I. INTRODUCTION

The fifth generation (5G) wireless communication techniques aim at connecting everything in the physical world as neurons in a neural network. To achieve this goal, 5G systems should be able to interact with environment and jointly optimize ever-increasing numbers of key performance indicators (KPIs) such as user experiences of virtual reality (VR) and internet of things (IoT). While network optimization becomes more complicated, the air interface is also facing the same problem taking into consideration the increasing antennas and higher frequency. Recently, artificial intelligence (AI) techniques are becoming the state of the art in many fields. While many researchers have already used AI algorithms in air interface blocks, most of air interface technologies are perfectly modeled by information theory, hence their replacements with AI should be carefully checked [1]. Other than replacement, we believe the AI-enhanced Air Interface (AleAI) technology will be promising, which combines the benefits of certainty and uncertainty.

In modern radio systems, the most uncertain part of air interface is the wireless channel. The estimated channel quickly become outdated due to rapid channel variation, and using the outdated information in transmission will degrade the system performance [2], [3]. Also, precoding of massive multiple-input multiple-output (mMIMO) requires excessive channel state information (CSI) feedback, if future CSI is predictable, feedback overhead and pilot resources can be partially saved. However, the wireless channel is a superposition of sinusoids contributed by changing reflectors and scatters, which is time-varying and intangible, moreover, the estimated channel has error and noises, these problems make the channel prediction hard to realize. Therefore, we need an appropriate channel model, which can not only fit the statistical properties of channel, but also memorize some small scale properties to some extent. As the environment around a base station will not change rapidly, the memorization capability should be helpful. This situation encountered in channel prediction reminds us the similar problems in language processing, where a word can have a statistical word frequency and many different meanings. In this paper, we take wireless channel model as a language model, and the time-varying channel as talking in this language, while the realistic noisy estimated channel can be compared with mumbling. Accordingly, to predict the mumble of wireless channel, is to do channel prediction under a noisy situation in a language modeling way.

To the best of our knowledge, in previous channel prediction works, only one or two features are extracted from a complex-valued channel coefficient, while some researchers focus only on the power [4], two features are normally real and imaginary parts or power and phase values. Though the power of channel is the only interest for some applications, it’s beneficial to predict complex-valued channel coefficients and obtain power from the squared magnitude of the complex value [5]. Previous channel prediction works are performed with various models, such as auto-regressive (AR) model [6], sinusoidal model [7], complex-valued neural networks [8], deep neural networks [9] and so on. However, we are wondering if it is sufficient to extract only two features from a complex-valued channel coefficient. The features should be complex enough, so that they can capture the underlying small scale property while the long term property is kept. In language modeling, word embedding [10] or word vector is the most popular technique. Word embedding is a vector of numbers, which can capture
The main contributions of this paper are as follows: seq2seq models are trained to translate sequence from one domain (e.g., sentences in English) to sequence in another domain. For prediction task, as future elements should be known information at past, the encoder side is channel in the past so that it can be bi-directional, and the decoder side is channel in the future. The numerical results with simulation and realistic data show robust, and realistic channel prediction with superior performance relative to channel estimation is attainable.

Recurrent neural network (RNN) is good at solving sequential problems. For prediction task, as future elements should not be seen, one RNN can only train an input sequence in unidirectional way. However, sequence-to-sequence (seq2seq) models can realize bi-directional training by introducing two RNNs for encoder and decoder respectively. Normally, seq2seq models are trained to translate sequence from one domain (e.g., sentences in English) to sequence in another domain (e.g., voices in Mandarin), while in this paper, we creatively exploit seq2seq models in the channel prediction task, the encoder side is channel in the past so that it can be bi-directional, and the decoder side is channel in the future. The main contributions of this paper are as follows:

- The first proposal and implementation of channel prediction algorithms with hundreds of features representing each complex-valued channel coefficient instead of conventional two features.
- The first demonstration of using seq2seq models and its variants in time series channel prediction. It turns out that the encoder and decoder of seq2seq models with different lengths can be perfect containers for different time spans of past and future signals.
- The numerical results with simulation and realistic data indicate the channel prediction model is reliable and robust, and realistic channel prediction with superior performance relative to channel estimation is attainable by using firstly proposed prediction diversity technique.

The rest of this paper is organized as follows. The algorithms for channel predictor are presented in Section II. The modeling and predicting method is introduced in Section III. This is followed by numerical results and discussions in Section IV. In Section V, conclusions are given.

II. CHANNEL PREDICTOR

Training a language model with neural networks is popular for modern natural language processing (NLP) tasks. Normally, a vocabulary should be extracted from the corpus to be learned, which comprises all high-frequency words. For a RNN-based language modeling, the conditional probability of each word is computed after all the previous words passing through the RNN cell, where Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) is typically implemented. Corresponding to each word in the vocabulary, a word embedding representing this word is a vector of learnable numbers, which are gradually changed during training until the meaning of this word is encoded in these numbers. Thanks to the high dimensional representation, a word embedding can carry much more information than a single word.

As we see the noisy estimated channel as mumbling with channel language, a vocabulary should also be extracted from the channel. While the combinations of power and phase of channel coefficients are too many for vocabulary, the Channel Changes (CC) is finite with a practical precision, if we can predict CC from the past, we can predict future channel. Channel coefficients in a past time span can be expressed as $h(t - M : t - 1)$, which contains the past $M$ sampled channel, its prediction target is the future $N$ samples, i.e., $h(t : t + N - 1)$. In this work, CC are calculated in the first place, CC sequence can be expressed as $h'(t-x) = h(t-x) - h(t-x-1)$, where $x = 1 : M - 1$, then $h'$ is used to predict $h'$ with future CC. At time $t$, as $h'(t) = h(t) - h(t-1)$, and $h(t-1)$ is known information at past, $h(t)$ can be obtained with $h(t) = h(t-1) + h'(t)$. Analogously, the $(t+y)^{th}$ sample in the future can be calculated by $h(t+y) = h(t-1) + \sum_{k=0}^{y} h'(t+k)$ with predicted $h'$, where $y = 0 : N - 1$.

As each CC in $h'$ can be regarded as a word, we introduce Vocabulary of Channel Changes (VCC) to map $h'$ into IDs. VCC includes top $X$ frequently appearing CC in $h'$ while the rest $L$ CC are out-of-vocabulary (OOV). Taking into consideration the numerical precision of $h'$, the size of data, the expressive power of model and GPU memories, an appropriate $X$ should be chosen. Small $X$ makes modeling inaccurate while large $X$ slows down convergence speed. Small $L$ introduces interference while large $L$ leads to too many unknown predictions. In this work, $X \approx 2000$ and $L \approx 500$ are chosen, and length of $h'$ is in the order of tens to hundreds of millions. In VCC, an unique ID (usually an integer) is assigned to each CC, embeddings of these integers are chosen, and length of $X$ is in the order of tens to hundreds of millions. In VCC, an unique ID (usually an integer) is assigned to each CC, embeddings of these integers are inputs of neural networks instead of channel coefficients. By using these IDs, channel coefficients are transformed from numbers to categories. Table I shows the top 10 most and least frequently occurring CC and their corresponding IDs in a VCC extracted from a realistic measured channel, where $X$ is the size of this vocabulary. From the top 10 most frequently occurring CC and their frequencies, we can observe pairs of CC, such as $t + 0.02 - 0.02i$ and $t' - 0.02 + 0.02i$, each pair of CC with similar frequencies is symmetrical about the origin or axis, if the occurring frequencies of a pair of CC are imbalanced, the distribution of channel will move away from the origin. CC with occurring frequency higher than 10 are kept in this VCC, and therefore the least frequency of CC in this vocabulary is 11. An ‘unk’ token is usually added to vocabulary whose ID is zero, so that all occurrences of OOV CC can be replaced with this ‘unk’ token. As mentioned above, every CC is represented by a $e$-length embedding, and $e$ is around 400 in this work. Accordingly, number of parameters utilized to represent this channel is $X \times e \approx 2000 \times 400 = 800k$ without considering the weights in neural networks. It is noteworthy that this work provides a fundamental method to enlarge the expressive power of channel predictor, the model used here may be over-parameterized and the ideas such as model pruning can be implemented for practice to speed up the computation. By looking up the VCC, each CC in $h'$ is
replaced by its corresponding ID, then \( h' \) is transformed into a new sequence with integers, which can be fed into neural networks for training and predicting. The predicted integers are then transformed back into future \( h' \) using VCC. With \( h' \), \( h \) in future can be obtained.

### III. MODELING AND PREDICTING

To achieve the purpose of predicting the \( N \) following CC given the \( M \) preceding CC, two solutions are proposed. The natural language generation (NLG) solution uses one RNN combined with backpropagation through time (BPTT) algorithm while neural machine translation (NMT) solution uses seq2seq models comprising two RNNs. Normally, these two RNNs belong to different domains, e.g., two kinds of languages, which have different vocabularies. However, in our solution, we use the same vocabulary for both RNNs as they are modeling the same channel, but only encoder is for past and decoder is for future. In Fig. 1, a block diagram is shown to explain the NMT based channel predictor. At the encoder side, time series with integer IDs (7, 4, 15), representing their CC counterparts in VCC, are fed into the reusable RNN cell, which consists an embedding (EMD) layer and stacked LSTM or GRU networks, the embedding layer contains the embedding vectors for all IDs. The hidden states (solid lines) are transferred through time steps in forward and backward direction, this bidirectional variant can provide better understanding of signals but should only stay at encoder side, because time order in future cannot be reversed. Usually, only the last hidden state is forwarded to decoder, while in seq2seq models with attention variant, intermediate hidden states with different attention values (weights) are also sent to decoder (dashed lines). Apparently, the attention and bidirectional variants should work together to achieve the largest performance gain. Except the hidden states input, IDs at the last time step of encoder should also be sent to decoder (dashed dotted lines), alternatively, forwarding number zero also works as the hidden states already possess enough information. The output of decoder is a predicted sequence, by comparing it to ground truth sequence, loss or prediction error can be obtained. It is noteworthy that the lengths of input and output sequence can be different. Furthermore, size of train set can be increased by sliding time series with a window. For example, the input can be (4, 15, 237) when output is (159, 1463) by sliding the original data with one time step.

![Block diagram of seq2seq model based channel predictor.](image)

**Fig. 1.** Block diagram of seq2seq model based channel predictor.

| Top10 most in VCC | CC | Freq. | ID value | Top10 least in VCC | CC | Freq. |
|-------------------|----|-------|---------|-------------------|----|-------|
| 1 \( +0.02+0.02i \) | 535211 | X-9 | \( +0.03+0.05i \) | 12 |  |
| 2 \( -0.02+0.02i \) | 536795 | X-8 | \( -0.03+0.02i \) | 12 |  |
| 3 \( -0.02+0.02i \) | 535761 | X-7 | \( -0.03+0.02i \) | 12 |  |
| 4 \( -0.02+0.02i \) | 534726 | X-6 | \( -0.1+0.2i \) | 12 |  |
| 5 \( -0.02+0.02i \) | 371729 | X-5 | \( -0.06+0.2i \) | 11 |  |
| 6 \( -0.01+0.02i \) | 371946 | X-4 | \( -0.05+0.2i \) | 11 |  |
| 7 \( +0.01+0.02i \) | 371856 | X-3 | \( +0.01+0.2i \) | 11 |  |
| 8 \( -0.02+0.01i \) | 371778 | X-2 | \( +0.2+0.01i \) | 11 |  |
| 9 \( -0.01+0.02i \) | 371682 | X-1 | \( +0.05+0.2i \) | 11 |  |
| 10 \( +0.01+0.02i \) | 371673 | X | \( +0.2+0.05i \) | 11 |  |

**TABLE I**

| Top10 most in VCC | CC | Freq. | ID value | Top10 least in VCC | CC | Freq. |
|-------------------|----|-------|---------|-------------------|----|-------|
| 1 \( +0.02+0.02i \) | 338211 | X-9 | \( +0.03+0.05i \) | 12 |  |
| 2 \( +0.02+0.02i \) | 536795 | X-8 | \( -0.03+0.02i \) | 12 |  |
| 3 \( +0.02+0.02i \) | 535761 | X-7 | \( -0.03+0.02i \) | 12 |  |
| 4 \( +0.02+0.02i \) | 534726 | X-6 | \( -0.1+0.2i \) | 12 |  |
| 5 \( +0.02+0.02i \) | 371729 | X-5 | \( -0.06+0.2i \) | 11 |  |
| 6 \( +0.01+0.02i \) | 371946 | X-4 | \( -0.05+0.2i \) | 11 |  |
| 7 \( +0.01+0.02i \) | 371856 | X-3 | \( +0.01+0.2i \) | 11 |  |
| 8 \( +0.01+0.02i \) | 371778 | X-2 | \( +0.2+0.01i \) | 11 |  |
| 9 \( +0.01+0.02i \) | 371682 | X-1 | \( +0.05+0.2i \) | 11 |  |
| 10 \( +0.01+0.02i \) | 371673 | X | \( +0.2+0.05i \) | 11 |  |
figure, attentions through different tests are similar, which is essential for a robust channel prediction model. For the 10th predicted CC, the main attention is on the last hidden state, which means the latest signal is more and more important when predicting samples further ahead.

To simplify, $M:N$ is used to represent the operation of utilizing $M$ preceding CC to predict $N$ following CC. In practice, this $M:N$ prediction operation should be repeated to ensure the prediction accuracy as the prediction error will always accumulate in time. Though predicting channels infinitely into the future is impossible, a better prediction algorithm can extend $N$ under a limited prediction error. In channel prediction task, regarding the time interval between sampled channel or Doppler shift, the unit of $M$ and $N$ can also be time span or wavelength. For example, task in Fig. 2 can be expressed as a 30:10 samples channel prediction task, as the time interval between sampled channel is 1 millisecond (ms), it’s also a 30:10 ms channel prediction task. As the user speed in this case is 3km/h, the highest Doppler frequency is about 10Hz at center frequency 3.45GHz, the distance traveled during the prediction can be measured in wavelengths $W$,

$$W = T f_d = \frac{T v}{\lambda} = \frac{T v f_c}{c}$$  \hspace{1cm} (1)

where $T$ is prediction time span, $f_d$ is doppler frequency, $v$ is user velocity in m/s, $\lambda$ is carrier wavelength, $f_c$ is the carrier frequency, and $c$ is the speed of light. By calculating equation 1, the previous task can be expressed as a 0.3:0.1 wavelengths in space.

In this paper, each RNN cell has a two-layered GRU with hidden size equals to 1000. For the sake of fairness, we train 2 epochs for every model with Adam as optimizer and same annealing recipe for learning rate. The training time for one epoch is around 6 hours with 12GB memory. After training, the predictor is tested with an $M:N$ predicting operation, then every $N$-length segmented data in channel sequence $h_t$ is replaced with $N$ predicted samples, so that a new predicted channel sequence $\hat{h}_t$ is obtained. The normalized mean square error (NMSE) of the prediction can be expressed as equation 2,

$$NMSE = \frac{E\{||h_t - \hat{h}_t||_F^2\}}{E\{||h_t||_F^2\}}$$  \hspace{1cm} (2)

where $\| \cdot \|_F$ is Frobenius norm. For a radio system, what really matters is throughput. Therefore, in some of the following experiments, block error rate (BLER) performance is calculated with the predicted channel coefficients.

**IV. NUMERICAL RESULTS AND DISCUSSIONS**

Performance evaluation of this channel predictor is carried out on two scenes: link level simulation (LLS) in section IV-A and realistic measured scene in section IV-B. Orthogonal Frequency Division Multiplexing (OFDM) system with 20MHz bandwidth and Quadrature Phase Shift Keying (QPSK) modulation is used for both scenes. All the data and results of LLS presented here are based on the tapped-delay line (TDL) channel model [13], which takes care of mmWave propagation ranging from 6 to 100GHz. Specifically, TDL-C is designed for non-line-of-sight (NLOS) propagation. In this paper, a long delay spread, 300ns, is set in the TDL-C model and noise-free channel coefficients on 31 propagation paths are introduced for training. For realistic measured scene, indoor and outdoor measurements are carried out at 3.45GHz carrier frequency.

**A. Simulation**

Channel predictor is investigated on LLS without impacts of noise, interference and hardware imperfections. NLG and NMT solutions are compared at user speed 100km/h, the unit of $M:N$ is symbol, where 14 symbols equal to one millisecond. In this experiment, train set is channel impulse response (CIR) from 31 paths in time domain for 100 seconds, while test set is obtained from another 10 seconds with a different random seed. In Fig. 3a, performance of decoding with ideal channel coefficients, i.e., ideal channel estimation (ICE), is shown as a baseline, NLG and NMT solution with varying $M$ are investigated. When $N$ is fixed to 14 symbols, larger $M$ introduces more preceding symbols for learning and predicting. From Fig. 3a we can tell that larger $M$ leads to better performance. From Fig. 3b, it is shown that the performance gains are caused by higher accuracy of prediction according to NMSE value. However, reducing prediction error by increasing $M$ will finally saturate at $M = M_s$, which is around 30 samples shown in previous work [2]. Based on the neural-network algorithm we used, the memory span of this channel predictor is six times larger, i.e., 196 samples for NLG solution.

The saturate point shows fitting and memorizing capabilities of the model, while for data-driven models, size of train set may also limit these capabilities. In Fig. 3b, results of NMT solution trained using the same dataset with NLG solution are shown as solid line and plus-sign markers, we can see that when $M$ is larger than 98, prediction is getting worse. This indicates the dataset is not big enough to define this NMT model with large $M$. Fortunately, time series sequence is continuous, if we slide the original dataset with a time offset, i.e., sliding window technique, an augmented dataset is obtained. Though this new dataset is all from the old sequence, it’s beneficial for NMT based channel predictor to understand that $M$ and $N$ samples are all from a continuous long sequence. In this experiment, a sliding window is chosen
solution. The predicted symbols are set for this experiment, i.e., a 30:30 ms NMT source model. Parameters with rate. It is important to use the same VCC with that from the for NLG, NMT and NMT combined with sliding window (SW) technique. with varying M

Fig. 3. (a) BLER performance of decoding under NLG and NMT solutions with varying M at N = 14 symbols (b) NMSE vs. M at N = 14 symbols for NLG, NMT and NMT combined with sliding window (SW) technique.

to make the dataset five times larger, and the prediction result is shown as dashed line, we can see that prediction error is going down at 196 samples and nearly saturate with this size of dataset. In the comparison of NMT and NLG results, NMT result with 14:14 symbols outperforms all NLG results on BLER performance and NMSE value, furthermore, NMT result of 98:14 symbols can reach the ICE performance bound. It means the prediction error of this solution is below the critical value for perfectly decoding with imperfect channel coefficients [14]. Comparing with NLG channel prediction, the fitting and memorizing capabilities of NMT solution are much stronger.

Transfer learning is efficient in solving a problem by using the knowledge of a related problem. As we have already learned channel model at a user speed of 100km/h, prediction of a slower or faster time-variant channel should be natural. In this experiment, we work on a channel with 3km/h user speed, as the channel changes slower, we can predict a longer time span with the same number of samples, time span can be increased to M × S : N × S, where S is the sampling rate. It is important to use the same VCC with that from the source model. Parameters with S = 30 and M = N = 14 symbols are set for this experiment, i.e., a 30:30 ms NMT solution. The predicted h’ can be recovered to original time interval by interpolating operation. As a result of the sampling operation, train set is 30 times smaller than original dataset.

For data-driven models, the size of dataset is essential, which we have already proved in the previous experiment. If we learn from scratch with this small train set, even after 20 epochs training, around 1dB performance loss at BLER = 0.1 is shown in Fig. 4. However, if the well-trained 100km/h channel model is directly used to predict sampled 3km/h channel, 0.8dB performance loss can be reduced. From these results, we can see that even without learning, predicting slowly-varying channel with rapidly-varying channel model is feasible. With transfer learning, the performance is further improved, where the 100km/h channel model is being modified to fit the sampled 3km/h channel according to the input data. In addition, further performance gains are achievable with attention variants as shown in Fig. 4. It is noteworthy that for higher SNR, the performance gap between prediction and estimation is larger, which is reasonable because estimation error become smaller and prediction error dominates the performance. However, for cases such as adaptive transmission, estimated channel is outdated, performance difference should be compared between predicted channel and outdated channel, where predicted one should be much better.

B. Measurement

In realistic scene, measured channel are forwarding to channel predictor. Indoor and outdoor channel measurements are conducted in an environment shown as a schematic diagram in Fig. 5, which includes a 20m × 20m room at 4th floor for indoor measurement, a 50m × 50m garden with surrounding buildings for outdoor measurement, and lines with arrowhead indicating routes and directions of moving user equipment (UE). In the room, floor to ceiling height is 3m and the heights of base station (BS) and UE are 2m and 1m. UE speed for routes 1 to 12 is 3km/h while route 13 is blocked behind a pillar, and route 14 is walking in the corridor. For outdoor scene, BS is pointed to the garden through a window 15 meters above the ground and routes 15 to 16 is moving in circle with 3km/h. In addition, UE uses 1 antenna while BS uses 1 antenna for outdoor, 2 antennas for indoor. Different from simulation scene, channel frequency response (CFR) is collected. The least-square estimated channel is truncated at time domain while significant taps are kept. The time-varying
channel from different subcarriers are put together to form one long sequence, which can largely increase the train set.

For indoor scene, channel predictor with attention based 30:10 ms NMT solution is used and the generalization capability of this channel predictor is being investigated. Not only the channel from different subcarriers are trained together, channel from different ports corresponding to 2 transmitting antennas are also trained together, the goal is to generalize the statistical channel model from training on limited route and transferring the knowledge on predicting other routes. Imagine such a scenario, where the AI-enhanced indoor distributed antenna system (IDAS) was trained with channel of an UE in a room and is capable to predict channel of other UEs in other rooms with similar layout. In this experiment, training is carried out with train set composed of data from routes 1 to 9, which are following the same route, but only collected separately. Data from routes 10 to 14 are directly predicted using the well-trained model, i.e., transferring knowledge directly without transfer learning (see section IV-A). The prediction error is shown in table II, for comparison, routes 8 and 9 are also predicted with this model. We can tell from the table that prediction error of routes 10 to 13 are close to trained routes 8 and 9, moreover, though route 10 is following the same route with train set, route 12 and NLOS route 13 have better prediction accuracy than it, which shows the generalization capability of this model is good. These results also indicate that transferring channel model among different locations, ports and times is feasible. However, route 14 in the corridor seems following different properties of propagation as the NMSE is much larger than the routes inside the room. Fortunately, transfer learning can be easily done on partial data of route 14, it can be seen that the prediction error of the rest of data is reduced. However, to further reduce the prediction error, more data in the corridor should be collected and trained. For a building with AI-enhanced IDAS, it would be easy to collect CSI in all the corridors having antennas in a very short time and generate a large enough dataset for training a generalized corridor channel model.

Compared with indoor, BLER performance of the channel predictor for outdoor is studied at various signal-to-noise ratio (SNR) values by using an adjustable attenuator. In this experiment, channel predictor with attention based 20:20 ms NMT solution is used. For comparison, channel sampling rate (CSR) of estimator, i.e., frequency of channel estimation, is designed to be 400Hz or 2kHz. Channel coefficients on 1320 subcarriers estimated with 400Hz CSR are used for training of channel predictor, and the total length of train set is around 0.2 billion, which is from around one minute CSI collection. Routes 15 to 16 are following the same circling route, data from route 15 is used for training and route 16 for testing. It is noteworthy that though channel predictor is trained with N = 20 ms, we still can use this predictor to inference on different time spans, e.g., 10ms or 30ms, nothing but the predictor’s understanding of small scale properties is slightly different.

The decoding performance with predicted and estimated channel coefficients for route 16 is shown in Fig. 6. We can tell that longer prediction time span introduces larger performance loss, and at BLER = 0.1 the decoding performance of N = 10 ms predictor is around 0.1dB inferior to decoding performance of estimator with 400Hz CSR. For M = 20 ms and N = 10 ms, at least one third pilot or CSI feedback resources can be saved, here is a trade-off between improving BLER performance and saving time-frequency resources, both of them can finally increase the system throughput.

The second trade-off is between improving BLER performance and saving computing resources. As shown in Fig. 6, improving BLER performance by increasing CSR is not easy due to the inherent estimation error (estimator 2). Fortunately, predictor can outperform estimator with the cost of some computing resources. For example, we can pick the maximum CFR from three predictors for each subcarrier and

---

**TABLE II**

| Route | 8    | 9    | 10   | 11   |
|-------|------|------|------|------|
| NMSE  | 0.0051 | 0.0057 | 0.0052 | 0.0064 |
| NMSE  | 0.0058 | 0.0054 | 0.0108 | 0.0075 |

---

![Fig. 5. Schematic diagram of environment for indoor and outdoor channel measurement.](image)

![Fig. 6. BLER performance of decoding with two estimators and three predictors. Estimator 1 and 2 use 400Hz and 2000Hz CSR respectively, and predictors with varying N are trained on 400Hz estimated channel coefficients when M is 20 ms. Results of prediction diversity (PD) by calculating maximum CFR of all three predictors are shown with around 2dB gain.](image)
combine them into a new prediction, as this operation is similar to antenna diversity, we name this new prediction as prediction diversity (PD). In Fig. 7, channel amplitudes versus subcarriers in a random frame are compared for three predictors, estimator 2 and PD. In this frame, only channel of predictor 2 can decode this frame correctly while other predictors or estimators will fail, though channel predicted by predictor 2 is the most different from channel estimated by estimator 2, the randomness of prediction can provide more possibilities to overcome estimation error. Especially for deep fades surrounded by the dashed circles, where estimation error is largest and the phase features of CFR are seriously influenced by noise, however, predictors with different settings may or may not fall into the deep fades, and when some predictor does not fall into the deep fades, more reasonable phase features can be kept, even it’s not the real channel that data go through, it’s much better than using a wrong estimated results. By using PD for decoding at each frame, around 2dB performance gain can be achieved as shown in Fig. 6. When improving BLER performance is critical and robust, and realistic channel prediction with superior performance relative to channel estimation is attainable by using firstly proposed prediction diversity technique. These results show us a promising future for AI-enhanced channel prediction.

In future works, the influence of memorizing capability of neural network based channel predictor on small scale fading should be further investigated. More antenna and user equipments are needed. Prediction diversity technique should be verified on various cases. Deal with the difficulties of deployment of this algorithm in a real system.

V. CONCLUSIONS

In this paper, we proposed a new channel prediction algorithms with hundreds of learnable features representing each complex-valued channel coefficient instead of conventional two features. The additional features are essential for a channel model to balance fitting the statistical model and memorizing small scale properties. To implement this algorithm, we tried recurrent neural networks as a start, furthermore, seq2seq models and its variants with better performance are proved for channel prediction as time series task. It turns out that the encoder and decoder of seq2seq models with different lengths but the same Vocabulary of Channel Changes can be perfect containers for different time spans of past and future signals. The numerical results with simulation and realistic data indicate the channel prediction model is reliable and robust, and realistic channel prediction with superior performance relative to channel estimation is attainable by using firstly proposed prediction diversity technique. These results show us a promising future for AI-enhanced channel prediction.

REFERENCES

[1] X. You, C. Zhang, X. Tan, S. Jin, and H. Wu, “Ai for 5g: research directions and paradigms,” Science China Information Sciences, vol. 62, no. 2, p. 21301, 2019.
[2] A. Duel-Hallen, “Fading channel prediction for mobile radio adaptive transmission systems,” Proceedings of the IEEE, vol. 95, no. 12, pp. 2299–2313, 2007.
[3] G. Oien, H. Holm, and K. J. Hole, “Impact of channel prediction on adaptive coded modulation performance in rayleigh fading,” IEEE Transactions on Vehicular Technology, vol. 53, no. 3, pp. 758–769, 2004.
[4] A. Duel-Hallen, S. Hu, and H. Hallen, “Long-range prediction of fading signals,” IEEE Signal processing magazine, vol. 17, no. 3, pp. 62–75, 2000.
[5] T. Ekman, “Prediction of mobile radio channels: modeling and design,” Ph.D. dissertation, Institutionen for materialvetenskap, 2002.
[6] M. Sternad and D. Aronsson, “Channel estimation and prediction for adaptive ofdm downlinks,” in IEEE Vehicular Technology Conference, vol. 2. IEEE: 1999, 2003, pp. 1283–1287.
[7] M. Chen, T. Ekman, and M. Viberg, “New approaches for channel prediction based on sinusoidal modeling,” EURASIP Journal on Advances in Signal Processing, vol. 2007, no. 1, p. 049393, 2006.
[8] T. Ding and A. Hirose, “Fading channel prediction based on combination of complex-valued neural networks and chirp z-transform,” IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 9, pp. 1686–1695, 2014.
[9] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, “Channel state information prediction for 5g wireless communications: A deep learning approach,” IEEE Transactions on Network Science and Engineering, 2018.
[10] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” Journal of machine learning research, vol. 3, no. Feb, pp. 1137–1155, 2003.
[11] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in Advances in neural information processing systems, 2014, pp. 3104–3112.
[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105.
[13] J. Meredith, “Study on channel model for frequency spectrum above 6 ghz,” 3GPP TR 38.900, Jun, Tech. Rep., 2016.
[14] S. Zhou and G. B. Giannakis, “How accurate channel prediction needs to be for transmit-beamforming with adaptive modulation over rayleigh mimo channels?” IEEE Transactions on Wireless Communications, vol. 3, no. 4, pp. 1285–1294, 2004.