

Towards Speed-Agnostic Time-Series Forecasting for Proactive Handover

Junseo Lee*

Department of Artificial Intelligence
Korea University
Seoul, Korea
junseo@korea.ac.kr

Suwan Yoon*

Department of Artificial Intelligence
Chung-Ang University
Seoul, Korea
swyoon0312@cau.ac.kr

Changhee Lee

Department of Artificial Intelligence
Korea University
Seoul, Korea
changheele@korea.ac.kr

Abstract—In modern cellular networks, the handover (HO) of selecting the optimal base station according to the user equipment (UE) mobility is indispensable. However, conventional HO mechanisms primarily rely on reactive methods, which face inherent limitations in dealing with network uncertainties and real-time adaptability. To overcome these challenges, recent studies have introduced proactive HO schemes that leverage deep learning-based time-series forecasting models. Nevertheless, these approaches still suffer from degraded prediction performance under diverse UE speed environments due to global distribution shift, and naively constructing separate models for each mobility speed is highly inefficient. In this study, we address this inefficiency by proposing a speed-agnostic unified model. Specifically, we incorporate the existing Reversible Instance Normalization (RevIN) technique and introduce our newly proposed method, Residual Normalization (RN). We apply these normalization schemes to deep learning-based time-series forecasting models (DLinear, NLinear, and TCN), and through training and evaluation on datasets collected at varying mobility speeds, we demonstrate improved generalization capability of the forecasting models.

Index Terms—Mobility management, RSRP measurement prediction, proactive handover, time-series forecasting, deep learning

I. INTRODUCTION

The handover (HO) process is critical for maintaining seamless connectivity for mobile user equipment (UE) in modern cellular networks. Conventional HO mechanisms, however, are fundamentally reactive, initiating a HO only after the serving cell's signal quality drops below a predefined threshold. By addressing link degradation after the fact, this approach suffers from an inherent latency that leads to increased rates of HO failures (HOFs), radio link failures, and unnecessary ping-pong effects. The performance limitations of these legacy methods underscore the urgent need for a paradigm shift toward proactive mobility management capable of anticipating the UE's future state and network conditions.

To address these shortcomings, proactive HO offers a compelling solution by shifting the decision-making paradigm from reactive to predictive [1]. The core idea is to forecast future signal conditions, such as the reference signal received power (RSRP), to determine the optimal HO timing and target cell in advance. The success of this strategy, therefore,

fundamentally depends on the accuracy of the underlying time-series forecasting.

While recent deep learning models have demonstrated promising results in time-series forecasting at crucial HO-related time points such as time-to-trigger (TTT) and minimum time-of-stay (MTS), their practical deployment faces significant hurdles, especially in realistic scenarios with variable UE mobility [2]. A primary issue is the poor generalization across different speeds; a model trained at one velocity exhibits performance degradation when the UE's speed changes, a well-known issue of distribution shift (see Fig. 1). This failure to adapt to dynamic mobility patterns is a major drawback. Furthermore, the naive brute-force alternative of training a distinct model for every possible speed is operationally infeasible, as it would require a prohibitive amount of training data and create an unmanageable deployment architecture.

To address these issues, we propose and evaluate a deep learning-based speed-agnostic forecasting model for proactive HO. Our investigation begins with a statistical analysis of signal distributions across various mobility speeds (e.g., 60, 90, and 120 km/h), characterizing the instance variance and residuals that challenge model generalization. Based on these insights, we introduce a novel residual-based normalization technique and compare its effectiveness against Reversible Instance Normalization (RevIN) [3]. Our rigorous evaluation demonstrates that a single model equipped with our proposed method can accurately forecast RSRP measurements at TTT and MTS. Critically, we show that this approach generalizes well to unseen test speeds, achieving significant and more stable performance improvements over baseline models that lack normalization in most scenarios.

II. RELATED WORKS

A. Proactive HO

Proactive HO aims to enhance network performance by minimizing service interruptions and reducing signaling overhead [1]. Unlike conventional reactive approaches, where HO is triggered once a specific signal threshold is crossed, proactive HO leverages predictive modeling of future signal conditions to trigger the HO to the target cell at the optimal time. Two critical parameters in this process are TTT, which predicts imminent signal degradation to prevent HOFs, and MTS,

*These authors contributed equally to this work.

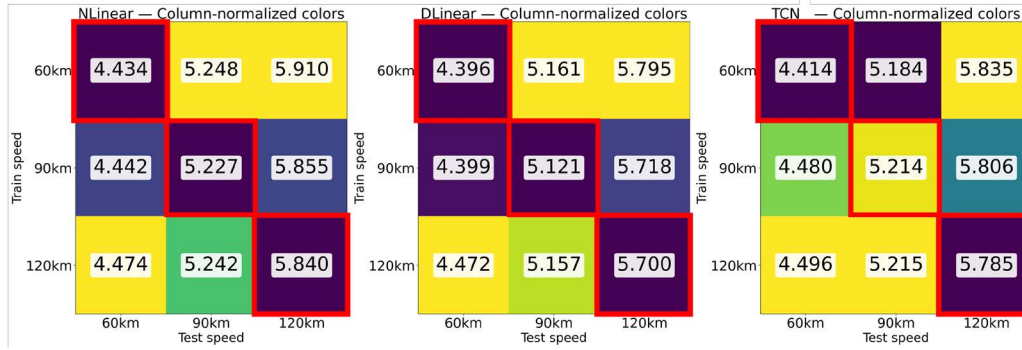


Fig. 1. Signal prediction performance at MTS across mismatched training and testing speeds (column-wise normalized). The metric is MAE; the x-axis denotes the test dataset speed and the y-axis denotes the training dataset speed. The red box highlights evaluations under matched training and testing speeds, where performance is generally superior.

which mitigates the ping-pong effect by avoiding unnecessary consecutive HOs. Recent work [2] has focused on improving the prediction accuracy of signals (e.g., RSRP) at TTT and MTS points, thereby reducing HOF rates, minimizing latency, and ultimately improving user-perceived throughput.

B. Deep Learning Models for Time-Series Forecasting

Time-series forecasting refers to the task of predicting future sequences (look-forward window) based on previously observed sequences (look-back window). Recently, deep learning-based forecasting models have been extensively studied [4]. Recurrent architectures such as Long Short-Term Memory (LSTM) [5], Gated Recurrent Units (GRU) [6], and SegRNN [7] effectively capture temporal dependencies through sequential processing. Convolutional models, including Temporal Convolutional Networks (TCN) [8] and Patch-Mixer [9], are also widely applied in time-series forecasting tasks. Furthermore, Transformer-based architectures such as Informer [10] and Autoformer [11] have demonstrated strong performance in long-term sequence forecasting. However, despite the introduction of Transformer-based models, simple linear approaches such as NLinear and DLinear [12] have been reported to outperform Transformer variants.

In this work, we focus on three time-series forecasting models previously reported to achieve strong performance in forecasting RSRP measurements at TTT and MTS [2] — i.e., **DLinear**, **NLinear**, and **TCN** — as the baseline methods for our evaluation.

C. Instance-Wise Normalization and Denormalization

Instance-wise normalization has emerged as a powerful technique for improving time-series forecasting by addressing the distribution shift between model inputs and prediction targets. A prominent example is Reversible Instance Normalization (**RevIN**) [3], which standardizes each input sequence before feeding it to the forecasting model and subsequently reverses the process on the model's output using learned statistics. This forces each instance to have zero mean and unit variance, enhancing training stability and predictive accuracy. While newer methods like SAN [13] and FAN [14] have since been proposed, RevIN remains particularly well-suited for our

study of time-series forecasting for improving UE mobility, as it is less sensitive to input sequence length compared to its counterparts.

III. PRELIMINARIES

A. Time-Series Forecasting for Proactive HO

Failure to select the proper HO target results in degraded performance and connection failures, such as HOFs and ping-pong effects [1]. The challenge of making an optimal HO decision can be addressed by accurately forecasting the RSRP of relevant cells based on the history of RSRP measurements collected by the UE.

Let x_t be the univariate RSRP measurement at time t . We define a subsequence of contiguous measurements from time t_1 to t_2 as:

$$x_{t_1:t_2} \triangleq (x_{t_1}, x_{t_1+1}, \dots, x_{t_2}).$$

Then, a forecasting model f_θ takes as input the look-back window of length W_{in} , i.e., $x_{t-W_{in}+1:t}$, and predicts the look-forward window of length W_{out} , i.e., $\hat{x}_{t+1:t+W_{out}}$, as follows:

$$\hat{x}_{t+1:t+W_{out}} = f_\theta(x_{t-W_{in}+1:t}), \quad (1)$$

where f_θ represents a forecasting model with learnable parameters θ . To optimize the model's predictive accuracy, we train it to minimize the mean squared error (MSE) between the predicted and actual future values. The optimal model f_θ^* can be found by solving the following:

$$\theta^* = \arg \min_{\theta} \mathbb{E} \left[\|\hat{x}_{t+1:t+W_{out}} - x_{t+1:t+W_{out}}\|_2^2 \right]. \quad (2)$$

B. Dataset Configuration

The dataset is constructed based on the 3GPP TR 38.744 specifications, reflecting a large-scale cellular system scenario consisting of 19 base stations (BSs) and 57 cells configured in a three-sector model. The UE follows a random way-point mobility model, traversing a single trajectory for a total duration of 11 hours. The simulation is conducted under three speed conditions: 60, 90, and 120 km/h.

L3-RSRP (dB) measurements are collected in a 1Tx-1Rx beamforming environment with an L3 filter coefficient of 4 and a measurement interval of 40 ms. The collected UE mobility

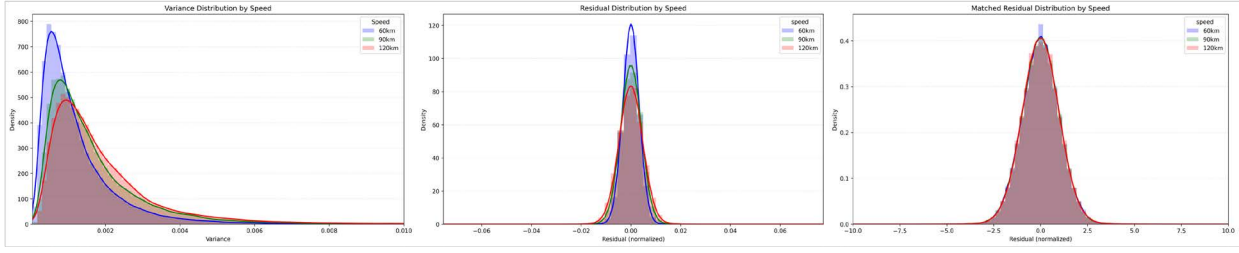


Fig. 2. Distributional characteristics according to UE speed. (left) Instance-wise variance distribution showing separability by speed, (middle) residual distribution across different speeds, and (right) normalized residual distribution obtained by dividing with the speed-specific standard deviation.

traces are segmented into non-overlapping sequences and split into training, validation, and test sets with ratios of 64%, 16%, and 20%, respectively. Each sequence is further processed using a sliding window approach with a stride 1, where the input and output window lengths are set to 32.

To ensure a fair and controlled evaluation of forecasting performance, our analysis is based on the signal from a single cell that primarily acts as the serving cell across all three speed datasets. Furthermore, we apply min-max scaling for stable model training.

IV. METHOD

In this section, we investigate the distributional characteristics of each dataset under three different UE speed conditions (60, 90, and 120 km/h).

A. Variance Distribution According to the UE Speed

This subsection examines whether signal variance can be used to differentiate UE speeds. As shown in Fig. 2 (left), the variance distributions for different speeds are distinct enough to be separated, even with similar overall shapes. This key observation implies that an input window's variance contains implicit, speed-related information. Our strategy leverages this insight. By removing the variance from the input signal (i.e., normalizing it), the forecasting model is encouraged to learn a general, speed-agnostic representation. The model's output can then be made speed-specific by restoring the original variance information. This process allows a single model to operate effectively across various UE speeds without being retrained to a specific one.

However, Fig. 2 also reveals that signal variance presents significant temporal variability even within a single UE speed. This intra-speed fluctuation implies that a naive normalization approach can degrade model performance. To properly address this, we employ RevIN [3], a technique designed to handle such distribution shifts, as a key method for mitigating the effect of input variance in our performance evaluation. More specifically, RevIN is applied to each time-series forecasting models based on the following steps:

$$\hat{\mathbf{x}}_t = \gamma \left(\frac{\mathbf{x}_t - \mathbb{E}_t[\mathbf{x}_t]}{\sqrt{\text{Var}[\mathbf{x}_t] + \epsilon}} \right) + \beta, \quad (3)$$

$$\tilde{\mathbf{y}}_t = f_\theta(\hat{\mathbf{x}}_t), \quad (4)$$

TABLE I
STANDARD DEVIATION OF RESIDUALS FOR EACH UE SPEED

speed v	σ_v
60	≈ 0.4783
90	≈ 0.5919
120	≈ 0.6836

$$\hat{\mathbf{y}}_t = \sqrt{\text{Var}[\mathbf{x}_t] + \epsilon} \cdot \left(\frac{\tilde{\mathbf{y}}_t - \beta}{\gamma} \right) + \mathbb{E}_t[\mathbf{x}_t], \quad (5)$$

where $\mathbf{x}_t = x_{t-W_{\text{in}}+1:t}$, denotes the input instance and $\mathbf{y}_t = x_{t+1:t+W_{\text{out}}}$ its paired target; γ and β are learnable parameters, while $\text{Var}[\cdot]$ and $\mathbb{E}_t[\cdot]$ represent the variance and mean of the instance, respectively [3].

Through this process, the forecasting model f_θ receives input signals with reduced speed dependency, thereby learning predictions less biased toward specific UE speeds. Subsequently, RevIN restores the statistical properties of the predicted signal, enabling the model to adaptively generalize across various speed environments.

B. Residual Distribution According to UE Speed

In this subsection, we investigate whether UE speed can be characterized by the distributional properties of signal variations, namely residuals. A residual represents the change between consecutive signals and is formally defined as

$$r_t = x_t - x_{t-1}. \quad (6)$$

The distribution of residuals across the entire dataset is illustrated in the middle plot of Fig. 2, while the corresponding standard deviations are summarized in Table I.

It is observed that the standard deviation σ_v monotonically increases with UE speed. Moreover, when each residual is normalized by its respective σ_v , the resulting distributions collapse into an almost identical form, as shown in the rightmost plot of Fig. 2. Based on this observation, we propose a global normalization and denormalization strategy, which we refer to as residual normalization (RN).

To achieve speed-agnostic forecasting, we introduce a methodology centered on residual prediction. Motivated by our previous observations, the approach first transforms the original signal into its residual form using (6). A normalization-denormalization procedure is then applied to this residual

series, and the forecasting model is subsequently trained to predict future residuals instead of the raw signal:

$$\tilde{r}_{t+1:t+W_{\text{out}}} = f_{\theta}(r_{t-W_{\text{in}}+2:t}), \quad (7)$$

$$\hat{x}_{t'} = \hat{x}_{t'-1} + \tilde{r}_{t'}, \quad (\hat{x}_t = x_t, \quad t \leq t' \leq t + W_{\text{out}}). \quad (8)$$

Our proposed RN method is designed to remove speed-dependent information from the data. During preprocessing, we normalize the residuals by dividing them by a global standard deviation σ_v , which is calculated for each specific UE speed. After the model makes a prediction, the output is denormalized by multiplying it by the same scaling factor:

$$r_t = \frac{r_t}{\sigma_v}, \quad \tilde{r}_t = \tilde{r}_t \cdot \sigma_v. \quad (9)$$

This approach is fundamentally different from RevIN. While RevIN uses instance-wise normalization to handle distribution shifts in every single input, RN performs a global normalization across an entire speed category. The goal here is not to model instance-level shifts but simply to remove the overall statistical bias associated with a specific speed.

V. EXPERIMENT

As described earlier, we conduct experiments under three distinct UE speed scenarios. We train and evaluate three time-series forecasting models — **DLinear**, **NLinear**, and **TCN** — using univariate RSRP signals from the specific cell. In addition to evaluating each model under the same UE speed as used in training, we further assess their performance on data collected at different UE speeds. This cross-speed evaluation enables us to determine the extent to which the proposed normalization methods enhance the *speed-agnostic* property of the models.

A. Experimental Setup

The detailed configurations of the models used in our experiments are as follows: **DLinear**, **NLinear**: We employed the official implementation provided in [12]. **TCN**: The model is constructed with a kernel size of 5 and a stack of 6 convolutional layers. For all models, the hidden dimension is fixed to 256. Training is performed with the Adam optimizer, starting from an initial learning rate of 0.001, which decayed exponentially by a factor of 0.9 at each epoch. The look-back and look-forward window lengths are both set to 32. For evaluation, the TTT and MTS are defined as the 4th (160 ms) and 25th (1000 ms) time steps within the look-forward window, respectively.

The loss function is MSE in all cases. Every model is trained only on the two specific points corresponding to TTT and MTS. Therefore, the residual-based approach is designed to predict the residual between the signal at the last time step of the look-back window and the signal at the TTT point, as well as the residual between the signals at the TTT point and the MTS point. Performance evaluation is conducted using mean absolute error (MAE) at the TTT and MTS points. All experiments are conducted on a single GPU machine¹.

¹CPU: Intel Xeon Gold 6526Y (16 cores, 32 threads); GPU: NVIDIA A6000 (48GB VRAM).

TABLE II

RESULTS OF DLINEAR, NLINEAR, AND TCN TRAINED ON 60/90/120 KM/H AND TESTED ON 60/90/120 KM/H (FULL CROSS-EVALUATION). EACH CELL LISTS TTT (160 MS) / MTS (1000 MS). LOWER VALUES INDICATE BETTER PERFORMANCE; THE BEST IS IN BOLD AND THE SECOND BEST IS UNDERLINED.

Method	DLinear	NLinear	TCN
<i>Train speed = 60 km/h, Test speed = 60 km/h</i>			
Raw	0.822 / 4.396	0.802 / 4.434	0.902 / 4.414
Raw+RevIN	0.723 / 4.417	0.790 / 4.434	<u>0.727</u> / 4.413
Resi	<u>0.602</u> / <u>4.338</u>	<u>0.667</u> / 4.341	0.814 / 4.472
Resi+Norm (RN)	0.592 / 4.330	0.651 / 4.338	0.550 / 4.323
<i>Train speed = 60 km/h, Test speed = 90 km/h</i>			
Raw	1.054 / 5.161	1.035 / 5.248	1.150 / 5.184
Raw+RevIN	0.954 / 5.235	1.017 / 5.246	<u>0.954</u> / 5.221
Resi	0.782 / 5.126	0.832 / 5.155	0.996 / 5.298
Resi+Norm (RN)	0.765 / 5.125	0.816 / 5.153	0.715 / 5.111
<i>Train speed = 60 km/h, Test speed = 120 km/h</i>			
Raw	1.252 / 5.795	1.235 / 5.910	1.359 / <u>5.835</u>
Raw+RevIN	1.141 / 5.889	1.207 / 5.905	<u>1.139</u> / 5.879
Resi	<u>0.947</u> / <u>5.778</u>	<u>0.972</u> / <u>5.830</u>	1.151 / 5.978
Resi+Norm (RN)	0.917 / 5.772	0.954 / 5.827	0.855 / 5.755
<i>Train speed = 90 km/h, Test speed = 60 km/h</i>			
Raw	0.756 / 4.399	0.782 / 4.442	0.936 / 4.480
Raw+RevIN	0.738 / 4.424	0.789 / 4.442	<u>0.726</u> / 4.420
Resi	<u>0.616</u> / <u>4.364</u>	<u>0.672</u> / <u>4.345</u>	0.776 / 4.431
Resi+Norm (RN)	0.592 / 4.347	0.630 / 4.337	0.567 / 4.352
<i>Train speed = 90 km/h, Test speed = 90 km/h</i>			
Raw	0.971 / <u>5.121</u>	1.001 / 5.227	1.188 / 5.214
Raw+RevIN	0.957 / 5.214	1.010 / 5.229	<u>0.943</u> / <u>5.210</u>
Resi	0.784 / 5.133	0.831 / 5.155	0.946 / 5.241
Resi+Norm (RN)	0.751 / 5.118	0.793 / 5.149	0.724 / 5.119
<i>Train speed = 90 km/h, Test speed = 120 km/h</i>			
Raw	1.157 / 5.718	1.190 / 5.855	1.395 / <u>5.806</u>
Raw+RevIN	1.137 / 5.842	1.193 / 5.854	1.121 / 5.836
Resi	<u>0.936</u> / 5.746	<u>0.968</u> / <u>5.824</u>	<u>1.091</u> / 5.875
Resi+Norm (RN)	0.890 / <u>5.736</u>	0.931 / 5.822	0.863 / 5.734
<i>Train speed = 120 km/h, Test speed = 60 km/h</i>			
Raw	0.770 / 4.472	0.792 / 4.474	0.917 / 4.496
Raw+RevIN	0.755 / 4.443	0.779 / 4.460	0.738 / <u>4.442</u>
Resi	<u>0.611</u> / <u>4.395</u>	<u>0.736</u> / <u>4.359</u>	<u>0.712</u> / 4.443
Resi+Norm (RN)	0.606 / 4.377	0.673 / 4.353	0.594 / 4.383
<i>Train speed = 120 km/h, Test speed = 90 km/h</i>			
Raw	0.978 / <u>5.157</u>	1.003 / 5.242	1.156 / <u>5.215</u>
Raw+RevIN	0.977 / 5.226	0.999 / 5.244	0.951 / 5.221
Resi	0.768 / 5.157	0.881 / 5.171	0.882 / 5.241
Resi+Norm (RN)	0.764 / 5.139	0.835 / 5.169	0.744 / 5.140
<i>Train speed = 120 km/h, Test speed = 120 km/h</i>			
Raw	1.156 / 5.700	1.183 / 5.840	1.356 / <u>5.785</u>
Raw+RevIN	1.159 / 5.831	1.182 / 5.852	1.127 / 5.826
Resi	<u>0.908</u> / 5.754	<u>1.009</u> / 5.836	<u>1.029</u> / 5.862
Resi+Norm (RN)	0.903 / <u>5.736</u>	0.974 / 5.836	0.875 / 5.735

B. Results

Table II summarizes the forecasting performance based on the intra and inter UE speed scenarios (60, 90, and 120 km/h). Here, the *raw input* refers to the RSRP measurements preprocessed only by min-max scaling, while the *residual input* denotes the signal obtained by applying the residual transformation defined in (6) on top of the raw input. When normalization is applied, RevIN is applied to the raw input, whereas for the residual input, normalization is conducted by

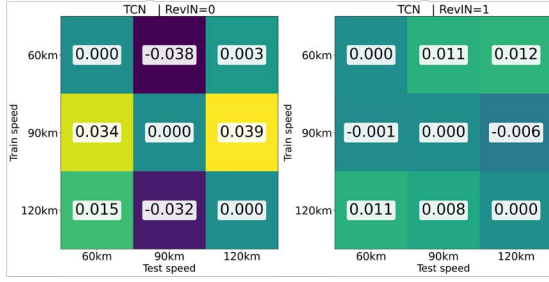


Fig. 3. Effect of RevIN on TCN model performance at the TTT prediction point. Each column is adjusted by subtracting its diagonal entry, enabling an evaluation of the model’s generalization capability across different UE speeds.

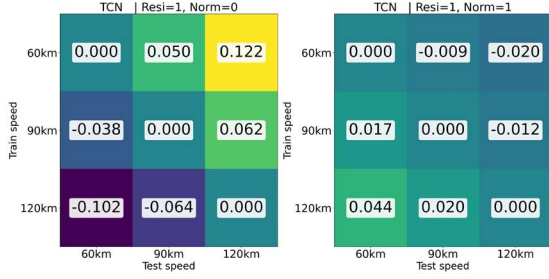


Fig. 4. Effect of RN on TCN model performance at the TTT prediction point. Each column is adjusted by subtracting its diagonal entry, enabling an evaluation of the model’s generalization capability across different UE speeds.

dividing each residual by the global standard deviation σ_v for each UE speed in the training set.

Forecasting Performance. As presented in Table II, both RevIN and our proposed RN consistently improve prediction accuracy over the baseline at the TTT. The RN method, when applied to signal residuals, yields the most significant gains across all evaluated backbones (i.e., DLinear, NLinear, and TCN), indicating that residual-based forecasting is highly effective for short-term prediction. Furthermore, for the longer-horizon forecasting at the MTS, our proposed approach also demonstrates superior performance compared to the baseline, confirming its superiority for both short and long-term forecasting.

Experiment on Different UE Speed. Our cross-speed evaluation results suggest that a single, unified forecasting model can be robust to varying UE speeds. This is evidenced by Figs. 3 and 4, which show that both RN and RevIN significantly reduce the performance gap when a model is tested on speeds unseen during training. Both normalization methods effectively mitigate the degradation from speed variations, with our residual-based approach consistently providing performance gains (Table II). This capability eliminates the practical need for deploying and maintaining separate, speed-specific forecasting models.

Comparison of Normalization Methods. A key distinction between RevIN and RN emerges in long-horizon forecasting. While both methods improve short-term (TTT) prediction accuracy, Table II shows that only RN maintains this advantage for the long-term (MTS) task; RevIN’s performance, in contrast, degrades. We hypothesize that RevIN’s learnable

parameters become biased toward short-term dependencies, hindering the model’s ability to capture the long-range patterns necessary for MTS prediction. This suggests that RN is a more robust and reliable normalization strategy for speed-agnostic mobility management across varying forecast horizons.

VI. CONCLUSION

In this work, we have demonstrated the feasibility of a single, speed-agnostic forecasting model for proactive HO. We discover that training on signal residuals consistently improves performance and that normalization techniques, namely RN and RevIN, enable robust generalization across diverse user speeds unseen during training. This obviates the need for multiple, speed-specific models. However, we identify two limitations for future work. First, the proposed RN method requires sufficient samples to estimate statistics specific to each UE speed, a challenge in real-time scenarios; future work could investigate models that predict these statistics for unseen speeds. Second, long-horizon prediction accuracy remains limited. To address this, we will explore multivariate time series from neighboring cells and advanced architectures capable of integrating this information to build more reliable data-driven HO frameworks.

ACKNOWLEDGMENTS

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-00972, Development of Intelligent Wireless Access Technologies), and by the Institute of Information & Communications Technology Planning & Evaluation (IITP) funded by the Korea government (MSIT) under Grant RS-2019-II190079 by the AI Graduate School Program (Korea University). We thank Hyun-Seo Park for providing the simulation data and for insightful discussions that improved the quality of this work.

REFERENCES

- [1] H.-S. Park, H. Kim, C. Lee, and H. Lee, “Mobility management paradigm shift: From reactive to proactive handover using ai/ml,” *IEEE Network*, vol. 38, no. 2, pp. 18–25, 2024.
- [2] H. Park, E. Kim, and C. Lee, “A comprehensive evaluation of time-series forecasting methods for proactive handover,” in *2024 15th International Conference on Information and Communication Technology Convergence (ICTC)*, 2024, pp. 1447–1452.
- [3] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, “Reversible instance normalization for accurate time-series forecasting against distribution shift,” in *International conference on learning representations*, 2021.
- [4] Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, and J. Wang, “Deep time series models: A comprehensive survey and benchmark,” *arXiv preprint arXiv:2407.13278*, 2024.
- [5] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

- [6] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.
- [7] S. Lin, W. Lin, W. Wu, F. Zhao, R. Mo, and H. Zhang, “Segrnn: Segment recurrent neural network for long-term time series forecasting,” *arXiv preprint arXiv:2308.11200*, 2023.
- [8] Y. Lin, I. Koprinska, and M. Rana, “Temporal convolutional attention neural networks for time series forecasting,” in *International joint conference on neural networks (IJCNN)*, 2021.
- [9] Z. Gong, Y. Tang, and J. Liang, “Patchmixer: A patch-mixing architecture for long-term time series forecasting,” in *Proceedings of the International Conference on Learning Representations (ICLR 2024)*, vol. 12, 2024.
- [10] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, “Informer: Beyond efficient transformer for long sequence time-series forecasting,” in *Proceedings of the AAAI conference on artificial intelligence (AAAI 2021)*, vol. 35, 2021.
- [11] H. Wu, J. Xu, J. Wang, and M. Long, “Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting,” in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS 2021)*, vol. 35, 2021.
- [12] A. Zeng, M. Chen, L. Zhang, and Q. Xu, “Are transformers effective for time series forecasting?” in *Proceedings of the AAAI conference on artificial intelligence (AAAI 2023)*, vol. 37, 2023.
- [13] Z. Liu, M. Cheng, Z. Li, Z. Huang, Q. Liu, Y. Xie, and E. Chen, “Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 14 273–14 292, 2023.
- [14] W. Ye, S. Deng, Q. Zou, and N. Gui, “Frequency adaptive normalization for non-stationary time series forecasting,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 31 350–31 379, 2024.