Deep Learning for Synchronization and Channel Estimation in NB-IoT Random Access Channel

Jespersen, M.H.; Pajovic, M.; Koike-Akino, T.; Wang, Y.; Popovski, P.; Orlik, P.V.

TR2019-140 December 09, 2019

Abstract

The central challenge in supporting massive IoT connectivity is the uncoordinated, random access by sporadically active devices. The random access protocol and activity detection have been widely studied, while the auxiliary procedures, such as synchronization, channel estimation and equalization, have received much less attention. However, once the protocol is fixed, the access performance can only be improved by a more effective receiver, through more accurate execution of the auxiliary procedures. This motivates the pursuit of joint synchronization and channel estimation, rather than the traditional approach of handling them separately. The prohibitive complexity of the conventional analytical solutions leads us to employ the tools of deep learning in this paper. Specifically, the proposed method is applied to the random access protocol of Narrowband IoT (NB-IoT), preserving its standard preamble structure. We obtain excellent performance in estimating Time-of-Arrival (ToA), Carrier-Frequency Offset (CFO), channel gain and collision multiplicity from a received mixture of transmissions. The proposed estimator achieves a ToA Root-Mean-Square Error (RMSE) of 0.99 microsecond and a CFO RMSE of 1.61 Hz at 10 dB Signal-to-Noise Ratio (SNR), whereas a conventional estimator using two cascaded stages have RMSEs of 15.85 microsecond and 8.05 Hz,respectively.

IEEE Global Communications Conference (GLOBECOM)

Deep Learning for Synchronization and Channel Estimation in NB-IoT Random Access Channel

Mads H. Jespersen*, Milutin Pajovic†, Toshiaki Koike-Akino†, Ye Wang†, Petar Popovski*, and Philip V. Orlik†
*Department of Electronic Systems, Aalborg University, Denmark
†Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA

***Abstract*—The central challenge in supporting massive IoT connectivity is the uncoordinated, random access by sporadically active devices. The random access protocol and activity detection have been widely studied, while the auxiliary procedures, such as synchronization, channel estimation and equalization, have received much less attention. However, once the protocol is fixed, the access performance can only be improved by a more effective receiver, through more accurate execution of the auxiliary procedures. This motivates the pursuit of joint synchronization and channel estimation, rather than the traditional approach of handling them separately. The prohibitive complexity of the conventional analytical solutions leads us to employ the tools of deep learning in this paper. Specifically, the proposed method is applied to the random access protocol of Narrowband IoT (NB-IoT), preserving its standard preamble structure. We obtain excellent performance in estimating Time-of-Arrival (ToA), Carrier-Frequency Offset (CFO), channel gain and collision multiplicity from a received mixture of transmissions. The proposed estimator achieves a ToA Root-Mean-Square Error (RMSE) of 0.99 µs and a CFO RMSE of 1.61 Hz at 10 dB Signal-to-Noise Ratio (SNR), whereas a conventional estimator using two cascaded stages have RMSEs of 15.85 µs and 8.05 Hz, respectively.**

***Index Terms*—Deep learning, IoT standards, massive random access, joint estimation**

I. Introduction

A massive number of devices are expected to be connected to the Internet and several standards have been proposed to enable connectivity of low-complexity devices operating over a shared wireless channel. Most prominent technologies are Sigfox, LoRa and Narrowband IoT (NB-IoT) [1]. In Internet of Things (IoT) applications, the random access procedure has a high impact on device battery life and number of devices that can be supported concurrently [2]. Random access is used to request uplink allocation from the base station without requiring users to be constantly connected to the base station. Most IoT data packets are on the order of bits and users transmit them sporadically by establishing a new connection for every transmission. Establishing a connection using random access is a four step procedure [3], [4], which is initiated by a user that has packet to transmit by sending a random access preamble. The random access preamble is designed such that the base station is able to efficiently detect the transmitting user and estimate any timing offset between the user and base-station from the received signal. The timing offset comprises of propagation time, downlink synchronization errors and channel delay spread [5].

NB-IoT is a recent standard proposed by the 3rd Generation Partnership Project (3GPP) to accommodate the emerging number of wireless devices connected to the Internet. It is designed to co-exist with Long-Term Evolution (LTE) and provide low-cost and low-power devices with low throughput connectivity. The random access procedure in the NB-IoT is initiated by the Narrowband Physical Random Access CHannel (NPRACH). The NB-IoT has a system bandwidth of 180 kHz that accommodates 48 orthogonal channels from which a user attempting to establish a connection chooses one at random. If NB-IoT users choose different (i.e., orthogonal) preambles, the base station is able to estimate Time of Arrival (ToA) and Carrier Frequency Offset (CFO) of each user [5], [6]. However, given a possibly large number of users and relatively small number of orthogonal preambles, it is likely that two or more users choose the same preamble. The resulting collision may lead to a user back-off time of up to almost 9 minutes [7], [8]. In order to avoid unnecessary backoff periods and consequently improve channel utilization and overall capacity of the NB-IoT system, we propose in this paper a Deep Learning (DL)-based method for separating colliding users, detecting their number and estimating their respective ToAs and CFOs. We validate the proposed method using simulations and demonstrate significantly improved performance compared to the conventional approaches.

A. Related Work

Several papers have explored methods for activity detection, ToA and CFO estimation using the NB-IoT NPRACH preamble structure. As such, [5] estimates the ToA by searching for highest correlation between the received signal and delayed/frequency-shifted preamble on a grid of possible delays and frequencies. To reduce the complexity of the algorithm in [5], the ToA and CFO are estimated using the residual phase difference between symbol groups and channel hops in a two-stage procedure in [6]. With the goal to improve the ToA estimation, [4] suggests a novel hopping pattern that renders more accurate ToA estimation compared to that achieved with the already defined NB-IoT preamble.

We consider in this paper a problem of separating colliding NB-IoT users that choose the same random access preamble in the NPRACH scheme, and propose a method to detect the number of colliding users and estimate their ToA, CFO

The first author performed this work as an intern at MERL.

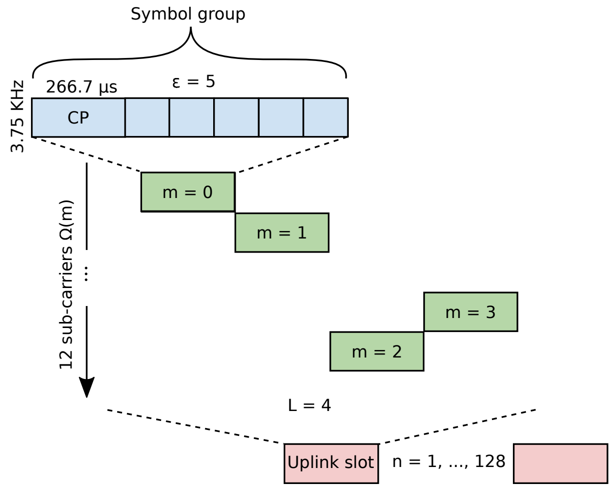

Fig. 1. Overview of NPRACH preamble and packet structure.

and channel gains. Motivated by recent success in leveraging learning-based methods for addressing problems related to physical layer communications [9], our method builds upon deep learning framework. In particular, we jointly detect the number of active users and estimate their parameters, with the aim to improve the capacity of the critical random access phase by not discarding interfering signals in order to utilize channel resources better, which in turn reduces back-off periods. In addition to handling much richer class of scenarios, the proposed method outperforms [6] in their own scenario where users transmit orthogonal preambles and do not collide. In comparison to [4], the random access preamble in this work is as suggested by the NB-IoT standard, ensuring the proposed method is practical in the NB-IoT systems currently being deployed. Finally, looking outside the NB-IoT scope, we believe that this work is the first application of deep learning techniques for user separation in massive connectivity systems.

II. NB-IoT Random Access Preamble Design

The preamble format and packet structure are illustrated in Fig. 1. The preamble is divided into symbol groups, where each group consists of a Cyclic Prefix (CP) and ε identical symbols. The value of ε depends on preamble format. The preamble format is chosen by the user based on the downlink power measurement to estimate its coverage area [3].

The most common preamble format is format 1 with preamble frame structure 0 or 1, which has $\varepsilon = 5$ and a symbol time $T_{\mathrm{SYM}} = 266.7\,\mu\mathrm{s}$. The CP period for frame format 0 is $T_{\mathrm{CP}} = 66.7\,\mu\mathrm{s}$ and $T_{\mathrm{CP}} = 266.7\,\mu\mathrm{s}$ for frame format 1 [10]. The CP is designed such that it is long enough to cover the maximum round trip delay to suppress Inter-Symbol Interference (ISI). Therefore one interpretation of allowing adaptive CP selection is for the user to use the short CP in the range 0–8 km and the long CP in the range 8–35 km [5].

The full preamble consists of 4 repetitions of the symbol group which is again repeated $n = 2^J, J = 0, \ldots, 7$ times for a full preamble length of $L = 4 \times 2^J$ symbol groups. The repetition of the symbol groups occurs within an uplink slot, and the number of repetitions is decided by the upper Medium Access Control (MAC)-layer depending on estimated link quality [10]. For simplicity we only consider the arbitrarily chosen case where $J = 2$, i.e., four symbol groups are repeated 4 times.

Before transmission, the user chooses a contiguous set of $N = 12, 24, 36$ or 48 subcarriers with 3.75 kHz spacing out of the available 48 subcarriers. This paper focuses on the preamble frame structure type 1 where $N = 12$. At the start of the NPRACH preamble transmission, the subcarrier of the first symbol group is chosen at random. After each symbol group the subcarrier will change using a deterministic channel hopping sequence so in the duration of a preamble there will be L subcarrier hops. Since the hopping pattern is deterministic, several users choosing the same initial subcarrier will thus collide for the entirety of the NPRACH preamble sequence. The number of orthogonal preamble sequences is therefore the number of allocated NPRACH subcarriers [7].

For frame structure type 1 and preamble format 0, two “levels” of hopping are employed as shown in Fig. 1. The hopping pattern is deterministic within a cell, but the subcarrier of every 4th symbol group appears random to neighbouring cells. The hopping procedure aids in the estimation of ToA and also reduces inter- and intra-cell interference [5]. The ToA should be estimated by the base station for successful uplink signal decoding and it further enables device positioning. Error in the ToA estimation results in the user being unable to receive the response sent by the base station. ToA estimation therefore has a great impact on performance in NB-IoT [4].

III. System Model

The received signal at the base station is a superposition of signals from multiple users, given by

$$y[n] = \sum_{k=0}^{K-1} a_k s_k[n] + w[n], \tag{1}$$

where K is the maximum number of concurrent users, $a_k \in \{0, 1\}$ indicates whether the kth user is active or not, and $w[n] \sim \mathcal{CN}(0, 1/\rho_n)$ denotes the additive noise with a per symbol Signal-to-Noise Ratio (SNR) of ρ_n.

At the receiver, the phase of each symbol depends on the ToA τ, the CFO Δf (which gives the frequency of the user's chosen channel with respect to the receiver's uplink carrier frequency f), and the channel rotation given by $\arg(h)$, where h is the complex-valued channel coefficient. These parameters are assumed to be independent across users and denoted by τ_k, Δf_k and h_k for each user k.

The signal from the kth user is given by

$$s_k[n] = h_k \, \mathrm{e}^{-\jmath 2\pi (f_n + \Delta f_k)(n T_{\mathrm{sym}} - \tau_k)}, \tag{2}$$

where T_{sym} is the symbol duration. The signal model is limited to only considering a single preamble sequence for the

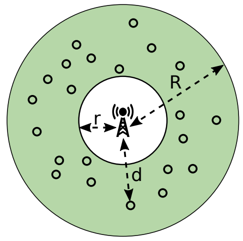

Fig. 2. Geometry of users distribution.

sake of simplicity. This means that the sub-carrier frequency pattern f_n is predetermined and identical for all instances of s.

The typical FFT length in LTE is 512 [6], but for simplicity we describe that each sample, n, corresponds to a symbol. In this model, the contents of the CP are interpreted as a symbol and therefore no distinction is made between the CP and the $\varepsilon = 5$ repeated symbols in a symbol group. This signal model may be valid only for the long CP which corresponds to distances between the user and base station within a minimum of $r = 8\,\mathrm{km}$ and a maximum of $R = 35\,\mathrm{km}$ [5]. The users are assumed to be uniformly distributed in the coverage area of the base station as illustrated in Fig. 2. The distance from the base station to the users d has the following Probability Density Function (PDF) [11]:

$$f_D(d) = \frac{2d}{R^2 - r^2}, \quad r \leq d \leq R, \tag{3}$$

which is used to model the ToA $\tau_k = \frac{d}{c}$, where c is the propagation speed.

The channel coefficient h_k of the signal model in (2) is a complex-valued constant which accounts for small scale fading: $h_k \sim \mathcal{CN}(0, 1)$. This means that the average received signal power is normalized to one. The narrowband channel is modeled as a slowly varying single-tap Rayleigh fading channel and for this reason, modeled as a single coefficient [6]. Large scale fading is not included in the model since users already have knowledge of the downlink SNR and adjust their transmit power accordingly using power control.

The CFO in (2) is chosen uniformly at random between -20 and $+20$ Hz [6]. For the sake of simplicity, the CFO and ToA are assumed to be constant throughout an entire NPRACH transmission for each user.

The activity indicator a_k is modeled as Bernoulli random variable with the probability of transmitting p and $a_1, \ldots, a_K$ are independent and identically distributed (i.i.d.). The number of concurrent active users is

$$N_a = \sum_{k=1}^{K} a_k \sim \mathcal{B}(K, p), \tag{4}$$

where $\mathcal{B}$ is the binomial distribution. We consider the case with $K = 4$ and $p = 0.5$ throughout the paper.

IV. Deep Learning Estimator

The goal of the estimator is to use the discrete signal $y[n]$ to estimate the activity indicator a_k, ToA τ_k, CFO Δf_k, and channel coefficient h_k of each user. Since the activity indicator of each user is a random variable, the total number of active users in the received signal is unknown. This boils down to a notoriously challenging problem of source separation with unknown number of users [12]. Deep learning has significantly improved the field of source separation and the general idea of using deep learning is to capture non-linear relationship between inputs and corresponding targets that is often difficult to model with analytically tractable expressions [12]. In this paper, estimating the unknown parameters is dealt with by splitting the problem into:

- Classification of the number of active users; and
- Estimation of ToAs, CFOs and channel coefficients given the number of users.

The two separate tasks are combined such that the synchronization parameters are accurately estimated for each detected user.

A. Estimation of the Number of Users

Finding the number of active users, N_a, is formulated as a classification problem where $\mathbf{p} = \mathsf{OneHot}(N_a)$ is a categorical random variable encoded as a one-hot vector specifying N_a. With a one-hot encoding, the true target $\mathbf{p} = [p_0, p_1, \ldots, p_K]$ has entry one at index N_a, and zero entries everywhere else. This is different from a typical way of representing active users where users are ordered in a vector and each index indicates the activity of a unique user. The number of users N_a can then be estimated as the l_0 norm of that sparse vector. In this collision scenario users are transmitting using the same spreading sequence and are not uniquely distinguishable. For this reason, only the information on the number of active users is represented in $\mathbf{p}$.

Cross-entropy loss is typically used in classification problems [13], and [14] suggests that the cross-entropy loss in classification problems leads to faster convergence and better generalization compared to the Mean Squared Error (MSE). For nonbinary classification, we typically use softmax cross entropy loss (or negative log-likelihood) expressed as:

$$\ell_{\mathrm{NLL}}(\mathbf{p}, \mathbf{q}) = -\sum_{k=0}^{K} p_k \log q_k, \tag{5}$$

where $\mathbf{q}$ is a continuous differentiable softmax function:

$$q_k = \frac{\exp(\pi_k)}{\sum_i \exp(\pi_i)}, \tag{6}$$

where $[\pi_0, \pi_1, \ldots \pi_K]$ are the outputs from the last layer of the neural network and $[q_0, q_1, \ldots q_K]$ represent the *a posteriori* class probabilities. A hard class prediction could then be found as $\arg\max_i[\pi_i]$. The simple $\arg\ max$ is not differentiable, and thus the softmax approximation of argmax is often used [15].

B. Parameter Estimation

The parameters to be estimated are collected in a vector

$$\mathbf{x}_k = \left[\tau_k, \Delta f_k, \Re[h_k], \Im[h_k]\right]^T. \tag{7}$$

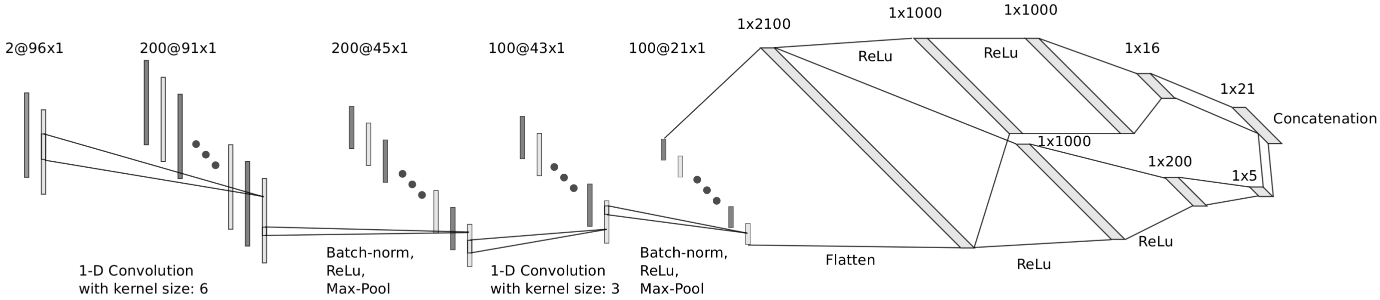

Fig. 3. Overview of Deep Neural Network (DNN) architecture for estimating synchronization parameters of up to 4 colliding users.

Note that it was found that representing the complex-valued channel coefficient h_k by Cartesian coordinate (i.e., real and imaginary parts) shows superior performance to phasor representation (i.e, amplitude and phase) as seen in Fig. 7. For N_a active users, the respective vectors are collected in a matrix

$$\mathbf{X} = [\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{N_a-1}]. \tag{8}$$

The neural network seeks to find an estimate $\hat{\mathbf{X}}$ such that $\mathbb{E}\|\mathbf{X} - \hat{\mathbf{X}}\|_2^2$ is minimal which is equivalent to a Minimum Mean-Square Error (MMSE) estimator.

The above formulation is sufficient to derive an estimation procedure. However, $\mathbf{X}$ consists of multiple parameters which have values on different scales. When using a practical optimization algorithm to find an estimate, any scaling difference between the parameters will affect the impact each value has on the gradient descent step.

To circumvent possible issues arising from error variations across parameters, we minimize the reconstruction error instead. The actual received signal without additive noise, $\mathbf{s}$, with the parameters in matrix $\mathbf{X}$ can be reconstructed using (2). The reconstruction is conveniently represented using function $f(\cdot)$ such that $\mathbf{s} = f(\mathbf{X})$.

For each estimate $\hat{\mathbf{X}}$, the equivalent noise-free signal $\hat{\mathbf{s}}$ is reconstructed and compared to the actual noise-free received signal $\mathbf{s}$. The noise-free signal is known during the training procedure and is used so the output of the neural network does not account for the distribution of the noise. The data fidelity (i.e., reconstruction loss) is quantified using the MSE metric such that

$$\ell_r(\mathbf{X}, \hat{\mathbf{X}}) = \mathbb{E}\big\|f(\mathbf{X}) - f(\hat{\mathbf{X}})\big\|_2^2 = \mathbb{E}\big\|\mathbf{s} - \hat{\mathbf{s}}\big\|_2^2. \tag{9}$$

The number of concurrent users in each sample is known during training so when reconstructing the signal $\hat{\mathbf{s}}$, the contributions from the correct number of users are taken into account when calculating the reconstruction loss ℓ_r for each sample.

The loss function which the neural network seeks to minimize is simply the sum of (5) and (9)

$$\mathsf{loss} = \ell_p(k, \mathbf{q}) + \ell_r(\mathbf{X}, \hat{\mathbf{X}}). \tag{10}$$

C. Network Implementation

An overview of the neural network that estimates both the number of users and synchronization parameters is illustrated in Fig. 3. The input to the network is the received signal which consists of 4 NPRACH repetitions each with $L(\epsilon+1)$ symbol periods. The total number of samples in the received signal is: $N_{\text{rep}}L(\epsilon+1) = 4 \cdot 4 \cdot (5+1) = 96$, where the real and imaginary parts are represented in 2 individual channels.

The output of the network is the flattened matrix $\mathbf{X}$ and the probability vector $\boldsymbol{\pi}$. For 4 users there are $4 \cdot 4 = 16$ parameters in $\mathbf{X}$ and 5 possible classes in the number of users (including the zero users case). The input to the network is processed so as to extract common features that are subsequently used for multi-task learning, that is, to detect the number of users and estimate their parameters. The first layer performs a 1-dimensional convolution over the input signal. Since the number of users, ToA, CFO and channel coefficient all are assumed to be constant throughout a transmission, a convolution layer is chosen so as to extract translationally invariant features of the input time-domain signal.

Following a typical Convolutional Neural Network (CNN) structure, batch normalization, non-linear activation and max-pooling are employed. The convolution layers, activations and pooling layers are repeated to form a deep neural network. The features found by the convolution layers are reshaped to a single vector which is then used as input to two individual feedforward neural networks. One of the networks performs classification and detects the number of users based on the output of the feature extraction layers. The other network performs regression with the goal to yield parameters so that the reconstructed signal is as close as possible to the received signal in the MSE sense. Each feedforward network has two fully connected layers followed by the Rectified Linear Unit (ReLU) activation and a linear output layer. The network and automatic differentiation are implemented using the PyTorch framework [16] and trained using multiple Graphics Processing Units (GPUs).

In the simulation ToA, CFO and channel coefficient are all drawn according to the distributions given in the system model and N_a is drawn according to $\Pr(k)$ for each sample. The input to the network $\mathbf{y}$ and each parameter in the output $\mathbf{X}$ is scaled to have zero mean and unit variance. In general the convergence of a neural network is faster if all inputs to all layers have zero-mean and unit covariance between training examples in the case when all examples are of equal importance [17]. From the system model the variance and mean of each parameter (CFO, ToA and h_k) are known and used to normalize the parameters to have mean zero and unit variance. The mean and variance of τ_k can be derived from (3) and the standardized ToA is given by

$$\tau'_k = \frac{\tau_k - \mathbb{E}[\tau_k]}{\mathsf{Var}(\tau_k)}. \tag{11}$$

The CFO, Δf_k, is scaled similarly. No normalization is necessary for the channel coefficients since $h_k \sim \mathcal{CN}(0, 1)$ and thus no scaling is necessary for the signal $\mathbf{y}$.

V. Estimation Results

A. Traditional Methods

The Phase-Difference (PD)-based method proposed in [6] utilizes the relationship between the phase trace of the received signal and the ToA and CFO. This method is used a benchmark comparison and is a two-step procedure where first the CFO-induced phase is estimated from phase-differences between symbol groups in time-domain. This contribution is then subtracted from the phase of the received signal to estimate the ToA-induced phase which is found from phase-differences between symbol groups on different frequencies.

The phase-trace of a noise-free received signal for user k can be expressed [6]:

$$\beta_k[n] = -2\pi\tau_k f_n - 2\pi\Delta f_k n T_{\text{sym}} + C \tag{12}$$

where C is a random constant phase offset. In practice the phase-trace of the received signal is not straightforward to obtain due to 2π-ambiguity but the unwrap-function and complex argument function of the received signal provides a good approximation [6]:

$$\beta_k[n] = \text{unwrap}\left(\arg(s_k[n])\right). \tag{13}$$

Phase differences between symbols with the same subcarrier frequency f_n can be used to estimate the phase contribution of the CFO. Symbols groups contain five consecutive identical symbols which phase-differences should be averaged to reduce noise variance. The average of all these estimates is used is used to estimate the CFO-induced phase [6]:

$$\beta_{k,\Delta f} = \frac{1}{N}\frac{1}{4}\sum_{n=0}^{N-1}\sum_{i=1}^{4}\beta_k[5n+i+1] - \beta_k[5n+i]. \tag{14}$$

The CFO estimate of the kth user is then simply

$$\hat{\Delta f}_k = \frac{1}{2\pi}\beta_{k,\Delta f}. \tag{15}$$

The estimated CFO-induced phase is subtracted from the phase-trace of the received signal and the ToA can be estimated similarly. This estimate is only valid if the phase-trace of the received signal only contains the contribution from a single user.

As a benchmark for the detection of the number of users, a crude amplitude-based estimator is devised. The mean amplitude of the received signal for different number of colliding users is compared to the amplitude of the received signal. The closest match then yields an estimate of the number of colliding users present in the received signal.

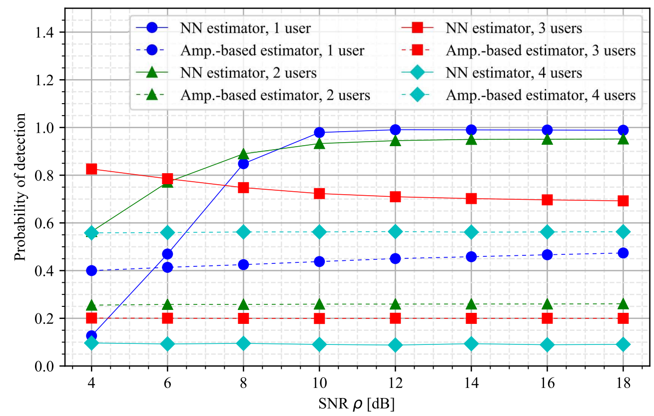

Fig. 4. Accuracy of estimating the number of colliding users. A signal is deemed correctly detected if the number of users are estimated correctly. The NN estimator is trained for signals with 10 dB SNR.

B. Simulation

The neural network is trained using samples generated with up to $K = 4$ concurrent users and at an SNR of 10 dB. New batches are generated for every step in the training procedure. The learning rate is 0.0001 and each batch consists of 50,000 realizations of $\mathbf{y}$ from (1). The stochastic optimization method based on adaptive momentum (ADAM) [18] is used and a total of 20,000 different batches are used in training.

In Fig. 4, the estimation of collision multiplicity is shown for the proposed classification method compared to a simple amplitude-based method. As colliding signals will add non-coherently, the amplitude of the signal is not a good indicator on collision multiplicity. 1 and 2 users are successfully identified with 98.0 % and 93.2 % at an SNR of 10 dB and the estimation accuracy decreases with the number of concurrent users. The proposed method often miss-classifies a signal containing 4 colliding users as resulting from transmissions of 3 users. Further it is counterintuitive to see a decrease in probability of detection for 3 users as SNR increases. However, using a data-driven approach the performance can only be guaranteed under the same conditions as used during training. In this case all the training signals has 10 dB SNR.

Since the loss function only depends on the reconstruction error, the estimated parameters in $\hat{\mathbf{X}}$ are arbitrarily ordered across users. To compare the output with the target $\mathbf{X}$ the parameters are ordered according to the estimated amplitudes. In cases where the estimated amplitudes are similar, the ordering may be wrong which leads to an artificially high error when evaluating performance for multiple users.

The RMSE of each parameter in $\mathbf{X}$ is calculated as:

$$\mathsf{RMSE}_k = \sqrt{\mathbb{E}\left[\|e_k\|_2^2\right]}, \tag{16}$$

where e.g. the estimation error of τ is: $e_k = \tau_k - \hat{\tau}_k$. The RMSE of the proposed neural network-based estimator is the

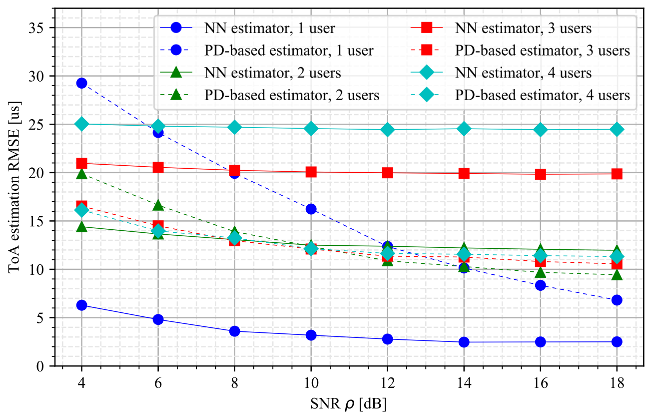

Fig. 5. Root-Mean-Square Error (RMSE) of ToA estimation across SNRs.

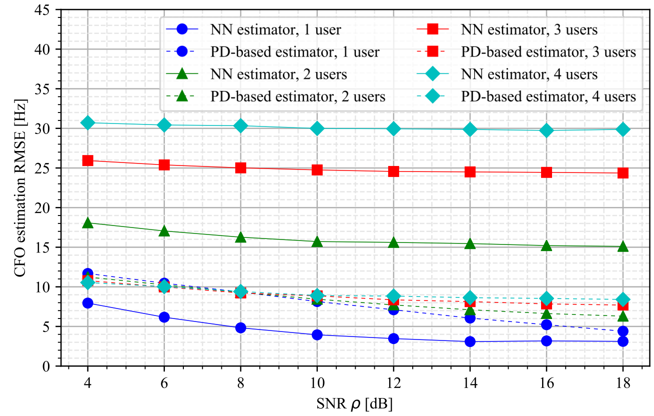

Fig. 6. RMSE of CFOs estimation across SNRs.

average of all RMSEs up to user k:

$$\mathsf{RMSE}_{\mathrm{NN},k} = \frac{1}{k}\sum_{i=1}^{k}\mathsf{RMSE}_i. \tag{17}$$

The conventional estimator is only able to estimate a single set of parameters, regardless of the actual number of users k. The error of the conventional estimator is therefore measured as the estimate which has the smallest error over all actual sets of parameters in $\mathbf{X}$, e.g. the estimated ToA error is

$$e_{\tau,\mathrm{PD}} = \min_k(|\tau_k - \hat{\tau}_{PD}|). \tag{18}$$

This gives the conventional estimator an artificial advantage.

The RMSE of ToA and CFO estimation with a varying number of users are shown in Figures 5 and 6. The neural network-based estimator shows lower estimation error for both ToA and CFO compared to the phase-difference-based estimator even for a single user. The DNN approach achieves better accuracy by jointly considering ToA and CFO instead of the cascaded structure of the conventional method. For two users the proposed estimator is superior to the conventional estimator when estimating ToA. At 10 dB the proposed estimator has an RMSE of 0.99 µs and 1.61 Hz for a single user compared to 15.85 µs and 8.05 Hz for the conventional estimator. The relatively high RMSE of the conventional estimator is likely due to the noise which causes wrong phase unwrapping at low SNRs [6].

The accuracy in estimating the channel coefficient h is shown in Fig. 7. The RMSE is 0.101 for the in-phase part and 0.103 for the quadrature part for a single user. The RMSE shows a similar trend as in ToA and CFO estimation with deteriorating performance as the number of concurrent users increases.

Overall the proposed method presents considerably improved performance compared to the traditional estimator in scenarios with a single, as well as multiple users.

VI. Discussion and Conclusion

We proposed a novel approach to synchronization and channel estimation. The system model consists of a superposition

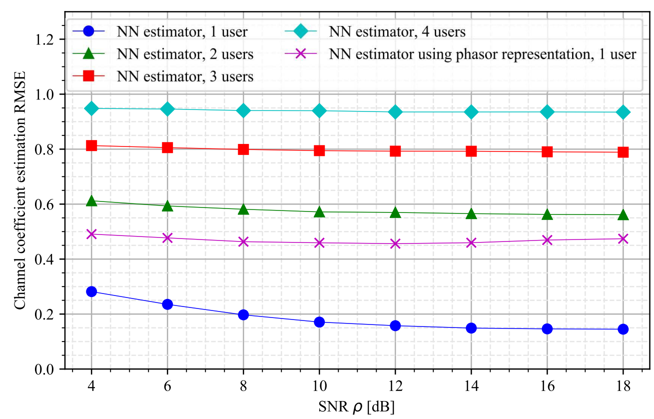

Fig. 7. RMSE of channel coefficient estimation across SNRs.

of an unknown number of users transmitting with the same preamble sequence. Deep learning is used to classify the multiplicity of collisions and estimate ToA, CFO and the channel coefficients for all user simultaneously.

The method is demonstrated in NB-IoT NPRACH where the number of orthogonal preambles is limited. The estimation error of a conventional approach in NB-IoT is compared to the performance of the proposed scheme. Traditional synchronization methods fail in the case of collisions with high Signal-to-Interference Ratio (SIR) whereas, with the proposed algorithm users can be distinguished and respective synchronization parameters can still be estimated with a reasonable performance.

Deep learning is a promising tool for developing joint estimation procedures, which are notoriously difficult in traditional model-based methods, and enables separation of synchronization parameters even when users transmit using the same preamble. Although deep learning-based estimation will lead to sub-optimal estimators compared to an analytically derived joint estimator, it allows for practical, straightforward development and efficient computation.

ACKNOWLEDGMENT

The work of Petar Popovski has been in part supported the European Research Council (ERC) under the European Union Horizon 2020 research and innovation program (ERC Consolidator Grant Nr. 648382 WILLOW) and Danish Council for Independent Research (Grant Nr. 8022-00284B SEMIOTIC).

REFERENCES

[1] I. Catherine, R. Tardy, N. Aakvaag, B. Myhre, R. Bahr, I. Catherine, R. Tardy, N. Aakvaag, B. Myhre, and R. Bahr, "Comparison of wireless techniques applied to environmental sensor monitoring," SINTEF Digital, Trondheim Norway, Tech. Rep., 2017.

[2] A. Azari, P. Popovski, G. Miao, and C. Stefanovic, "Grant-free radio access for short-packet communications over 5G networks," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–7.

[3] Y. . E. Wang, X. Lin, A. Adhikary, A. Grovlen, Y. Sui, Y. Blankenship, J. Bergman, and H. S. Razaghi, "A primer on 3GPP narrowband Internet of things," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 117–123, March 2017.

[4] W. S. Jeon, S. B. Seo, and D. G. Jeong, "Effective frequency hopping pattern for ToA Estimation in NB-IoT Random Access," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 10 150–10 154, 2018.

[5] X. Lin, A. Adhikary, and Y. P. Eric Wang, "Random access preamble design and detection for 3GPP narrowband IoT systems," *IEEE Wireless Communications Letters*, vol. 5, no. 6, pp. 640–643, 2016.

[6] J. Hwang, C. Li, and C. Ma, "Efficient detection and synchronization of superimposed NB-IoT NPRACH preambles," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 1173–1182, Feb 2019.

[7] L. Feltrin, G. Tsoukaneri, M. Condoluci, C. Buratti, T. Mahmoodi, M. Dohler, and R. Verdone, "Narrowband IoT: A survey on downlink and uplink perspectives," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 78–86, February 2019.

[8] 3GPP TS, *36.321 - Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification, V. 15.4.0*, 2018.

[9] O. Simeone, "A Very Brief Introduction to Machine Learning With Applications to Communication Systems," *arXiv e-prints*, p. arXiv:1808.02342, Aug 2018.

[10] 3GPP TS, *36.211 - Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation, V. 15.4.0*, 2018.

[11] T. Hien, Z. Wang, S. Kim, J. Nielsen, and P. Popovski, "Preamble detection in NB-IoT random access with limited-capacity backhaul," in *Proceedings of IEEE ICC'19*, 2 2019.

[12] J. Chien, *Source Separation and Machine Learning*. Elsevier Science, 2018.

[13] C. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. Springer New York, 2016.

[14] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, Aug 2003, pp. 958–963.

[15] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-softmax," *arXiv e-prints*, p. arXiv:1611.01144, Nov 2016.

[16] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS-W*, 2017.

[17] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*. Springer-Verlag, 1998.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv e-prints*, p. arXiv:1412.6980, Dec 2014.