Machine Learning-Based Channel Estimation in Massive MIMO with Channel Aging

Machine Learning-Based Channel Estimation in Massive MIMO with Channel Aging

Jide Yuan*, Hien Quoc Ngo*, and Michail Matthaiou*
*Institute of Electronics, Communications and Information Technology (ECIT), Queen's University Belfast, U.K.
e-mail: {y.jide, hien.ngo, m.matthaiou}@qub.ac.uk

***Abstract*—To support the ever increasing number of devices in massive multiple-input multiple-output systems, an excessive amount of overhead is required for conventional orthogonal pilot-based channel estimation (CE) schemes. To relax this stringent constraint, we design a machine learning (ML)-based time-division duplex scheme in which channel state information (CSI) can be obtained by leveraging the temporal channel correlation. The proposed ML-based predictors involve a pattern extraction and CSI predictor, which can be implemented via either a convolutional neural network (CNN) and autoregressive (AR) predictor or an autoregressive network with exogenous inputs recurrent neural network (NARX-RNN), respectively. Numerical results demonstrate that ML-based predictors can remarkably improve the prediction quality, and the optimal CE overhead is provided for practical reference.**

I. Introduction

With the exponential growth of devices, the conventional channel estimation (CE) using orthogonal pilots is undoubtedly incompetent considering the limited overhead resources. Meantime, in practice, channel state information (CSI) is correlated over time [1], a phenomenon known as *channel aging*. Leveraging this intrinsic phenomenon, CE overhead has tremendous potential to be reduced by rigorous CSI prediction.

Channel aging is the variation of the channel caused by the user movement,the impact of which has been characterized in prior literature [2, 3]. Paper [3] points out that the performance degradation caused by such phenomenon can be partly compensated by applying channel prediction, which implies that this practical impairment can be learned and used for estimating CSI. An effective method to model an aging channel is as autoregressive (AR) stochastic model whose parameters are computed based on the channel correlation matching property among adjacent coherence intervals [4]. However, according to the Levinson-Durbin recursion which is used for computing model parameters, the model order is bounded by the data amount of previous CSI samples.

Recently, machine learning (ML) based non-linear methods have been successfully applied in wireless communications [5], which motivates us to adopt relevant techniques to forecast CSI. Considering that the CSI forecasting is a typical time series learning problem, which has been fully discussed in the field of financial analysis, a recurrent neutral network (RNN) is a perfectly suitable NN for exploring the hidden pattern within CSI variations. Moreover, in massive multiple-input multiple-output (mMIMO) scenarios, CSI series from each antenna at the base station (BS) has the same autocorrelation pattern for a particular terminal. Leveraging this property, and by mapping multiple CSI series into a matrix, we are able to apply a similar technique from the field of image recognition, i.e, convolutional NN (CNN) to detect the pattern of CSI variation, and with the aging pattern as a prior knowledge, the prediction accuracy can be substantially improved.

In particular, we aim to reduce CE overhead via CSI prediction by taking advantage of the autocorrelation across CSI series. We first provide a ML-based time division duplex (TDD) scheme in which CSI is obtained via a ML-based predictor instead of conventional pilot-based channel estimator. Then, two ML-based structures are designed to improve the CSI prediction, namely, CNN combined with AR predictor (CNN-AR) and autoregressive network with exogenous inputs (NARX) RNN (CNN-RNN). The main idea is to use CNN to identify the channel aging pattern, and adopt AR predictor or NARX-RNN to forecast CSI. Numerical results demonstrate that the CNN-AR outperforms other architectures, including CNN-RNN, in terms of prediction accuracy, and provides the optimal CE overhead with respect to accuracy requirements.

This work was supported by the RAEng/The Leverhulme Trust Senior Research Fellowship LTSRF1718\14\2.

II. System Model

A TDD single-cell multi-user mMIMO system is considered, where a BS having N antennas serves K single-antenna users. We assume that the channel is static during each coherence interval, but *it does not change independently from one interval to the next*. More precisely, there is a correlation over the channel coherence intervals. This is reasonable because the scattering environment shares a high degree of similarity across several intervals [6].

The $N \times 1$ channel vector between the BS and the kth user at the lth coherence interval is modeled as

$$\mathbf{g}_k[l] = \mathbf{h}_k[l]\sqrt{\beta_k}, \tag{1}$$

where β_k represents large-scale fading, and $\mathbf{h}_k[l]$ is the small-scale fading. The overall channel from K users to the BS can be represented in matrix form as

$$\mathbf{G}[l] = \mathbf{H}[l]\mathbf{B}^{\frac{1}{2}}, \tag{2}$$

where $\mathbf{B}$ is a diagonal matrix whose kth element is β_k, and $\mathbf{H}[l] = [\mathbf{h}_1[l], \ldots, \mathbf{h}_K[l]] \in \mathbb{C}^{N\times K}$.

A. ML-Based TDD

The frame structure in conventional TDD mainly consists of CE, uplink (UL) and downlink (DL) phases, in which the

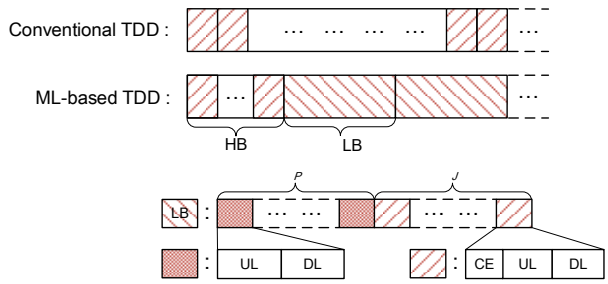

Fig. 1. Conventional TDD versus ML-based TDD. In learning-based block (LB), CE overhead is removed from frame structure for P intervals due to the adoption of ML-based CSI prediction.

channels estimated during the CE phase are further used for UL and DL transmission. Different from conventional TDD, the proposed ML-based TDD scheme increases the resources for data transmission by reducing the CE overhead from the frame structure, and CSI is obtained using a ML technique via exploring the correlation among adjacent intervals. The ML-based TDD scheme contains two types of blocks, namely, head block (HB) and learning-based block (LB), shown in Fig. 1. The following assumptions are made in the ML-based TDD scheme:

- A HB consists of V conventional TDD coherence intervals; A LB consists of P ML-based coherence intervals (without CE) and J $(J < V)$ conventional TDD coherence intervals.
- In a HB, channels of V intervals are estimated using the minimum mean square error (MMSE) estimator. After a HB, CSI is predicted via ML-based predictors for P intervals, and is then updated for the following J intervals via the MMSE estimator to improve the prediction accuracy for the subsequent LB.

B. Channel Aging Model

In general, aging property is mainly caused by the movement of the users, and such feature can be approximately characterized via the second order statistics of the channel, i.e., autocorrelation function (ACF) [4].

We assume that the propagation path experiences a two-dimensional isotropic scattering, whose corresponding normalized continuous-time ACF at the BS is

$$R\left(t\right) = J_0\left(2\pi f_{\mathrm{d}} t\right), \tag{3}$$

where $J_0\left(\cdot\right)$ is the zeroth-order Bessel function of the first kind, f_{d} is the maximum Doppler frequency given by $f_{\mathrm{d}} = v f_c / c$ with v is the velocity of user, c is the speed of light, and f_c is the carrier frequency. Although the formula indicates that the channel impulse response varies continuously, we notice that the variation is nearly imperceptible over period of dozens of channel samples. Therefore, we consider the discrete-time ACF of fading channel coefficients as

$$R\left[l\right] = J_0\left(2\pi f_n \left|l\right|\right), \tag{4}$$

where $|l|$ is the delay in terms of the number of coherence intervals, and $f_n = \nu T_s f_{\mathrm{d}}$ represents the normalized Doppler shifts with sampling duration T_s and the number of samples in a coherence interval ν.

In this paper, we assume the same autocorrelation among all channels from a particular user to the BS antennas. Hence, given the desired ACF as (4) for $l \geqslant 0$, we model the small-scale fading series as [4]

$$\mathbf{h}_k\left[l\right] = -\sum\nolimits_{q=1}^{Q} a_q \mathbf{h}_k\left[l-q\right] + \boldsymbol{\omega}\left[l\right], \tag{5}$$

where $\boldsymbol{\omega}\left[l\right]$ is the uncorrelated complex white Gaussian noise vector with zero mean and variance

$$\sigma_\omega^2 = R\left[0\right] + \sum\nolimits_{q=1}^{Q} a_q R\left[-q\right], \tag{6}$$

and $\{a_q\}_{q=1}^{Q}$ are the AR coefficients which are evaluated via the Levinson-Durbin recursion [4]. As Levinson-Durbin recursion is a well known algorithm, we skip the details for saving space.

Remark 1: Given a desired ACF, the fitting accuracy of AR model improves with higher order Q. However, according to the Levinson-Durbin recursion, Q is upper bounded by the amount of date of previous CSI samples, which implies that the performance of channel prediction via the AR estimator is limited by the number of coherence intervals V in a HB for the proposed ML-based TDD scheme.

Intuitively, the small-scale fading vector in (5) for a particular user follows the Gaussian distribution with zero mean and same variance. Denote by $\dot{h}_k$ the small-scale fading in a typical interval from the kth user to a typical antenna at the BS; its variance can be calculated via the Green's function [7]

$$\sigma_{\dot{h}_k}^2 = \sum\nolimits_{j=1}^{\infty} G_j^2 \sigma_\omega^2, \tag{7}$$

where

$$G_j \triangleq \begin{cases} 1, & j = 0, \\ \sum_{q=1}^{j} a_q G_{j-q}, & j \leq Q, \\ \sum_{q=1}^{Q} a_q G_{j-q}, & j > Q, \end{cases}$$

i.e., $\mathbf{h}_k[l] \sim \mathcal{CN}(0, \sigma_{\dot{h}_k}^2 \mathbf{I}_N), \forall l$.

C. MMSE Estimation

We assume that in a conventional coherence interval, the orthogonal pilots are used, and the channel is estimated using the MMSE estimator. For ease of analysis, we suppose that the length of pilot signal is equal to number of users, i.e., $\boldsymbol{\Psi} = \left[\boldsymbol{\psi}_1^T, \ldots, \boldsymbol{\psi}_K^T\right]^T \in \mathbb{C}^{K \times K}$, where $\boldsymbol{\psi}_k$ for $k = 1, \ldots, K$ with $\boldsymbol{\Psi}\boldsymbol{\Psi}^H = \mathbf{I}_K$. The operation $(\cdot)^T$ and $(\cdot)^H$ denote the matrix transpose and conjugate transpose, respectively. The users use the same power p_{p} to transmit pilots, and the received training signal at the BS is

$$\mathbf{Y}_{\mathrm{p}}\left[l\right] = \sqrt{K p_{\mathrm{p}}} \mathbf{G}\left[l\right] \boldsymbol{\Psi} + \mathbf{N}\left[l\right], \tag{8}$$

where $\mathbf{N}\left[l\right]$ is white additive Gaussian noise matrix whose elements have variance σ_n^2. Correlating $\mathbf{Y}_{\mathrm{p}}\left[l\right]$ with the pilot matrix $\boldsymbol{\Psi}$, the BS obtains

$$\mathbf{R}_{\mathrm{p}}\left[l\right] = \frac{1}{\sqrt{K p_{\mathrm{p}}}} \mathbf{Y}_{\mathrm{p}}\left[l\right] \boldsymbol{\Psi}^H, \tag{9}$$

and the received noisy channel vector from the kth user at the lth interval is

$$\mathbf{r}_{\mathrm{p},k}[l] = \mathbf{g}_k[l] + \frac{1}{\sqrt{Kp_\mathrm{p}}}\mathbf{N}[l]\boldsymbol{\psi}_k^H. \tag{10}$$

Recalling the channel model in (1), the channel vectors from the kth user to the BS is distributed as $\mathbf{g}_k[l] \sim \mathcal{CN}\left(\mathbf{0}, \dot{\beta}_k \mathbf{I}_N\right)$ according to (7) with $\dot{\beta}_k = \beta_k \sigma_{\dot{h}_k}^2$. Thus, the MMSE estimate of $\mathbf{g}_k[l]$ follows

$$\hat{\mathbf{g}}_k^{\mathrm{mmse}}[l] \sim \mathcal{CN}(\mathbf{0}, \dot{\beta}_k \gamma_k^{\mathrm{mmse}} \mathbf{I}_N), \tag{11}$$

where $\gamma_k^{\mathrm{mmse}} = \frac{\dot{\beta}_k}{\dot{\beta}_k + \mu}$ with $\mu = \frac{\sigma_n^2}{p_\mathrm{p} K}$, and the variance of the estimator error $\mathbf{e}_k^{\mathrm{mmse}}$ follows $\mathcal{CN}(\mathbf{0}, (1-\gamma_k^{\mathrm{mmse}})\dot{\beta}_k \mathbf{I}_N)$.

III. ML-based Channel Forecasting Approaches

We aim to implement multi-step prediction for CSI to minimize the CE overhead. In this section, two types of NN architectures, i.e., CNN-AR and CNN-RNN, are discussed for CSI forecasting. The idea behind two ML-based architectures is identical; that is the time-series predictor collaborates with the CNN which is used to extract the ACF pattern.

A. CNN-AR Approach

CNNs have been proved to have satisfactory performance in image classification problems [8]. Their key feature is that they conduct different function units alternatively, e.g., convolution layers, pooling layers, full connection layers. More importantly, CNNs treat the feature extraction and the classification identically; in particular, feature extraction is implemented by convolution layers and classification is approached by full-connection layers. As the shared weights in convolution layers and the weights in full-connection layers are trained together, the total classification error of a well designed CNN can be significantly minimized.

The mechanism of CNNs inspires us to adopt such architecture to extract the ACF pattern. As N CSI series for a particular user vary according to the same ACF, by mapping multiple CSI series into a matrix, the input of CNN is

$$_{op}\left(\ddot{\mathbf{G}}_k\right) = \left[_{op}\left(\hat{\mathbf{g}}_k^{\mathrm{mmse}}[1]\right), \ldots, _{op}\left(\hat{\mathbf{g}}_k^{\mathrm{mmse}}[V]\right)\right]. \tag{12}$$

This can be thought as a 2D image data, and the corresponding ACF is thought as label λ. The operator $_{op}(\cdot)$ is an designed manipulation to map the complex-valued CSI vector into a $2N$-dimensional real-valued vector, i.e., $_{op}\left(\mathbf{g}_k[l]\right) = [\mathrm{Re}\{\mathbf{g}_k[l]\}^T, \mathrm{Im}\{\mathbf{g}_k[l]\}^T]^T$. By classifying the pattern of ACF from $_{op}(\ddot{\mathbf{G}}_k)$, we are able to regenerate the channel series using pre-trained CSI predictor without real time calculation.

We choose the adaptive moment estimation (ADAM) as the optimizer, and use the minimum square error (MSE) as the loss function, which is defined as

$$\mathsf{C}_{\mathrm{cnn}} = \frac{1}{2}\sum\nolimits_{m=1}^{M}\sum\nolimits_{l_\mathrm{P}=1}^{L_\mathrm{P}}\left(\lambda_{l_\mathrm{P}}^m - \hat{\lambda}_{l_\mathrm{P}}^m\right)^2, \tag{13}$$

where M represents the training data amount, L_P represents the total number of ACF patterns, $\lambda_{l_\mathrm{P}}^m$ represents the l_Pth dimension of pattern label for the mth input data, and the estimates of which are denoted by $\tilde{\lambda}_{l_\mathrm{P}}^m$.

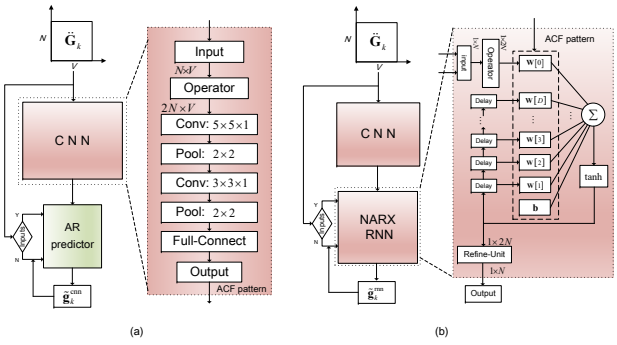

Fig. 2. (a). CNN-AR architecture in which CNN comprises an operator with two convolution layers and two max pooling layers and a full-connect layer for extraction, and an AR predictor whose coefficients are pre-computed. (b). CNN-RNN architecture in which CNN has same structure as CNN-AR, and NARX-RNN comprises an operator with D delays and a refine unit for CSI recovery.

The procedure for CNN-AR scheme is described in Fig. 2(a). Given $\ddot{\mathbf{G}}_k$ as inputs, CNN transforms the complex matrix into a real-valued matrix and identifies the CSI ACF pattern. Then, system loads the pre-computed AR coefficients of the corresponding aging pattern, and predicts CSI for the subsequent interval as

$$\hat{\mathbf{g}}_k^{\mathrm{cnn}}[l] = -\sum\nolimits_{q=1}^{Q} a_q \hat{\mathbf{g}}_k^{\mathrm{mmse}}[l-q]. \tag{14}$$

According to the proposed ML-based TDD scheme, for the first P intervals in LB, the NN output of the current interval is used as the input to forecast CSI for the next interval. Mathematically speaking,

$$\begin{aligned}\hat{\mathbf{g}}_k^{\mathrm{cnn}}[l+l'] = &-\sum\nolimits_{q=l'+1}^{Q} a_q \hat{\mathbf{g}}_k^{\mathrm{mmse}}[l+l'-q] \\ &-\sum\nolimits_{q'=1}^{l'} a_{q'} \hat{\mathbf{g}}_k^{\mathrm{cnn}}[l+l'-q'], \quad l' \in P,\end{aligned} \tag{15}$$

until the next conventional coherence interval.

Note that the given CNN structure is a simple NN which can only distinguish dozens of ACF patterns with acceptable accuracy. As ACF is dominated by the Doppler shift, which has hundreds of patterns, the engineering implementation of such architecture should be much deeper. In this paper, we aim to emphasize the feasibility of our scheme, and simplify the system structure for ease of training.

B. CNN-RNN Approach

As CNN in the CNN-RNN structure is identical to that in CNN-AR, we only introduce the CSI predictor, i.e., NARX-RNN in this part.

The general form of NARX-RNN is commonly described as

$$\mathbf{f}[l] = f\left(\mathbf{x}[l], \mathbf{f}[l-1], \mathbf{f}[l-2], \ldots, \boldsymbol{\theta}\right), \tag{16}$$

where a one-step prediction of $\mathbf{f}[l]$ depends on the previous several outputs, input $\mathbf{x}[l]$, and some parameters $\boldsymbol{\theta}$. Such architecture is implemented by introducing *delays* in the mechanism where the output has direct connections to the

past. In this paper, we adopt a widely used NARX RNN form, specifically given in [9]

$$\mathbf{f}\left[l\right]=\tanh\left(\mathbf{W}\left[0\right]\mathbf{x}\left[l\right]+\sum\nolimits_{d=1}^{D}\mathbf{W}\left[d\right]\mathbf{f}\left[l-d\right]+\mathbf{b}\right), \quad (17)$$

where D is the maximum number of delays, the weight matrix $\mathbf{W}[d] \in \mathbb{R}^{2N\times 2N}$, $\mathbf{W}[0] \in \mathbb{R}^{2N\times 2N}$, and bias vector $\mathbf{b} \in \mathbb{R}^{2N\times 1}$ are the parameters trained in the NN.

As there is no input from the MMSE estimator at the first P intervals in LB, to fit our problem, we make a minor revision in (17). Taking the channel of kth user as example, the NARX RNN is described as

$$\begin{aligned}{}_{op}\left(\hat{\mathbf{g}}_k^{\mathrm{rnn}}\left[l\right]\right)=\tanh\Big(\mathbf{W}\left[0\right]{}_{op}\left(\hat{\mathbf{g}}_k^{\mathrm{mmse}}\left[l-1\right]\right)\\ +\sum\nolimits_{d=1}^{D}\mathbf{W}\left[d\right]{}_{op}\left(\hat{\mathbf{g}}_k^{\mathrm{mmse}}\left[l-d\right]\right)+\mathbf{b}\Big), \quad (18)\end{aligned}$$

where $\hat{\mathbf{g}}_k^{\mathrm{rnn}}[l]$ is the NARX-RNN prediction. Therefore, the corresponding refine-unit for transforming the output from a real value into a complex value is given by

$${}_{ru}\left({}_{op}\left(\mathbf{g}_k\left[l\right]\right)\right)_n=\left({}_{op}\left(\mathbf{g}_k\left[l\right]\right)\right)_n+i\left({}_{op}\left(\mathbf{g}_k\left[l\right]\right)\right)_{n+N},$$

where $\left({}_{op}\left(\mathbf{g}_k\left[l\right]\right)\right)_n$ is the nth element of ${}_{op}\left(\mathbf{g}_k\left[l\right]\right)$, and $i=\sqrt{-1}$.

Consistent with typical RNNs, the training of this network is based on minimizing the sum-of-squared error cost function

$$\mathsf{C}_{\mathrm{rnn}}=\frac{1}{2}{}_{op}\left(\hat{\mathbf{g}}_k^{\mathrm{rnn}}\left[l\right]-\mathbf{g}_k\left[l\right]\right)^H{}_{op}\left(\hat{\mathbf{g}}_k^{\mathrm{rnn}}\left[l\right]-\mathbf{g}_k\left[l\right]\right). \quad (19)$$

The weight matrix $\mathbf{W}[0]$ is updated via its gradient

$$\triangle\mathbf{W}\left[0\right]=\eta\nabla_{\mathbf{W}[0]}\mathsf{C}_{\mathrm{rnn}}, \quad (20)$$

where η is a learning rate and $\nabla_{\mathbf{W}[0]}$ is the Jacobian in the derivative whose (i,j)th element is $\frac{\partial}{\partial w[0]_{i,j}}$ with $w\left[0\right]_{i,j}$ being the (i,j)-th element of matrix $\mathbf{W}\left[0\right]$. By assuming that the weights at different time instances are independent, the gradient can be expanded over $l-d$ time steps via the chain rule

$$\begin{aligned}\nabla_{\mathbf{W}[0]}C=\sum\nolimits_{n=1}^{N}\left(\hat{\mathbf{g}}_k^{\mathrm{rnn}}\left[l\right]-\mathbf{g}_k\left[l\right]\right)^H\nabla_{\hat{\mathbf{g}}_k^{\mathrm{mmse}}[l]}\hat{g}_{k,n}^{\mathrm{rnn}}\left[l\right]\\ \cdot\left(\sum\nolimits_{d=1}^{D}\nabla_{\mathbf{W}[d]}\hat{\mathbf{g}}_k^{\mathrm{mmse}}\left[l\right]\right), \quad (21)\end{aligned}$$

where $\hat{g}_{k,n}^{\mathrm{rnn}}$ represents the estimated CSI from kth user to nth antenna at BS. The methodology of training is called backpropagation through time (BPTT) algorithm, ans is detailed in [10].

The procedure for CNN-RNN is described in Fig. 2(b). At the beginning, NARX-RNN loads the pre-trained parameters according to the received ACF pattern from CNN, and use $\tilde{\mathbf{g}}_k^{\mathrm{mmse}}$ as input to predict the CSI for the next interval. In the subsequent interval, same as CNN-AR, the NN output of current interval is used as input to predict the CSI, and we repeat this procedure for P intervals.

Note that NARX-RNN also suffers from the vanishing gradient and long-term dependencies problem [10]. However, such drawback will not cause a major issue to our problem since channel series only have strong relation within adjacent intervals.

TABLE I
SYSTEM PARAMETERS.

Number of ACF patterns L_{P}	10
Number of intervals in HB V	8
Transmit power p_{p}	0 dBm
Background noise power σ_{n}^2	-174 dBm/Hz
Carrier frequency	2 GHz

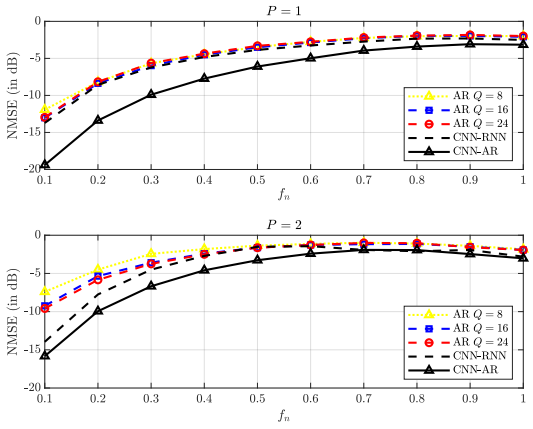

Fig. 3. Comparison of prediction NMSE among AR predictors, CNN-AR and CNN-RNN with respect to normalized Doppler shift f_n. Results are shown for $J=4$ and $U=10$.

IV. Numerical Results

In our simulations, the BS deploys 128 antennas, and K users are randomly distributed in a $1\,\mathrm{km}^2$ area. We also set a guard zone of 100 meters for each user, i.e., the distance between any user and BS is no less than 100 m. The large-scale fading β_k is modeled as a function of user at distance d_k, and is given as

$$\beta_k\left(d_k\right)=30.2+23.5\log_{10}\left(d_k\right). \quad (22)$$

Regarding the CE overhead, we consider the ratio of pilot length to the number of samples in a coherence interval ν as our metric, which is defined by

$$\mathcal{O}^{\mathrm{con}}=K/\nu \quad (23)$$

for a conventional TDD system, and

$$\mathcal{O}^{\mathrm{ML}}=K\phi/\nu \quad (24)$$

for ML-based TDD system, where

$$\phi\triangleq\frac{JU+V}{PU+JU+V}, \quad (25)$$

where U is the number of LBs.

The NMSE is chosen to evaluate the prediction performance, which is defined as

$$\mathsf{NMSE}\left[l\right]=\mathsf{E}\left\{\frac{1}{K}\sum_{k=1}^{K}\left\|\hat{\mathbf{g}}_k\left[l\right]-\mathbf{g}_k\left[l\right]\right\|_2^2\Big/\left\|\mathbf{g}_k\left[l\right]\right\|_2^2\right\} \quad (26)$$

for the lth step prediction. Some of the important parameters related to the simulation are shown in Table I.

We first verify the accuracy of the CSI prediction for the proposed ML-based architecture, and choose the AR estimator

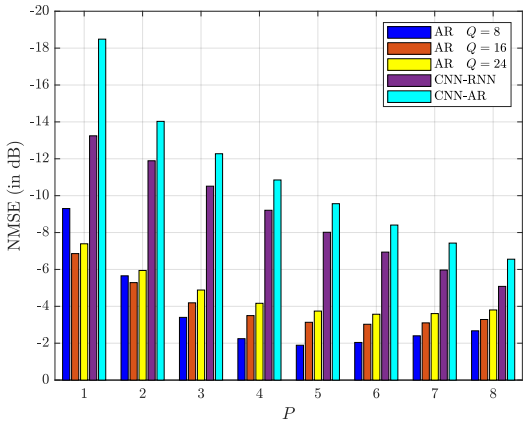

Fig. 4. Convergence of the prediction NMSE among AR predictors, CNN-AR and CNN-RNN with respect to P. Results are shown for $J = 4$, $U = 10$, and $f_n = 0.1$.

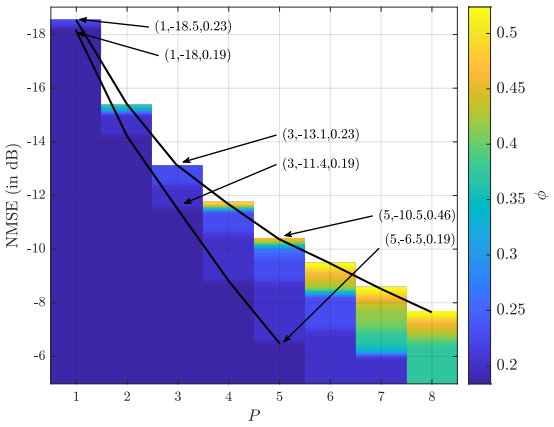

Fig. 5. Optimal ϕ with respect to P under different NMSE requirements. Results are shown as (P, NMSE, ϕ) with $f_n = 0.1$ and $\mathcal{O}^{\text{con}} = 0.3$.

as the benchmark to illustrate the performance improvement. Fig. 3 compares the NMSE of estimation for different predictors in terms of the normalized Doppler shift f_n. It is intuitive that the CNN-AR structure outperforms other predictors in all situations. Compared with simple AR predictors, significant gains can be observed due to the fact that pre-computed AR coefficients are much more precise than real-time computation. Moreover, the performance of CNN-RNN is slightly superior to that of AR predictor which indicates that RNNs indeed support functionalities similar to those provided by AR predictors. Compared with the performance of CNN-RNN in one-step prediction, the accuracy improvement in the second step prediction improves remarkably for small f_n. More importantly, for large f_n, all structures performs poorly. The reason is that the independency of CSI over intervals increases with larger Doppler shifts, which implies that the proposed ML-based TDD scheme is not suitable for super high mobility scenarios.

Fig. 4 shows the average NMSE over 10 LBs against the number of intervals in LB P. Obviously, both ML-based structures outperform the AR predictors, while CNN-AR can further yield at least 1.5 dB gain on every step prediction. The reason is two-fold: One is that the channel series is modelled strictly according to its ACF, and with CNN extracting the aging pattern correctly, the coefficients loaded for AR predictor are precisely accurate. Another one is that the designed NARX-RNN may be not powerful enough to explore the hidden feature within the CSI series; in this case, other time-series architectures, such as long short-term memory RNN, should be considered. It is worth noting that L_p used in simulations is small. In practice, the number of ACF pattern can be hundreds which requires to extend ML-based architecture to a much deeper and larger structure for recognizing. To best of our knowledge, there is no general criterion to design the NN size, and the choose of parameters that depend on L_p remains an implementation-level.

Finally, Fig. 5 illustrates the tradeoff between the CE overhead and prediction accuracy with CNN-AR predictor for ML-based TDD. First, the CE overhead can be sharply reduced by adopting the ML-based TDD scheme. For example, to achieve -18.5 dB of NMSE for $P = 1$ case, the ML-based TDD scheme can save 77% amount of overhead; and given $P = 5$, the ML-based TDD scheme can save more than a half amount of overhead while achieving an NMSE less than -10 dB. Also, the figure illustrates the limits of the proposed ML-based TDD scheme, where a strict prediction requirement is not achievable for multi-step prediction.

V. Conclusion

In this paper, we designed a ML-based TDD scheme as well as the corresponding ML-based architecture to estimate the channels in massive MIMO systems under channel aging effects. Combining the CNN with AR predictor or NARX-RNN, the proposed architecture achieves significant gains in prediction quality, and remarkable tradeoff between prediction quality and CE overhead by leveraging the ACF pattern.

References

[1] N. Palleit and T. Weber, "Time prediction of non flat fading channels," in *Proc. IEEE ICASSP*, May 2011, pp. 2752–2755.

[2] A. K. Papazafeiropoulos, "Impact of general channel aging conditions on the downlink performance of massive MIMO," *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 1428–1442, Feb. 2017.

[3] C. Kong, C. Zhong, A. K. Papazafeiropoulos, M. Matthaiou, and Z. Zhang, "Sum-rate and power scaling of massive MIMO systems with channel aging," *IEEE Trans. Commun*, vol. 63, no. 12, pp. 4879–4893, Dec. 2015.

[4] K. E. Baddour and N. C. Beaulieu, "Autoregressive modeling for fading channel simulation," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1650–1662, July 2005.

[5] T. Wang, C. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep learning for wireless physical layer: Opportunities and challenges," *China Communications*, vol. 14, no. 11, pp. 92–111, Nov. 2017.

[6] K. T. Truong and R. W. Heath, "Effects of channel aging in massive MIMO systems," *Journal of Communications and Networks*, vol. 15, no. 4, pp. 338–351, Aug. 2013.

[7] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015.

[8] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE CVPR*, Jun. 2016, pp. 1646–1654.

[9] R. DiPietro, N. Navab, and G. D. Hager, "Revisiting NARX recurrent neural networks for long-term dependencies," *CoRR*, vol. abs/1702.07805, 2017. [Online]. Available: http://arxiv.org/abs/1702.07805

[10] T. Lin, B. G. Horne, P. Tino, and C. L. Giles, "Learning long-term dependencies in NARX recurrent neural networks," *IEEE Trans. Neural Netw.*, vol. 7, no. 6, pp. 1329–1338, Nov. 1996.