Localisation in Wireless Networks using Deep Bidirectional Recurrent Neural Networks

David Lynch*, Lester Ho†, Michael MacDonald‡, and Michael O'Neill§
*§Natural Computing Research & Applications Group
School of Business
University College Dublin
Email: * david.lynch@ucd.ie,§ m.oneill@ucd.ie

†‡ Nokia Bell Laboratories
Email: † lester.ho@nokia-bell-labs.com, ‡ mike.macdonald@nokia-bell-labs.com

***Abstract*—The accurate localisation and tracking of objects is crucial in many domains. In this paper, we focus on location tracking in wireless networks. Reliable localisation will be essential for self-driving, future factories, and beamforming in 5G deployments. Time-of-arrival (TOA) based localisation systems use synchronised nodes to receive radio signals sent by transmitters at the object to be located. The time it takes for the signal to propagate in a straight line to the receiver is used to trilaterate the position of the transmitter. However, the correct TOA can be difficult to identify due to multiple reflected copies of the same signal arriving at different times. This paper presents a bidirectional recurrent neural network (BiRNN) for estimating the TOA from the channel impulse response (CIR) of the signal, and a multilayer perceptron (MLP) to trilaterate the position of the transmitter from the TOA. The BiRNN and MLP are trained using measured CIR data, and outperform the conventional approaches for TOA estimation and trilateration.**

I. Introduction

Accurate positional tracking of people and objects is crucial in commercial and industrial applications [6][10], and in sport's analytics [4]. In recent years, there has been a push towards the use of advanced technologies such as robotics and augmented reality to increase automation in industrial settings like factories, warehouses, agriculture and utilities. This digitisation of industry, given the term "Industry 4.0" [15], has brought forward the need for highly accurate localisation.

Localisation based on the reception of radio frequency signals is one of the most common approaches. The location of a wireless transmitter can be inferred based on signal strength [2][7], calculating the signal's angle of arrival using antenna arrays [9], or obtaining timing information to calculate the signal's propagation time [5][11].

Localisation using timing information is often employed to achieve the high accuracy needed in Industry 4.0 scenarios. In this approach, the signal propagation time (also called the time of flight, or time of arrival) is used to calculate the distances between a transmitter (the object to locate) and several receivers. The physical distance between the transmitter and a receiver can be inferred simply by multiplying the time of flight with the speed of light, c. Given at least three reference distances, trilateration can be used to estimate the transmitter's location, by determining the intersection of the three or more spheres with radii equal to the reference distances.

Two factors impact the accuracy of TOA based localisation systems. Firstly, localisation accuracy is bounded by the signal sampling rate of the receivers. A low sampling rate reduces the TOA resolution, as the signal may arrive in between sampled intervals. A second source of inaccuracy is due to the effects of multipath propagation. A wireless signal from the transmitter can be reflected off different surfaces before arriving at the receiver. Consequently, multiple copies of the signal arrive at different times. For TOA localisation, the signal that travels directly from the transmitter to receiver gives us the correct distance estimation. This direct signal is referred to as the first arriving path. However, a time-delayed copy of a reflected signal can be incorrectly identified as the first arriving path, and this misidentification is especially likely when the late arriving path has a strong signal strength.

In order to achieve high accuracies, the issues of low sampling and multipath propagation must be addressed when performing TOA estimation. The main contribution of this paper is a data-driven solution based on deep learning. We tackle the TOA estimation problem using deep bidirectional recurrent neural networks (BiRNNs) [14]. RNNs [13] have been successfully applied to time series classification [8] and natural language understanding [1]. However, the utility of a BiRNN for TOA estimation has not been explored in previous work. The BiRNN is augmented with a denoising multilayer perceptron for location trilateration. The combined system is more accurate and robust than conventional approaches.

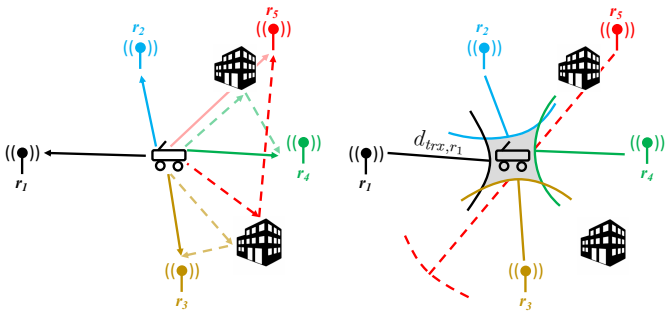
The rest of the paper is organised as follows. The coupled problems of TOA estimation and trilateration are defined in Section II. Conventional heuristics for localisation are reviewed in Section III, and the neural models

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

are specified. Experiments carried out using data from a real deployment scenario are outlined in Section IV. The results presented in Section V illustrate the suitability of deep neural networks for localisation. Finally, the paper concludes with directions for future work in Section VI.

II. Problem Definition

Consider the toy network depicted in Fig. 1. The environment could be a factory, sports arena, or industrial park. The transmitter could be a robot on the factory floor, a tracker carried by an athlete, or a smartphone in an area without GPS coverage. Our goal is to locate the transmitter in real time.



(a) Enterprise Environment. (b) Trilateration.

Fig. 1: In plot (a), signals from the transmitter reach receivers $r_1, r_2, \ldots, r_5$ directly (solid arrows), and also via indirect paths (dashed arrows). The travel time along the direct or 'first arriving' path implies the distance between the transmitter and receiver. In plot (b), the transmitter's location is estimated by trilateration.

Receivers are installed at fixed points in the environment. They are labelled $r_1, r_2, \ldots, r_5$ in Fig. 1. The transmitter sends five unique signals, one for each receiver, at intervals of one second. At the receiver, the correlation of the received signal against the sent signals is performed. The result is five channel impulse response (CIR) time series, each with 250 values spanning $250 \times 16.27 = 4067.5$ [ns]. Large peaks in the CIR occur when the expected signal is received. Exemplary CIR time series for receivers $r_1, r_2, \ldots, r_5$ are visualised in Fig. 2.

For example, the leftmost plot in Fig. 2 peaks after approximately 927 [ns]. This peak occurs when the signal produced by the transmitter for receiver r_1 arrives at r_1. Therefore, the signal travelled a distance from the transmitter to r_1 given by:

$$d_{trx,r_1} = 299792458 \text{ [m/s]} \times 0.000000927 \text{ [s]} \approx 278 \text{ [m]}, \quad (1)$$

where $c = 299792458$ [m/s] is the speed of light. The transmitter can be located given its distance from at least three receivers. The arcs in Fig. 1b illustrate where the transmitter could be located relative to individual receivers. The shaded region enclosed by the arcs contains the transmitter.

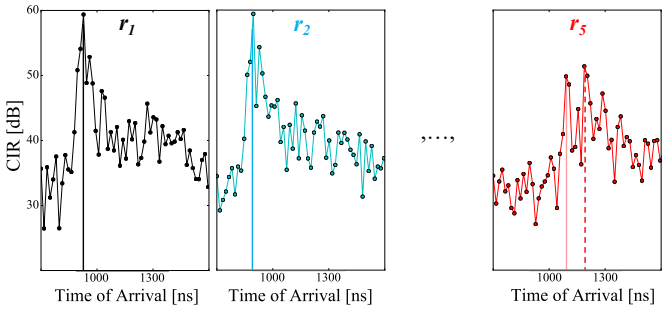


Fig. 2: Receivers listen for signals sent by the transmitter. Partial channel impulse response (CIR) time series for receivers $r_1, r_2, \ldots, r_5$ are displayed. Peaks occur when the signal produced for receiver r_i reaches r_i. Solid lines indicate when a signal is first received. The red dashed line indicates a stronger secondary peak associated with a reflected copy of a signal.

Three main sources of error preclude all five arcs in Fig. 1b from intersecting at a single point:

1) Imperfect surveying may render the receiver locations inaccurate.
2) The sampling period of the receivers is 16.27 nanoseconds, so the received signal will typically not coincide with a peak in the CIR. Any algorithm that does not interpolate between peaks will have its time resolution limited by this sampling period, which translates to a ranging distance resolution of 4.88m.
3) Lastly, radio signals emitted by the transmitter may take multiple indirect paths to a receiver. The dashed arrows in Fig. 1a show how some signals are reflected off buildings before reaching the receivers. These reflected copies manifest as secondary peaks in Fig. 2 (dashed red line). The secondary peak measured at receiver r_5 (rightmost plot) is stronger than the first peak. An algorithm that naively selects the strongest peak would overestimate the distance between the transmitter and r_5 (red dashed arc in Fig. 1b).

In summary, the transmitter's location can be estimated given its distance from each receiver and the receiver locations. Distances are inferred from the time of arrival (TOA) of signals sent by the transmitter. TOA estimation is non-trivial because the peak associated with the first arriving path may be obscured by stronger secondary peaks. Furthermore, predicting the x,y,z coordinates of the transmitter is challenging because the estimated TOAs are noisy. Our goal is to learn TOA estimation and trilateration algorithms from data. The neural network models are introduced in the next section.

III. Methods

Benchmark algorithms for localisation are described in this section. The ad-hoc manually designed benchmarks are not tailored to the deployment context, resulting in

suboptimal performance. This motivates a more flexible data-driven approach based on deep learning. Recurrent neural networks for time of arrival (TOA) estimation are presented. Finally, a multilayer perceptron is proposed for mapping the estimated TOAs to the transmitter's location.

A. Benchmark TOA Estimation Algorithm

The steps of the benchmark algorithm for estimating TOAs are illustrated in Fig. 3. All peaks (red boxes) in the CIR are first identified. Peaks are defined as points such that both neighbouring points have a lower value. The highest peak is then marked (peak 4). The TOA corresponding to this peak is returned if no earlier peak is within a small threshold α [dB], otherwise the TOA of the earlier peak (peak 2) is returned. Thus, the 'find peaks' algorithm attempts to identify the peak associated with the first arriving path.

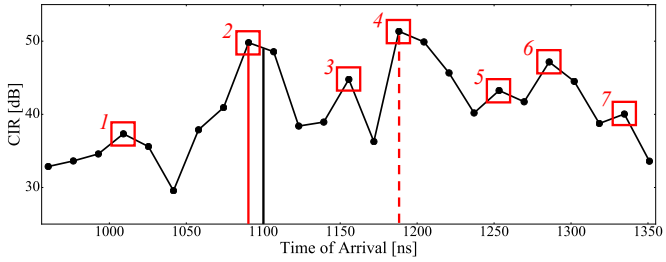


Fig. 3: Part of a typical CIR time series is displayed. The ground truth TOA is marked by a vertical black line. The 'find peaks' algorithm correctly selects peak 2 (solid red line) when the threshold $\alpha = 5$ [dB]. However, it commits a large error by selecting peak 4 if $\alpha = 1$ [dB] (dashed red line).

The find peaks algorithm has two main sources of error. Firstly, performance is sensitive to the threshold α. The red dashed line in Fig. 3 shows how a stronger secondary peak is selected when α is too small. The second source of error is due to the sampling frequency of receivers, which only take a sample every 16.27 nanoseconds. Therefore, the ground truth TOA of a signal rarely coincides exactly with a peak. The discrepancy between the predicted TOA (solid red line) and true TOA (black line) is evident in Fig. 3. Sampling error could be reduced by interpolating between points.

B. Benchmark Trilateration Algorithm

The benchmark trilateration algorithm takes the TOAs from five receivers as input, and returns the transmitter's location. The algorithm uses the downhill simplex method [12].

The logic underlying the benchmark is illustrated by Fig. 4. A randomly selected coordinate $L' = (x, y, z)$ serves as an initial guess for the transmitter's location. TOAs that would be observed by the receivers if the transmitter was located at L' are computed. These inferred

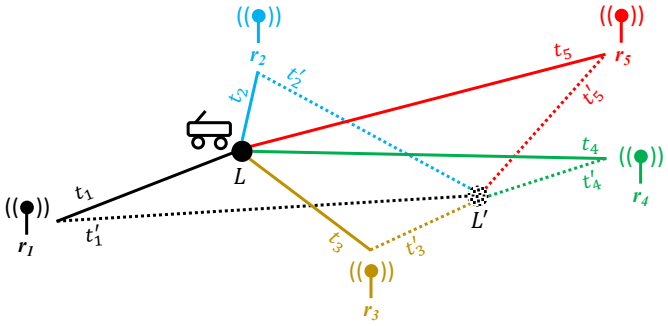


Fig. 4: The true location of the transmitter is marked L. The downhill simplex algorithm iteratively improves upon an initial guess L'. Solid lines represent the TOAs (t_i) that are reported by the receivers. Dotted lines represent the TOAs (t'_i) that would be reported if the transmitter was instead at location L'.

TOAs (t'_i) are compared with the TOAs (t_i) actually reported by the receivers. If both sets of TOAs agree it follows that $L' = L$, and L' must be the transmitter's true location. Otherwise, the degree of inconsistency reflects how far L' is from the transmitter. Let the error associated with L' be quantified by:

$$error\,(L') = \sum_{i=1}^{5} (t'_i - t_i)^2 . \tag{2}$$

The simplex procedure minimises $error\,(L')$ by iteratively updating L' until convergence occurs. This algorithm converges quickly to a good local optimum, but it is sensitive to outliers. An incorrect location for the transmitter will be returned if one or more of the reported TOAs are significantly off.

C. Deep Learning for TOA Estimation

Recurrent neural networks (RNNs) are distinguished from feedforward networks by their adoption of state. State enables an RNN to model temporal structure in a time series. Recall that the CIR generated by a receiver exhibits peaks when a signal from the transmitter is detected. It should be possible to train an RNN to identify the peak associated with the first arriving path. A bidirectional recurrent neural network (BiRNN) [14] lends itself well to this task.

The flow of information through a BiRNN is illustrated in Fig. 5. The network consists of an input layer, one or more hidden layers, and an output layer. The BiRNN evaluates the following expressions at every timestep t:

$$\overrightarrow{h}_t = tanh\left(W_{x\overrightarrow{h}} x_t + W_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right) \tag{3}$$

$$\overleftarrow{h}_t = tanh\left(W_{x\overleftarrow{h}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}\right) \tag{4}$$

$$y_t = W_{hy}\left(\overrightarrow{h}_t \odot \overleftarrow{h}_t\right) + b_y, \tag{5}$$

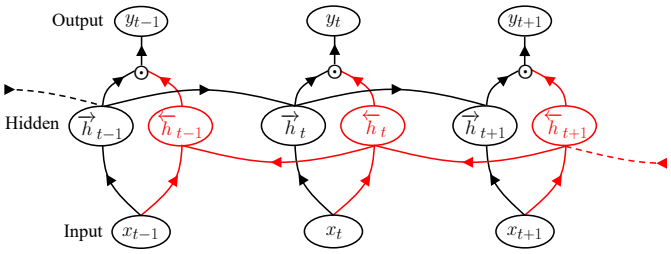


Fig. 5: Arrows indicate the flow of computation. Information from earlier in the time series is propagated through forward hidden states $\overrightarrow{h}$. Future context is carried through backward hidden states $\overleftarrow{h}$. Thus, the output y_t depends implicitly on the entire time series, in addition to the current input x_t.

where x_t is the current input vector, $\overrightarrow{h}_t$ is the forward hidden state, $\overleftarrow{h}_t$ is the backward hidden state, y_t is the output vector, and $\odot$ denotes vector concatenation. All hidden states are initialised to zero vectors. The weight matrices W and bias vectors b are optimised using back-propagation combined with stochastic gradient descent. Intuitively, the output y_t depends not only on the current input x_t, but also on all previous values in the time series (through $\overrightarrow{h}_t$) and all subsequent values (through $\overleftarrow{h}_t$).

We instrument a BiRNN for TOA estimation as follows. All 250 values in the CIR are associated with a timestep t. Let CIR_t denote the value at timestep t. The input to the BiRNN is given by $x_t = [CIR_{t-1}, CIR_t, CIR_{t+1}]$[1]. As such, x_t captures the local slope of the time series around t. Longer term structure is preserved in the hidden states. The BiRNN is executed for all timesteps, yielding 250 output vectors y_t.

The outputs are interpreted as follows. All 250 vectors y_t are concatenated into a single vector Y. A softmax is then applied to convert Y into a probability distribution. Hence, the TOA is estimated via:

$$TOA = \frac{argmax(Y)}{|Y|} \times 4067.5 \text{ [ns]},$$

where $argmax(\cdot)$ returns the index of the maximum value, $|\cdot|$ gives the dimension, and 4067.5 [ns] is the time spanned by the entire CIR. The achievable precision is 4067.5 [ns] $\div$ $|Y|$, where $|Y| = |y_t| \times 250$. Denote $|y_t|$ as the 'interpolation factor'. An interpolation factor of 1 implies that the TOA can be estimated to within a precision of 16.27 [ns], or the sampling frequency of a receiver. An interpolation factor of two implies a precision of 16.27 $\div$ 2 [ns], and so forth.

RNNs are vulnerable to vanishing and exploding gradients during training. Gated recurrent networks (GRUs) are much less prone to the numerical instabilities encountered by vanilla RNNs [3]. A GRU network is identical to an RNN, except for the equations used to update its hidden state. In this paper, we compare the performance of both versions on the TOA estimation task. Furthermore, we compare BiRNNs with forward RNNs. A forward RNN is simply a BiRNN without any backward recurrence.

[1]Missing values for CIR_{t-1} and CIR_{t+1} at $t = 0$ and $t = 250$ respectively are assigned the average of all values in the CIR.

D. Deep Learning for Trilateration

A multilayer perceptron (MLP) for trilateration is illustrated in Fig. 6. The inputs at layer i are the five TOAs reported by receivers $r_1, r_2, \ldots, r_5$. The outputs at layer o are the predicted x, y, z coordinates of the transmitter. The inputs are transformed through two hidden layers h_1 and h_2 with hyperbolic tangent activation functions. There are 25 neurons in each hidden layer.

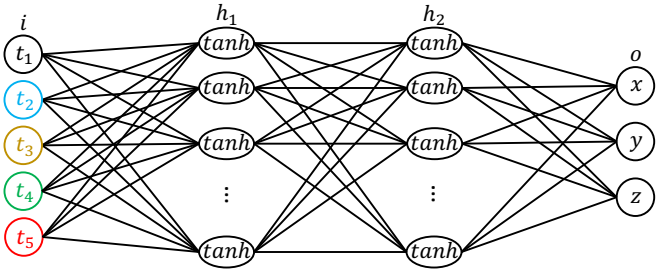


Fig. 6: An MLP maps TOAs $t_1, t_2, \ldots, t_5$ reported by five receivers to yield the transmitter's x, y, z coordinates.

$$a_{h_1} = tanh\left(W_{h_1,i} \cdot [t_1, t_2, t_3, t_4, t_5] + b_{h_1}\right) \quad (6)$$

$$a_{h_2} = tanh\left(W_{h_2,h_1} \cdot a_{h_1} + b_{h_2}\right) \quad (7)$$

$$[x, y, z] = W_{o,h_2} \cdot a_{h_2} + b_o \quad (8)$$

Equations 6, 7, and 8 define the forward pass of the MLP. The weights W and biases b are optimised via back-propagation combined with stochastic gradient descent. The MLP should cope better with erroneous and noisy TOAs than the benchmark trilateration algorithm from Section III-B. The benchmark attempts to reconcile all five TOAs when predicting the transmitter's location. A large localisation error will occur if one or more of the reported TOAs are incorrect. By contrast, the MLP can learn to ignore a subset of inconsistent TOAs at the input layer. This extra robustness should make the MLP less prone to committing large errors.

IV. Experiments

The experiments were designed to address two main research questions. Firstly, can the proposed approach based on deep learning outperform the benchmarks? Secondly, how do network hyperparameters such as depth and hidden size affect performance?

A. Deployment Scenario

The deployment scenario is illustrated in Fig. 7. The environment comprised of an open courtyard surrounded by buildings and trees. Fifteen measurement points were marked out around the courtyard. The transmitter was placed at each point for several minutes. During this time

it communicated with five receivers $r_1, r_2, \ldots, r_5$ which were installed on nearby buildings. The x, y, z coordinates of the measurement points and receivers were recorded. Hence, two datasets were generated; one for training TOA estimation networks, and another for training a trilateration network.

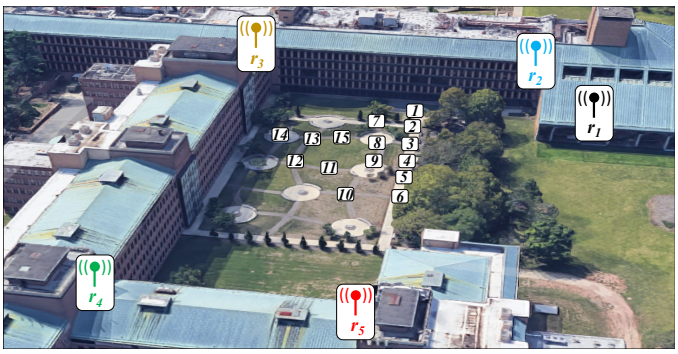


Fig. 7: The transmitter was moved between 15 measurement points in the Nobel Garden at Bell Laboratories. Receivers $r_1, r_2, , \ldots r_5$ were installed on nearby buildings.

Datasets for training the TOA network were obtained as follows. The transmitter was initially placed at measurement point 1 for 576 seconds. Here, it sent signals to the five receivers at intervals of one second. Each receiver saved 576 CIR time series like those displayed in Fig. 2. Ground truth TOAs were obtained by dividing the distance between measurement point 1 and each receiver by c. Next, the transmitter was moved to measurement point 2 for several minutes, and the preceding steps were repeated. Thus, 15 datasets containing CIR time series labelled with ground truth TOAs were obtained for the 15 measurement points.

Artificial training and development datasets were generated for training a trilateration network. The coordinates of 1,000,000 randomly sampled points were saved. Points were sampled in a cuboid enclosing the courtyard of dimensions 50 [m] $\times$ 50 [m] $\times$ 2 [m]. The distances between a point and the five receivers were computed, and divided by the c to yield the corresponding TOAs. A single training case consisted of the ground truth location of a point, and the five TOAs that would be reported by receivers if the transmitter was located at that point. A test set was obtained using real data from the 15 measurement points in Fig. 7. TOAs were estimated from the real CIRs reported by receivers.

B. Training the Models

1) TOA Estimation:: Experiments were carried out to assess various design alternatives for the TOA neural network. A design alternative could be the use of a bidirectional network versus a forward network for example. A given alternative was assessed by training 15 different network instances, one for each of the 15 measurement points p. This evaluation strategy prevented snooping on test data. A TOA network that would later be tested on point p was trained as follows:

1) A training, development, and test set were formed using the CIR time series from the 15 measurement points in Fig. 7. The test set contained CIRs from point p only. The training and development sets contained CIRs from 11 and 3 of the other points respectively.
2) The network was trained over 15 epochs using a mini batch size of 64. The training and development loss was given by the binary cross entropy.
3) The weights and biases were optimised via backpropagation combined with stochastic gradient descent. The learning rate was set to 0.0001. Weight decay was implemented in order to prevent overfitting. The penalty term was set to 0.0001.
4) Models were evaluated on the development set after every epoch. The model with the lowest development loss was returned after the run terminated.
5) The gradient descent algorithm is stochastic. As such, three independent runs (steps 2–4) were carried out. The best model from all three runs was saved, and the other two models were discarded.

The 15 trained network instances were evalauted on the 15 held out test sets. TOA errors on each test set were then aggregated. Hence, the design choice for the network was assessed, based on the aggregated test errors.

The parameter α in the benchmark TOA algorithm was tuned via random search. The benchmark was executed with randomly sampled thresholds α drawn from the interval $[0, 10]$. Whatever setting achieved the lowest TOA errors over the training and development sets was saved.

2) Trilateration:: Trilateration networks were trained in a similar fashion. The training and development sets were obtained as outlined in the previous section. The benchmark trilateration algorithm did not require any training.

V. Results and Discussion

Neural networks for TOA estimation and trilateration are benchmarked against conventional heuristics in this section. A variety of neural network design choices are compared.

A. Time of Arrival Estimation

1) Impact of Interpolation:: As outlined in Section II, a signal's time of arrival (TOA) at a receiver can be determined using the channel impulse response (CIR) time series. Peaks in the CIR occur when direct and reflected copies of a signal from the transmitter arrive at a receiver. The first peak is associated with the first arriving path, whereas secondary peaks are due to reflected copies of the signal. Two factors constrain the TOA estimation accuracy; the sampling frequency of receivers, and the aforementioned multipath effects. Here, we focus on

how interpolating between sampling intervals can improve TOA estimation accuracy.

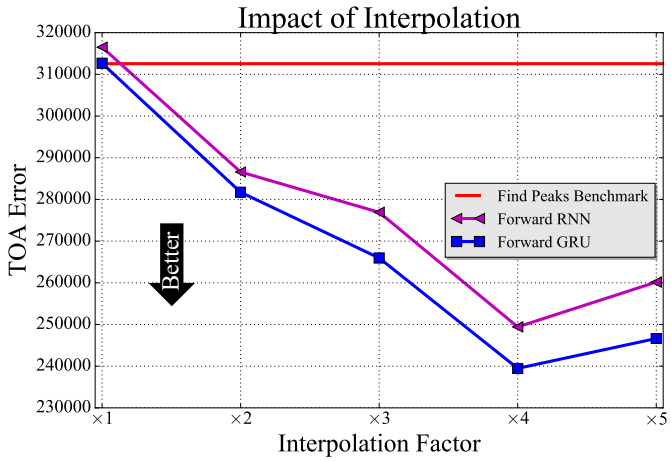


Fig. 8: The performance of forward RNN and GRU networks is sensitive to the interpolation factor. Both of the neural networks are significantly better than the benchmark at higher resolutions.

Forward RNN and GRU networks are compared with the benchmark 'find peaks' algorithm (see Section III-A) in Fig. 8. Each technique is assessed using CIR time series from the 15 development sets. For each CIR c, the estimated quantity TOA_c^{est} is compared with the ground truth value $TOA_c^{ground\ truth}$. The errors across all 15 development sets are integrated to yield a scalar:

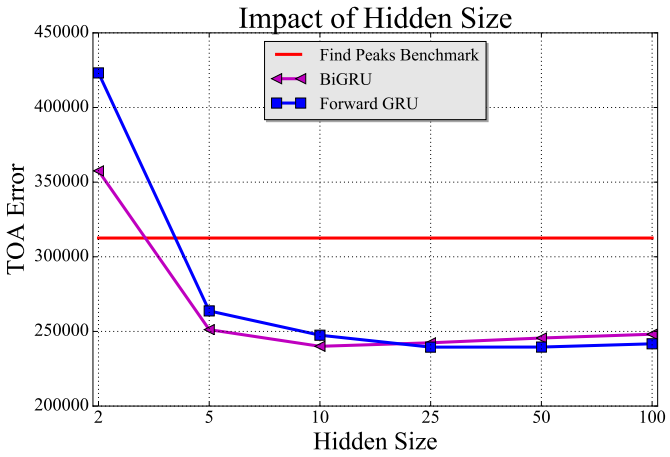
$$TOA\ Error = \sum_c \log_e \left(\left| TOA_c^{ground\ truth} - TOA_c^{est} \right| + 1 \right), \tag{9}$$

where the logarithm reduces sensitivity to outliers, and $|\cdot|$ returns the absolute value. The summation makes Equation 9 sensitive to large errors. The integrated error is plotted on the ordinate in Fig. 8.

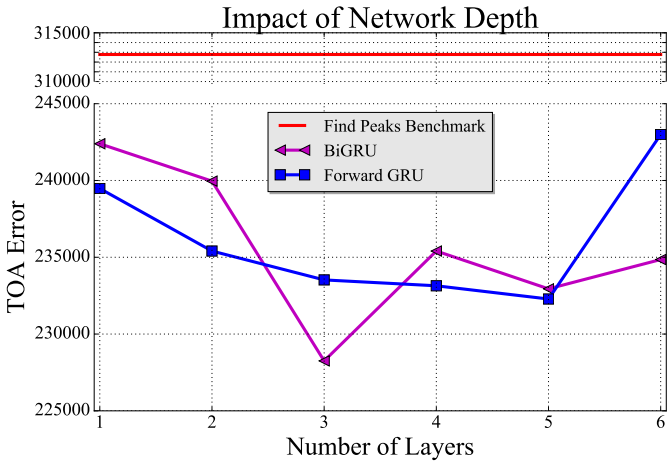
The neural networks and benchmark achieve a similar error when the interpolation factor is $\times 1$. This finding is not surprising since neither approach can interpolate between sampled intervals. However, the neural networks outperform the benchmark at larger interpolation factors. The best performance is observed when the network's resolution is four times that of the benchmark.

Finally, the GRU network outperforms its RNN counterpart. This result is consistent with experiments carried out by the authors in [3]. Henceforth, we restrict our attention to GRU networks with an interpolation factor of $\times 4$.

2) Impact of Network Topology:: A GRU's topology is defined by its hidden size (dimension of h) and the number of hidden layers. A larger hidden size gives the network greater capacity. Increased depth enables the network to tap more complex patterns in a time series. Intuitively, successive layers in a deep GRU can extract progressively more holistic features. For instance, the first hidden layer might capture the slope of the CIR time series around a given timestep. The second hidden layer might aggregate this information to understand where peaks, plateaus, and valleys occur.



(a)



(b)

Fig. 9: The best models are both wide and deep.

Forward and bidirectional GRUs with a variety of different topologies were trained. In one set of experiments, the number of layers was fixed at 1 and the hidden size was varied. In a second set, the hidden size was fixed at 25 and the depth was varied. The results are displayed in Figs. 9a and 9b respectively.

Both networks fail to beat the benchmark when their hidden size is only 2. However, they do outperform the benchmark if the hidden size is greater than 2. This result implies that wider models propagate more useful information through time. The forward GRU takes account of earlier inputs at timestep t, while the BiGRU captures future and past context.

Increasing the hidden size above a moderate value does not translate to better performance. However, adding more hidden layers is beneficial. The best forward GRU has 5 layers and the best BiGRU has 3 layers. Depth enables the

GRUs to compose hierarchical features at different layers. Thus, deeper networks recognise subtle patterns in a CIR that may be indiscernible to shallower networks, even if their capacity (hidden size) is large.

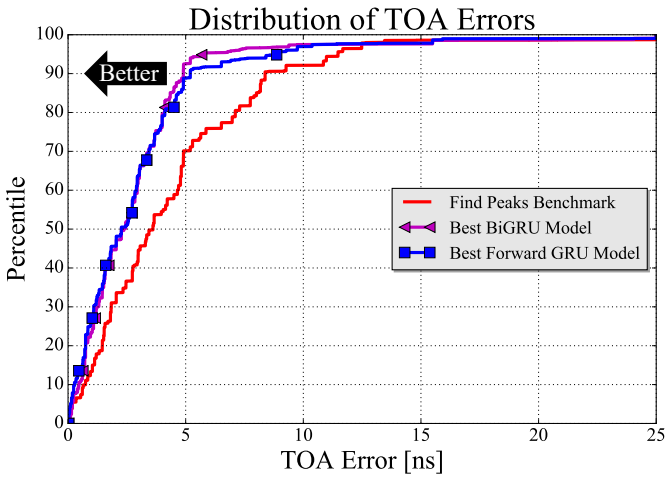


Fig. 10: The forward GRU and BiGRU networks achieve lower TOA errors than the benchmark. The BiGRU's worst errors are much lower than those of the forward GRU and benchmark. Outliers are clipped at 25 [ns] for clarity.

The performance metric defined by Equation 9 is convenient for comparing alternative models. However, it is an abstract quantity that does not lend itself well to intuition. The cumulative distribution plots in Fig. 10 reveal more clearly how TOA errors are reduced versus the benchmark. The TOA errors were computed on all CIRs in the 15 test sets. The best forward GRU and BiGRU networks realise lower errors than the benchmark. The utility of a localisation system is judged by its worst errors. It is striking that the neural networks commit much lower errors than the benchmark at high percentiles. The BiGRU achieves a 90^{th} percentile TOA error of 4.90 [ns], versus the benchmark's 8.38 [ns] – a reduction of 63%.

In summary, the best neural architecture for TOA estimation is a 3-layer BiGRU with a hidden size of 25. In the next section, this TOA network is combined with a denoising MLP for location trilateration.

B. Trilateration

Localisation consists of two coupled phases. In the downstream phase, TOAs from multiple receivers are estimated based on CIR time series. In the upstream phase, TOAs are mapped to an x, y, z coordinate using a trilateration algorithm. The estimated TOAs always embody errors due to the transmitter's finite sampling frequency, and multipath effects. Thus, overall system accuracy depends on a robust upstream trilateration algorithm, which must adequately denoise the input TOAs.

Artificial training and development datasets were created for training a trilateration algorithm as outlined in

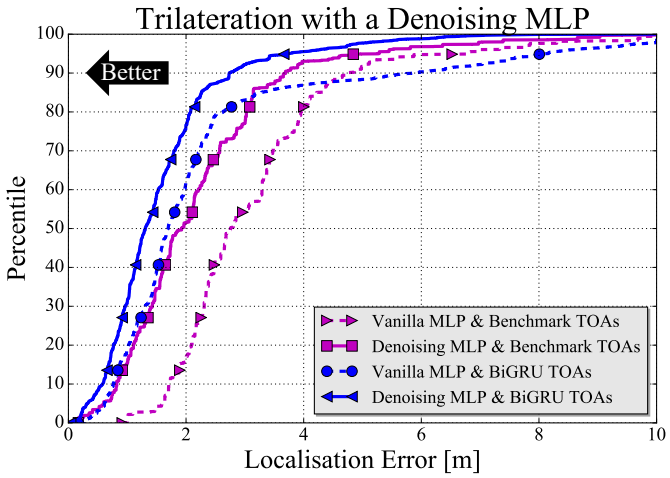


Fig. 11: Denoising MLPs (solid lines) outperform the vanilla MLPs (dotted lines) regardless of how the input TOAs are obtained. Outliers are clipped at 10 [m].

Section IV. The datasets consisted of randomly sampled locations in the environment and their associated ground truth TOAs to all five receivers. A vanilla multilayer perceptron (MLP) was trained using the unaltered TOAs. A denoising MLP was trained by adding noise to some of the input TOAs. A small amount of Gaussian noise was added to 10% of the TOAs (simulating sampling error), and a larger shift was added to 2% of the TOAs (simulating a misidentified peak in the CIR). The trained vanilla and denoising MLPs were evaluated on real test data. The test set was formed by obtaining estimated TOAs to all 15 measurement points in the environment (see Fig. 7). The TOAs were estimated using the find peaks benchmark and BiGRU algorithm.

The distributions of errors generated by both MLPs are visualised in Fig. 11. The solid blue line implies that the lowest errors are attained when the denoising MLP is combined with the BiGRU. Comparing the dotted blue and solid blue lines suggests that adding noise to the input TOAs during training is crucial. In particular, the vanilla MLP commits increasingly larger errors than the denoising MLP at higher percentiles. The vanilla MLP is prone to large errors because it does not ignore an incorrect TOA (due to a misidentified peak in the CIR). By contrast, the denoising MLP is trained to recognise when one of the five input TOAs is inconsistent with the other four. Lastly, consider the large gulf between the solid blue and solid magenta lines. This result implies that the denoising MLP works better if TOAs are obtained using the BiGRU, as opposed to the benchmark find peaks algorithm.

The neural networks are compared with benchmark algorithms in Fig. 12. Consider first the comparison between both solid lines and both dotted lines. The more accurate TOAs from the BiGRU usually lead to better localisation, regardless of the upstream trilateration al-

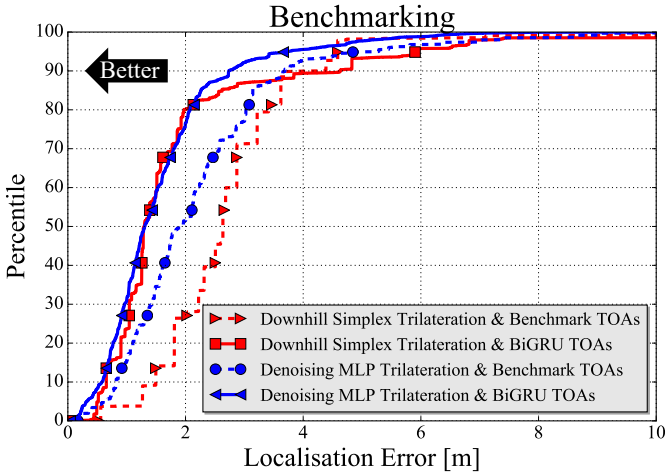


Fig. 12: The lowest localisation errors are observed when the BiGRU is combined with the denoising MLP (solid blue line). Outliers are clipped at 10 [m].

gorithm. Comparing the solid red and solid blue lines implies that the worst errors are lower when a denoising MLP is employed for trilateration, as opposed to the benchmark downhill simplex algorithm. Once again the benchmark cannot ignore an inconsistent TOA, and it is more sensitive to noisy TOAs than the denoising MLP. Finally, the main result is inferred by comparing the solid blue and dotted red lines. The proposed approach based on deep learning reduces median localisation error from 2.64 [m] to 1.30 [m], and the 90^{th} percentile error from 4.37 [m] to 2.74 [m].

VI. Conclusions and Future Work

In this paper, a novel TOA localisation technique based on deep learning was presented. A bidirectional recurrent neural network (BiGRU) was instrumented to obtain an interpolated TOA from the channel impulse response, and combined with a denoising multilayer perceptron (MLP) that performed trilateration. This end-to-end application of deep learning was shown to perform better than conventional approaches in both line-of-sight and non-line-of-sight conditions, lowering the median localisation error by 49% and the 90^{th} percentile error by 63%. Conventional heuristics with ad-hoc rules easily break down in new scenarios. Neural networks learn tailored strategies using training data collected in-situ.

The BiGRU benefits from multiple hidden layers. This finding suggests that the deeper models learn hierarchical structure in the CIR time series, structure that is opaque to a shallow model. Both components of the proposed approach are necessary to achieve state of the art performance. The combination of a denoising MLP with the BiGRU lends the system robustness. The more brittle benchmarks are easily confused by errors that propagate from the downstream TOA estimation phase.

If applied to a scenario with more non-line-of-sight conditions, and with the availability of larger datasets, we anticipate that the performance gains would be higher. Verifying this hypothesis is reserved as a subject for future work. Accurate localisation will enable new technologies in 5G wireless communications networks, such as beamforming and proactive resource allocation. The increasing complexity and heterogeneity of 5G deployments will compel operators to adopt a data-driven approach to network optimisation based on machine learning.

Acknowledgements

This research is based upon works supported by the Science Foundation Ireland under grant 13/IA/1850.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] P. Bahl and V. N. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In *Proc. 19th IEEE Conference on Computer Communications (INFOCOM'00)*, volume 2, pages 775–784 vol.2, March 2000.

[3] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[4] V. De Silva, M. Caine, J. Skinner, S. Dogan, A. Kondoz, T. Peter, E. Axtell, M. Birnie, and B. Smith. Player tracking data analytics as a tool for physical performance management in football: A case study from Chelsea Football Club Academy. *Sports*, 6(4):130, 2018.

[5] D. Giustiniano and S. Mangold. CAESAR: Carrier Sense-based Ranging in Off-the-shelf 802.11 Wireless LAN. In *Proceedings of the Seventh COnference on Emerging Networking EXperiments and Technologies*, CoNEXT '11, pages 10:1–10:12, 2011.

[6] D. Gorecky, M. Schmitt, M. Loskyll, and D. Zühlke. Human-machine-interaction in the industry 4.0 era. In *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, pages 289–294, July 2014.

[7] A. Goswami, L. E. Ortiz, and S. R. Das. Wigem: A learning-based approach for indoor localization. In *Proc. 7th COnference on Emerging Networking Experiments and Technologies*, CoNEXT '11, pages 3:1–3:12, 2011.

[8] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, pages 1764–1772, 2014.

[9] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti. SpotFi: Decimeter Level Localization Using WiFi. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, pages 269–282, 2015.

[10] L.Gergs. 5G for Industrial Applications. *ABI Research market report*, July 2019.

[11] D. D. McCrady, L. Doyle, H. Forstrom, T. Dempsey, and M. Martorana. Mobile ranging using low-accuracy clocks. *IEEE Transactions on Microwave Theory and Techniques*, 48(6):951–958, June 2000.

[12] J. A. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[14] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[15] M. Wollschlaeger, T. Sauter, and J. Jasperneite. The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0. *IEEE Industrial Electronics Magazine*, 11(1):17–27, March 2017.