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

Jide, Y., Ngo, H-Q., & Matthaiou, M. (2020). Machine Learning-Based Channel Prediction in Massive MIMO with Channel Aging. *IEEE Transactions on Wireless Communications*, *19*(5), 2960. https://doi.org/10.1109/TWC.2020.2969627

Published in:
IEEE Transactions on Wireless Communications

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

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

Jide Yuan, Hien Quoc Ngo, *Member, IEEE*,
and Michail Matthaiou, *Senior Member, IEEE*

***Abstract*—To support the ever increasing number of devices in massive multiple-input multiple-output (mMIMO) systems, an excessive amount of overhead is required for conventional orthogonal pilot-based channel estimation schemes. To circumvent this fundamental 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 presence of the temporal channel correlation is due to the stationarity of the propagation environment across time. The proposed ML-based predictors involve a pattern extraction implemented via a convolutional neural network, and a CSI predictor realized by an autoregressive (AR) predictor or an autoregressive network with exogenous inputs recurrent neural network. Closed-form expressions for the user uplink and downlink achievable spectral efficiency and average per-user throughput are provided for the ML-based time division duplex schemes. Our numerical results demonstrate that the proposed ML-based predictors can remarkably improve the prediction quality for both low and high mobility scenarios, and offer great performance gains on the per-user achievable throughput.**

***Index Terms*—Achievable spectral efficiency, channel estimation, machine learning, massive multiple-input multiple-output.**

I. Introduction

Channel estimation (CE) is an essential procedure to obtain channel state information (CSI) which is required for the uplink and downlink transmission in massive multiple-input multiple output (mMIMO) [2]. However, with the exponential growth of devices, as well as the spectrum of new applications, an excessive amount of overhead is required to support an ever increasing number of devices. According to [3], the number of devices serviced in one cell may be up to in the order 10^5. As a result, the conventional orthogonal pilot based CE schemes are undoubtedly incompetent considering the limited overhead resources and the latency constraints.

Non-orthogonal multiple access (NOMA) can be considered as one of the possible solutions for this issue [4, 5]. In NOMA, users are partitioned into clusters, and users in one cluster are encouraged to share the same channel or bandwidth resource. Compared to orthogonal CE methods, the pilot length can be reduced by multiple times as it only needs to be larger than the number of clusters rather than the number of users. The main drawback of CE in NOMA is that the CSI of different users in one cluster are coupled, which severely limits the system performance due to the channel mismatch. As an alternative, grant-free user access schemes have been proposed, where user activity detection and CE are performed in one shot employing compressed sensing techniques [6–8]. A key observation made in [6, 7] is that the user activity pattern is sparse, therefore the pilot length can be reduced to the number of activated users instead of all users. However, the performance of grant-free scheme highly depends on the sparsity of user activity pattern. Specifically, if the proportion of the active users is high, the grant-free schemes may fail due to the inherent drawback of compressed sensing techniques [8]. Furthermore, when the number of users exceeds the length of the pilot sequences, the CSI prediction accuracy of such scheme deteriorates significantly.

In practice, the channel does not vary independently across time due to the nearly stationary scattering environment [9]. This phenomenon is known as *channel aging*. The impact of such feature has been characterized in prior literature. For example, [10] studied the achievable rate at the uplink and downlink in mMIMO with channel prediction over an aging channel; [11] showed that channel aging does not impact the power scaling law in mMIMO; [12] points out that the performance degradation caused by such phenomenon can be partially compensated by applying channel prediction, which implies that this practical impairment can be learned and used for estimating CSI. By leveraging this critical observation, the CE overhead has tremendous potential to be reduced by rigorous CSI prediction. An effective method to model an aging channel is an autoregressive (AR) stochastic model whose parameters are computed based on the channel correlation matching property among adjacent coherence intervals [12, 13]. A key observation on the one-step prediction of AR predictor is that the prediction accuracy improves with increasing the AR model order [13]. However, according to the Levinson-Durbin recursion, which is used for computing the model parameters, the model order is bounded by the data amount of previous CSI samples, and the computational complexity is proportional to the square of model order. As it is impractical for a mMIMO base station (BS) to collect and buffer large datasets of previous CSI samples, the performance

A conference version of this paper has appeared in [1].

Manuscript received May 14, 2019; revised November 08, 2019; accepted January 21, 2020. The work of J. Yuan and M. Matthaiou was supported by the RAEng/The Leverhulme Trust Senior Research Fellowship LTSRF1718\14\2. The work of H. Q. Ngo was supported by the UK Research and Innovation Future Leaders Fellowships under Grant MR/S017666/1.

J. Yuan, H. Q. Ngo, and M. Matthaiou are with the Institute of Electronics, Communications and Information Technology (ECIT), Queen's University Belfast, Belfast, BT3 9DT, U.K. (e-mail: y.jide, hien.ngo, m.matthaiou@qub.ac.uk).

of the existing CSI predictors can be rather limited.

Recently, machine learning (ML) based non-linear methods have been successfully applied in wireless communications [1, 14, 15], which motivates us to adopt relevant techniques to forecast CSI. First, by considering the *channel aging* property, the CSI forecasting becomes a typical time series learning problem. In the case of time series learning, recurrent neural networks (RNNs) have been proved to be a potentially powerful tool, which are considered as non-linear approximators to map the undetermined feature within the data. [16–18] RNNs belong to a class of neural networks which are naturally designed for learning sequential data. However, it is extremely difficult for simple RNNs to learn from distant data due to the vanishing gradient problem [16]. Therefore, several powerful architectures, which are able to track the long-term correlation within the sequential data, are proposed, including autoregressive network with exogenous inputs (NARX) [16], long short-term memory (LSTM) [17], gated recurrent unit (GRU) [18], etc. In [19], the authors use LSTM for addressing the power allocation problem in the downlink of mMIMO networks. The use of deep learning significantly reduces the complexity of power allocation, and is able to guarantee near-optimal performance. Also, [20] proposes a real-time CSI feedback framework for point-to-point mMIMO by extending the CsiNet, a deep learning-based NN for reconstructing the CSI, with LSTM. The simulation results demonstrate that the proposed architecture can achieve excellent recovery quality without considerably increasing the feedback overhead. Second, as the temporal correlation is related to the Doppler-shift, in mMIMO scenarios, it is reasonable to assume that CSI series from each antenna at the BS have the same autocorrelation pattern for a particular terminal [12]. By 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 and sentence classification [21–23], i.e, convolutional NN (CNN) to detect the pattern of CSI variation. 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 [24, 25]. 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 [26]. In [27], the authors construct a CNN-based network, called PowerNet, to approximate the reweighted minimum mean-square error (MMSE) algorithm for power allocation. With the help of the PowerNet, the runtime of power allocation reduces to the millisecond level, demonstrating the feasibility of DL for real-time power control in mMIMO. In [28], a CNN-based scheme for predicting downlink (DL) CSI from observed uplink (UL) CSI for frequency division duplex (FDD) is proposed. The new scheme outperforms the classic Wiener filter-based approach in both single-input single-output and MIMO scenarios.

In this paper, we aim to reduce CE overhead via CSI prediction by taking advantage of the autocorrelation across CSI series. By leveraging the same aging pattern of CSI series from massive antennas to a particular user, we are able to create a simple structure NN which can significantly improve the tradeoff between prediction accuracy and CE overhead only by some simple training. Our work is motivated by [29] in the field of video representation and reconstruction, in which a CNN and an RNN are used to extract spatial features and interframe correlation, respectively. Specifically, the main contributions of this paper are summarized as follows.

- We provide an ML-based time-division duplex (TDD) scheme in which CSI is obtained via an ML-based predictor instead of conventional pilot-based channel estimator.
- Two ML-based structures are designed to improve the CSI prediction, namely, CNN combined with AR predictor (CNN-AR) and 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.
- To give a full picture of the proposed ML-based TDD scheme, we provide a closed-form expression for the per-user achievable spectral efficiency (SE) for the ML-based TDD scheme, and consider the tradeoff between CE overhead and achievable throughput. Note that our derivations differ substantially from the bulk of mMIMO literature (e.g. [11, 30]) as we leverage tools of AR prediction.
- We numerically evaluate the performance of the proposed TDD scheme, as well as the ML-based CSI predictors. The results demonstrate that the CNN-AR outperforms other architectures, including CNN-RNN, in terms of prediction accuracy for low and medium mobility scenarios. Regarding the achievable throughput, the proposed ML-based TDD scheme exhibits a remarkable tradeoff between throughput and CE overhead. Even for high mobility scenarios, a significant performance gain can be observed due to the reduced CE overhead.

The rest of this paper is organized as follows: Section II presents the system model including the channel model and the proposed ML-based scheme. Section III presents the proposed ML-based CSI predictors. We provide the discussion of our scheme in Section IV, and the numerical results in Section V, respectively. Section VI summarizes the main observations and proofs are relegated to Appendices.

Notation—Throughout this paper, vectors and matrices are denoted in bold lowercase letters and bold uppercase letters, respectively. The operation $\|\mathbf{A}\|_p$ denotes the p-norm of the matrix $\mathbf{A}$, and $\mathsf{diag}\,(\mathbf{a})$ denotes the diagonal matrix of vector $\mathbf{a}$. The superscripts $(\cdot)^*$ and notation $\mathsf{E}\{\cdot\}$ denote the conjugate transpose and the expectation operations, respectively.

II. System Model

A TDD single-cell multi-user mMIMO system is considered, where a BS having N antennas serves K single-antenna users simultaneously. We assume that the channel is static during each coherence interval, but *it changes from one interval to the next.* Furthermore, the channel in a given coherence interval is correlated with the channels in previous coherence intervals, a phenomenon known as channel aging [10]. More precisely, there is a so called autocorrelation pattern over the channel coherence intervals [31, 32]. The reasoning for this

assumption is two fold: 1) The scattering environment in many scenarios is nonisotropic, which strongly affects the second-order statistics of the channel. 2) The scattering environment shares a high degree of stationarity across several intervals, over which the real and imaginary Gaussian sequences in channel response exhibit cross-correlations [12, 13].

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 (LSF), which remains constant over many coherence time intervals,[1] and $\mathbf{h}_k[l]$ is the small-scale fading. The effective 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 (k,k)th element is β_k, and $\mathbf{H}[l] = [\mathbf{h}_1[l], \ldots, \mathbf{h}_K[l]] \in \mathbb{C}^{N\times K}$.

A. Channel Aging Model

In general, the 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) [13].

We assume that the propagation path experiences two-dimensional isotropic scattering, whose corresponding normalized discrete-time ACF at the BS is [13]

$$R[l] = J_0(2\pi f_n |l|), \tag{3}$$

where $J_0(\cdot)$ is the zeroth-order Bessel function of the first kind, $|l|$ is the delay in terms of the number of coherence intervals, and $f_n = \nu T_s f_{\mathrm{d}}$ represents the normalized Doppler shift, with the maximum Doppler frequency f_{d}, the sampling duration T_s, whilst the number of samples in a coherence interval is ν.

In this paper, we assume the same temporal autocorrelation among all channels from a particular user to the BS antennas.[2] Hence, given the desired ACF as (3) for $l \geqslant 0$, we model the small-scale fading series as [13]

$$\mathbf{h}_k[l] = -\sum\nolimits_{q=1}^{Q} a_{k,q}\mathbf{h}_k[l-q] + \boldsymbol{\omega}[l], \tag{4}$$

where $\boldsymbol{\omega}[l]$ is the complex white Gaussian noise vector independent of $\mathbf{h}_k$ with zero mean and variance

$$\sigma_\omega^2 = R[0] + \sum\nolimits_{q=1}^{Q} a_{k,q}R[-q], \tag{5}$$

and $\{a_{k,q}\}_{q=1}^{Q}$ are the AR coefficients which are evaluated via the Levinson-Durbin recursion via [13]

$$\boldsymbol{a}_k = -\mathbf{R}^{-1}\mathbf{w}, \tag{6}$$

where

$$\boldsymbol{a}_k = [a_{k,1}, \ldots, a_{k,Q}]^T,$$

$$\mathbf{R} = \begin{bmatrix} R[0] & R[-1] & \cdots & R[1-Q] \\ R[1] & R[0] & \cdots & R[2-Q] \\ \vdots & \vdots & \ddots & \vdots \\ R[Q-1] & R[Q-2] & \cdots & R[0] \end{bmatrix},$$

and

$$\mathbf{w} = [R[1], \ldots, R[Q]]^T,$$

with $R[l] = R[-l]$ and $R[0] = 1$.

Remark 1: Given a desired ACF, the fitting accuracy of the AR model improves with higher order Q. However, according to the Levinson-Durbin recursion, Q is upper bounded by the amount of collected CSI samples, which implies that the performance of channel prediction via the AR estimator is limited by the number of coherence intervals used for collecting CSI.

Intuitively, according to (4), the small-scale fading vector is generated as the weighted sum of independent complex white Gaussian vectors in an iterative manner. Thus, the small-scale fading vector follows the Gaussian distribution with zero mean and same variance. Denote by 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 [33]

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

where

$$G_j \triangleq \begin{cases} 1, & j=0, \\ \sum_{q=1}^{j} a_{k,q}G_{j-q}, & j\leqslant Q, \\ \sum_{q=1}^{Q} a_{k,q}G_{j-q}, & j>Q. \end{cases}$$

B. Conventional TDD Scheme

The frame structure in conventional TDD consists of three main blocks that correspondingly represent: CE, UL payload and DL payload phases, in which the channels estimated during the CE phase are further used for UL and DL transmission.

1) CE scheme: We assume that orthogonal pilots are used in the CE phase, and the channel is estimated using the MMSE estimator. Consider the pilot vector assigned to the kth user is $\boldsymbol{\psi}_k$ with $\|\boldsymbol{\psi}_k\|_2^2 = 1$; by considering that the length of pilot signal is equal to number of users, the overall pilot matrix $\boldsymbol{\Psi} = \left[\boldsymbol{\psi}_1^T, \ldots, \boldsymbol{\psi}_K^T\right]^T \in \mathbb{C}^{K\times K}$ satisfies $\boldsymbol{\Psi}\boldsymbol{\Psi}^H = \mathbf{I}_K$. The users use the same power p_{p} to transmit pilots, and the received training signal at the BS is

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

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

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

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

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

[1]This assumption is reasonable since the path-loss is inherently related to the distances between the users and the BS, and thus, the value of β_k changes very slowly with time.

[2]The temporal channel correlation is determined by the scattering environment [13]. For a particular user, the N channels from a user to the BS antennas experience nearly identical scattering environment, which justifies our assumption.

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 $\dot{\beta}_k = \beta_k \sigma^2_{\dot{h}_k}$. Thus, the MMSE estimate of $\mathbf{g}_k[l]$ follows

$$\hat{\mathbf{g}}_k^{\text{mmse}}[l] = \gamma_k^{\text{mmse}} \mathbf{r}_{\text{p},k}[l] \quad \sim \mathcal{CN}(\mathbf{0}, \dot{\beta}_k \gamma_k^{\text{mmse}} \mathbf{I}_N), \tag{11}$$

where $\gamma_k^{\text{mmse}} = \frac{\dot{\beta}_k}{\dot{\beta}_k + \mu}$ with $\mu = \frac{\sigma_n^2}{p_{\text{p}} K}$, and the variance of the estimation error

$$\mathbf{e}_k^{\text{mmse}} = \hat{\mathbf{g}}_k^{\text{mmse}} - \mathbf{g}_k \quad \sim \mathcal{CN}(\mathbf{0}, (1 - \gamma_k^{\text{mmse}}) \dot{\beta}_k \mathbf{I}_N). \tag{12}$$

We recall that the performance of pilot-based schemes deteriorates significantly when the number of devices is higher than the pilot length, due to the pilot contamination phenomenon [34]. Allocating more resources for pilot training will indeed sustain the CE accuracy, however, it will compromise the resources for UL/DL payload transmissions. Thus, a fundamental tradeoff between pilot length and throughput always exists.

2) UL Data Transmission: Without loss of generality, we assume that the K users send their data $s_{\text{u},k}$ $\left(\mathsf{E}\left\{|s_{\text{u},k}|^2\right\} = 1\right)$ to the BS simultaneously with the same power p_{u}. The received signal $\mathbf{y}_{\text{u},k}[l] \in \mathbb{C}^{N \times 1}$ from the kth user at the BS is given by

$$\mathbf{y}_{\text{u},k}[l] = \sqrt{p_{\text{u}}} \sum\nolimits_{k=1}^{K} \mathbf{g}_k[l]\, s_{\text{u},k} + \mathbf{n}_{\text{u}}, \tag{13}$$

where $\boldsymbol{n}_{\text{u}}$ is additive zero-mean Gaussian noise whose elements has variance σ_n^2. Consider maximal-ratio combining (MRC) receiver at the BS, we multiply received signal with the conjugate of estimated CSI, and the detected signal at the BS is given by

$$\begin{aligned} r_{\text{u},k}^{\text{mmse}}[l] = \sqrt{p_{\text{u}}} \sum\nolimits_{n=1}^{N} \sum\nolimits_{k'=1}^{K} (\hat{g}_{k,n}^{\text{mmse}}[l])^* g_{k',n}[l]\, s_{\text{u},k'} \\ + \sum\nolimits_{n=1}^{N} (\hat{g}_{k,n}^{\text{mmse}}[l])^* n_{\text{u},n}, \end{aligned} \tag{14}$$

where $\hat{g}_{k,n}^{\text{mmse}}$ represents the CSI estimated from the kth user to the nth antenna at the BS.

3) DL Data Transmission: In the downlink, the BS treats the estimated channel as the true channel and adopts conjugate beamforming to transmit its signal. Hence, the transmitted signal with power p_{d} from nth antennas at the BS is given by

$$x_{\text{d},n}^{\text{mmse}}[l] = \sqrt{\frac{p_{\text{d}}}{\eta_k^{\text{mmse}}}} \sum\nolimits_{k=1}^{K} \left(\hat{g}_{k,n}^{\text{mmse}}[l]\right)^* s_{\text{d},k}, \tag{15}$$

where $\mathsf{E}\left\{|s_{\text{d},k}|^2\right\} = 1$, and

$$\eta_k^{\text{mmse}}[l] = \mathsf{E}\left\{\left\|\sum\nolimits_{k=1}^{K} \hat{g}_{k,n}^{\text{mmse}}[l]\right\|_2\right\} = \sum\nolimits_{k=1}^{K} \dot{\beta}_k \gamma_{k,n}^{\text{mmse}}[l], \tag{16}$$

is the power control coefficient that normalizes the power of the precoding signal. The received signal at the kth user is given by

$$r_{\text{d},k}^{\text{mmse}}[l] = \sqrt{\frac{p_{\text{d}}}{\eta_k^{\text{mmse}}}} \sum_{n=1}^{N} \sum_{k'=1}^{K} \left(\hat{g}_{k',n}^{\text{mmse}}[l]\right)^* g_{k,n}[l]\, s_{\text{d},k'} + n_{\text{d},k}, \tag{17}$$

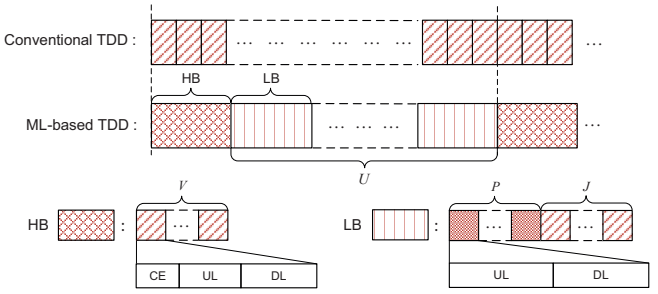

Fig. 1. Conventional TDD versus ML-based TDD. In learning-based block (LB), the CE overhead is removed from the frame structure for P intervals due to the introduction of ML-based CSI prediction.

where $n_{\text{d},k}$ is the additive $\mathcal{CN}(0,1)$ noise at the kth user.

One main drawback of conventional TDD is that the CE overhead becomes extremely large to support massive connectivity demands. Therefore, an ML-based TDD scheme, as well as the corresponding ML-based CSI predictor, are proposed in the following section aiming to reduce the CE resources.

III. ML-BASED CHANNEL FORECASTING APPROACHES

We aim to implement multi-step prediction for CSI to minimize the CE overhead. We first propose an ML-based TDD scheme in which the CE phase is removed from parts of TDD intervals. Then, two types of NN architectures, i.e., CNN-AR and CNN-RNN, are discussed for CSI forecasting. The idea behind the two architectures is identical; more specifically, the architecture adopts a CNN to extract the ACF pattern across the channels, and then, loads the pretrained time-series predictors according to the ACF pattern to forecast the propagation channel.

A. ML-based TDD Scheme

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, while CSI is obtained using an 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 considerations are made in the ML-based TDD scheme:

- In the ML-based TDD scheme, one HB and U LBs form a loop. After each loop, the system restarts a new loop to track the variation of the environment.
- A HB consists of V conventional TDD coherence intervals, in which channels are estimated using the MMSE estimator. These channel estimates are the CSI data used for extracting the aging pattern.
- A LB consists of P coherence intervals without CE phase and J $(J < V)$ conventional TDD coherence intervals, in which the CSI of first P intervals is predicted by the ML-based channel predictor.
- The overall procedure of ML-based TDD is described as follows: i) Collecting CSI data in HBs using MMSE estimator; ii) After a HB, CSI is predicted for P intervals,

and is then updated for the following J intervals via the MMSE estimator in order to further improve the prediction accuracy for the subsequent LB; iii) After U LBs, the system restarts to track the change of the propagation environment.

Similarly to conventional TDD scheme, the uplink and downlink received uplink for the kth user are given by

$$r_{\mathrm{u},k}^{\mathrm{ml}}[l] = \sqrt{p_{\mathrm{u}}} \sum\nolimits_{n=1}^{N} \sum\nolimits_{k'=1}^{K} (\hat{g}_{k,n}^{\mathrm{ml}}[l])^{*} g_{k',n}[l]\, s_{\mathrm{u},k'} + \sum\nolimits_{n=1}^{N} (\hat{g}_{k,n}^{\mathrm{ml}}[l])^{*} n_{\mathrm{u},n} \quad (18)$$

and

$$r_{\mathrm{d},k}^{\mathrm{ml}}[l] = \sqrt{\frac{p_{\mathrm{d}}}{\eta_k^{\mathrm{ml}}}} \sum_{n=1}^{N} \sum_{k'=1}^{K} \left(\hat{g}_{k',n}^{\mathrm{ml}}[l]\right)^{*} g_{k,n}[l]\, s_{\mathrm{d},k'} + n_{\mathrm{d},k}, \quad (19)$$

respectively, where

$$\eta_k^{\mathrm{ml}}[l] = \mathsf{E}\left\{ \left\| \sum\nolimits_{k=1}^{K} \hat{g}_{k,n}^{\mathrm{ml}}[l] \right\|_2 \right\} \quad (20)$$

with $\hat{g}_{k,n}^{\mathrm{ml}}[l]$ representing the CSI estimated by the ML-based scheme, which is detailed in the next section.

B. ML-based channel predictors

Accuracy and timeliness are the major demands for online prediction. A CNN is therefore adopted in both NN architectures for its ability in extracting spatial correlation and its low complexity.

1) CNN-AR Approach: For the proposed architecture shown in Fig. 2(a), a correct detection of the ACF pattern is of paramount importance for accurate channel prediction. The great success of CNNs in image recognition application motivates us to adopt such architecture to extract the ACF pattern by treating the CSI data as the image data. More importantly, as the channels from N antennas to a particular user vary according to the same ACF, and by mapping multiple CSI series into a matrix, the input data can thus be regarded as 2D image pixels.

Therefore, by collecting the CSI data for V intervals in a HB, we separate them into real part and imaginary part, and reform the data as

$$\ddot{\mathbf{G}}_k = \left[{}_{op}\left(\hat{\mathbf{g}}_k^{\mathrm{mmse}}[1]\right), \ldots, {}_{op}\left(\hat{\mathbf{g}}_k^{\mathrm{mmse}}[V]\right) \right], \quad (21)$$

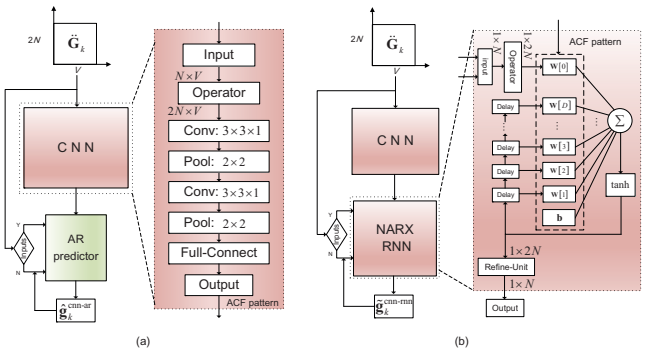

Fig. 2. The proposed CNN-AR and CNN-RNN CSI predictor.

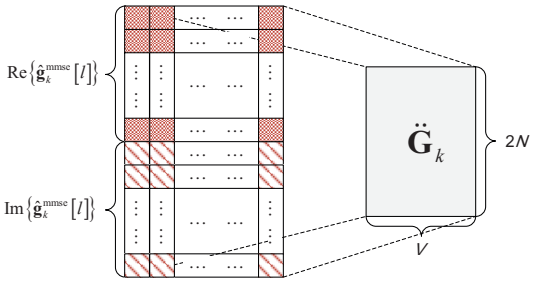

Fig. 3. The schematic diagram of the input structure.

shown in Fig. 3. The corresponding ACF is thought as label λ. The operator ${}_{op}(\cdot)$ is a designed manipulation to map the complex-valued CSI vector into a $2N$-dimensional real-valued vector, i.e., ${}_{op}(\mathbf{g}_k[l]) = \left[\mathrm{Re}\{\mathbf{g}_k[l]\}^T, \mathrm{Im}\{\mathbf{g}_k[l]\}^T\right]^T$. By classifying the pattern of the ACF from $\ddot{\mathbf{G}}_k$, we are able to regenerate the channel series using pre-trained CSI predictor without real time calculation.

The CNN-AR predictor can boost the prediction quality because of two main reasons. First of all, the simple AR predictor has an order which is upper bounded by V according to *Remark* 1; this fundamental constraint limits the achievable performance. Another one is that, a simple AR predictor requires a huge amount of data to obtain the accurate AR coefficients $\{a_q\}_{q=1}^{Q}$, where a CSI series with length of V is obviously insufficient. In contrast, the proposed CNN-AR predictor can avoid these two problems since the predictor can load the pre-computed accurate AR coefficients with arbitrary orders as long as the network recognizes the variation pattern correctly.

We employ *tanh* and *sigmoid* as the activation functions in the convolutional layers and full-connection layers, respectively. We choose the adaptive moment estimation as the optimizer, and use the mean-square error (MSE) as the loss function, which is defined by

$$\mathsf{C}_{\mathrm{cnn}} = \frac{1}{2} \sum_{m=1}^{M} \sum_{l_{\mathrm{P}}=1}^{L_{\mathrm{P}}} \left(\lambda_{l_{\mathrm{P}}}^{m} - \tilde{\lambda}_{l_{\mathrm{P}}}^{m}\right)^2, \quad (22)$$

where M represents the training data amount, L_{P} represents the total number of ACF patterns, $\lambda_{l_{\mathrm{P}}}^{m}$ represents the l_{P}th dimension of pattern label for the mth input data, and $\tilde{\lambda}_{l_{\mathrm{P}}}^{m}$ is the estimates of $\lambda_{l_{\mathrm{P}}}^{m}$.

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, the system loads the pre-computed AR coefficients of the corresponding aging pattern, and predicts the CSI for the subsequent interval as

$$\hat{\mathbf{g}}_k^{\mathrm{cnn-ar}}[l] = -\sum\nolimits_{q=1}^{Q} a_q \hat{\mathbf{g}}_k^{\mathrm{mmse}}[l-q]. \quad (23)$$

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,

$$\hat{\mathbf{g}}_k^{\mathrm{cnn-ar}}[l+l'] = -\sum\nolimits_{q=l'+1}^{Q} a_q \hat{\mathbf{g}}_k^{\mathrm{mmse}}[l+l'-q] - \sum\nolimits_{q'=1}^{l'} a_{q'} \hat{\mathbf{g}}_k^{\mathrm{cnn-ar}}[l+l'-q'], \quad l' \in P, \quad (24)$$

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 demonstrate the feasibility of our scheme, and simplify the system structure for ease of training.

C. 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 RNN is commonly described as [35]

$$\mathbf{f}[l] = f(\mathbf{x}[l], \mathbf{f}[l-1], \boldsymbol{\theta}), \quad (25)$$

where an one-step prediction of $\mathbf{f}[l]$ depends on the previous $\mathbf{f}[l-1]$, input $\mathbf{x}[l]$, and some parameters $\boldsymbol{\theta}$. This simple RNN cannot fit our problem properly since the channel aging model in (4) indicates that the there is a long-term correlation within the sequential CSI data. Hence, we adopt the NARX-RNN to track the correlation, which is generally described as

$$\mathbf{f}[l] = f(\mathbf{x}[l], \mathbf{f}[l-1], \mathbf{f}[l-2], \ldots, \boldsymbol{\theta}). \quad (26)$$

Such an architecture is implemented by introducing *delays* in the original simple RNN where the output has direct connections to the past. In this paper, we adopt a widely used NARX RNN form, specifically given in [35]

$$\mathbf{f}[l] = \tanh\left(\mathbf{W}[0]\mathbf{x}[l] + \sum\nolimits_{d=1}^{D} \mathbf{W}[d]\mathbf{f}[l-d] + \mathbf{b}\right), \quad (27)$$

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 the 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 modification in (27). Taking the channel of the kth user as example, the NARX RNN is described as

$${}_{op}\left(\hat{\mathbf{g}}_k^{\mathrm{cnn-rnn}}[l]\right) = \tanh\Big(\mathbf{W}[0]\,{}_{op}\left(\hat{\mathbf{g}}_k^{\mathrm{mmse}}[l-1]\right) + \sum\nolimits_{d=1}^{D} \mathbf{W}[d]\,{}_{op}\left(\hat{\mathbf{g}}_k^{\mathrm{mmse}}[l-d]\right) + \mathbf{b}\Big), \quad (28)$$

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

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

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

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

$$\mathsf{C}_{\mathrm{cnn-rnn}} = \frac{1}{2}{}_{op}\left(\hat{\mathbf{g}}_k^{\mathrm{cnn-rnn}}[l] - \mathbf{g}_k[l]\right)^H {}_{op}\left(\hat{\mathbf{g}}_k^{\mathrm{cnn-rnn}}[l] - \mathbf{g}_k[l]\right). \quad (29)$$

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

$$\triangle\mathbf{W}[0] = \eta\nabla_{\mathbf{W}[0]}\mathsf{C}_{\mathrm{cnn-rnn}}, \quad (30)$$

where η is a learning rate and $\nabla_{\mathbf{W}[0]}$ is the matrix operator

$$\nabla_{\mathbf{W}[0]} = \begin{bmatrix} \frac{\partial}{\partial w[0]_{1,1}} & \cdots & \frac{\partial}{\partial w[0]_{1,2N}} \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial w[0]_{2N,1}} & \cdots & \frac{\partial}{\partial w[0]_{2N,2N}} \end{bmatrix},$$

where $w[0]_{i,j}$ is the (i,j)-th element of matrix $\mathbf{W}[0]$. 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

$$\nabla_{\mathbf{W}[0]}C = \sum_{n=1}^{N}\left(\hat{\mathbf{g}}_k^{\mathrm{cnn-rnn}}[l] - \mathbf{g}_k[l]\right)^H \nabla_{\hat{\mathbf{g}}_k^{\mathrm{mmse}}[l]}\hat{g}_{k,n}^{\mathrm{cnn-rnn}}[l] \cdot \left(\sum\nolimits_{d=1}^{D}\nabla_{\mathbf{W}[d]}\hat{\mathbf{g}}_k^{\mathrm{mmse}}[l]\right), \quad (31)$$

where $\hat{g}_{k,n}^{\mathrm{cnn-rnn}}$ represents the estimated CSI from kth user to nth antenna at BS. The methodology of training is called backpropagation through time algorithm, and is detailed in [16].

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 the 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 [16]. However, this drawback will not cause a major issue to our formulation since the channel series only have strong relation within adjacent intervals.

IV. Performance Metrics

The following parameters are considered as our performance metrics:

- Prediction quality of ML-based CSI predictors;
- Trade-off between the CE overhead and the prediction quality;
- Trade-off between the average per-user throughput of ML-based TDD scheme and the CE overhead.

We now provide some important definitions for further discussion.

A. Prediction Quality

The normalized MSE (NMSE) is chosen to evaluate the prediction performance, which is defined as

$$\mathsf{NMSE}[l] = \mathsf{E}\left\{\frac{1}{K}\sum_{k=1}^{K}\|\hat{\mathbf{g}}_k[l] - \mathbf{g}_k[l]\|_2^2 \Big/ \|\mathbf{g}_k[l]\|_2^2\right\}, \quad (32)$$

where $\hat{\mathbf{g}}_k[l]$ represents the estimated CSI vector via both the MMSE and ML-based predictor.

B. CE Overhead

As orthogonal pilots are used for conventional TDD, we consider the average ratio of pilot length to the number of samples in a coherence interval ν as our metric, which is defined by

$$\mathsf{O}^{\mathrm{con}} = K/\nu \tag{33}$$

for a conventional TDD system, and

$$\mathsf{O}^{\mathrm{ML}} = \phi K/\nu \tag{34}$$

for ML-based TDD system, where

$$\phi \triangleq \frac{JU+V}{PU+JU+V}. \tag{35}$$

Note that the introduced factor ϕ reveals the reduced CE overhead for the proposed ML-based TDD scheme.

C. Uplink Throughput

Assume that the channel statistics are known at the BS [36], while $\hat{g}^{\mathrm{ce}}_{k,n}$ represents the CSI estimated via either MMSE estimator or ML-based predictors; then, the received signal at lth interval $r^{\mathrm{ce}}_{\mathrm{u},k}[l]$ can be written as

$$r_{\mathrm{u},k}[l] = \mathsf{DS}_k[l]\, s_{\mathrm{u},k} + \mathsf{BU}_k[l]\, s_{\mathrm{u},k} + \sum_{k'\neq k}^{K} \mathsf{UI}_{k'}[l]\, s_{\mathrm{u},k'} + \mathsf{N}_k, \tag{36}$$

where

$$\mathsf{DS}_k[l] = \sqrt{p_{\mathrm{u}}}\mathsf{E}\left\{\sum\nolimits_{n=1}^{N} (\hat{g}^{\mathrm{ce}}_{k,n}[l])^* g_{k,n}[l]\right\}, \tag{37}$$

$$\mathsf{BU}_k[l] = \sqrt{p_{\mathrm{u}}}\left(\sum\nolimits_{n=1}^{N} (\hat{g}^{\mathrm{ce}}_{k,n}[l])^* g_{k,n}[l] - \mathsf{E}\left\{(\hat{g}^{\mathrm{ce}}_{k,n}[l])^* g_{k,n}[l]\right\}\right), \tag{38}$$

$$\mathsf{UI}_{k'}[l] = \sqrt{p_{\mathrm{u}}}\sum\nolimits_{n=1}^{N} (\hat{g}^{\mathrm{ce}}_{k,n}[l])^* g_{k',n}[l], \tag{39}$$

$$\mathsf{N}_k[l] = \sum\nolimits_{n=1}^{N} (\hat{g}^{\mathrm{ce}}_{k,n}[l])^* n_{\mathrm{u},n}, \tag{40}$$

which represents the desired signal, the channel estimation uncertainty, the interference caused by the k'th user, and AWGN, respectively.

We treat the sum of the second to the fourth terms in (36), as the noise-plus-interference (NPI) power, to which the desired signal is uncorrelated, i.e.,

$$\mathsf{E}\{\mathsf{DS}_k \times \mathsf{NPI}_k\} = 0. \tag{41}$$

As the proposed CNN-AR architecture uses the AR predictor to forecast CSI, and by recalling that the uncorrelated Gaussian noise represents the worst case, the achievable uplink SE of the kth user at lth interval for MMSE and CNN-AR estimator is given by (42) at the top of next page. Note that the expression in (42) represents the individual achievable SE for each interval.[3] As the AR predictor has been applied to the estimate channel, we first prove that the channel estimates via AR predictor are uncorrelated with the estimation error.

Lemma 1: Denote by $\hat{g}^{\mathrm{ar}}_{k,n}[l]$ and $e^{\mathrm{ar}}_{k,n}[l]$ the estimated CSI and the estimation error from the kth user to the nth antenna at the BS in the lth interval via the AR predictor, respectively. It can be proved that

$$\mathsf{E}\left\{(\hat{g}^{\mathrm{ar}}_{k,n}[l])^* e^{\mathrm{ar}}_{k,n}[l]\right\} = 0, \quad l \in P. \tag{43}$$

Proof: See Appendix A. ∎

Proposition 1: The worst-case (and thus achievable) uplink SE of the kth user at lth interval using MMSE or CNN-AR predictor is given by

$$R_{\mathrm{u},k}(\gamma^{\mathrm{ce}}_k[l]) = \log_2\left(1 + \frac{N p_{\mathrm{u}} \dot{\beta}_k \gamma^{\mathrm{ce}}_k[l]}{p_{\mathrm{u}}\sum_{k'=1}^{K}\dot{\beta}_{k'} + \sigma^2_{\mathrm{n}}}\right), \tag{44}$$

where $\gamma^{\mathrm{ce}}_k[l] = \gamma^{\mathrm{mmse}}_{k,n=1,\ldots,N}$ when we use MMSE, and

$$\gamma^{\mathrm{ce}}_k[l] = \gamma^{\mathrm{cnn-ar}}_{k,n=1,\ldots,N}[l] = 1 - \mathsf{NMSE}^{\mathrm{cnn-ar}}[l]$$

represents the prediction accuracy of CSI at lth interval when we use the CNN-AR predictor.

Proof: Following a similar method as in [36], and by harnessing the fact that the channel estimates and estimation error for the MMSE estimator and AR predictor are uncorrelated (as proved in *Lemma* 1), we have $\mathsf{DS}_k[l] = \sqrt{p_{\mathrm{u}}} N \dot{\beta}_k \gamma^{\mathrm{ce}}_k[l]$. The power of the channel uncertainty can be then obtained as $\mathsf{E}\left\{|\mathsf{BU}_k[l]|^2\right\} = N p_{\mathrm{u}} \dot{\beta}^2_k \gamma^{\mathrm{ce}}_k[l]$. The power of interference from the k'th user and the power of the noise can be derived as $\mathsf{E}\left\{|\mathsf{UI}_k[l]|^2\right\} = N p_{\mathrm{u}} \dot{\beta}_{k'} \dot{\beta}_k \gamma^{\mathrm{ce}}_k[l]$, and $\mathsf{E}\left\{|\mathsf{N}_k|^2\right\} = N\sigma^2_n \dot{\beta}_k \gamma^{\mathrm{ce}}_k[l]$, respectively. Substituting these results into (42), we complete the proof. ∎

Note that (1) cannot represent the worst-case achievable SE for the architecture using CNN-RNN predictor. This is because it is challenging, and if not impossible, to analytically determine, whether the estimation error of RNN is correlated with the RNN output.

We now consider that the durations of uplink and downlink transmission are identical. The average per-user uplink throughput for the conventional TDD scheme and ML-based TDD schemes are provided in the following proposition.

Proposition 2: The average per-user uplink throughput (in bit/s) for the conventional TDD scheme is given by

$$\mathsf{TP}^{\mathrm{con}}_{\mathrm{u}} = \frac{(1-\mathsf{O}^{\mathrm{con}})W}{2K}\sum_{k=1}^{K} R_{\mathrm{u},k}(\gamma^{\mathrm{mmse}}_k), \tag{45}$$

where W is the bandwidth.

Proof: The result can be obtained directly from (33) and *Proposition* 1. ∎

Proposition 3: The average per-user uplink throughput of the ML-based TDD scheme with CNN-AR predictor is given

[3] According to (42), it is required to estimate the LSF at each coherence interval, which, unfortunately, is not possible for a single subcarrier system since small scale fading also remains constant during a coherence interval. However, in the widely used OFDM systems, the expectation over small scale fading is still reasonable since the LSF across all subcarriers is the same [30, 37].

$$R_{\mathrm{u},k}\left(\gamma_k^{\mathrm{ce}}\left[l\right]\right)=\log_2\left(1+\frac{\left|\mathsf{DS}_k\left[l\right]\right|^2}{\mathsf{E}\left\{\left|\mathsf{BU}_k\left[l\right]\right|^2\right\}+\sum_{k'\neq k}^{K}\mathsf{E}\left\{\left|\mathsf{UI}_{k'}\left[l\right]\right|^2\right\}+\mathsf{E}\left\{\left|\mathsf{N}_k\right|^2\right\}}\right). \tag{42}$$

TABLE I
SIMULATION PARAMETERS.

Number of ACF patterns L_{P}[4]	10
Number of intervals in HB V	8
Transmit power p_{p}, p_{u}, p_{d}	0 dBm, 0 dBm, 0 dBm
Bandwidth W	10 MHz
Background noise power σ_{n}^2	-174 dBm/Hz
Hidden layers in full-connection	3
Nodes in layers	1024, 1024, 256

by

$$\mathsf{TP}_{\mathrm{u}}^{\mathrm{cnn-ar}}\left(P,J,V,U\right)=\phi\mathsf{TP}_{\mathrm{u}}^{\mathrm{con}}+\frac{\phi UW}{2\left(JU+V\right)K}\sum_{p=1}^{P}\sum_{k=1}^{K}R_{\mathrm{u},k}\left(\gamma_k^{\mathrm{cnn-ar}}\left[p\right]\right). \tag{46}$$

Proof: The result can be obtained directly from (34) and *Proposition* 1. ■

D. Downlink Throughput

Using a similar methodology as in *Proposition* 1, we obtain the worst-case (achievable) downlink SE in the following proposition.

Proposition 4: The worst-case (and thus achievable) downlink SE of the kth user at lth interval using MMSE and CNN-AR predictor is given by

$$R_{\mathrm{d},k}\left(\gamma_k^{\mathrm{ce}}\left[l\right]\right)=\log_2\left(1+\frac{Np_{\mathrm{d}}\left(\dot{\beta}_k\gamma_k^{\mathrm{ce}}\left[l\right]\right)^2}{\eta_k^{\mathrm{ce}}\left[l\right]\left(p_{\mathrm{d}}\dot{\beta}_k+\sigma_{\mathrm{n}}^2\right)}\right). \tag{47}$$

Similarly, the average per-user downlink throughput is given in the next proposition.

Proposition 5: The average per-user downlink throughput of the conventional TDD scheme is given by

$$\mathsf{TP}_{\mathrm{d}}^{\mathrm{con}}=\frac{\left(1-\mathsf{O}^{\mathrm{con}}\right)W}{2K}\sum_{k=1}^{K}R_{\mathrm{d},k}\left(\gamma_k^{\mathrm{mmse}}\right). \tag{48}$$

Proposition 6: The average per-user downlink throughput of the ML-based TDD scheme with CNN-AR predictor is given by

$$\mathsf{TP}_{\mathrm{d}}^{\mathrm{cnn-ar}}\left(P,J,V,U\right)=\phi\mathsf{TP}_{\mathrm{d}}^{\mathrm{con}}+\frac{\phi UW}{2\left(JU+V\right)K}\sum_{p=1}^{P}\sum_{k=1}^{K}R_{\mathrm{d},k}\left(\gamma_k^{\mathrm{cnn-ar}}\left[p\right]\right). \tag{49}$$

Therefore, the overall transmission throughput can be calculated as

$$\mathsf{TP}^{\mathrm{cnn-ar}}\left(P,J,V,U\right)=\mathsf{TP}_{\mathrm{u}}^{\mathrm{cnn-ar}}\left(P,J,V,U\right)+\mathsf{TP}_{\mathrm{d}}^{\mathrm{cnn-ar}}\left(P,J,V,U\right), \tag{50}$$

and the maximum total throughput among all possible configurations is described by

$$\mathsf{TP}_{\max}^{\mathrm{cnn-ar}}=\max_{P,J,V,U}\mathsf{TP}_{\max}^{\mathrm{cnn-ar}}\left(P,J,V,U\right). \tag{51}$$

V. 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 meter for each user, i.e., the distance between any user and the 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 [38]

$$\beta_k\left(d_k\right)=30.8+24.2\log_{10}\left(d_k\right). \tag{52}$$

Some of the important parameters related to the simulation are shown in Table I. Training, and testing sets have 80,000, and 20,000 samples, respectively, for offline training. The epochs and the batch size are set as 300 and 50 with learning rate equals 0.001. 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. However, to best of our knowledge, there is no general criterion to design the NN size, and the choice of parameters that depend on L_p remains an implementation-level.

We first verify the performance of feature extraction of CNN included in the proposed ML-based architecture. Fig. 4 shows the prediction accuracy with respect to number of epochs. Clearly, after 300 of epochs training, the CNN can recognize the aging pattern with over 95% accuracy for both kernel sizes. More importantly, we observe that the proposed NN structure with the 3×3 kernel outperforms that with 5×5 kernel. The reason is that, in contrast to image recognition problem, only the columns of input data series $\ddot{\mathbf{G}}_k$ are correlated while the rows of $\ddot{\mathbf{G}}_k$ are independent. Therefore, a kernel with larger size may not improve the decision accuracy because of increased "interference".

[4]While $L_p=10$ is seemingly a low value, our simulations indicate that the ACF pattern can be extracted by the NN and the channel prediction can be improved as long as the network recognizes the variation pattern accurately enough. Moreover, a larger value of L_p will not substantially improve the achievable throughput. This is because by reducing the "spacing" between normalized Doppler shifts, the distinction between each set of AR coefficients becomes marginal, making the prediction results considerably similar. At the cost of increased complexity of NN structure, such slight improvement on throughput becomes questionable. Nevertheless, a more powerful NN structure that can recognize more ACF patterns can further improve the prediction quality, and we prefer to leave this to our future work.

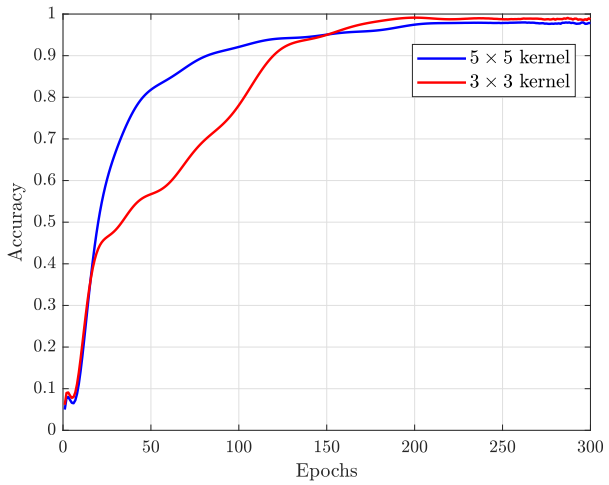

Fig. 4. Comparison of performance with different kernel size of CNN for $V = 8$.

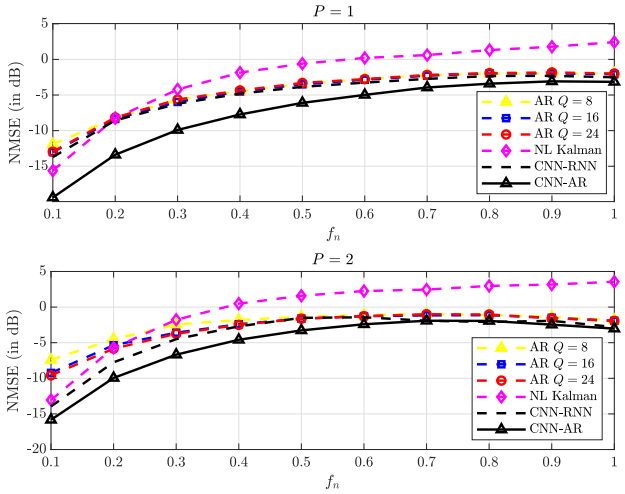

Fig. 5. Comparison of prediction NMSE among the AR predictors, NL Kalman predictor, CNN-AR and CNN-RNN with respect to the normalized Doppler shift f_n. Results are shown for $J = 4$ and $U = 10$.

Then, we verify the accuracy of the CSI prediction for the proposed ML-based architecture, and choose the AR estimator and nonlinear (NL) Kalman predictor [37, 39] as the benchmarks to illustrate the performance improvement. It is worth noting that the parameters in NL Kalman predictor require real-time training, and the results for NL Kalman in Fig. 5 and Fig. 6 are shown for $V = 500$. Fig. 5 compares the NMSE of estimation of different predictors against the normalized Doppler shift f_n. It is intuitive that the CNN-AR structure outperforms other predictors in all situations. Compared with simple AR and NL Kalman predictors, significant gains can be observed due to the fact that the pre-computed AR coefficients are much more precise than real-time computations based on limited CSI information. 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 perform poorly. The reason is that the independency of CSI over different intervals increases with larger Doppler shifts. This implies that the proposed ML-based TDD scheme is not so suitable for super high mobility scenarios.

Fig. 6 shows the NMSE of estimation of different predictors against the prediction step P. Obviously, the performance of CNN-AR architecture is superior than that of other predictors on every step prediction, and significant gains can be observed for CNN-RNN compared with AR predictors from 2nd to 12th step prediction. Moreover, via the comparison between $Q = 16$ and $Q = 24$ of the AR predictor, the limited improvement implies that the prediction accuracy can be hardly improved by expanding the order of AR predictor. This is because the real-time calculation of AR coefficients is not accurate enough with only a small number of CSI data as inputs. In addition, the poor performance of the simple AR predictor implies the infeasibility of such architecture in multi-step prediction. Also, the CNN-AR predictor provides at least 3 dB of gain than CNN-RNN for each step prediction, which shows that the CNN-AR is a better ML-based predictor on CSI prediction for an aging channel. This is due to the fact the aging patterns are formulated by the AR model, thereby making AR predictors represent the variation of channel more precisely than the approximation used by NARX-RNN. It must be pointed out that, although the prediction accuracy of NL Kalman outperforms AR predictors when $P < 14$, such predictor since it requires real-time training, thereby requires a huge amount of overhead.

According to the proposed ML-based TDD scheme shown in Fig. 1, U is an important parameter that determines how often the system resets itself. Fig. 7 illustrates the NMSE of estimation for different predictors against U. First, we observe that the NMSE of all predictors converge. Specifically, for the ML-based architectures, the performance of CNN-AR

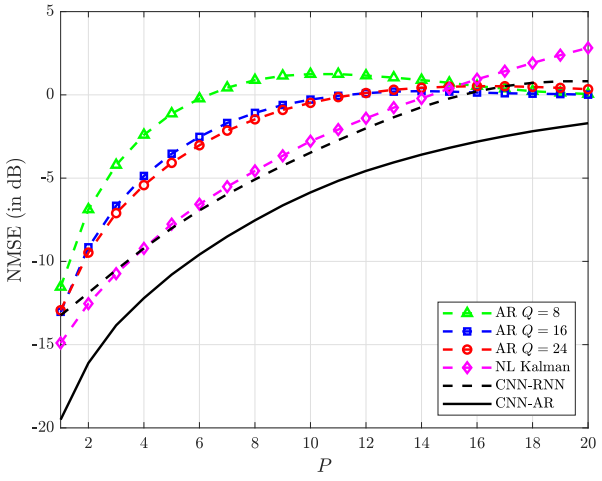

Fig. 6. Comparison of prediction NMSE among the AR predictors, NL Kalman predictor, CNN-RNN and CNN-AR for the first P intervals, i.e., $U = 1$. Results are shown for $f_n = 0.1$, $J = 4$ and $V = 8$.

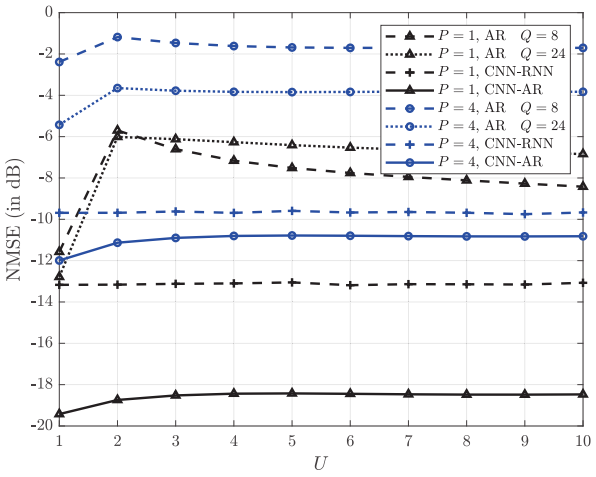

Fig. 7. Comparison of prediction NMSE among the AR model, CNN-RNN and CNN-AR with respect to U. Results are shown for $J = 4$, $V = 8$ and $f_n = 0.1$.

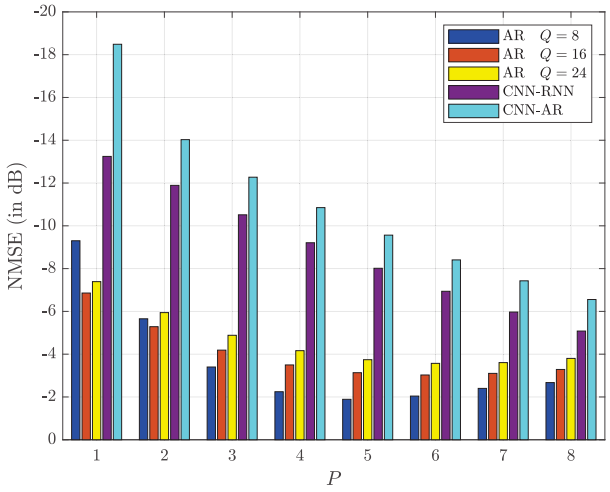

Fig. 8. Convergence of the prediction NMSE among the AR predictors, CNN-AR and CNN-RNN with respect to P. Results are shown for $J = 4$, $U = 10$, and $f_n = 0.1$.

predictor slightly deteriorates with increasing U, while that of CNN-RNN predictor remains constant. For AR predictors, the prediction performance experiences a slight improvement after a sharp decline. This is because AR coefficients can be updated continuously via the Levinson-Durbin recursion during the communication. However, since the real-time computation of AR coefficients depends strongly on the CSI data, using predicted CSI as true CSI to update coefficients may cause more error. ML-based architectures can avoid this fundamental problem since the predictor parameters are pre-computed. More importantly, the slight performance decline indicates that it is not necessary to reset the system frequently for ML-based TDD schemes which can further reduce the CE overhead.

Fig. 8 shows the average NMSE over 10 LBs against the number of intervals in a LB P. Obviously, both ML-based structures outperform the AR predictors, while CNN-AR can

TABLE II
ESTIMATION ERROR NMSE (in dB)

J	P	AR model			CNN-RNN	CNN-AR
		$Q = 8$	$Q = 16$	$Q = 24$		
1	1	-3.89	-4.61	-4.91	-12.5	-19.1
	2	-3.76	-4.58	-4.84	-10.7	-15.9
	4	-3.56	-3.78	-4.08	-8.15	-11.6
2	1	-6.73	-5.29	-4.93	-12.9	-18.9
	2	-3.89	-4.35	-4.64	-11.1	-14.7
	4	-2.98	-3.74	-4.03	-8.66	-10.5
	8	-2.42	-3.29	-3.35	-4.83	-4.29
4	1	-8.44	-6.91	-6.14	-13.3	-18.4
	2	-5.53	-5.75	-5.98	-11.9	-13.9
	4	-3.43	-3.63	-3.93	-9.16	-10.8
	8	-2.23	-3.23	-3.38	-5.23	-6.54
8	1	-15.7	-8.77	-8.15	-14.1	-19.4
	2	-10.9	-7.80	-7.30	-12.4	-15.8
	4	-5.54	-5.45	-5.55	-9.73	-12.1
	8	-0.82	-1.02	-2.26	-5.66	-7.17
	16	-0.12	-0.28	-0.54	-0.21	-2.31

further yield at least 1.5 dB gain on every step prediction. The reason is two-fold: One is that the channel series is modeled 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 LSTM RNN [17], should be considered. We also provide more results in TABLE II. From the table, a significant observation is that there is slight improvement of the prediction quality by increasing J for CNN-RNN and CNN-AR, which implies that the prediction results are mainly determined by the latest input, meaning that the ML-based TDD can achieve most of the performance gain with $J = 1$. Therefore, we conclude that, the CE overhead can be further scaled down by reducing the conventional TDD intervals in a LB.

The above results showcase that CNN-AR outperforms the other architectures in any circumstances. Therefore, we inves-

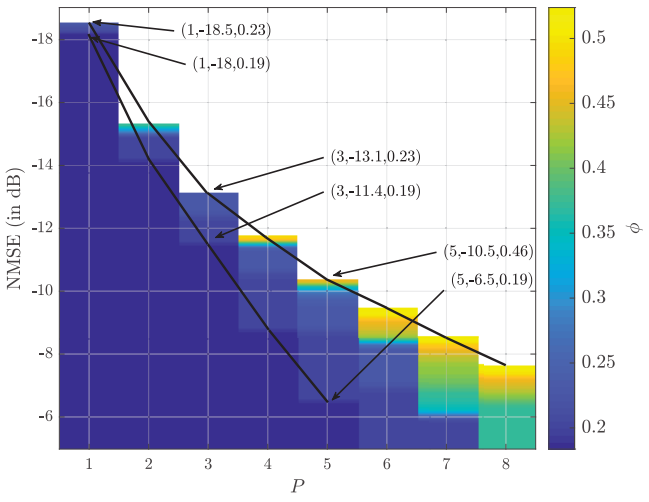

Fig. 9. Optimal ϕ with respect to P under different NMSE requirements. Results are shown as (P, NMSE, ϕ) with $f_n = 0.1$ and $\mathsf{O}^{\mathrm{con}} = 0.3$.

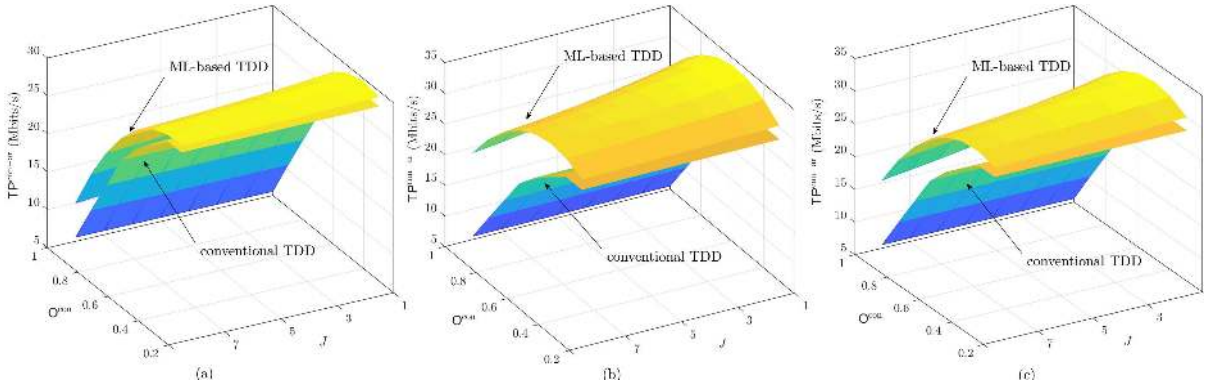

Fig. 10. The average per-user throughput $\mathsf{TP}^{\mathrm{cnn-ar}}$ under different CNN-AR configurations. (a). $P = 2$; (b). $P = 5$; (c). $P = 8$. Results are shown for $f_n = 0.1$.

tigate the tradeoff between the CE overhead and prediction accuracy with CNN-AR predictor for ML-based TDD, and is shown in Fig. 9. 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 for $P = 5$, 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.

We now verify the average per-user throughput by considering the CNN-AR in ML-based TDD scheme in Fig. 10. The results of conventional TDD are included in the figure as benchmark. All the results are averaged over 1000 runs. A key observation from the three subfigures is that the per-user throughput $\mathsf{TP}^{\mathrm{cnn-ar}}$ undergoes a significant decline after a slight increase with increasing $\mathsf{O}^{\mathrm{con}}$. The phenomenon is different from the conventional TDD in which the average per-user throughput decreases monotonically with respect to the number of users. The reason is that by reducing the CE overhead, the BS can allocate more resources for data transmission, thereby we can improve $\mathsf{TP}^{\mathrm{cnn-ar}}$ by reasonably increasing the number of users. In some extreme cases, such as, when all resources are used for CE to support massive users in conventional TDD, i.e., $\mathsf{O}^{\mathrm{con}} = 1$, the ML-based TDD still provides considerable serving quality, e.g., at least 10 Mbits/s and 20 Mbits/s for $P = 2$ and $P = 5$, respectively. Also, we note that $\mathsf{TP}^{\mathrm{cnn-ar}}$ decreases more or less with increasing J in all simulations, which implies that the configuration of J that maximizes $\mathsf{TP}^{\mathrm{cnn-ar}}$ is 1 which is consistent with our previous results in TABLE II. Moreover, by comparing the average per-user throughput across the three subfigures, we note that an optimal P that maximizes $\mathsf{TP}^{\mathrm{cnn-ar}}$ exists. However, the optimal P highly depends on prediction quality which, in turn, varies with respect to the ACF pattern.

To evaluate the performance gain of ML-based TDD scheme, we illustrate the joint impact of P and $\mathsf{O}^{\mathrm{con}}$ on the ratio of $\mathsf{TP}^{\mathrm{cnn-ar}}$ to $\mathsf{TP}^{\mathrm{con}}$ in Fig. 11. First, different from the behavior of per-user achievable throughput, we observe that the performance gain increases monotonically with increasing $\mathsf{O}^{\mathrm{con}}$. This is reasonable because of the inherently poor behavior of conventional TDD for massive user scenarios. Moreover, regarding the parameter P, when $\mathsf{O}^{\mathrm{con}} = 0.2$, the ratio of the per-user achievable throughput between ML-based TDD scheme and conventional TDD scheme decreases, and reduces below than 1. In contrary, when $\mathsf{O}^{\mathrm{con}} > 0.6$, the ML-based TDD scheme can obtain significant performance gain. This result indicates that, the proposed ML-based TDD scheme is most suitable for future dense wireless networks in which the support of massive user scenarios is a prerequisite.

To further illustrate the performance gain provided by the optimal configuration of CNN-AR, we introduce an indicator $\delta = \frac{\mathsf{TP}^{\mathrm{cnn-ar}}_{\mathrm{max}}}{\mathsf{TP}^{\mathrm{con}}}$ as our metric. Fig. 12 illustrates δ against the normalized Doppler-shift f_n. First of all, a significant gain can be observed even for the high speed scenarios. The reason is, although the small-scale fading is hard to track in high mobility scenarios, the channel statistics, i.e., the LSF can be observed from predicted CSI. More specifically, the performance improvement decreases with higher f_n since the temporal channel correlation gradually vanishes in high speed

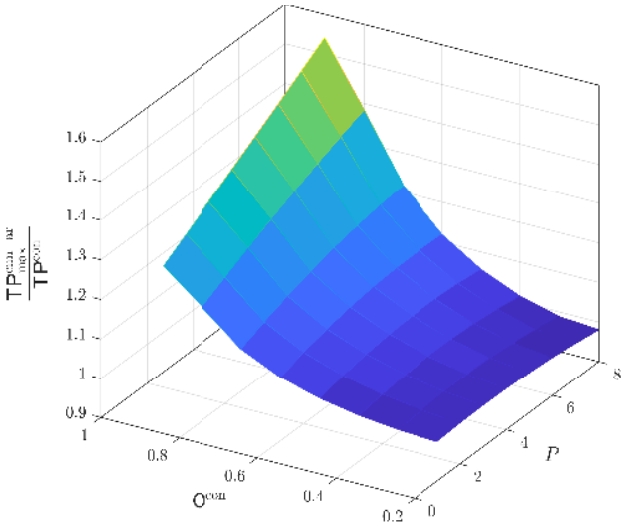

Fig. 11. Comparison of $\frac{\mathsf{TP}^{\mathrm{cnn-ar}}}{\mathsf{TP}^{\mathrm{non}}}$ with respect to P and $\mathsf{O}^{\mathrm{con}}$. Results are shown for $f_n = 0.1$ and $J = 1$.

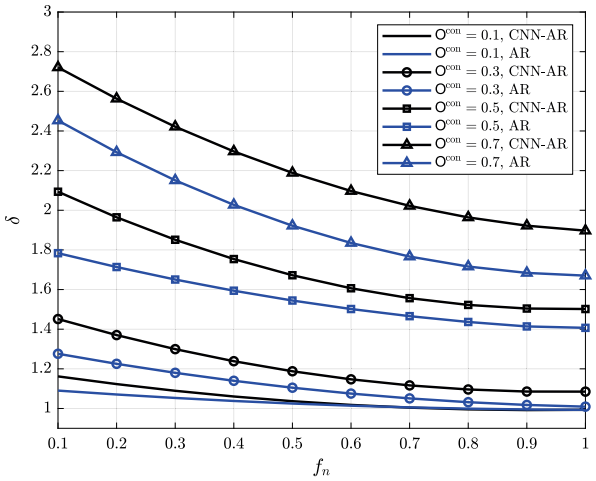

Fig. 12. Comparison of δ with respect to f_n.

scenarios. Also, we observe more performance gain for an increasing number of users. For instance, when $f_n = 0.1$, the ML-based TDD scheme can achieve more than 2 times per-user achieve throughput than conventional TDD does for $\mathsf{O}^{\mathrm{con}} = 0.5$, while only a slight improvement can be obtained for $\mathsf{O}^{\mathrm{con}} = 0.1$. Besides, the performance of ML-based architecture, i.e., CNN-AR, is significantly superior to that of conventional AR predictor, and such benefit gets bigger with increasing number of users. This phenomenon demonstrates that our proposed ML-based TDD scheme can support massive users for both low and high mobility scenarios.

VI. Conclusion

To reduce the excessive amount of CE overhead, we designed an ML-based TDD scheme as well as the corresponding ML-based architecture to predict the channels in massive MIMO systems under channel aging effects. Different from conventional CE schemes, the proposed ML-based architecture extracts the ACF pattern via a CNN, and loads a pre-trained NARX-RNN or an AR predictor to forecast the CE. The simulation results demonstrate that the proposed architecture achieves significant gains in prediction quality, and remarkable tradeoff between prediction quality and CE overhead by leveraging the ACF pattern. In terms of the average per-user throughput, the proposed ML-based TDD scheme can offer remarkable gains for both low and high mobility scenarios, and such improvements are even more significant for massive user cases. These characteristics showcase the great potential of the proposed ML-based TDD scheme for future wireless networks in the 5G era.

Note that the proposed ML-based TDD scheme, as well as the ML-based CSI predictor, can be extended by considering the effects of shadowing and time delaying in the channel model. These extensions will be part of our future work.

Appendix A
Proof of Lemma 1

To obtain *Lemma 1*, we first prove the following lemma as preliminary knowledge.

Lemma 2: For an aging channel modeled as in (4), the estimated CSI via MMSE at any interval is uncorrelated with the estimation error at any interval; mathematically speaking,

$$\mathsf{E}\left\{\left(\hat{g}_{k,n}^{\mathrm{mmse}}[i]\right)^{*} e_{k,n}^{\mathrm{mmse}}[j]\right\} = 0, \qquad \forall i,j. \tag{53}$$

Proof: We can prove this lemma using a recursive procedure. Firstly, consider the channel estimation for a typical channel coefficient $g_{k,n}$ in the first interval. It is well known that, in a coherent interval, the channel estimate via MMSE estimator is uncorrelated with estimation error, i.e.,

$$\mathsf{E}\left\{\left(\hat{g}_{k,n}^{\mathrm{mmse}}[1]\right)^{*} e_{k,n}^{\mathrm{mmse}}[1]\right\} = 0. \tag{54}$$

Then, for the second interval, substituting the channel model (2) and (4) into (11), we reach

$$\hat{g}_{k,n}^{\mathrm{mmse}}[2] = \gamma_k^{\mathrm{mmse}}\left(-a_1 g_{k,n}[1] + \omega[2] + \frac{1}{\sqrt{K p_{\mathrm{p}}}}\mathbf{n}[2]\,\boldsymbol{\psi}_k^{H}\right). \tag{55}$$

Rewriting $\hat{g}_{k,n}[1]$ as a function of $\hat{g}_{k,n}^{\mathrm{mmse}}[1]$ using (11), we have

$$g_{k,n}[1] = \frac{\hat{g}_{k,n}^{\mathrm{mmse}}[1]}{\gamma_k^{\mathrm{mmse}}} - \frac{1}{\sqrt{K p_{\mathrm{p}}}}\mathbf{n}[1]\,\boldsymbol{\psi}_k^{H}. \tag{56}$$

Substituting (56) into (55), we can observe that

$$\mathsf{E}\left\{\left(\hat{g}_{k,n}^{\mathrm{mmse}}[2]\right)^{*} e_{k,n}^{\mathrm{mmse}}[1]\right\} = 0. \tag{57}$$

Similarly, the estimation error at the second interval can be described as

$$\begin{aligned} e_{k,n}^{\mathrm{mmse}}[2] &= \hat{g}_{k,n}^{\mathrm{mmse}}[2] - g_{k,n}[2] \\ &= \left(\gamma_k^{\mathrm{mmse}} - 1\right)\left(-a_1 g_{k,n}[1] + \omega[2]\right)\frac{\gamma_k^{\mathrm{mmse}}}{\sqrt{K p_{\mathrm{p}}}}\mathbf{n}[2]\,\boldsymbol{\psi}_k^{H}. \end{aligned} \tag{58}$$

Again, rewriting $e_{k,n}^{\mathrm{mmse}}[2]$ as a function of $e_{k,n}^{\mathrm{mmse}}[1]$ using (11), we have

$$\begin{aligned} e_{k,n}^{\mathrm{mmse}}[2] &= -a_1 e_{k,n}^{\mathrm{mmse}}[1] \\ &\quad + \gamma_k^{\mathrm{mmse}}\left(\frac{1}{\sqrt{K p_{\mathrm{p}}}}\left(a_1\mathbf{n}[1] + \mathbf{n}[2]\right)\boldsymbol{\psi}_k^{H} + \omega[2]\right), \end{aligned} \tag{59}$$

which leads to

$$\mathsf{E}\left\{\left(\hat{g}_{k,n}^{\mathrm{mmse}}[1]\right)^{*} e_{k,n}^{\mathrm{mmse}}[2]\right\} = 0. \tag{60}$$

Regarding to the subsequential intervals, we can prove that the estimated CSI at any interval is uncorrelated with estimation error at any interval using same procedure. ■

Having established *Lemma* 2, we are ready to prove *Lemma* 1. Denoting $p_1, \ldots, p_P$ the P intervals in a LB, we can estimate the channel according to (23)

$$\hat{g}_{k,n}^{\mathrm{ar}}[p_1] = -\sum\nolimits_{q=1}^{Q} a_q \hat{g}_{k,n}^{\mathrm{mmse}}[p_1 - q]. \tag{61}$$

Recalling that the real channel is given by

$$g_{k,n}[p_1] = -\sum\nolimits_{q=1}^{Q} a_q g_{k,n}[p_1 - q] + \omega[p_1], \tag{62}$$

the estimation error $e_{k,n}^{\mathrm{ar}}[p_1]$ can be calculated as

$$e_{k,n}^{\mathrm{ar}}[p_1] = \sum_{q=1}^{Q} a_q \hat{g}_{k,n}^{\mathrm{mmse}}[p_1 - q] - \sum_{q=1}^{Q} a_q g_{k,n}[p_1 - q] + \omega[p_1]$$
$$= \sum_{q=1}^{Q} a_q e_{k,n}^{\mathrm{mmse}}[p_1 - q] + \omega[p_1], \quad (63)$$

which leads to

$$\mathsf{E}\left\{(\hat{g}_{k,n}^{\mathrm{ar}}[p_1])^* e_{k,n}^{\mathrm{ar}}[p_1]\right\}$$
$$= \mathsf{E}\left\{ - \sum_{q'=1}^{Q} \sum_{q=1}^{Q} a_q a_{q'} (\hat{g}_{k,n}^{\mathrm{mmse}}[p_1 - q])^* e_{k,n}^{\mathrm{mmse}}[p_1 - q'] \right\}$$
$$\overset{(a)}{=} 0, \quad (64)$$

where (a) is obtained via *Lemma* 2.

Now, consider the prediction of the second interval p_2, we have

$$\hat{g}_{k,n}^{\mathrm{ar}}[p_2] = -\sum\nolimits_{q=2}^{Q} a_q \hat{g}_{k,n}^{\mathrm{mmse}}[p_2 - q] - a_1 \hat{g}_{k,n}^{\mathrm{ar}}[p_1], \quad (65)$$

and the estimation error can be addressed as

$$e_{k,n}^{\mathrm{ar}}[p_2] = \sum_{q=2}^{Q} a_q e_{k,n}^{\mathrm{mmse}}[p_2 - q] + a_1 e_{k,n}^{\mathrm{ar}}[p_1] + \omega[p_2]. \quad (66)$$

After some manipulations, the expectation of channel estimates and the estimation error can be computed as

$$\mathsf{E}\left\{(\hat{g}_{k,n}^{\mathrm{ar}}[p_2])^* e_{k,n}^{\mathrm{ar}}[p_2]\right\} = \mathsf{E}\left\{-a_1^2 (\hat{g}_{k,n}^{\mathrm{ar}}[p_1])^* e_{k,n}^{\mathrm{ar}}[p_1]\right\}$$
$$= 0. \quad (67)$$

Similarly, we can obtain $\mathsf{E}\left\{(\hat{g}_{k,n}^{\mathrm{ar}}[p_i])^* e_{k,n}^{\mathrm{ar}}[p_i]\right\} = 0,\ i > 2$ step by step. Thus, we complete the proof.

REFERENCES

[1] J. Yuan, H. Q. Ngo, and M. Matthaiou, "Machine learning-based channel estimation in massive MIMO with channel aging," in *Proc. IEEE SPAWC*, Jul. 2019, pp. 1–6.

[2] S. Sesia, I. Toufik, and M. Baker, Eds., *LTE: The UMTS Long Term Evolution: From Theory To Practice*. John Wiley and Sons, 2011.

[3] 5G PPP Architecture Working Group, "View on 5G architecture," *White Paper*, Jun. 2016.

[4] Y. Sait, Y. Kishiyama, A. Benjebbour, T. Nakamura, A. Li, and K. Higuchi, "Non-orthogonal multiple access (NOMA) for cellular future radio access," in *Proc. IEEE VTC*, Jun. 2013, pp. 1–5.

[5] L. Dai, B. Wang, Y. Yuan, S. Han, C. I, and Z. Wang, "Non-orthogonal multiple access for 5G: Solutions, challenges, opportunities, and future research trends," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 74–81, Sep. 2015.

[6] H. Zhu and G. B. Giannakis, "Exploiting sparse user activity in multiuser detection," *IEEE Trans. Commun*, vol. 59, no. 2, pp. 454–465, Feb. 2011.

[7] X. R. X. Xu and V. K. N. Lau, "Active user detection and channel estimation in uplink CRAN systems," in *Proc. IEEE ICC*, Jun. 2015, pp. 2727–2732.

[8] L. Liu and W. Yu, "Massive connectivity with massive MIMO Part I: Device activity detection and channel estimation," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2933–2946, Jun. 2018.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] H. Ye, G. Y. Li, and B. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.

[16] 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.

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

[18] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. EMNLP*, Oct. 2014, pp. 1724-1734.

[19] L. Sanguinetti, A. Zappone, and M. Debbah, "Deep learning power allocation in massive MIMO," 2018. [Online]. Available: https://arxiv.org/abs/1812.03640

[20] T. Wang, C. We, S. Jin, and G. Y. Li, "Deep learning-based CSI feedback approach for time-varying massive MIMO channels," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 416–419, Apr. 2019.

[21] 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.

[22] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proc. ACL*, Jun. 2014, pp. 655–665.

[23] Yoon Kim, "Convolutional neural networks for sentence classification," in *Proc. EMNLP*, Oct. 2014, pp. 1746–1751.

[24] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," in *Proc. ACM MM*, 2015, pp. 689–692.

[25] M. Danelljan, G. Hager, F. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE ICCVW*, Dec. 2015, pp. 58–66.

[26] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of CNN activations," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1511.05879

[27] T. V. Chien, T. N. Canh, E. Björnson, and E. G. Larsson, "Power control in cellular massive MIMO with varying user activity: A deep learning solution," 2019. [Online]. Available: http://arxiv.org/abs/1901.03620

[28] M. Arnold, S. Dörner, S. Cammerer, S. Yan, J. Hoydis, and S. Brink, "Enabling FDD massive MIMO through deep learning-based channel prediction," 2019. [Online]. Available: http://arxiv.org/abs/1901.03664

[29] K. Xu and F. Ren, "Csvideonet: A real-time end-to-end learning framework for high-frame-rate video compressive sensing," in *Proc. IEEE WACV*, Mar. 2018, pp. 1680–1688.

[30] Q. Bao, H. Wang, Y. Chen, and C. Liu, "Downlink sum-rate and energy efficiency of massive MIMO systems with channel aging," in *Proc. IEEE WCSP*, Oct. 2016, pp. 1–5.

[31] A. K. Papazafeiropoulos and T. Ratnarajah, "Uplink performance of massive MIMO subject to delayed CSIT and anticipated channel prediction," in *Proc. IEEE ICASSP*, May 2014, pp. 3162–3165.

[32] ——, "Deterministic equivalent performance analysis of time-varying massive MIMO systems," *IEEE Trans. Wireless Communi.*, vol. 14, no. 10, pp. 5795–5809, Oct. 2015.

[33] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time series analysis: Forecasting and control*. John Wiley & Sons, 2015.

[34] J. Jose, A. Ashikhmin, T. L. Marzetta, and S. Vishwanath, "Pilot contamination and precoding in multi-cell TDD systems," *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2640–2651, Aug. 2011.

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

[36] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1834–1850, Mar. 2017.

[37] S. Kashyap, C. Mollén, E. Björnson, and E. G. Larsson, "Performance analysis of TDD massive MIMO with Kalman channel prediction," in *Proc. IEEE ICASSP*, Mar. 2017, pp. 3554–3558.

[38] J. Hoydis, K. Hosseini, S. t. Brink, and M. Debbah, "Making smart use

of excess antennas: Massive MIMO, small cells, and TDD," *Bell Labs Techn. J.*, vol. 18, no. 2, pp. 5–21, Aug. 2013.

[39] S. Gifford, C. Bergstrom, and S. Chuprun, "Adaptive and linear prediction channel tracking algorithms for mobile OFDM-MIMO applications," in *Proc. IEEE MILCOM*, Oct. 2005, pp. 1298–1302.

Jide Yuan (S'16–M'19) received the B.S. degree in communication engineering from the Ocean University of China, Qingdao, China in 2011. He then received the M.S. degree from the Harbin Engineering University, Harbin, China in 2014 and Ph.D. degree in Communication and Information Systems from the Southeast University, China in 2018, respectively. From October 2018 through September 2019, he was with the Institute of Electronics, Communications and Information Technology, Queen's University Belfast, U.K. working as a Postdoctoral Research Fellow. He is currently a Research Fellow with the Department of Electronic Systems, Aalborg University, Denmark. His current research interests include random matrix theory, massive MIMO, machine learning and large intelligent surface.

Hien Quoc Ngo received the B.S. degree in electrical engineering from the Ho Chi Minh City University of Technology, Vietnam, in 2007, the M.S. degree in electronics and radio engineering from Kyung Hee University, South Korea, in 2010, and the Ph.D. degree in communication systems from Linköping University (LiU), Sweden, in 2015. In 2014, he visited the Nokia Bell Labs, Murray Hill, New Jersey, USA. From January 2016 to April 2017, Hien Quoc Ngo was a VR researcher at the Department of Electrical Engineering (ISY), LiU. He was also a Visiting Research Fellow at the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, UK, funded by the Swedish Research Council.

Hien Quoc Ngo is currently a Lecturer at Queen's University Belfast, UK. His main research interests include massive (large-scale) MIMO systems, cell-free massive MIMO, physical layer security, and cooperative communications. He has co-authored many research papers in wireless communications and co-authored the Cambridge University Press textbook *Fundamentals of Massive MIMO* (2016).

Dr. Hien Quoc Ngo received the IEEE ComSoc Stephen O. Rice Prize in Communications Theory in 2015, the IEEE ComSoc Leonard G. Abraham Prize in 2017, and the Best PhD Award from EURASIP in 2018. He also received the IEEE Sweden VT-COM-IT Joint Chapter Best Student Journal Paper Award in 2015. He was an *IEEE Communications Letters* exemplary reviewer for 2014, an *IEEE Transactions on Communications* exemplary reviewer for 2015, and an *IEEE Wireless Communications Letters* exemplary reviewer for 2016. He was awarded the UKRI Future Leaders Fellowship in 2019. Dr. Hien Quoc Ngo currently serves as an Editor for the IEEE Wireless Communications Letters, Digital Signal Processing, and IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. He was a Guest Editor of IET Communications, special issue on "Recent Advances on 5G Communications" and a Guest Editor of IEEE Access, special issue on "Modelling, Analysis, and Design of 5G Ultra-Dense Networks", in 2017. He has been a member of Technical Program Committees for several IEEE conferences such as ICC, GLOBECOM, WCNC, and VTC.

Michail Matthaiou (S'05–M'08–SM'13) was born in Thessaloniki, Greece in 1981. He obtained the Diploma degree (5 years) in Electrical and Computer Engineering from the Aristotle University of Thessaloniki, Greece in 2004. He then received the M.Sc. (with distinction) in Communication Systems and Signal Processing from the University of Bristol, U.K. and Ph.D. degrees from the University of Edinburgh, U.K. in 2005 and 2008, respectively. From September 2008 through May 2010, he was with the Institute for Circuit Theory and Signal Processing, Munich University of Technology (TUM), Germany working as a Postdoctoral Research Associate. He is currently a Reader (equivalent to Associate Professor) in Multiple-Antenna Systems at Queen's University Belfast, U.K. after holding an Assistant Professor position at Chalmers University of Technology, Sweden. His research interests span signal processing for wireless communications, massive MIMO systems, hardware-constrained communications, mm-wave systems and deep learning for communications.

Dr. Matthaiou and his coauthors received the IEEE Communications Society (ComSoc) Leonard G. Abraham Prize in 2017. He was awarded the prestigious 2018/2019 Royal Academy of Engineering/The Leverhulme Trust Senior Research Fellowship and recently received the 2019 EURASIP Early Career Award. His team was also the Grand Winner of the 2019 Mobile World Congress Challenge. He was the recipient of the 2011 IEEE ComSoc Best Young Researcher Award for the Europe, Middle East and Africa Region and a co-recipient of the 2006 IEEE Communications Chapter Project Prize for the best M.Sc. dissertation in the area of communications. He has co-authored papers that received best paper awards at the 2018 IEEE WCSP and 2014 IEEE ICC and was an Exemplary Reviewer for IEEE COMMUNICATIONS LETTERS for 2010. In 2014, he received the Research Fund for International Young Scientists from the National Natural Science Foundation of China. He is currently the Editor-in-Chief of Elsevier Physical Communication and a Senior Editor for IEEE WIRELESS COMMUNICATIONS LETTERS. In the past, he was an Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, Associate Editor/Senior Editor for IEEE COMMUNICATIONS LETTERS and was the Lead Guest Editor of the special issue on "Large-scale multiple antenna wireless systems" of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS.