethrone GNSS-based methods as the default low-power commercial positioning system.

VIII. CONCLUSION

In the context of 5G, millimeter wave communications will release a massive amount of bandwidth and introduce significant theoretical improvements. However, the transformation of such improvements into practice is far from being trivial, as the physics underlying the radiation propagation change dramatically.

In the context of outdoor positioning, the use of millimeter waves implies that the typical geometrical approaches are no longer reliable for NLOS positions. The concept of beamformed fingerprint, which was coined recently, enabled the application of deep learning techniques so as to achieve accurate outdoor positioning. The result is state-of-the-art accuracy for NLOS millimeter wave outdoor positions, while using a moderate bandwidth, binary data samples and a single anchor. In this paper, it was shown that the newly proposed system is far more energy efficient than conventional positioning strategies, while preserves similar precision, paving the way for smaller positioning-enabled autonomous devices.

A. Future Work

A significant portion of recent machine learning papers concern more complex DL architectures, leveraging the increasing amount of computing power available in the systems. However, it is also possible to alter the model in the opposite direction, sacrificing some accuracy in order to obtain higher model energy efficiency. In [50], the authors show that it is possible to train a smaller model from a larger model, retaining most of the larger model's precision, in a technique called distillation. The drawback of using distillation is that not only some precision is lost, but also training becomes more expensive – given that a model per BS is needed, the cost can be non-negligible. Finally, if the hardware used for the DL inference allows it, it is possible to trade some accuracy for energy efficiency through operations with fewer bit resolution. In fact, it was shown in [51] that representing the weights of a DNN with a single bit can result in very competitive results. This particular solution is particularly beneficial if the BFF position inference is to be computed at the mobile device, as the cost for transmitting and storing the networks' weights is drastically reduced.

IX. Acknowledgment

This work is funded by FCT/MCTES, Instituto de Telecomunicaes and INESC-ID through national funds and when applicable co-funded EU funds under projects UIDB/EEA/50008/2020, UIDB/CEC/50021/2020 and PTDC/EEI-HAC/30485/2017.

References

[1] L. A. Tawalbeh, A. Basalamah, R. Mehmood, and H. Tawalbeh, "Greener and smarter phones for future cities: Characterizing the impact of GPS signal strength on power consumption," *IEEE Access*, vol. 4, pp. 858–868, 2016.

[2] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter wave mobile communications for 5G cellular: It will work!" *IEEE Access*, vol. 1, pp. 335–349, 2013.

[3] J. A. del Peral-Rosado, R. Raulefs, J. A. Lpez-Salcedo, and G. Seco-Granados, "Survey of cellular mobile radio localization methods: From 1G to 5G," *IEEE Communications Surveys Tutorials*, vol. 20, no. 2, pp. 1124–1148, Secondquarter 2018.

[4] K. Witrisal, P. Meissner, E. Leitinger, Y. Shen, C. Gustafson, F. Tufvesson, K. Haneda, D. Dardari, A. F. Molisch, A. Conti, and M. Z. Win, "High-accuracy localization for assisted living: 5G systems will turn multipath channels from foe to friend," *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 59–70, March 2016.

[5] M. Koivisto, A. Hakkarainen, M. Costa, P. Kela, K. Leppanen, and M. Valkama, "High-efficiency device positioning and location-aware communications in dense 5G networks," *IEEE Communications Magazine*, vol. 55, no. 8, 2017.

[6] O. Kanhere and T. S. Rappaport, "Position Locationing for Millimeter Wave Systems," *2018 IEEE Global Communications Conference*, Dec. 2018. [Online]. Available: https://arxiv.org/abs/1808.07094

[7] J. Gante, G. Falco, and L. Sousa, "Beamformed fingerprint learning for accurate millimeter wave positioning," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, Aug 2018, pp. 1–5.

[8] J. Gante, G. Falcao, and L. Sousa, "Enhancing beamformed fingerprint outdoor positioning with hierarchical convolutional neural networks," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 1473–1477.

[9] J. Gante, G. Falcão, and L. Sousa, "Deep Learning Architectures for Accurate Millimeter Wave Positioning in 5G," *Neural Processing Letters*, Aug 2019. [Online]. Available: https://doi.org/10.1007/s11063-019-10073-1

[10] D. Franklin, "Nvidia developer blog: Nvidia jetson tx2 delivers twice the intelligence to the edge," https://devblogs.nvidia.com/jetson-tx2-delivers-twice-intelligence-edge/, accessed: 11th of January, 2020.

[11] "Intel(r) movidius(tm), intel movidius neural compute stick," https://software.intel.com/en-us/neural-compute-stick, accessed: 11th of January, 2020.

[12] F. van Diggelen, *A-GPS: Assisted GPS, GNSS, and SBAS*. Artech House, 2009.

[13] N. Vallina-Rodriguez, J. Crowcroft, A. Finamore, Y. Grunenberger, and K. Papagiannaki, "When assistance becomes dependence: Characterizing the costs and inefficiencies of A-GPS," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 17, no. 4, pp. 3–14, Dec. 2013.

[14] "Super-E: low power and good performance (white paper)," https://www.u-blox.com/en/white-papers , accessed: 19th of February, 2019; Requires registration.

[15] "MediaTek MT 3339 datasheet," https://labs.mediatek.com/en/chipset/MT3339 , accessed: 19th of February, 2018.

[16] S. C. Wu, W. I. Bertiger, D. Kuang, S. Nandi, L. J. Romans, and J. M. Srinivasan, "MicroGPS for low-cost orbit determination," *TDA Progress Report*, vol. 42-131, Nov. 1997.

[17] T. Nguyen Dinh and V. La The, "A novel design of low power consumption GPS positioning solution based on snapshot technique," in *2017 International Conference on Advanced Technologies for Communications (ATC)*, Oct 2017, pp. 285–290.

[18] S. Fischer, "Observed time difference of arrival (OTDOA) positioning in 3GPP LTE,," in *Qualcomm Technologies Inc., White Paper*, Jun 2014.

[19] S. Hu, A. Berg, X. Li, and F. Rusek, "Improving the performance of OTDOA based positioning in NB-IoT systems," in *2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–7.

[20] G. T. 37.857, "Study on indoor positioning enhancements for UTRA and LTE,," in *Rel. 13, V13.1.0*, Jan 2016.

[21] Z. Z. M. Kassas, J. Khalife, K. Shamaei, and J. Morales, "I hear, therefore I know where I am: Compensating for GNSS limitations with cellular signals," *IEEE Signal Processing Magazine*, vol. 34, no. 5, pp. 111–124, Sep. 2017.

[22] C. Chen and W. Wu, "Three-dimensional positioning for LTE systems," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3220–3234, April 2017.

[23] E. Lei, O. Castaeda, O. Tirkkonen, T. Goldstein, and C. Studer, "Siamese neural networks for wireless positioning and channel charting," 2019.

[24] H. Zhang, H. Du, Q. Ye, and C. Liu, "Utilizing CSI and RSSI to achieve high-precision outdoor positioning: A deep learning approach," in *2019 IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–6.

[25] S. Hu, F. Rusek, and O. Edfors, "Beyond massive MIMO: The potential of positioning with large intelligent surfaces," *IEEE Transactions on Signal Processing*, vol. 66, no. 7, pp. 1761–1774, April 2018.

[26] Z. Wei, Y. Zhao, X. Liu, and Z. Feng, "DoA-LF: A location fingerprint positioning algorithm with millimeter-wave," *IEEE Access*, vol. 5, 2017.

[27] X. Han, J. Wang, W. Shi, Q. Niu, and L. Xu, "Indoor precise positioning algorithm using 60GHz pulse based on compressive sensing," *Journal of Mathematics and Computer Science*, 2016.

[28] A. Shahmansoori, G. E. Garcia, G. Destino, G. Seco-Granados, and H. Wymeersch, “Position and orientation estimation through millimeter-wave MIMO in 5G systems,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1822–1835, March 2018.

[29] Z. Abu-Shaban, X. Zhou, T. Abhayapala, G. Seco-Granados, and H. Wymeersch, “Error bounds for uplink and downlink 3D localization in 5G mmwave systems,” *IEEE Transactions on Wireless Communications*, pp. 1–1, 2018.

[30] V. Savic and E. G. Larsson, “Fingerprinting-based positioning in distributed massive MIMO systems,” in *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, Sept 2015.

[31] L. R. W. Mohinder S. Grewal and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration, 2nd ed.* Wiley, 2007.

[32] A. Guerra, F. Guidi, and D. Dardari, “Single-anchor localization and orientation performance limits using massive arrays: MIMO vs.beamforming,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5241–5255, Aug 2018.

[33] Y. Bengio, Y. LeCun, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, 2015.

[34] F. Lemic, J. Martin, C. Yarp, D. Chan, V. Handziski, R. Brodersen, G. Fettweis, A. Wolisz, and J. Wawrzynek, “Localization as a feature of mmWave communication,” in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Sep. 2016, pp. 1033–1038.

[35] Y. Azar, G. N. Wong, K. Wang, R. Mayzus, J. K. Schulz, H. Zhao, F. Gutierrez, D. Hwang, and T. S. Rappaport, “28 GHz propagation measurements for outdoor cellular communications using steerable beam antennas in New York city,” in *2013 IEEE International Conference on Communications (ICC)*, June 2013, pp. 5143–5147.

[36] W. B. Abbas, F. Gomez-Cuba, and M. Zorzi, “Millimeter wave receiver efficiency: A comprehensive comparison of beamforming schemes with low resolution ADCs,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 8131–8146, Dec 2017.

[37] S. Buzzi and C. DAndrea, “Energy efficiency and asymptotic performance evaluation of beamforming structures in doubly massive MIMO mmwave systems,” *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 2, pp. 385–396, June 2018.

[38] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[39] S. Bai, M. Yan, Y. Luo, and Q. Wan, “RFedRNN: An end-to-end recurrent neural network for radio frequency path fingerprinting,” in *Recent Trends and Future Technology in Applied Intelligence*, M. Mouhoub, S. Sadaoui, O. Ait Mohamed, and M. Ali, Eds. Cham: Springer International Publishing, 2018, pp. 560–571.

[40] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ser. ICML’13, 2013, pp. III–1310–III–1318. [Online]. Available: http://dl.acm.org/citation.cfm?id=3042817.3043083

[41] S. Bai, J. Zico Kolter, and V. Koltun, “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling,” *arXiv e-prints*, p. arXiv:1803.01271, Mar. 2018.

[42] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3431–3440.

[43] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[44] J. Gante, G. Falciao, and L. Sousa, “Data-aided fast beamforming selection for 5G,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 1183–1187.

[45] “Wireless InSite web-page,” https://www.remcom.com/wireless-insite-em-propagation-software/ , accessed: 19th of February, 2019.

[46] T. Obara, Y. Inoue, Y. Aoki, S. Suyama, J. Lee, and Y. Okumurav, “Experiment of 28 GHz band 5G super wideband transmission using beamforming and beam tracking in high mobility environment,” in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sep. 2016, pp. 1–5.

[47] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, “Tensorflow: A system for large-scale machine learning,” in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

[48] R. Caruana, S. Lawrence, and L. Giles, “Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping,” in *Proceedings of the 13th International Conference on Neural Information Processing Systems*, ser. NIPS’00. Cambridge, MA, USA: MIT Press, 2000, pp. 381–387. [Online]. Available: http://dl.acm.org/citation.cfm?id=3008751.3008807

[49] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[50] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” 2015.

[51] M. Kim and P. Smaragdis, “Bitwise neural networks,” 2016.

Joao Gante is a Ph.D. candidate in Electrical and Computer Engineering at Instituto Superior Tcnico, Universidade de Lisboa (IST-UL), and a researcher of Engenharia de Sistemas e Computadores Investigao e Desenvolvimento (INESC-ID). He was awarded a doctoral grant by Fundao para a Ciłncia e Tecnologia (FCT), and was the recipient of the top 3% students award at the University of Coimbra in 2009, 2010, 2011, and 2012.

His current research activities focus on the development of an energy-efficient location system for 5G networks, making use of machine learning and MIMO antenna beamforming techniques.

Leonel Sousa (M01SM03) received a Ph.D. degree in Electrical and Computer Engineering from the Instituto Superior Tecnico (IST), Universidade de Lisboa (UL), Lisbon, Portugal, in 1996, where he is currently Full Professor. He is also a Senior Researcher with the R&D Instituto de Engenharia de Sistemas e Computadores (INESC-ID). His research interests include VLSI architectures, computer architectures, parallel computing, computer arithmetic, and signal processing systems. He has contributed to more than 200 papers in journals and international conferences, for which he got several awards - such as, DASIP’13 Best Paper Award, SAMOS’11 ’Stamatis Vassiliadis’ Best Paper Award, DASIP’10 Best Poster Award, and the Honorable Mention Award UTL/Santander Totta for the quality of the publications in 2009. He has contributed to the organization of several international conferences, namely as program chair and as general and topic chair, and has given keynotes in some of them. He has edited two special issues of international journals, and he is currently Associate Editor of the IEEE Transactions on Computers, IEEE Access and Springer JRTIP. He is Fellow of the IET and a Distinguished Scientist of ACM.

Gabriel Falcao (S’07–M10–SM’14) received the Ph.D. degree from the University of Coimbra in 2010. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering of the University of Coimbra and a Researcher with Instituto de Telecomunicaes. His research interests include parallel computer architectures, GPU- and FPGA-based accelerators, energy-efficient processing, and compute-intensive signal processing applications related with communications and imaging. In 2011/12 and again in 2017/18 he was a Visiting Professor with EPFL, and in the summer of 2018 he was a Visiting Academic at ETHZ, Switzerland. Gabriel is a member of the IEEE Signal Processing Society, a Senior member of the IEEE and a full member the HiPEAC network of excellence.