Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

TSNet: a foundation model for wireless network status prediction in digital twin

Siyao SONG, Guoao SUN, Yifan CHANG, Nengwen ZHAO†‡, Yijun YU

Huawei Technologies Co., Ltd., ShenZhen 518129, China

†E-mail: zhaonengwen@huawei.com

Received Apr. 16, 2024; Revision accepted Dec. 3, 2024; Crosschecked

Abstract: Predicting future network status is a key capability in digital twin networks and can assist operators in estimating network performance and taking proactive measures in advance. Existing methods, including statistical methods, machine learning-based methods, and deep learning-based methods, suffer from several limitations in generalization ability and data dependency. To overcome these drawbacks, inspired by the success of pretraining and the fine-tuning framework in the natural language processing and computer vision domains, in this work, we propose TSNet, a transformer-based foundation model for predicting various network status measurements in digital twins. To adapt transformer architecture to time series data, frequency learning attention and time series decomposition blocks are implemented. We also design a fine-tuning strategy to enable TSNet to adapt to new data or scenarios. Experiments demonstrate that zero-shot prediction using TSNet without any training data performs better than fully supervised baseline methods and achieves higher accuracy. In addition, the prediction accuracy could be further improved by equipping the proposed model with fine-tuning. Overall, TSNet exhibits strong capability and accuracy in various datasets. For future work, we plan to explore TSNetâĂŹs generalization abilities on various downstream tasks (such as anomaly detection and pattern matching) using few-shot learning approaches

Unedited

Key words: Digital twin; Communication network; Foundation model; Network status prediction
https://doi.org/10.1631/FITEE.2400295 **CLC number:** TP

1 Introduction

In recent years, digital twin (DT) technologies have been increasingly important and useful in wireless communications. A digital twin network (DTN) is a virtual representation of a physical communication network. Given the current network status as inputs, DTN can predict the future network state change, and then predict the future network performance (such as latency and reliability) based on the results and the network optimization strategies(Khan et al., 2022). Thus, network status prediction is a basic DTN capability.

In wireless networks, there are many kinds of measurements that characterize the network status, such as radio resource utilization, user equipment (UE) throughput, and the radio resource control (RRC) connection number. These measurements are collected as the time series data with the format of (timestamp, value). Therefore, network status prediction is fundamentally a problem of time series forecasting.

In the literature, many efforts have been devoted to time series prediction, including traditional statistical methods, machine learning based methods, and deep learning based methods. However, these methods suffer from several limitations in practice. First, statistical methods require strong assumptions concerning data and careful parameter tuning. Considering hundreds of network status measurements and a large number of base stations, it is time-consuming and tedious for engineers to manually select a suit-

‡ Corresponding author
© Zhejiang University Press 2025

able method with the best parameters for each sequence. Second, learning-based methods require long-term historical data for training. However, obtaining adequate data in the operating communication network may not always be feasible. Third, real-world networks tend to change frequently due to configuration changes, resulting in concept drift in data distribution. The model trained on old data will be aged, and sufficient new data needs to be collected to train new models. Therefore, we aim to design a general and robust prediction approach that could be applied to various network status measurements.

Motivated by the pretraining and fine-tuning framework, several time series foundation models have been proposed, for example, TimesNet (Wu et al., 2023), GPT4TS (Zhou T et al., 2023), LLM4TS (Chang et al., 2024), and TimeFM (Das et al., 2024). However, in the face of various and massive network status indicators with different monitoring granularity in communication networks, these methods, which are trained on open-source datasets, do not perform well. Thus, we propose TSNet, a general framework for network status prediction in DTNs. Specifically, TSNet comprises three key components: data preprocessing, pretraining, and fine-tuning. In the data preprocessing stage, we collected numerous historical sequences from multiple wireless experimental networks as training datasets, aiming to improve the generality and robustness of the model using diverse data. In the pretraining stage, we designed a novel neural network architecture with 0.1B parameters. To adapt traditional transformer architecture to time series data, frequency learning attention and time series decomposition were adopted to capture the temporal dependency. The model can be trained in a self-supervised manner, in a prediction task with a mean square error (MSE) loss function. After pretraining, the trained model can be used to predict a new sequence directly (zero-shot). Furthermore, we also designed a minimal fine-tuning (MinFT) method that freezes the majority of parameters to fine-tune the pretrained model with a few days of historical data (few-shot), to make TSNet adapt to new data or scenarios better and further enhance the accuracy.

The contributions of this paper are summarized as follows:

1. To tackle the limitations in generalization and data dependency of existing methods, motivated by the framework of pretraining and fine-tuning, we propose a foundation model named TSNet for network status prediction in wireless DTs. TSNet aims to require all measurements share a prediction model. In addition, equipping TSNet with a larger model and larger training datasets could achieve better accuracy and generality compared with baseline methods.

2. We designed three key components in TSNet: data preprocessing, self-supervised pretraining, and few-shot fine-tuning. Specifically, we import frequency learning attention and time series decomposition into the traditional transformer, to model time series data better.

3. We conducted extensive experiments by comparing TSNet with several existing approaches. Our results demonstrate that zero-shot TSNet achieved the best prediction accuracy compared with baseline methods using long-term historical data for fully supervised training.

Unedited

2 Related works

2.1 Traditional statistical methods

These classical methods are typically based on statistical techniques, such as ARIMA (Mehrmolaei and Keyvanpour, 2016) and Holt-Winter (HoHyndman and Athanasopoulos, 2018). However, statistical methods tend to make strong assumptions on data distribution (such as seasonality and stationarity), whereas the network status in cellular networks is dynamic, which can be too complex for the statistical models to fit on most occasions.

2.2 Traditional machine learning methods

In general, time series prediction could be regarded as a classical regression problem in machine learning. Thus, any regression algorithms could be adopted to model the relationship between the historical and the future data points. Ensemble methods such as random forest (Breiman et al., 2001) are effective learners of time series data and combine multiple base tree models. However, these methods require engineers to design effective features carefully to achieve high accuracy.

2.3 Deep neural networks

In recent years, researchers have explored deep neural networks to leverage the capability of data-

driven modeling for time series prediction tasks. Time convolutional network (TCN)-based models (Liu YJ et al., 2019) extract and fuse meaningful temporal features by learnable convolutional kernels. Recurrent neural networks (RNNs) (Salinas et al., 2020) are capable of capturing long-term dependencies within sequential data, by continuously updating the internal states at each time step. Long short-term memory (LSTM) is a variant of the RNN that addresses the vanishing and exploding gradient issues in RNNs by introducing a gating mechanism to control the information flow. Inspired by the success of transformers in natural language processing (NLP) (Lin et al., 2022), recent works have been exploring time series transformers (Zhou HY et al., 2021; Challu et al., 2022; Liu S et al., 2022; Liu Y et al., 2022; Zhou T et al., 2022). Thanks to the attention mechanism, time series transformers are more efficient and effective in capturing long-term context information, with the benefit of parallel computation.

2.4 Foundation models

Foundation models have demonstrated remarkable performance in the NLP domain (PaaÃ§ and Giesselbach, 2023). TimesNet (Wu et al., 2023) leveraged multi-periodicity to capture time variations, and projects the 1D time series into a 2D space. Subsequently, it capitalizes on existing pretrained vision backbone networks as features extractors. Finally, the extracted features are applied to any of the five downstream time series tasks. GPT4TS(Zhou T et al., 2023) and LLM4TS (Chang et al., 2024) repurpose pretrained language models for the time series task and consider textual data and time series to be similar as sequential characteristics. TimesFM (Das et al., 2024) used a decoder-only structure and pretrained with time series datasets from scratch.

Specifically, pure time series pretrained models are preferred for the following reasons. First, only a small part of parameters could be fine-tuned when using the CV or NLP model for time series prediction, leading to poor flexibility. Second, there are noticeably fewer parameters for pure time series pretrained models than those for the language area. Thus, using a larger model will bring high inference costs. Third, the latest research (Das et al., 2024) shows that pure time series models work better. In summary, this paper aims to design and train a time series foundation model in the wireless communication network domain.

3 Approach

3.1 Overall framework

Fig. 1 is an overview of TSNet. First, we collected numerous historical network status sequences (more than 300,000) as our training datasets for pretraining. To avoid the low-quality data destroying model accuracy, we designed a data quality measurement strategy to ensure the quality of datasets. For the pretraining stage, we propose a novel network architecture for TSNet with 0.1B parameters, including frequency learning attention and time series decomposition components, which enhance the traditional transformer for capturing temporal relationships. The model can be trained in a self-supervised manner using a prediction task with an MSE loss function. In the inference stage, the trained TSNet model can be used for direct prediction (zero-shot). In addition, we propose a freezing method to fine-tune the pretrained model with a few historical data (few-shot), to help TSNet better adapt to new data and scenarios.

3.2 Data preprocessing

We collected more than 300,000 time sequences from multiple experimental sites. Data preprocessing aims to filter low-quality data and standardize the data, so the data can be fed into the neural network for training.

3.2.1 Data quality measurement

Completeness: During data collection and transmission phases between network elements and the DTN, monitoring data may be lost. As the result of such potential data loss, the temporal sequence may be incomplete, and thereby introduce null values into the sequence. In the pretraining stage, time sequences with a missing data ratio exceeding 5% are deemed hazardous for representation learning and are consequently discarded.

Variation: In typical scenarios, network status indicators are expected to vary across each monitoring interval, genuinely reflecting the dynamic operation status of communication networks. However, the measurements sometimes appear constant. The

Unedited

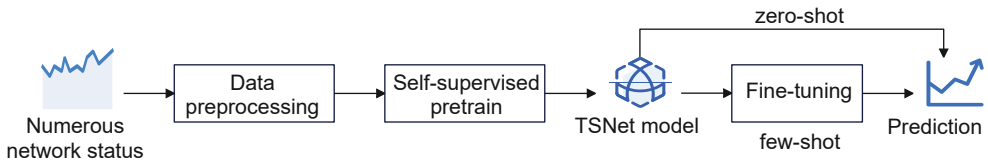

Fig. 1 Pipeline of TSNet

possible reasons include the related measurement not being supported by the network element, or the network element might go offline. Such sequences without any variation provided limited temporal information, rendered them ineffective for pretraining. Given the normalized sequence with the range of 0–1, the sequences with variance <0.05 were filtered out.

3.2.2 Data normalization

Network status measurements exhibit diverse units and data range. For instance, the average number of users typically ranges from 10 to 100, whereas downlink traffic volume is measured in bits, often reaching values in the millions. Thus, all input time series sequences fed into TSNet are normalized to a predefined scale, ranging from 0 and 1 using min-max scaling.

3.3 Architecture Details and Pretraining

TSNet is a transformer-based time-series foundation model, which is able to extract the universal features from any sequence and provide representation for various downstream tasks. Fig. 2 illustrates the model details. We selected transformer architecture for TSNet, with minor adaption to time series data. The vanilla attention blocks are substituted with frequency learning attention layers, which perform attention in frequency domain. Additionally, time series decomposition blocks are inserted before attention blocks to extract time patterns.

3.3.1 Input embedding

The input data for TSNet consists of scalar sequences paired with their corresponding timestamps. A value embedding is used to project the scalar time sequence into the model's dimension space. Sinusoidal positional encoding is added to incorporate local contextual information and enable sequential processing by the attention mechanisms. Additionally, timestamp embedding proposed by Zhou HY et al. (2021) is employed to provide absolute temporal information, enriching the model with global temporal features.

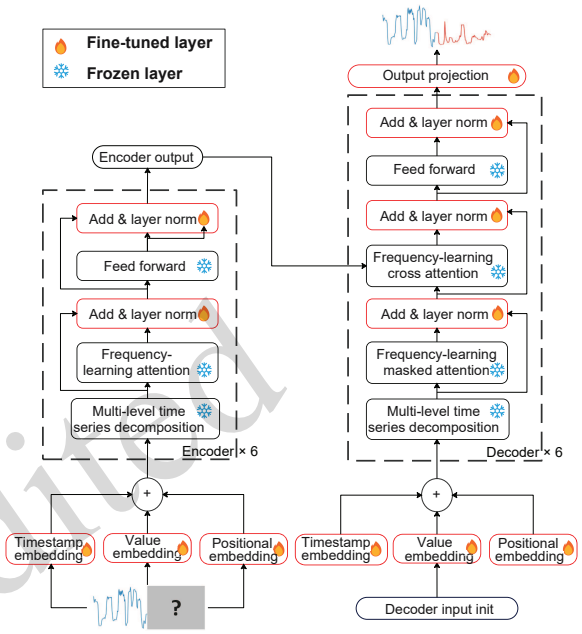

Fig. 2 Overall TSNet model architecture. The fine-tuned components in MinFT are marked in red, and the frozen components are marked in black.

3.3.2 Channel independence

In TSNet, multivariate time series data are processed independently as separate univariate sequences, employing the channel independence policy (Nie et al., 2023; Chang et al., 2024). Considering different network status measurements from the same cell may have potential mutual influence on each other; such cross-sequence information might be limited and nonuniversal. One notable advantage of TSNet is its ability to handle diverse types of input sequences concurrently, whether they represent the same performance for neighboring cells or are correlated metrics within the same cell. Consequently, TSNet can process any group of input sequences simultaneously yet independently, ensuring consistent

Unedited

feature extraction and forecasting results for successive inputs.

3.3.3 Time series decomposition

We introduce the time series decomposition modules within the transformer block, which enhances its adaptability and applicability for learning complex time series data. These modules are designed to separate the origin time series and hidden states into distinct seasonal and trend components. The decompositions enable the model to capture and use both short-term fluctuations and long-term trends more effectively. Network status measurements often exhibit pronounced seasonal behaviors. For example, the average user numbers typically increase in the morning and decrease after 5 pm on weekdays. Conversely, the average user numbers may remain low during weekends, particularly from those cells near office buildings. Therefore, we incorporate time series decomposition layers in the network to separate the seasonal and trend components and enable more effective capture of such features.

The decomposition process begins with trend extraction. We employed a series of smoothing filters, implemented as average pooling layers with different kernel sizes. These pooling layers serve as low-pass filters, smoothing the sequence over varying time scales to capture the long-term trend. The kernel sizes are learnable parameters, which allows the model to adapt the level of smoothing based on data features. Padding is applied to ensure the output maintains the same length.

Once the trend is extracted from the sequence, the seasonal component can be isolated by subtracting the trend from the origin sequence. The residual sequence highlights the periodic fluctuations, effectively capturing the recurring patterns in the data. The seasonal component isolation allows the subsequent layers to focus on learning the periodic behaviors independently from the long-term trend.

By incorporating these decomposition modules, TSNet can better learn from both short-term seasonal variations and long-term trends, improving its capability to forecast complex time series with multiple periodicities. This decomposition approach not only enhances feature representation but also mitigates the challenge of capturing mixed-frequency temporal patterns, resulting in more accurate and reliable predictions.

3.3.4 Frequency learning attention

Instead of conducting attention directly in the time domain, we implement multi-head attention in the frequency domain and achieve improved computational efficiency and the ability to capture implicit features. The primary limitation of attention in the temporal domain lies in its restriction to computing dependencies solely between discrete time steps, which might fail to capture local semantic contexts such as peaks or sudden increases, particularly in sequences with complex and nonlinear dependencies. Moreover, densely connected attention in the time domain has a complexity of N^2, allowing each time step to attend any other time step along the sequence.

Given that network status measurements normally demonstrate useful periodic properties and contain stochastic noise, points in the frequency domain may provide more useful information compared to time domain representations. We leverage such characteristics and address shortcomings of the attention mechanism with sparse frequency learning attention, enabling effective and efficient universal representation learning of time series data. The proposed frequency learning attention involves several time sequence procedures: Fourier Transform, sparse multi-head attention, and inverse Fourier transform. Considering the time sequence as a discrete signal, we use the discrete Fourier transform to project it from the time domain to the frequency domain. Subsequently, the frequency sequence undergoes a sparse multi-head attention with k sampled data points. This frequency-based projection highlights periodic features and may serve as a noise filter when combined with frequency domain sampling. Ultimately, we fill the unsampled points with the sequence mean, and then reconstruct it in the time domain through inverse Fourier transform.

3.3.5 Pretraining

We pretrained TSNet with a combination of public available time series datasets and a non-public network status indicators dataset collected from multiple experimental sites and laboratory environments. The public datasets include the M4 hourly dataset (Makridakis et al., 2018) and the electricity transformer dataset (ETT)(Zhou HY et al., 2021). The M4 hourly dataset, characterized by ev-

Unedited

idently daily seasonality with minimal noise, is considered suitable for warmup training and aligning the transformer architecture with the inherent characteristics of time series data. Meanwhile, the ETT dataset, which contains 2-year data points, is considered favorable for TSNet to capture long-term time-sequence variations. We then trained TSNet with the wireless domain dataset, which comprises more than 300,000 time sequences collected from multiple wireless experimental environments. During the pretraining stage, the sequences were mixed and fed into TSNet without differentiation based on indicator types. The purpose of pretraining was to establish common feature extraction capability for TSNet, enabling it to perform zero-shot forecasting tasks.

3.4 Fine-tuning

Fine-tuning basically refers to the process of adjusting and tweaking a pretrained model on smaller, specific datasets to enhance performance in a particular task or domain. It bridges the gap between generic pretrained models and the unique requirements of specific applications.

Based on the pretrained TSNet, we propose a MinFT method that only updates a small set of parameters and freezes the rest of the parameters in the model. Specifically, MinFT fine-tunes three specific components of TSNet, including the input embedding layers (value embedding, positional embedding, and timestamp embedding), the layer norm (LN), and the output layer, as framed in red in Fig. 2. The input layer and output layer are reinitialized because the model is transferred to the new data, and the LN is fine-tuned as is standard practice in other finetuning works (Rebuffi et al., 2017; Houlsby et al., 2019; Lu et al., 2022). Note that the attention layers are frozen because the target data is similar to the pretrained data and therefore the acquired knowledge preserved in the attention layers can be directly recapped in the new domains.

The advantages of MinFT are as follows: (1) Improved accuracy. The adjusted parameters can adapt to the new input data and help the model exploit the hidden features of the input. (2) Time-saving and memory-efficient. MinFT updates a small part of model, thus having a much smaller number of parameters involved than in the original model. (3) Overcoming catastrophic forgetting phenomenon. MinFT does not adjust all the weights in the original model and thus the previously learned key information is preserved.

4 Evaluation

4.1 Dataset and evaluation metrics

We collect the testing set from a new experimental site to prevent data leakage. The testing dataset contains eight measurements, namely downlink/uplink prb utilization, downlink/uplink traffic, average downlink/uplink user throughput, the average number of users, and the number of RRC connection attempts. The duration of the testing dataset spans 30 days with a constant interval of 1 h, thereby providing a sufficiently long period to train baseline models. The specific prediction task involves predicting the network status measurement for the subsequent 10 time steps, given the data from the preceding 90 time steps as input. We use the mean square error (MSE $= \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$) and mean absolute error (MAE $= \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i|$) as evaluation metrics, where y_i is the ground truth and $\hat{y}_i$ is the predicted value from the model.

4.2 Zero-shot performance

We selected several baseline models for comparison, including LSTM, NHiTS, Nbeats, and temporal fusion transformer (TFT). Each baseline method is trained individually for each indicator and each cell, using the first 75% sequence for training. Subsequently, the last 25% sequence is used for evaluation in a sliding window fashion. Finally, we compute the MAE and MSE for each subseries; the results are summarized in Table 1, with the best score of each group is highlighted in bold text. Table 1 also presents the zero-shot prediction results for TSNet, which only evaluated on the last 25%, without depending on the first 75% of each sequence for training.

Table 1 indicates that TSNet consistently outperforms the baseline models in most cases based on MSE and MAE. Specifically, TSNet achieved the best performance in 17 out of 24 cases measured by MSE and 15 out of 24 cases measured by MAE. Overall, TSNet achieves the lowest average MSE (0.0316) and MAE (0.1073) on all testing sets. The experimental results also indicate that the performance of the baseline methods is unstable on dif-

Unedited

Table 1 Predictions results of TSNet and compared methods on evaluation datasets

	Model	TSNet (zero-shot)		LSTM		Nbeats		TFT		NHiTS		TSNet (fine-tuned)	
	Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Cell 1	PRB.DL	0.0117	**0.0792**	0.0215	0.1073	**0.0100**	0.0811	0.0175	0.0986	0.0118	0.0845	*0.0084*	*0.0701*
	PRB.UL	0.0171	0.1038	0.0211	0.1078	**0.0132**	**0.0895**	0.0176	0.1057	0.0157	0.0969	*0.0142*	*0.0964*
	Traffic.DL	**0.0118**	**0.0782**	0.0205	0.1021	0.0133	0.0847	0.0413	0.1664	0.0155	0.0919	*0.0091*	*0.0726*
	Traffic.UL	**0.0215**	0.1065	0.0292	0.1172	0.0223	0.1062	0.0248	0.1201	0.0219	**0.1019**	*0.0155*	*0.0883*
	Num.user	0.0072	0.0634	0.0128	0.0822	**0.0036**	**0.0454**	0.0144	0.0854	0.0039	0.0488	*0.0042*	*0.0487*
	Thrp.DL	**0.0192**	0.1062	0.0265	0.1217	0.0203	**0.1010**	0.0389	0.1459	0.0241	0.1105	*0.0106*	*0.0851*
	Thrp.UL	**0.0313**	**0.1339**	0.0514	0.1796	0.0352	0.1482	0.0490	0.1797	0.0452	0.1640	*0.0210*	*0.1184*
	RRC.connect	0.0072	0.0614	0.0441	0.1782	0.0041	0.0505	0.0616	0.2093	**0.0039**	**0.0464**	*0.0049*	*0.0497*
Cell 2	PRB.DL	**0.0300**	**0.1143**	0.0579	0.1731	0.0428	0.1426	0.0490	0.1615	0.0425	0.1424	*0.0141*	*0.0892*
	PRB.UL	**0.0346**	**0.1255**	0.1169	0.2846	0.0756	0.2153	0.0871	0.2416	0.0979	0.2586	*0.0156*	*0.0884*
	Traffic.DL	**0.0228**	**0.1050**	0.0398	0.1427	0.0646	0.1974	0.0372	0.1416	0.0296	0.1227	*0.0134*	*0.0841*
	Traffic.UL	**0.0505**	**0.1041**	0.0700	0.1348	0.0594	0.1134	0.0699	0.1362	0.0616	0.1172	*0.0167*	*0.0624*
	Num.user	**0.0596**	**0.1664**	0.1452	0.2942	0.0994	0.2413	0.1403	0.2867	0.1082	0.2500	*0.0162*	*0.0983*
	Thrp.DL	0.0426	0.1553	0.0826	0.2324	0.0396	0.1616	**0.0377**	**0.1365**	0.0483	0.1745	*0.0315*	*0.1355*
	Thrp.UL	**0.0235**	0.0922	0.0252	**0.0879**	0.0247	0.0900	0.0238	0.0881	0.0320	0.1150	*0.0104*	*0.0735*
	RRC.connect	**0.1205**	**0.2151**	0.2839	0.3935	0.2503	0.3567	0.3762	0.4835	0.3245	0.4331	*0.0411*	*0.1242*
Cell 3	PRB.DL	**0.0123**	**0.0827**	0.0208	0.1070	0.0150	0.0983	0.0356	0.1542	0.0167	0.1024	*0.0071*	*0.0672*
	PRB.UL	**0.0318**	**0.0956**	0.0490	0.1414	0.0323	0.0999	0.0703	0.1727	0.0430	0.1252	*0.0086*	*0.0713*
	Traffic.DL	**0.0234**	**0.1100**	0.0328	0.1373	0.0249	0.1209	0.0566	0.1982	0.0265	0.1230	*0.0154*	*0.0952*
	Traffic.UL	0.0371	0.0993	0.0505	0.1274	**0.0331**	0.1041	0.0447	**0.0980**	0.0662	0.1319	*0.0092*	*0.0721*
	Num.user	**0.0209**	**0.1164**	0.0626	0.2047	0.0312	0.1381	0.0769	0.2276	0.0388	0.1555	*0.0133*	*0.0962*
	Thrp.DL	**0.0416**	**0.1242**	0.0484	0.1278	0.0427	0.1321	0.0720	0.1622	0.0484	0.1438	*0.0102*	*0.0771*
	Thrp.UL	0.0545	0.0994	0.0609	0.1066	0.0560	0.0916	**0.0542**	**0.0853**	0.0557	0.1046	*0.0074*	*0.0671*
	RRC.connect	**0.0249**	**0.1278**	0.1947	0.3747	0.0590	0.1967	0.3086	0.4989	0.0737	0.2310	*0.0145*	*0.1022*
	Average	**0.0316**	**0.1073**	0.0653	0.1694	0.0447	0.1336	0.0752	0.1827	0.0523	0.1448	*0.0139*	*0.0847*

MAE: mean absolute error; MSE: mean square error; TFT: temporal fusion transformer

ferent data. For certain indictors, TFT and NHiTS achieved marginally superior performance. This discrepancy could be influenced by factors such as the specific characteristics of the sequence or the randomness inherent in the training process.

More importantly, it is noteworthy that the baseline methods necessitate the collection of historical data for training covering a significantly long period, which is 22.5 days in our experiment setup. In addition, efforts are required to train each model for individual cells and indicators. In contrast, TSNet is directly evaluated without any prior provision of historical sequences. In conclusion, TSNet demonstrates strong generalization abilities in wireless network status prediction, achieving the best performance on average across difference measurements without depending on historical training datasets.

4.3 Effectiveness of fine-tuning

Table 1 presents the performance of fine-tuning TSNet with MinFt (see the last two columns). Overall, MinFt achieves the best average MSE and MAE compared with zero-shot TSNet and other baseline methods. The average MSE and MAE drop from 0.0316/0.1073 for zero-shot TSNet to 0.0139/0.0847, which demonstrates the effectiveness of the fine-tuning method.

5 Conclusions

In this paper, we propose TSNet, a novel time series foundation model designed for network status prediction in DTs. TSNet demonstrates competitive zero-shot prediction capability on diverse network status measurements, achieving the best MSE and MAE prediction on average compared with fully supervised prediction models trained with historical data. For future endeavors, we plan to explore TSNetâĂŹs capability for unified time-series representation, and its extension to various downstream tasks including time series classification, anomaly detection, and pattern matching.

Contributors

NengwenZhao and Yijun Yu designed the research. Siyao Song and Guoao Sun processed the data and conducted experiments. Siyao Song and Yifan Chang drafted the manuscript. Nengwen Zhao revised and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

Data available on request from the authors. The data that support the findings of this study are available from the corresponding author Nengwen Zhao, upon reasonable request.

Unedited

References

Breiman L, 2001. Random forests. *Mach. Learn.*, 45(1):5-32. https://doi.org/10.1023/A:1010933404324

Challu C, Olivares KG, Oreshkin BN, et al., 2022. N-HiTS: Neural hierarchical interpolation for time series forecasting. https://doi.org/10.48550/arXiv.2201.12886

Chang C, Wang WY, Peng WC, et al., 2024. LLM4TS: Aligning pre-trained LLMs as data-efficient time-series forecasters. https://doi.org/10.48550/arXiv.2308.08469

Das A, Kong WH, Sen R, et al., 2024. A decoder-only foundation model for time-series forecasting. https://doi.org/10.48550/arXiv.2310.10688

Ding N, Qin YJ, Yang G, et al., 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nat Mach Intell*, 5(3):220-235. https://doi.org/10.1038/s42256-023-00626-4

Drucker H, Burges CJC, Kaufman L, et al., 1996. Support vector regression machines. Proc 9th Int Conf on Neural Information Processing Systems, p.155-161.

Ekambaram V, Jati A, Nguyen N, et al., 2023. TSMixer: Lightweight MLP-mixer model for multivariate time series forecasting. Proc 29th ACM SIGKDD Conf on Knowledge Discovery and Data Mining, p.459-469. https://doi.org/10.1145/3580305.3599533

Goodfellow IJ, Bengio Y, Courville A, 2016. Deep Learning. MIT Press, Cambridge, MA, USA.

Hochreiter S, Schmidhuber J, 1997. Long short-term memory. *Neural Comput*, 9(8):1735-1780.

HoHyndman RJ, Athanasopoulos G, 2018. Forecasting: Principles and Practice 2nd ed. OTexts, Melbourne, Australia.

Houlsby N, Giurgiu A, Jastrzebski S, et al., 2019. Parameter-efficient transfer learning for NLP Proc 36th Int Conf on Machine Learning, p.2790-2799.

Khan LU, Han Z, Saad W, et al., 2022. Digital twin of wireless systems: Overview, taxonomy, challenges, and opportunities. *IEEE Commun Surv Tut*, 24(4):2230-2254. https://doi.org/10.1109/COMST.2022.3198273

Lea C, Vidal R, Reiter A, et al., 2016. Temporal convolutional networks: A unified approach to action segmentation Computer Vision-ECCV 2016 Workshops, p.47-54. https://doi.org/10.1007/978-3-319-49409-8_7

Lim B, Zohren S, 2021. Time-series forecasting with deep learning: A survey. *Philos Trans A Math Phys Eng Sci*, 379(2194):20200209. https://doi.org/10.1098/rsta.2020.0209

Lim B, ArÄśk SÃŰ, Loeff N, 2021. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *Int J Forecast*, 37(4):1748-1764. https://doi.org/10.1016/j.ijforecast.2021.03.012

Lin TY, Wang YX, Liu XY, et al., 2022. A survey of transformers. *AI Open*, 3:111-132. https://doi.org/10.1016/j.aiopen.2022.10.001

Lindemann B, MÃijller T, Vietz H, et al., 2021. A survey on long short-term memory networks for time series prediction. *Procedia CIRP*, 99:650-655. https://doi.org/10.1016/j.procir.2021.03.088

Liu S, Yu H, Liao C, et al., 2022. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. 10th Int Conf on Learning Representations, p.20.. https://doi.org/10.34726/2945

Liu Y, Wu HX, Wang JM, 2022. Non-stationary transformers: Exploring the stationarity in time series forecasting Proc 36th Int Conf on Neural Information Processing Systems, p.9881-9893.

Liu Y, Hu TG, Zhang HR, et al., 2024. iTransformer: Inverted transformers are effective for time series forecasting. https://doi.org/10.48550/arXiv.2310.06625

Liu YJ, Dong HB, Wang XM, et al., 2019. Time series prediction based on temporal convolutional network. IEEE/ACIS 18th Int Conf on Computer and Information Science, p.300-305. https://doi.org/10.1109/ICIS46139.2019.8940265

Lu K, Grover A, Abbeel P, et al., 2022. Frozen pretrained transformers as universal computation engines. Proc 36th AAAI Conf on Artificial Intelligence, p.7628-7636. https://doi.org/10.1609/aaai.v36i7.20729

Makridakis S, Spiliotis E, Assimakopoulos V, 2018. The M4 competition: Results, findings, conclusion and way forward. *Int J Forecast*, 34(4):802-808. https://doi.org/10.1016/j.ijforecast.2018.06.001

Mehrmolaei S, Keyvanpour MR, 2016. Time series forecasting using improved ARIMA. Artificial Intelligence and Robotics, p.92-97. https://doi.org/10.1109/RIOS.2016.7529496

Natekin A, Knoll A, 2013. Gradient boosting machines, a tutorial. *Front Neurorobot*, 7:21. https://doi.org/10.3389/fnbot.2013.00021

Nie YQ, Nguyen NH, Sinthong P, et al., 2023. A time series is worth 64 words: Long-term forecasting with transformers. https://doi.org/10.48550/arXiv.2211.14730

Oreshkin BN, Carpov D, Chapados N, et al., 2020. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. https://doi.org/10.48550/arXiv.1905.10437

PaaÃ§ G, Giesselbach S, 2023. Foundation Models for Natural Language Processing: Pre-Trained Language Models Integrating Media. Springer, Cham. https://doi.org/10.1007/978-3-031-23190-2

Papa L, Russo P, Amerini I, et al., 2024. A survey on efficient vision transformers: algorithms, techniques, and performance benchmarking. https://doi.org/10.48550/arXiv.2309.02031

Rebuffi SA, Bilen H, Vedaldi A, 2017. Learning multiple visual domains with residual adapters. Proc 31st Int Conf on Neural Information Processing Systems, p.506-516.

Salinas D, Flunkert V, Gasthaus J, et al., 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *Int J Forecast*, 36(3):1181-1191. https://doi.org/10.1016/j.ijforecast.2019.07.001

Shiblee M, Kalra PK, Chandra B, 2009. Time series prediction with multilayer perceptron (MLP): A new generalized error based approach. 15th Int Conf on Advances in Neuro-Information Processing, p.37-44. https://doi.org/10.1007/978-3-642-03040-6_5

Vaswani A, Shazeer N, Parmar N, et al., 2017. Attention is all you need. Proc 31st Int Conf on Neural Information Processing Systems, p.6000-6010.

Wu HX, Hu TG, Liu Y, et al., 2023. TimesNet: Temporal 2D-variation modeling for general time series analysis. https://doi.org/10.48550/arXiv.2210.02186

Unedited

Zhou HY, Zhang SH, Peng JQ, et al., 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. https://doi.org/10.48550/arXiv.2012.07436

Zhou T, Ma ZQ, Wen QS, et al., 2022. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. Proc 39th Int Conf on Machine Learning, p.27268-27286.

Zhou T, Niu PS, Wang X, et al., 2023. One fits all: Power general time series analysis by pretrained LM. https://doi.org/10.48550/arXiv.2302.11939

Å¡uniÄĞ E, KorjeniÄĞ K, HodÅ¿iÄĞ K, et al., 2020. Application of FacebookâĂŹs prophet algorithm for successful sales forecasting based on real-world data. *Int J Comput Sci Inf Technol*, 12(2):23-36. https://doi.org/10.5121/ijcsit.2020.12203

Unedited