

Received May 3, 2022, accepted May 22, 2022, date of publication May 25, 2022, date of current version June 1, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3177837

# Expert-Knowledge-Based Data-Driven Approach for Distributed Localization in Cell-Free Massive MIMO Networks

SIBREN DE BAST<sup>1</sup>, EVGENII VINOGRADOV<sup>1</sup>, (Member, IEEE), AND SOFIE POLLIN<sup>1,2</sup>

<sup>1</sup>Wavecore, ESAT, KU Leuven, 3001 Leuven, Belgium

<sup>2</sup>Imec, 3001 Leuven, Belgium

Corresponding author: Sibren De Bast (sibren.debast@kuleuven.be)

This work was supported in part by the Research Foundation Flanders (FWO) Strategic Basic (SB) Ph.D. Fellowship, under Grant 1SA1619N; and in part by the European Union's Horizon 2020 Research and Innovation Program, MARSAL Project, under Agreement 101017171.

**ABSTRACT** Massive Multiple-Input Multiple-Output (MaMIMO) communication networks are recently being investigated for their high potential for localisation services. This is enabled by the high-dimensional channel state information (CSI) captured by the many antennas in the system. Previously, it has been shown that these systems can achieve a very high localisation accuracy. However, many challenges still remain, we identified two of them. First, the recent trend towards cell-free MaMIMO with many highly distributed Access Points (AP), leads to the question of how this impacts the localisation methods. Current localisation methods process the signals in a central processing unit (CPU), resulting in a high fronthaul requirement when deploying these algorithms in a distributed network, limiting the deployment and scalability. Second, there exists a trade-off between using model-driven and data-driven localisation methods. In this work, we propose two new localisation methods which employ a distributed processing strategy and compare them against two centralised localisation methods. In addition, the four analysed methods explore the trade-off between being model- and data-driven. Moreover, the proposed ML-MUSIC method blurs the lines between the two by combining Machine Learning and traditional signal processing. Next to comparing the localisation accuracy, we evaluate the performance in a dynamic setting, the scalability and fronthaul requirement of the methods. The proposed Machine Learning-enhanced Multiple Signals Classification method, ML-MUSIC, reaches a median error of 34.2 mm on the test set while only using 500 training samples. Due to ML-MUSIC's distributed design, the fronthaul throughput requirement is reduced 1200-fold in comparison to the centralised methods. Furthermore, ML-MUSIC has the lowest computational complexity of all analysed methods, making it an ideal method to localise users in upcoming distributed cell-free MaMIMO networks.

**INDEX TERMS** Cell-free, massive MIMO, localization.

## I. INTRODUCTION

With the introduction of 5G, massive Multiple Input Multiple Output (MaMIMO) enabled communication systems will be deployed all around the world. Such MaMIMO systems are characterised by employing a large number of antennas at the base station (BS) to beamform the signal power towards the intended user [1]. In this way, multiple users can be served using the same time and frequency resource as they are multiplexed in the spatial domain. In order to effectively beamform towards the users, the channel state information (CSI) for each user has to be measured. This CSI

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Huo<sup>1</sup>.

contains all information about the wireless channel between the user and the BS and can be used to localise users. The challenge is now to estimate the position in the most effective way.

In [3], the authors envision that future distributed massive MIMO networks will enable six-dimensional localisation. In this vision, not only the 3D location will be estimated and provided by the cellular network, but also the orientation of the user. This opens interesting opportunities for unmanned aerial vehicles (UAVs), ground robots in industrial settings, autonomous cars, but as well other applications in the world of augmented and virtual reality. Ubiquitous six-dimensional localisation services will be a key technology linking the metaverse with the physical world.

### A. DISTRIBUTED LOCALISATION

The literature has seen a recent surge of research interest in localisation based on MaMIMO systems, resulting in highly accurate localisation methods. The main advantage of using MaMIMO systems for localisation is the high number of antenna elements at the base station. In [12], the authors show using a statistical channel model that the localisation performance increases with a rising number of antennas at the base station. In [10], we show the same trend, i.e. an increased localisation performance due to an increased number of base station antennas, while using a measured dataset.

However, next-gen MaMIMO communication systems are evolving towards distributed and cell-free configurations [2]. A cell-free system is divided in space by distributing the BS into multiple access points (AP) and spreading the APs over the targeted coverage area. It has been shown [4], [5], that the distribution of the antennas increases the spatial diversity, which decreases the channel correlation between users. As a result, inter-user interference decreases and the spectral efficiency of the system increases. These benefits are both true for outdoor and indoor scenarios. However, when adopting a different antenna configuration, the impact on the localisation performance has to be analysed.

Savic and Larsson [6] presented, based on simulations, that increased spatial diversity is beneficial for localisation purposes. Later, we showed a practical distributed antenna system that is indeed able to reliably locate users based on measured data [10]. Moreover, [7] and [11] show that for an indoor scenario, a distributed antenna deployment increases the localisation accuracy in comparison to co-located systems.

In addition, we have shown before [9] that when using a collocated antenna array, the localisation performance is strongly influenced by moving objects in the vicinity of the user. More specifically, when the line-of-sight (LoS) link between the user and the BS is blocked, the localisation accuracy reduces strongly. By distributing APs over an area, we increase the probability that at least some of the APs will have a LoS connection with the user. In general, when external factors influence the channel between the user and the BS antennas, a distributed system has the advantage that on average fewer links between the user and BS antennas are affected, resulting in a more reliable localisation system.

Previous methods processed the CSI centrally, which requires that the CSI is transmitted from each AP over a fronthaul link to a central processing unit (CPU). However, since the CSI is updated very frequently and a distributed MaMIMO system consists of many APs, the transmission of the CSI will require a high throughput of the fronthaul link. When this is the case, this may lead to increased latency of the position updates, resulting in potentially dangerous situations when the localisation service is used by autonomous drones or cars. A trade-off between latency and accuracy has to be made here. It can be solved by installing a higher capacity fronthaul link and increasing the processing power at the CPU, however, this induces an extra cost for the providers.

Therefore, in order to lower the required fronthaul throughput and computational requirements of the CPU, we propose to distribute the localisation task over the APs.

### B. MODEL-DRIVEN VERSUS DATA-DRIVEN LOCALISATION

When designing a localisation method based on a wireless communication system, multiple options can be considered. The literature classifies localisation algorithms into two categories [14]:

- 1) *Model-Driven*: The localisation method computes the location of the user based on how the channel is expected to behave based on channel models.
- 2) *Data-Driven*: The localisation method locates a user based on a channel dataset. The channels in the dataset need a positional label to link channels to locations. A data-driven localisation method links the channel of the user with channels in its dataset to estimate the user's position.

Between the two classes, there exists a trade-off, depending on the given system and targeted deployment. We will give a short overview of the two categories and discuss the advantages and disadvantages.

#### 1) MODEL-DRIVEN LOCALISATION

Model-driven localisation methods use the geometric properties of the communication system and the theoretical knowledge of the user's signal to triangulate the position of the user [18]. To do so, Time-of-Arrival (ToA), Time-Difference-of-Arrival (TDoA) or Angle-of-Arrival (AoA) can be used at the APs. ToA and TDoA estimate the distance between the user and the APs. AoA methods estimate the direction from which the received signal is originating.

In ToA systems, the time is measured between sending the signal at the user and receiving the signal at the receiver's antennas. The propagation speed is known, the speed of light, hence, the time difference can be used for ranging. When the range to different receivers is known, the user can be located using the geometry of the receivers. TDoA systems measure the time difference between receiving the signal at the different receivers. In this way, the source location of the signal can be calculated based on the location and time differences of the receivers. In ToA and TDoA based systems, accurate synchronisation in time is required. In addition, in multipath scenarios, it is difficult to extract a correct ToA [22].

When using AoA-based positioning methods, the received signal at an antenna array is used to compute the direction from which the signal originated. When multiple AoAs are estimated at different arrays, the exact origin of the signal can be calculated based on the geometry of the communication system. In [12], Garcia *et al.* localise the users by using the AoA, assisted by the ToA, measured at different distributed APs of a MaMIMO system.

In general, the advantage of model-driven methods is that they are very deterministic and require no training data, given that the antenna arrays are calibrated. However, the required

calibration is one of the biggest disadvantages of model-driven methods. Furthermore, in complex environments, e.g., where many reflections are present or users experience a non-line-of-sight connection to the APs, model-driven methods are inaccurate since these effects are hard to model.

## 2) DATA-DRIVEN LOCALISATION

Data-driven localisation methods rely on measured databases to estimate the location of a user. First, during an off-line step, a labelled database is gathered, containing information about the channel of the user. The information can consist, for example, of received signal strength indicators, power delay profiles or CSI. This information serves as the fingerprint of the user, hence, these methods are often referred to as fingerprinting techniques. Afterwards, in an online step, the measured fingerprint of a user is compared to the database to estimate the location. The localisation performance of data-driven methods is fundamentally limited by the size and label accuracy of the recorded database, a larger number of samples and a very accurate label are key.

For data-driven localisation methods, the biggest challenge is how you most effectively link the measured fingerprint with the fingerprints in the database. The authors of [20] use weighted K-nearest neighbour (WKNN) to localise a user in a MaMIMO network based on a power delay profile database. However, WKNN requires searching through big parts of the database, making it computationally inefficient. Furthermore, as a larger database is beneficial for accuracy, the practical use of methods that have to compare against the database is limited. In [6], Savic and Larsson proposed to use Machine Learning (ML), more specifically Gaussian Process Regression, to learn an unknown non-linear function to directly map the fingerprints to a location. Vieira *et al.* proposed in [13] the use of convolutional neural networks (CNN) in order to link the fingerprint of the user to the fingerprints in the database. The CNN learns the direct mapping between the fingerprint and the location of the user in an offline way. Afterwards, during the online step, the CNN can directly estimate the location of the user without accessing the database, resulting in a much more efficient online step.

Ever since Vieira *et al.* proposed the use of NNs for fingerprinting with MaMIMO CSI, many studies have adopted this technique. Arnold *et al.* showed a fingerprinting method using CNNs on measured data, including data of users in a non-line-of-sight scenario [14]. Ferrand *et al.* showed the performance in indoor and outdoor environments using a measured dataset, they explore the generalisation of the approach and data ageing [15]. In [16], Foliadis *et al.* combined NNs and careful feature design to reach cm-level accuracy in indoor environments.

The main drawback of using data-driven methods is that data-driven methods have to start learning how to locate a user without any prior knowledge. In practice, this translates into the need for a very large dataset. Gathering a large dataset containing both the wireless channel and the exact location of the corresponding user is very time consuming and, therefore,

an expensive process. As a result, the challenge for data-driven methods is to minimise the required size of the dataset, hence, improving the data efficiency.

## C. CONTRIBUTIONS AND OUTLINE

In this article, we study the localisation performance of a distributed MaMIMO system. We study the localisation accuracy, the computational complexity and the required fronthaul throughput of several proposed localisation methods. Our main contributions are:

- A new publicly available measured dataset, containing data of a nomadic environment;
- Introducing two new distributed localisation methods;
- A comparison of distributed and centralised localisation methods using multiple performance indicators;
- Exploring the advantages and disadvantages of model-driven and data-driven methods;
- The proposed ML-MUSIC method reduces the computational complexity and fronthaul requirements, while reaching a high localisation accuracy.

The paper is organised as follows: In Section II, the testbed used for the measurements is described as well as the measurement scenario. Section III introduces the different localisation methods used in this work. Afterwards, Section IV analyses the performance of the different methods. Next, in Section V, we delineate some implications on the privacy of the users using the proposed localisation methods. Finally, Section VI concludes this study.

## II. MEASURED MAMIMO DATASETS

In this section, the datasets used to study the performance of the different proposed methods are introduced. In order to study the localisation performance of different methods, a dataset of Massive MIMO channels is needed. To gather such a dataset, there are two possible options.

The first option is to simulate the channels. This can be done using various channel models or, preferably, using ray-tracing software. One of the advantages of using simulated data is the total control over the environment, which leads to a perfect knowledge of the ground truth. Having accurate positional labels is crucial when developing and evaluating localisation methods. Furthermore, when using simulated data, extra data is generated with ease, resulting in a possibly very large dataset.

The second option is to gather the data using a testbed and measuring real-life channels. Creating a large accurately labelled dataset is very hard with measured data. Measuring the channel at a large number of locations is very time consuming. Furthermore, acquiring an accurate positional label for such measurements is very hard. However, we argue that simulated data can not capture all the different non-idealities that we observe in the real world. Therefore, we decided to base this study on measured data. To do so, we have to overcome the challenges that such a measurement introduces.

First, we describe the distributed Massive MIMO testbed which we use for the measurements. Next, we delineate how

we extended the testbed in order to overcome the following challenges: *i*) Highly-accurate labels and *ii*) recording a very large dataset. To end this section, the two datasets collected using the testbed for this study are presented.

### A. DISTRIBUTED MASSIVE MIMO TESTBED

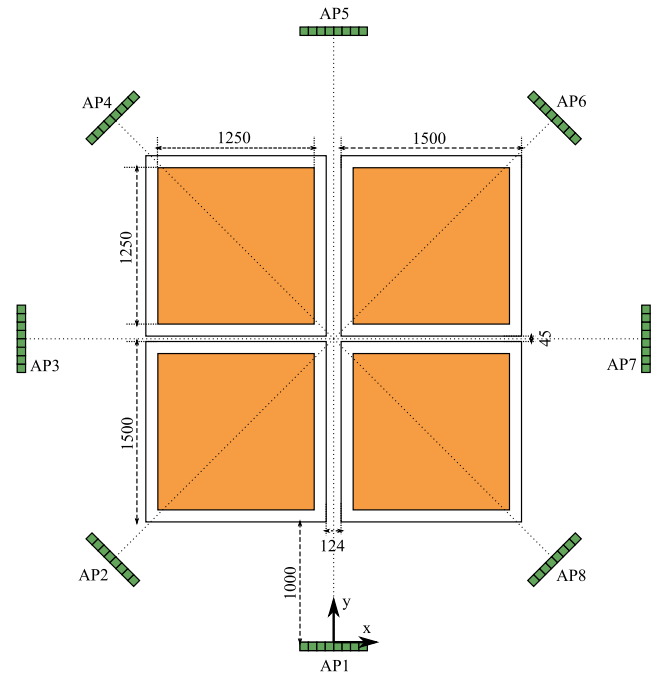
The KU Leuven Massive MIMO testbed is equipped with 64 patch antennas, that can be combined into an array by employing a flexible mounting system. This allows it to quickly change the antenna deployment from a centralised uniform rectangular array to a uniform linear array or a distributed array. When utilising the distributed array, the antennas are distributed into eight APs, each being a uniform linear array of 8 antennas.

For this study, the APs were placed in an octagonal shape around the region-of-interest (ROI), with the patch antennas facing the middle of the octagon. The users were placed inside the ROI. The placement of the distributed antennas is shown as the green rectangles on Fig. 1. This setup was build inside our lab, which is 6 m by 9.5 m. Around the ROI, the lab is very cluttered with desks, cabinets, measurement equipment and other miscellaneous object, resulting in an environment with a lot of scattering.

The testbed uses a time division duplexing frame structure, it switches between a down- and up-link phase. Since the channel is assumed to be reciprocal, we assume that the uplink and downlink CSI is the same. The testbed measures the channel by the use of uplink pilots. During a specific time slot, every 0.5 ms, each user sends a known uplink pilot. The base station receives these uplink pilots and estimates the channel by comparing the known pilot tones with the received pilot tones. The estimated channel is used to compute the combining and precoding vectors to enable MaMIMO communications. As a results, the CSI is already available for communication in the base station and can easily be used for localisation services without introducing extra overhead in the communication system. Afterwards, the CSI samples are saved to a hard disk at the base station during the collection of the dataset.

For each antenna, the testbed can measure the channel at 100 subcarriers. Therefore, while performing measurements, the measured channel has the form  $\mathbf{H} \in \mathbb{C}^{64 \times 100}$ . The measured data contains the I/Q sample of the channel for the corresponding antenna and subcarrier. The distance between subcarriers is 180 kHz, resulting in an effective bandwidth of 18 MHz. As a centre frequency, 2.61 GHz is used ( $\lambda = 11.49$  cm). The space between two antennas centres in the same subarray is 70 mm. These APs are attached to the BS by coax cables with a length of 8 m. This allows us to spread the APs over an indoor location. The antennas of the APs are placed at a height of 1 m above the floor.

Twelve users can be connected at the same time to the testbed. The user equipment is synchronised to the BS by the use of coax cables. This ensures a good time and frequency synchronisation between the devices. All devices are



**FIGURE 1.** Overview of the indoor measurement campaign. The 64 BS antennas (green boxes) are distributed over eight APs, each configured as a ULA, and were spread around the region-of-interest. Each of the four users scans an orange rectangle by the use of a CNC xy-table. Every 5 mm, the xy-table stops and the channel is recorded for the static user. In this way a highly accurate dense dataset is achieved, spanning a grid with 252,004 measurement points.

controlled using the NI MIMO application software running in LabView.

### B. CNC-POWERED MEASUREMENT EXTENSIONS

#### 1) ACCURATE POSITIONAL LABELS

In order to overcome the challenge of having accurate positional labels for our measured datasets, we focused our efforts in developing a robotically controlled automated measurement set-up. The general idea of the proposed set-up is to use Computer Numerical Control (CNC) XY-tables to move the antenna of the users. These machines have a sub-mm-accuracy when placing something in the XY-plane. In this way, the labels of the users locations are very accurate. For these experiments four CNC XY-positioners (OpenBuilds ACRO1515) were placed in the lab, each moving the antenna of one user. The antennas were connected to the corresponding devices using coax cables with a length of 5 m.

#### 2) LARGE AND DENSE DATASET

The second challenge to overcome when recording the dataset is the number of samples we can record. As moving the antenna and taking a measurement takes some time, doing this manually will take too much time, resulting in a dataset spanning a small area or with a very low density. Since the positioners can be controlled by a computer, an automated measurement set-up was developed. The four positioners are connected to a central control PC, which is able to send

commands and move the antennas to an arbitrary location inside the reach of the positioners. The LabView MIMO application framework running on the BS was extended to allow automated CSI measurements. This is done by sending TCP packets from the central PC to the BS. When a packet arrives, the BS takes one CSI snapshot for all connected users. The packet contains an identification number which is used to identify the samples of measured data. In this way, the correct positional label can be matched with the corresponding measured CSI sample. As a result, the testbed and XY-positioners can perform automated measurements, recording the CSI of users at defined positions with a high accuracy and without any human intervention.

### C. DATASETS

For this study, two different datasets are employed. The first one is a large and dense dataset recorded in a static environment where the users have an unobstructed LoS connection to the APs. This dataset is used to train or calibrate the employed methods. Furthermore, it is used to test the accuracy of the different methods with regards of the number of training/calibration samples used. The second dataset is recorded while a person was moving in the environment. This dataset is called the nomadic environment dataset. This dataset is used to test the methods in a changing environment. The moving person alters the channel of the users by changing the multipath. Furthermore, the moving person can even cause a loss of LoS with some APs. The two datasets are presented below in more detail are publicly available at IEEE Dataport.<sup>1</sup>

#### 1) DENSE CHANNEL SURVEY

The first dataset is a dense channel survey. By employing the CNC XY-tables, we recorded a very large dataset. We set the XY-tables to move over a grid with a grid-size of 5 mm and stop at each node of the grid for 0.5 s. While the XY-tables stopped, one CSI-sample was recorded by the APs for this position. The total grid spanned an area of 1.25 m by 1.25 m, resulting in 63,001 CSI-samples per user. We employed four users, therefore, this dataset contains the channel for 252,004 different locations in the region-of-interest. The full scenario for this measurement is shown in Fig. 1.

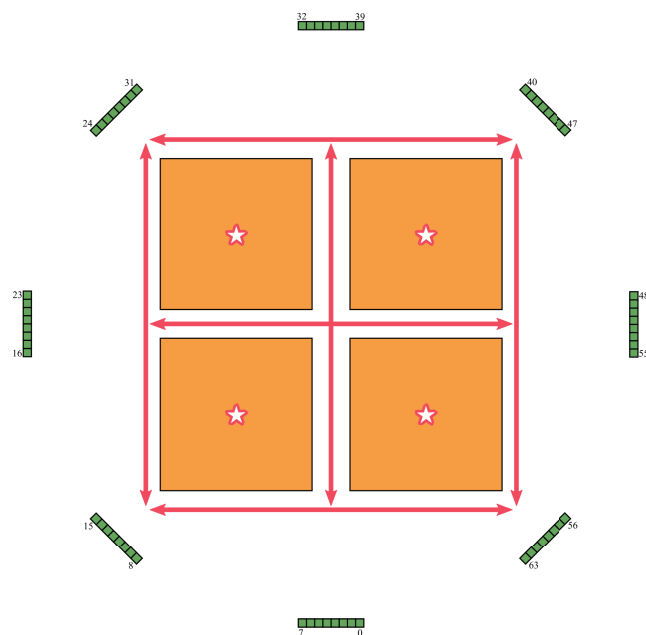
#### 2) NOMADIC ENVIRONMENT DATASET

When using ML to develop a localisation model, the accuracy is only as good as the dataset it was trained on. Therefore, when the CSI changes, for example by movement in the area of the users, and this affected CSI is not represented in the dataset, the localisation accuracy will suffer. To check the performance of the proposed methods in such nomadic environments, a second experiment was designed.

During this experiment, the users were put in static positions in the middle of their respective XY-table. Next, we recorded the CSI for two minutes with a half second

<sup>1</sup><https://iee-dataport.org/open-access/ultra-dense-indoor-mamimo-csi-dataset>

interval in between the CSI samples. While performing the measurements, a person was moving in the room, following predetermined paths. In total seven different scenarios were performed. During the first scenario, nothing was moving in the room, this measurement can be used as a reference to see how stable the localisation is over time when the scenario remains the same. During the other six scenarios, a person was walking back and forth along one of the edges of the positioners. The six paths can be seen in Fig. 2, depicted by the large pink arrows.



**FIGURE 2.** The scenario for the nomadic measurements. The users were placed at the middle position of the xy-tables, shown by stars on the figure. Next, a person walked back and forth along a given trajectory for two minutes while the channel for the user was captured every 0.5 s. In total six trajectories were recorded, shown by the red arrows on the figure. Also a reference measurement was taken when nothing was moving in the environment.

The nomadic measurement was performed right after the recording of the dense channel survey. This was done to ensure that the environment of the measurement campaign stayed exactly the same. The purpose of this dataset is to see if the proposed methods, which will be trained on the static data of the dense channel survey, are still valid when the environment slightly changes.

### III. LOCALISATION METHODS

This section outlines the different localisation methods proposed in this study. The methods are split into two categories. The first category, centralised localisation methods, delineates localisation methods which process the full CSI at the CPU. The second category, distributed localisation methods, handles the localisation task by partially distributing the processing of the CSI to the APs.

#### A. CENTRALISED LOCALISATION METHODS

Centralised localisation methods gather the CSI from every AP at a CPU, which combines the CSI to locate the user.

This leads to the full availability of the CSI, combining all the APs into one large array, therefore, moving the user to the near-field (NF) of the array [19]. The distance at which the near-field of an array stops, and the far-field begins, is often assumed to be the Fraunhofer distance

$$d_F = \frac{2D^2}{\lambda}.$$

The Fraunhofer distance depends on the size of the array  $D$  and the wavelength  $\lambda$  of the used carrier. When using only one AP, the ROI is located partially inside the NF region of that AP. However, when they are combined, the full ROI is located in the NF region of the array. Inside the NF region, the wavefront of the signal transmitted by the user has a spherical shape. Analysis of the curvature of this wavefront enables simultaneous ranging and AoA estimation. As a result, the location of the user can be estimated with high precision [17].

Since centralised localisation methods need all CSI at the CPU, they put a high required throughput on the fronthaul link. As cell-free MaMIMO networks scale, more APs will be deployed, resulting in more CSI that has to be sent to the CPU. As bigger antenna arrays lead to a higher localisation accuracy [10], being able to process the CSI from all these APs is beneficial. Therefore, the use of a centralised processing method, may limit the scalability of the system.

Here we outline two centralised localisation methods. The first method is data-driven, the second is model-driven.

### 1) END-TO-END MACHINE LEARNING

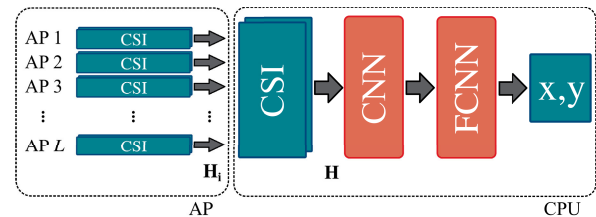
When using end-to-end (E2E) ML, a model learns all the steps between the input and the outputs of the model and all the different parts of the method are trained simultaneously. In recent years, E2E ML was proposed for position estimation based on the CSI of Massive MIMO systems. In a couple of years, the research field has made tremendous strides, going from validation work, based on simulations [13], to cm-level localisation accuracy on measured datasets [10], [14] [16].

The general idea for localisation with data-driven methods, is to feed the CSI measured at the APs to a neural network (NN) running at the CPU. The NN estimates the position of the associated user based on the CSI. To learn the mapping from CSI to a location estimate, the neural network is trained on a CSI database, labelled with corresponding positional labels. This is done during an online training phase. During the training phase, the NN learns the relevant features of the CSI to estimate the position of the user. The NN learns the direct relation between the CSI and the coordinates of the users.

E2E ML models that are currently used in literature gather the full CSI at a central node and process it in one NN. Therefore, we implemented such a centralised ML model as a benchmark. In order to achieve this, all APs send their CSI  $\mathbf{H}_i$  over the fronthaul link to the CPU. The CPU constructs the full CSI matrix  $\mathbf{H}$ , by joining the  $\mathbf{H}_i \in \mathbb{C}^{8 \times 100}$  matrices from each AP.

For this study, an ML model based on the combination of a convolutional neural network (CNN) and a fully connected

neural network (FCNN) was designed. The full implementation of this NN can be found in the code repository<sup>2</sup> accompanying this study. An overview of the architecture of this method can be seen in Fig. 3.



**FIGURE 3. End-to-end centralised machine learning architecture for distributed user localisation. The CSI is passed to the CPU which combines it and forwards it to a NN. The NN is a combination of a CNN and an FCNN. After an off-line training phase, the NN is able to directly estimate the position of the user based on the received CSI.**

### 2) CALIBRATED NEAR-FIELD MUSIC

Model-driven methods are based on how the channel is expected to behave and estimate the position of the users accordingly. This can be achieved by employing the well known MUSIC algorithm [23]. MUSIC is used to estimate the angle-of-arrival of a signal. Furthermore, since the ROI is in the near field of the antenna array, near-field (NF) MUSIC [24] can be adopted to estimate the exact location of the users. For NF MUSIC, all information needs to be processed in a centralised way, otherwise, the ROI would not be fully in the near field of the array.

Model-driven methods extract the location information from the theoretical knowledge of the signal and, therefore, expect a signal from a perfectly calibrated antenna array. In this case, the antenna array is not calibrated and the calibration parameters are not known. As a result, without extra processing to calibrate the data, NF MUSIC can not be used.

Traditionally, to calibrate an antenna array and estimate its calibration parameters, the array is placed in an anechoic chamber to measure the response of the array. However, by moving the array, the calibration can change due to small mechanical differences induced by the move. As a result, the calibration accuracy achieved is limited in practise. To combat this, [7] proposes a method to calibrate NF MUSIC using data measured during the dense channel survey presented in Section II.

Measuring the calibration parameters for an array is performed in an anechoic chamber for the reason that there is no multipath and the response can be measured perfectly. The main novelty of the method proposed in [7], is to use on-site measured data where a great deal of multipath is present. However, when using enough measured samples and applying mean-squared-error, the multipath components cancel each other out and the calibration parameters are correctly estimated. This calibration step is performed

<sup>2</sup><https://gitlab.kuleuven.be/networked-systems/public/localisation-in-distributed-mamimo-networks>

during an off-line step, before the system is used. Afterwards, NF MUSIC can estimate the position of the user very accurately.

## B. DISTRIBUTED LOCALISATION METHODS

Distributed localisation methods process the CSI at the APs before sending the extracted information to the CPU. Which processing is applied and which information is extracted depends on the used method. Next, the CPU estimates the exact location of the users based on the information received from the distributed APs. The goal is to compress the CSI by extracting relevant information that has a smaller dimension than the full CSI. In this way, less data has to be sent over the fronthaul link, lowering the throughput requirement of this link.

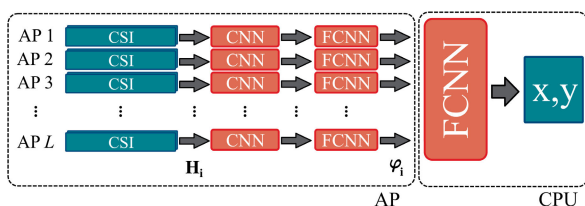
We outline two different distributed localisation methods. The first method, employs a fully data-driven approach. The second method combines data- and model-driven methods.

### 1) ML WITH DISTRIBUTED AoA LEARNING

Instead of sending the full CSI matrix from the AP to the CPU, first, the AoA is estimated at the AP. In this way, only the AoA for every AP has to be sent to the CPU, lowering the fronthaul requirement. Once the CPU receives all the AoA estimates, the location of the user can be estimated based on this information.

As shown in [8], the AoA of a user can be estimated by ML using the CSI of the user. To accomplish this, an NN is deployed at the AP to estimate the AoA of the users signal. This NN consists out of a CNN and an FCNN. It is trained using  $\mathbf{H}_i \in \mathbb{C}^{M \times N}$ , where  $i$  is the index of the AP,  $M$  is the number of antennas at the AP and  $N$  is the number of subcarriers in the CSI. The AoA  $\varphi_i$  is used as the label during the off-line training phase.  $\varphi_i$  is calculated based on the position and orientation of the  $i$ -th AP and the recorded location of the user.

Once the APs estimate the AoA  $\varphi_i$ , the estimate is transmitted over the fronthaul link to the CPU. At the CPU, each  $\varphi_i$  is gathered and combined in to the vector  $\varphi$ . Next, a simple FCNN is employed to estimate the location based on  $\varphi$ . The full compute graph for this method is shown in Fig. 4.



**FIGURE 4.** To distribute the localisation task, the APs estimate the AoA of the users signal in relation to the corresponding AP. Each AP uses its partial CSI  $\mathbf{H}_i$  to estimate the AoA  $\varphi_i$  using an NN. Next, all AoA estimations are gathered in a CPU to estimate the exact location of the user using an FCNN.

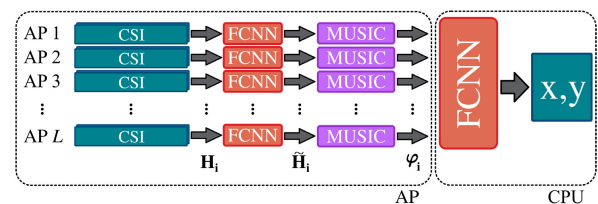
### 2) ML-MUSIC

In order to lower the amount of data samples needed to train the NNs, model-based expert knowledge can be included.

in order to do so, we use the MUSIC algorithm. However, this algorithm only works well when the antenna array is well calibrated. However, the array used for recording the channels is not calibrated.

#### a: CSI CALIBRATION

In order to use the MUSIC algorithm, a calibration step can be applied to the CSI. As mentioned before, this calibration step can be performed by analytical algorithms [7], [11]. However, these methods are far from trivial. Therefore, we saw an opportunity to employ neural networks to perform the calibration step. In this way, the CSI matrix  $\mathbf{H}_i \in \mathbb{C}^{M \times N}$ , with  $M$  the number of antennas at the AP and  $N$  the number of subcarriers, is first be passed through an autoencoder. An autoencoder is an NN that preserves the shape of the data. The autoencoder is trained to apply the calibration to  $\mathbf{H}_i$ . This results in the calibrated CSI,  $\tilde{\mathbf{H}}_i$ . Afterwards,  $\tilde{\mathbf{H}}_i$  is passed through the MUSIC algorithm to obtain the AoA vector  $\varphi_i$ . Once  $\varphi_i$  is estimated for every AP, they are combined into the vector  $\varphi$ . This vector is passed to another FCNN in order to estimate the exact location. Fig. 5 shows the overview of the ML-MUSIC compute graph.



**FIGURE 5.** Instead of only relying on data to train a NN and estimate the AoA, we can use some expert knowledge. Therefore, the MUSIC algorithm was implemented inside the NN framework. MUSIC is very effective in extracting the AoA from signals if the array is calibrated. Instead of calculating the calibration matrix using by traditional methods, a NN was placed in front of the MUSIC algorithm. This will make sure the a calibration step is performed on the CSI.

To calibrate the CSI, an NN was designed. For this application an FCNN was used. When building an autoencoder, it is easy to start from the simplest network first and gradually increase the depth and complexity of the network. However, we noticed that the proposed setup already worked really well by just employing an FCNN with one hidden layer. Therefore, for simplicity and minimisation of computational cost, this NN was kept.

#### b: APPLYING MUSIC

After calibration by the NN, MUSIC is applied to  $\tilde{\mathbf{H}}_i$ . In order for the NN to optimise its weights using back propagation, the MUSIC algorithm needs to be implemented in the ML framework. In this way, the derivative can propagate through the algorithm and the weights can be optimised using labelled CSI-samples. Furthermore, we propose to modify the MUSIC pseudo-spectrum and approximate it using a normalised range in order to map the ML labels to the pseudo-spectrum. Moreover, since ML-frameworks are optimised to run on graphical processing units (GPUs) and perform

matrix operations, we implemented the algorithm as much as possible to incorporate matrix operations instead of using code loops.

We will detail the implementation here. First, the covariance is calculated:

$$\mathbf{S}_i = \tilde{\mathbf{H}}_i \tilde{\mathbf{H}}_i^H.$$

Afterwards,  $\mathbf{S}_i$  is deconstructed using the eigenvalue decomposition

$$\mathbf{S}_i = \mathbf{U} \mathbf{\Gamma} \mathbf{U}^H.$$

The eigenvalues are located on the diagonal of  $\mathbf{\Gamma}$  in ascending order. The largest eigenvalues are attributed to the signal and the small ones to the noise. To extract the noise subspace, the noise eigenvectors are selected as

$$\mathbf{U}_n = \mathbf{U}_{(:,1:M-K)},$$

where  $K$  is the number of incident signals. We assume that there is only one signal in  $\mathbf{H}_i$ , therefore  $K$  is kept low. In this study, we assume  $K = 2$ . The eigenvectors in  $\mathbf{U}_n$  are normalised.

Next, the steering vector  $\mathbf{a}(\theta)$  is constructed as:

$$\mathbf{a}(\theta) = \left[ 1, e^{j\omega\theta}, e^{j2\omega\theta}, \dots, e^{j(M-1)\omega\theta} \right]^T,$$

where  $\omega = \frac{2\pi d}{\lambda}$ ,  $d$  is the spacing in between the antennas and  $\lambda$  is the wavelength of the carrier wave.

Using the steering vector, the MUSIC pseudo-spectrum can be evaluated for every arbitrary  $\theta$ . The angle  $\theta$  for which the pseudo-spectrum has the highest response is the estimated AoA  $\varphi_i$ . However, in practice, the number of angles along we can search for the highest response is limited. For this study,  $\theta$  ranges from 0 to 180 degrees with a 1 degree resolution, resulting in a vector with a length of 181. The vector containing these discrete angles is written as

$$\mathbf{b} = \left[ 0, \frac{1}{180}\pi, \frac{2}{180}\pi, \dots, \pi \right].$$

Afterwards, the pseudo-spectrum of MUSIC is calculated. This spectrum is also the output of the ML model at the APs. However, the traditional MUSIC algorithm does not have a pseudo-spectrum that is normalised to a specific range. Furthermore, supervised learning assumes that the exact label, i.e. the value of the pseudo-spectrum, is known in order to teach the NN this value. Since the exact value is not known for the MUSIC algorithm, we propose an alteration to the MUSIC algorithm to normalise the pseudo-spectrum. As a result, at the AoA, the pseudo-spectrum tends towards 1, and at the other angles it tends towards 0. In this way, we can label the training data by setting all angles to 0 except for the known AoA, which is set to 1, i.e., we one-hot encode the labels.

To explain the modification, we start from the unmodified pseudo-spectrum. The traditional MUSIC pseudo-spectrum is given by:

$$f(\theta) = \frac{1}{\mathbf{a}(\theta)^H \mathbf{U}_n \mathbf{U}_n^H \mathbf{a}(\theta)}.$$

When  $\theta$  is not the AoA of the signal  $\varphi_i$ , the denominator of  $f(\theta)$  will be maximised and can reach a value of maximum  $M$ . When  $\theta \approx \varphi_i$ , the denominator tends towards 0, hence, the pseudo-spectrum reaches a high response, with an unknown maximum value. The range that the spectrum can reach becomes defined when inverting the equation. The range becomes:

$$\begin{aligned} \theta \approx \varphi_i, \quad f(\theta)^{-1} &\Rightarrow 0, \\ \theta \neq \varphi_i, \quad f(\theta)^{-1} &\Rightarrow M. \end{aligned}$$

To normalise the range, a division by  $M$  is performed. As last step, the response is flipped by subtracting it from 1. The new pseudo-spectrum  $f$  is calculated for every  $\theta$  in  $\mathbf{b}$  as:

$$f(\theta) = 1 - \frac{\mathbf{a}(\theta)^H \mathbf{U}_n \mathbf{U}_n^H \mathbf{a}(\theta)}{M}.$$

As ML frameworks are optimised to run on GPUs, which prefer matrix operation, the full pseudo-spectrum  $f$  can be calculated for every  $\theta$  in one time using matrix operations. First, all steering vectors are combined in to the matrix  $\mathbf{A} \in \mathbb{C}^{181 \times M}$

Where the matrix  $\mathbf{A}$  is constructed as:

$$\mathbf{A} = \mathbf{a}(\mathbf{b}),$$

The new pseudo-spectrum  $f$  is calculated in the following way:

$$f = 1 - \frac{\text{diag}(\mathbf{A}^H \mathbf{U}_n \mathbf{U}_n^H \mathbf{A})}{M}.$$

and the  $\text{diag}()$  function selects the elements on the diagonal of the square input matrix and transforms them into a vector. The resulting pseudo-spectrum contains the response for every angle  $\theta$  in vector  $\mathbf{b}$ . The angle  $\theta$  with the highest response is selected as  $\varphi_i$ .

### c: COMBINING THE AoA ESTIMATIONS

For the next step, the CPU gathers all AoA estimates  $\varphi_i$  into the vector  $\varphi$ . Afterwards, another FCNN uses  $\varphi$  to estimate the position. The used FCNN consists out of 5 hidden layers. To train this network, a two step method is proposed. Since the task that the network should learn is very well defined, we can first train the NN on simulated model-based data. This has as the major advantage that we can simulate an infinite amount of model-based data, which will give the network a good starting point. This is very advantageous when only a low amount of training samples is available.

To create the model-based data, we use a geometry-based model of the environment. We sample random positions of the possible user positions and calculate  $\varphi$  based on the location and orientation of the APs. Next, we add some random errors to  $\varphi$  to emulate errors and blockages in the system. In this way, the FCNN learns how to deal with erroneous  $\varphi_i$  estimates. The number of angles  $\varphi_i$  that is affected by noise is decided by sampling integers from an exponential distribution with the rate parameter  $\lambda = 1$  and rounding this number to the nearest integer. Which of the AoAs  $\varphi_i$  are



affected is decided randomly using a uniform chance for each. To the selected angles we add a random angle, drawn from a normal distribution with a mean of  $0 \text{ rad}$  and a standard deviation of  $\frac{\pi}{4} \text{ rad}$ . In this way, the NN learns how to deal with noisy data that can contain errors.

In the second training step, we use the actual measured data, presented in Section II, to train the NN. This will teach the NN about the scenario specific imperfections and deviations from the theoretical deployment, further improving the accuracy of the model.

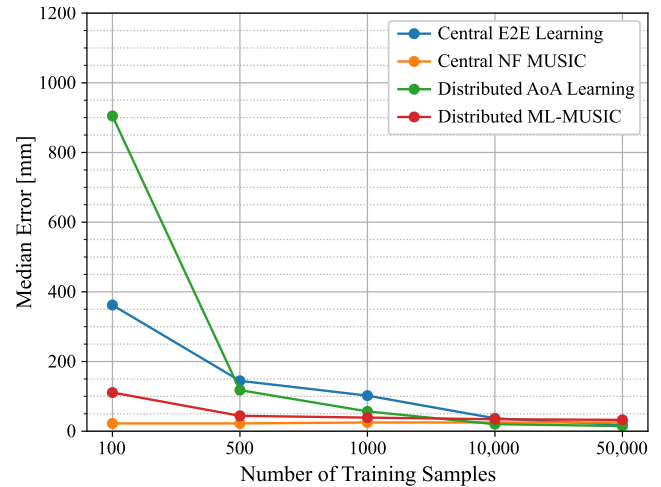
#### IV. PERFORMANCE ANALYSIS

In this section, the performance of the different proposed localisation methods is analysed. First, we study the localisation performance in a static environment. Second, we will analyse the performance in nomadic environments. In this case, there will be a moving person in the environment, interacting with the propagation of the signals sent by the users. Afterwards, we analyse the computational complexity of the different methods and how it scales for a rising number of APs in the communication system. To end, we calculate for the different methods the required fronthaul throughput.

##### A. PERFORMANCE IN A STATIC ENVIRONMENT

To analyse the performance of the four proposed methods, they are trained/calibrated on a couple of datasets, each with a different size. This was done to see which method is most data efficient, as gathering labelled data is very time consuming and, therefore, expensive. In total five training sets were created with sizes of 100, 500, 1000, 10,000 and 50,000 samples. These samples were randomly selected from the 252,004 labelled samples in the dense CSI dataset. At the same time, a test set of 20,000 randomly sampled CSI-samples was selected from the same dataset while making sure there is no overlap between the test set and the training sets. For each of the CSI-samples in the test set, the position was estimated by the four different methods after being trained on the different training sets.

The median localisation error is shown in Fig. 6. The median localisation error is computed by estimating the location for all samples in the test set and computing the error of these estimations. From these error values, the median was computed and reported. The first trend to notice is that all methods become better when more training samples are available. However, model-based methods require less data to achieve a good localisation performance. In general, when a low amount of samples is available in the training set, the two methods that (partially) rely on MUSIC perform best. This is due to the expert knowledge present in these methods. The algorithm did not have to learn from scratch how it can efficiently position users, as is the case with the methods that solely rely on ML. The pure data-driven methods only reach a similar performance when using 10,000 or more samples in the training set. However, when an even larger amount of data is available, the data-driven methods can perform better than the model-driven ones.



**FIGURE 6.** Localisation performance of the proposed methods in function of the size of the the training set. We can see a strong trend that when more data is used, the performance becomes better. Furthermore, model-driven methods need less data to train or calibrate the method. However, when ample data is provided, the data-driven methods match and even surpass the performance of model-driven methods.

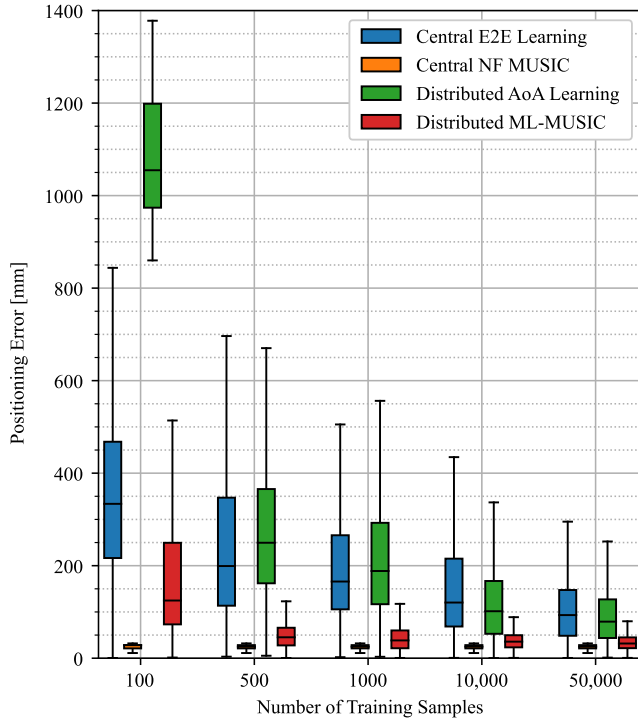
The figure shows that the calibrated NF MUSIC method has the best performance and already reaches a very high performance when only using 100 samples in the training set. The second best method is the ML-MUSIC algorithm, which reaches an adequate localisation performance with only 500 samples to train on. Both of these methods reach an accuracy lower than 50 mm while only using 500 training samples, which is 20 times less samples than required by the purely data-driven method. Therefore, these two methods have our preference from the view of sample efficiency. This shows the strength of model-driven methods. When a low amount of training data is provided, data-driven methods can not reach the same level of performance as model-driven methods.

##### B. PERFORMANCE IN A NOMADIC ENVIRONMENT

Next, these trained models were tested on the nomadic dataset presented in Section II. All CSI samples in the nomadic dataset were given to the four different methods. The full nomadic dataset was used here as the test set. The training sets are the same as before, randomly sampled from the dataset of the static environment. The results can be seen in Fig. 7.

First of all, the conclusions of the static environment are still true for the nomadic environment. As the number of training samples go up, the accuracy of the methods increases. Second, the accuracy for this test set is lower as for the one of the static environment, as would be expected since the CSI is impacted by the movement of the user and the methods are not trained/calibrated for this.

Surprisingly, the calibrated NF MUSIC method was not impacted by the movements of the user. This was the only method that was still able to estimate the users position with a very high accuracy due to the centralised combining and



**FIGURE 7.** Localisation performance in a nomadic environment of the proposed methods in function of the size of the the training set.

processing of the data. Moreover, the method never performs worse than 50 mm error, which is really impressive in comparison to the other methods. The second best method is ML-MUSIC, which is only slightly impacted by the nomadic environment. Again, both of these methods reached a median error of below 50 mm, while only using 500 training samples. Furthermore, the impact on the purely data-driven methods is higher than the impact on the methods that use model-based information.

These experiments showed that model-based approaches are still a very powerful tool. In addition, they show that centralised and distributed methods can both reach a high localisation performance. Hence, we do not have to send the full CSI to the CPU to reliably localise users in distributed MaMIMO systems.

### C. COMPUTATIONAL COMPLEXITY

To compare the different proposed methods in more depth, we assess the computational complexity. For a localisation system to function properly, this metric should be as low as possible, otherwise, it is not possible to estimate the location in a timely manner, which is important for future applications as autonomous vehicles and UAVs which require a low latency update of their location.

Since execution time is the most important aspect of computational when localising users, we will focus on the number of operations to be performed to localise a user. More specific, since multiplications are the most time consuming operation, we will count the number of multiplications that have to be performed in order to localise a user.

Furthermore, we also assess the scalability of the methods. The scalability of the system can be assessed in two ways. *i)* How the computational complexity scales with a growing number of APs and *ii)* how the complexity scales by a growing number of users to be located. Since all methods estimate the location for a user one-by-one, the computational complexity of the methods scales linearly with the number of users. Therefore, to compare the scalability of the different methods, the number of APs used for localisation is used as the main variable. We assume the network consists of a varying number of APs, each AP consisting of an array of eight antennas.

#### 1) CENTRAL E2E LEARNING

For the NNs, we can compute the computational complexity in following ways: For an FCNN, for every layer, the size of the input vector is multiplied by the number of hidden nodes in the layer. Afterwards, we sum these values for all the layers. For a CNN, the calculation becomes a bit more complex. For each layer, the computational complexity  $C$  can be calculated in the following way:

$$C_{CNN} = h_{in} \cdot w_{in} \cdot c_{in} \cdot k_h \cdot k_v \cdot c_{out}$$

with  $h_{in}$  the height,  $w_{in}$  the width, and  $c_{in}$  the number of channels of the input of the layer.  $k_v$  and  $k_h$  are the vertical and horizontal kernel size and  $c_{out}$  the number of output channels.

For the Central E2E Learning method, we have to compute the computational complexity by applying the above formulas for both the CNN and FCNN. In this case  $h_{in}$  depends on the number of used antennas, therefore, it will scale with the number of APs. Furthermore, also the FCNN to estimate the position based on the AoAs depends on the number of APs, however, here only the first layer will increase in size. As a result, the computational complexity of Central E2E Learning increases linearly with the number of APs.

#### 2) CENTRAL NF MUSIC

When computing the complexity of the centralised calibrated NF MUSIC method, we have to look at all the different algebraic operation that are being performed. First the covariance of the CSI is calculated, its complexity corresponds to the one of a matrix multiplication, resulting in a complexity of  $\mathcal{O}(M^2N)$ . Next, the eigenvalues should be calculated, this has a complexity of  $\mathcal{O}(M^3)$ . All following computations have to be applied to a steering vector. For NF MUSIC, there is a steering vector for each possible user location. We set the number of steering vectors to  $n$ . when calibrating the steering vectors,  $\mathcal{O}(MN)$  multiplications have to be performed. Afterwards, these steering vectors have to be multiplied with the noise space with dimension  $M - 1$ , computed using the eigenvalue decomposition, this results in a complexity of  $\mathcal{O}(M(M - 1))$ .

As a result, the computational complexity of the centralised calibrated NF MUSIC can be calculated as:

$$C = M^2N + M^3 + n(MN + M(M - 1)), \quad (1)$$

with  $M$  the number of antennas,  $N$  the number of subcarriers and  $n$  the number of positions that have to be evaluated. In [7], this was set to all possible user locations, i.e.  $n = 252,004$ . The computational complexity is strongly influenced by the number of antennas in the antenna array and the number of locations to be evaluated. When the number of antennas is increased, the computational complexity of this method rises in a cubic manner, strongly limiting the scalability of the method.

### 3) DISTRIBUTED AoA LEARNING

For the distributed AoA learning method, the CNN architecture is the same as the one for the central E2E learning, however, it is distributed per AP. Therefore, the complexity of the CNN at each AP is the same as the complexity of the central E2E for one AP. This complexity is multiplied by the number of APs, therefore, scaling linearly. Also the FCNNs of the distributed AoA learning method and the central E2E learning method are mostly the same. As a result, they have the same complexity and scale in the same way. However, in this case, the computations are distributed over the different APs and the CPU in the network, lowering the per device requirement of computation power.

### 4) DISTRIBUTED ML-MUSIC

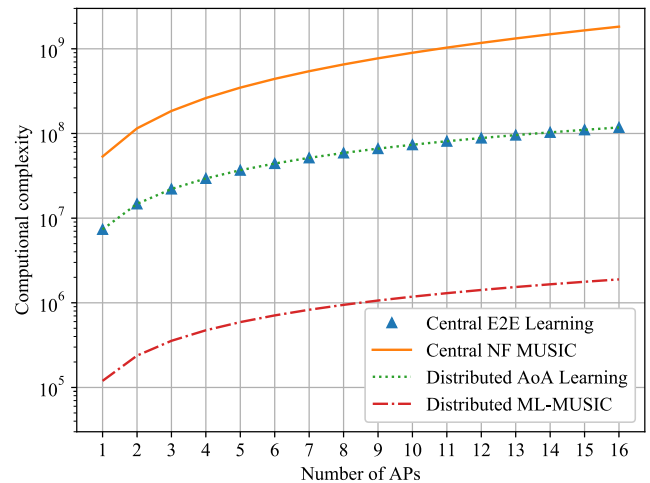
For the computational complexity of the distributed ML-MUSIC a comparable formula to equation (1) can be found for the complexity of the MUSIC block in the compute-graph:

$$C = N_{AP}(M_{AP}^2 N + M_{AP}^3 + n(M_{AP}(M_{AP} - 1))).$$

However, here we have to multiply everything by  $N_{AP}$ , the number of APs, but the number of antennas per AP,  $M_{AP}$ , is lower and fixed. Since  $M_{AP}$  is fixed and only  $N_{AP}$  rises when increasing the number of APs, the complexity rises in a linear way for this method. Furthermore, in this case,  $n$  is the number of angles we search along the array, in this study,  $n = 181$ . In comparison to NF MUSIC,  $n$  is much lower for ML-MUSIC, as in NF MUSIC  $n = 252,004$ . To complete the calculation, the complexity of the calibration NNs and of the NN to estimate the location, have to be added. For the calibration, a one-layer-deep FCNN is used per AP, therefore, the complexity of this rises linearly with the number of APs. Next, the other FCNN to combine the multiple AoAs to a 2D position also depends on the number of APs, but only the size of the input changes, increasing the complexity in a linear way. As a result, the total complexity of the ML-MUSIC method increases linearly with the number of APs.

### 5) COMPARISON OF THE COMPUTATIONAL COMPLEXITY

Fig. 8 shows the computational complexity for the four different evaluated methods in function of the number of APs. Each AP is equipped with eight antennas. So the case of eight APs corresponds with the one in this paper, where we use data of 64 antennas, divided over 8 distributed antenna arrays. The figure shows that Central NF MUSIC has by far the



**FIGURE 8. Computational complexity: the number of multiplications needed to estimate the position of one user for the different localisation methods. We analysed this amount in function of the number of APs. We assume an AP to have an antenna array with eight antennas. The scalability of Central NF MUSIC is cubic, therefore, problematic for a practical deployment, while Distributed ML-MUSIC scales only linearly.**

highest computational complexity, followed by Central E2E learning and Distributed AoA Learning. In the case of 16 APs, ML-MUSIC reduces the computational complexity by a factor 1000 in comparison with NF MUSIC.

Furthermore, the computational complexity of Central NF MUSIC also rises faster than the other methods, cubic in comparison to linear. As a result, calibrated NF MUSIC becomes an impractical method for future cell-free Massive MIMO networks. Note that the number of evaluated positions,  $n$  in equation 1, is kept constant in this figure. When the cell-free network grows and the number of APs rises, also the number of possible locations rises, increasing the complexity even further.

Note that the value shown in Fig. 8 is the total amount of computations that has to be performed. However, for the distributed methods, these computations are further divided over the distributed APs and the CPU, resulting in an even lower amount of computations per device. This in comparison to the centralised methods, which need all the computations to be performed at the CPU, putting a high requirement on a single device.

### D. FRONTHAUL REQUIREMENT

For the localisation methods to work, information has to be shared by the different APs to a central node. For the two centralised methods, all CSI is needed to be transmitted to the CPU. For the distributed methods, only a small amount of information is needed to be shared. In this part of the study, we will calculate the required information throughput from the APs to the CPU for the proposed methods.

We will base this analysis on the KU Leuven MaMIMO testbed presented in Section II. The presented MaMIMO system has 64 antennas and measures the channel for each

user for 100 subcarriers. Each measured symbol consists of a complex number, using 16 bit for the real and imaginary part. Therefore, the number of bytes needed for one sample for one user is equal to 25.6 kB. In the testbed, which is based on the LTE framework, the channel is measured 2000 times per second for up to 12 users.

For the distributed methods only the AoA is sent to the CPU. Since the AoA is discretised at 1 degree per step, and we consider a range from 0 to 180 degrees, 181 different values should be encoded. To do this, we need 8 bits or 1 byte. Therefore, for the studied system, 8 bytes are needed to be sent to the CPU per user per CSI update. As a result, the required fronthaul throughput for the central methods is 4.915 Gbps. While for the distributed methods, this is reduced to 1.536 Mbps. This is a 1200-fold reduction of the required throughput. The reduced fronthaul requirements of distributed localisation methods will enable a practical deployment of localisation services in cell-free MaMIMO systems.

## V. PRIVACY-CENTRIC LOCALISATION

There are rising concerns about user privacy in recent times as more people start to be aware that a big part of their digital life can be tracked. This has even given rise to multiple conspiracies about that 5G will be used to monitor and control persons. Therefore, it is important to address these concerns and think about how we will keep the location of the users safe. To do this, one can employ the principles of “privacy by design” [25].

When processing the information in one central place, e.g. a server connected to the BS, the mobile internet service provider will have the full knowledge of the exact location of the user. This might be vulnerable for malicious attacks of hackers stealing the valuable data. Furthermore, exact localisation data could be used for surveillance by governments that do not respect civil rights.

As we have shown, distributed localisation methods achieve a high accuracy while reducing the computational complexity and fronthaul requirement. This allows use to move the CPU that combines the information of the separate APs to be moved to the user equipment itself. In this case, the full information will only be available at the device of the user, keeping its exact location for itself, increasing the privacy of the user. In conclusion, we identify privacy-centric localisation as an interesting topic for future research.

## VI. CONCLUSION

In this study, we compared four user localisation methods based on a distributed MaMIMO network. We compared methods that process all data centrally and methods that off-load part of the computation to the distributed APs. Second, we compared model-driven and data-driven methods. The localisation performance of the different methods is compared using measured data from two scenarios. The computational complexity and fronthaul requirement were computed.

By employing distributed localisation methods, the fronthaul requirement was reduced 1200-fold in comparison to the centrally processed methods, making the technology scalable for future networks with tens of APs. In addition, the computational complexity of the distributed methods was the lowest of the analysed methods, reducing the computational requirements of the distributed APs and CPU. In addition, to achieving a localisation accuracy below 50 mm, by including model-driven techniques, we were able to reduce the amount of training samples needed by a factor of 20 in comparison to using purely data-driven methods.

We found that the proposed distributed ML-MUSIC algorithm, which combines machine learning and MUSIC to localise the users, is the best candidate for the presented task and scenario, as it is able to reach accurate localisation while using a moderate amount of training samples. It reaches an accuracy of 34.2 mm when only using 500 training samples. When using more training samples, the accuracy further improves. Furthermore, at the same time, it keeps the computational complexity and fronthaul requirement to a minimum, making it an ideal candidate to be used in large distributed MaMIMO networks.

## REFERENCES

- [1] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, “Massive MIMO for next generation wireless systems,” *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014, doi: [10.1109/MCOM.2014.6736761](https://doi.org/10.1109/MCOM.2014.6736761).
- [2] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, “Cell-free massive MIMO versus small cells,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1834–1850, Mar. 2017, doi: [10.1109/TWC.2017.2655515](https://doi.org/10.1109/TWC.2017.2655515).
- [3] E. Björnson, L. Sanguinetti, H. Wymeersch, J. Hoydis, and T. L. Marzetta, “Massive MIMO is a reality—What is next?” *Digit. Signal Process.*, vol. 94, pp. 3–20, Nov. 2019, doi: [10.1016/j.dsp.2019.06.007](https://doi.org/10.1016/j.dsp.2019.06.007).
- [4] L. Van der Perre, E. G. Larsson, F. Tufvesson, L. D. Strycker, E. Björnson, and O. Edfors, “RadioWeaves for efficient connectivity: Analysis and impact of constraints in actual deployments,” in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Nov. 2019, pp. 15–22, doi: [10.1109/IEEECONF44664.2019.9048825](https://doi.org/10.1109/IEEECONF44664.2019.9048825).
- [5] A. P. Guevara, S. De Bast, and S. Pollin, “Weave and conquer: A measurement-based analysis of dense antenna deployments,” in *Proc. IEEE Int. Conf. Commun.*, Jun. 2021, pp. 1–6, doi: [10.1109/ICC42927.2021.9500612](https://doi.org/10.1109/ICC42927.2021.9500612).
- [6] V. Savic and E. G. Larsson, “Fingerprinting-based positioning in distributed massive MIMO systems,” in *Proc. IEEE 82nd Veh. Technol. Conf. (VTC-Fall)*, Sep. 2015, pp. 1–5, doi: [10.1109/VTCFall.2015.7390953](https://doi.org/10.1109/VTCFall.2015.7390953).
- [7] Y. Pan, S. De Bast, and S. Pollin, “Indoor direct positioning with imperfect massive MIMO array using measured near-field channels,” *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021, doi: [10.1109/TIM.2021.3067963](https://doi.org/10.1109/TIM.2021.3067963).
- [8] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, “Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8549–8560, Sep. 2018, doi: [10.1109/TVT.2018.2851783](https://doi.org/10.1109/TVT.2018.2851783).
- [9] S. De Bast and S. Pollin, “MaMIMO CSI-based positioning using CNNs: Peeking inside the black box,” in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2020, pp. 1–6, doi: [10.1109/ICCWorkshops49005.2020.9145412](https://doi.org/10.1109/ICCWorkshops49005.2020.9145412).
- [10] S. D. Bast, A. P. Guevara, and S. Pollin, “CSI-based positioning in massive MIMO systems using convolutional neural networks,” in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–5, doi: [10.1109/VTC2020-Spring48590.2020.9129126](https://doi.org/10.1109/VTC2020-Spring48590.2020.9129126).
- [11] C. Li, S. De Bast, E. Tanghe, S. Pollin, and W. Joseph, “Toward fine-grained indoor localization based on massive MIMO-OFDM system: Experiment and analysis,” *IEEE Sensors J.*, vol. 22, no. 6, pp. 5318–5328, Mar. 2022, doi: [10.1109/JSEN.2021.3111986](https://doi.org/10.1109/JSEN.2021.3111986).

- [12] N. Garcia, H. Wymeersch, E. G. Larsson, A. M. Haimovich, and M. Coulon, "Direct localization for massive MIMO," *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2475–2487, May 2017, doi: [10.1109/TSP.2017.2666779](https://doi.org/10.1109/TSP.2017.2666779).
- [13] J. Vieira, E. Leitingner, M. Sarajlic, X. Li, and F. Tufvesson, "Deep convolutional neural networks for massive MIMO fingerprint-based positioning," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–6, doi: [10.1109/PIMRC.2017.8292280](https://doi.org/10.1109/PIMRC.2017.8292280).
- [14] M. Arnold, S. Dorner, S. Cammerer, and S. Ten Brink, "On deep learning-based massive MIMO indoor user localization," in *Proc. IEEE 19th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun. 2018, pp. 1–5, doi: [10.1109/SPAWC.2018.8446013](https://doi.org/10.1109/SPAWC.2018.8446013).
- [15] P. Ferrand, A. Decurgingne, and M. Guillaud, "DNN-based localization from channel estimates: Feature design and experimental results," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6, doi: [10.1109/GLOBECOM42002.2020.9348191](https://doi.org/10.1109/GLOBECOM42002.2020.9348191).
- [16] A. Foliadis, M. H. C. Garcia, R. A. Stirling-Gallacher, and R. S. Thoma, "CSI-based localization with CNNs exploiting phase information," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2021, pp. 1–6, doi: [10.1109/WCNC49053.2021.9417561](https://doi.org/10.1109/WCNC49053.2021.9417561).
- [17] A. Guerra, F. Guidi, D. Dardari, and P. Djuric, "Near-field tracking with large antenna arrays: Fundamental limits and practical algorithms," *IEEE Trans. Signal Process.*, vol. 69, pp. 5723–5738, 2021, doi: [10.1109/TSP.2021.3101696](https://doi.org/10.1109/TSP.2021.3101696).
- [18] O. Alamu, B. Iyaomolere, and A. Abdulrahman, "An overview of massive MIMO localization techniques in wireless cellular networks: Recent advances and outlook," *Ad Hoc Netw.*, vol. 111, Feb. 2021, Art. no. 102353, doi: [10.1016/j.adhoc.2020.102353](https://doi.org/10.1016/j.adhoc.2020.102353).
- [19] E. Björnson, Ö. T. Demir, and L. Sanguinetti, "A primer on near-field beamforming for arrays and reconfigurable intelligent surfaces," 2021, *arXiv:2110.06661*.
- [20] X. Sun, X. Gao, G. Y. Li, and W. Han, "Single-site localization based on a new type of fingerprint for massive MIMO-OFDM systems," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6134–6145, Jul. 2018, doi: [10.1109/TVT.2018.2813058](https://doi.org/10.1109/TVT.2018.2813058).
- [21] K. J. Krizman, T. E. Biedka, and T. S. Rappaport, "Wireless position location: Fundamentals, implementation strategies, and sources of error," in *Proc. IEEE 47th Veh. Technol. Conf. Technol. Motion*, vol. 2, May 1997, pp. 919–923, doi: [10.1109/VETEC.1997.600463](https://doi.org/10.1109/VETEC.1997.600463).
- [22] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Z. Win, "Ranging with ultrawide bandwidth signals in multipath environments," *Proc. IEEE*, vol. 97, no. 2, pp. 404–426, Feb. 2009, doi: [10.1109/JPROC.2008.2008846](https://doi.org/10.1109/JPROC.2008.2008846).
- [23] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. AP-34, no. 3, pp. 276–280, Mar. 1986, doi: [10.1109/TAP.1986.1143830](https://doi.org/10.1109/TAP.1986.1143830).
- [24] Y. D. Huang and M. Barkat, "Near-field multiple source localization by passive sensor array," *IEEE Trans. Antennas Propag.*, vol. 39, no. 7, pp. 968–975, Jul. 1991, doi: [10.1109/8.86917](https://doi.org/10.1109/8.86917).
- [25] A. Cavoukian, "Privacy by design," Office Inf. Privacy Commissioner, ON, Canada, Tech. Rep., 2009.



**SIBREN DE BAST** received the M.Sc. degree in electrical engineering from KU Leuven, Leuven, Belgium, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Electrical Engineering (ESAT). He is focusing on massive MIMO communications for aerial and ground robots, for which he created an experimental test and measurement environment. His current research interests include artificial intelligence for communications and localization.



**EVGENII VINOGRADOV** (Member, IEEE) received the Dipl.-Eng. degree in radio engineering and telecommunications from Saint Petersburg Electrotechnical University, Russia, in 2009, and the Ph.D. degree from UCL, Belgium, in 2017. After several years of working in the field of mobile communications, he joined UCL, in 2013. His doctoral research interests focused on multidimensional radio propagation channel modeling. In 2017, he joined the Electrical Engineering Department, KU Leuven, Belgium, where he is working on wireless communications with UAVs and UAV detection.



**SOFIE POLLIN** is a Professor with KU Leuven focusing on wireless communication systems. Before that, she worked at imec and UC Berkeley. Her research interests include wireless networks that require networks that are ever more dense, heterogeneous, battery powered, and spectrum constrained. She pioneered a 5G testbed for distributed Massive MIMO at KU Leuven and is currently leading the way toward 6G tests in multiple large EU projects.

...