Trend Encoder with Attentive Neural Process: Node Performance Extrapolation for Non-Time-Series Datasets

Kyota Hattori
NTT Network Service Systems Laboratories
NTT Corporation
Tokyo, Japan
kyota.hattori@ntt.com

Tomohiro Korikawa
NTT Network Service Systems Laboratories
NTT Corporation
Tokyo, Japan
tomohiro.korikawa@ntt.com

Chikako Takasaki
NTT Network Service Systems Laboratories
NTT Corporation
Tokyo, Japan
chikako.takasaki@ntt.com

***Abstract*— Future network infrastructures will support multiple heterogeneous networks to facilitate 6G and network disaggregation. This will require the verification of various types of devices and components. Consequently, the efficiency of performance verification needs to be enhanced for the combinations of numerous network nodes and components, considering unknown network conditions. This study focuses on improving the inference accuracy of network node performance in the extrapolation. To address this issue, we propose a trend encoder for non-time-series datasets, which collaborates with an Attentive Neural Process. Preliminary extrapolation results show that the coefficient of determination of router throughput is improved by paying more attention to the trend information, including the relationship between both router configurations and input traffic influencing router performance for non-time-series datasets.**

Keywords*— *Trend encoder for non-time-series datasets, Attentive Neural Process, Node modeling

I. INTRODUCTION

The sixth generation of mobile communications networks (6G) is expected to realize an Intelligent Internet of Intelligent Things (IIoIT). This evolution will provide ultra-fast mobile broadband services, low-latency applications such as remote surgery and autonomous vehicles, as well as support a wide range of connected devices for building a virtual digital world [1]. This necessitates 6G to offer significantly faster speeds, lower latency, and wider coverage than 5G. Hence, 6G networks will be able to seamlessly connect multiple heterogeneous networks based on a mix of different types of technologies, such as wireless, satellite, and optical networks, allowing for faster and more reliable communication between other technologies.

One candidate to support multiple heterogeneous networks for 6G is network disaggregation technology [2]. Network disaggregation involves disassembling previously integrated components into their individual network components. In the current network, network components often come with proprietary interfaces, software, and hardware, which need more flexibility and cost-efficiency, failing to meet the growing demands of telecommunication carriers. By breaking free from vendor lock-in constraints, telecommunication carriers will embrace network disaggregation technologies, allowing them to leverage the most suitable and advanced technology from various suppliers to meet their service requirements. However, implementing this technology will result in a surge in the number of network components, necessitating efficient management and verification of the performance of numerous combinations of network elements.

Hence, ensuring network quality and reliability across diverse network equipment and optimal combinations of numerous network components to support network disaggregation will become essential. In response to this challenge, we present a network digital replica (NDR) [3]. The NDR acts as a digital counterpart to the physical network, facilitating the verification of network node performance in the digital domain. To achieve this, network node modeling plays a crucial role as it replicates the performance of actual network nodes within the digital domain. In the NDR, the network node modeling is performed using machine learning (ML) based modeling for actual network node performance, such as the router throughput. However, the application of network node modeling has been considered only for known environments, i.e., where interpolation between the training datasets is possible. Neural networks (NNs), which are commonly used in ML, are not particularly effective at extrapolation [4]. Therefore, the issue of the NDR is to extrapolate the node performance in the digital domain.

Given the above background, this paper presents a novel trend encoder to improve the inference accuracy of network node performance in extrapolation. The main contribution of this study is a trend encoder for non-time-series datasets with a meta-learner: attention-based NP combined with trend information as follows:

1) Trend encoder for non-time-series datasets: We propose the trend encoder to enrich information for non-time-series datasets to extrapolate node performance. This encoder captures the trend information representing the relationship between both node configurations (including hardware specifications) and input traffic influencing node performance. To achieve this, the proposed trend encoder reorders the datasets, uses feature importance [5] to assess the importance of each dataset for node performance, and uses wavelet transform [6] to identify the trend with high-ranked feature importance.

2) Attention-based NP combined with trend information to extrapolate node performance: As the basic method to extrapolate the router performance, we improve an Attentive NP (ANP) [7], a kind of meta learner, to enable each node performance to focus on the extracted trend information.

To the best of our knowledge, few or no existing reports on node modeling utilize regression prediction to extrapolate

979-8-3503-7099-7/24/$31.00 ©2024 IEEE

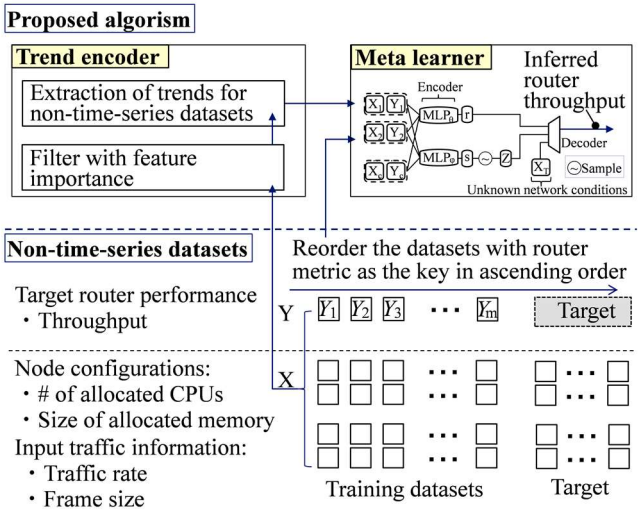

Fig. 1. Proposed trend encoder for non-time-series datasets with meta learner to extrapolate the router performance.

actual node performance by using trend information for non-time-series datasets.

II. Related Work

Several studies have evaluated the performance of ML techniques in extrapolation [8-9]. Decugis et al. [8] investigated the potential of implicitly defined NNs to extrapolate on mathematical tasks. In addition, a Gaussian process (GP) framework has been proposed [9]. The GP is used for modeling a distribution over regression functions. The GP would be better at extrapolation if the kernel were better defined for extrapolation. However, an appropriate prior for the GP is difficult to design.

As an alternative model to GP, a neural process (NP) has been introduced, which allows to learn a stochastic process from data using the flexibility of NNs [10]. NP is a kind of meta-learner applied to situations where each task has only a few examples. However, NP has not yet been applied to the inference of node performance for actual equipment. Also, the accuracy of extrapolation needs to be improved for applying only vanilla NP to the inference of network node performance that is not covered by training datasets, as evaluated in this paper.

Approaches similar to ours have been proposed in several studies that propose the integration of wavelet transform and deep learning techniques [11]. However, this analysis is for time series datasets, and correlations between the before and after time periods are relatively straightforward. Since the dataset assumed in this research is non-time-series, there is no correlation between adjacent data points. Therefore, we aim to extract trends from non-time-series datasets and incorporate them to improve the accuracy of extrapolating node performance.

III. Background

A. Interpolation and Extrapolation for Node Performance

The purpose of this study is to address extrapolation for network node performance in regression problems. In regression problems covered in this study, non-time-series datasets are used to generalize a function that maps a set of input variables X to node performance Y. Here, the input variables include node configurations and input traffic. When the input variables fall within the range of the training datasets, this process is defined as interpolation. Conversely, if the inferred point lies outside this range, it is defined as extrapolation. Extrapolating the node performance beyond its training domain is generally challenging due to its sensitivity to the training data and model parameters. This results in unpredictable behavior unless the model formulation incorporates implicit or explicit assumptions about the extrapolation process.

B. Neural Process

The primary objective of NP is to acquire knowledge of distributions over functions by learning from distributions over datasets [10]. First, to enable NP to learn distributions over functions, we consider a set of datasets, $D=\{(x_i, y_i)\}_{i=1}^{n}$ of n inputs x_i and outputs y_i. NP splits D into two disjoint subsets: a set of m context points $C=\{(x_i, y_i)\}_{i=1}^{m}$ and a set of targets $T=\{(x_j, y_j)\}_{j=m+1}^{n}$. The NP model is then presented with C to estimate the corresponding function values for T. Basic NP is characterized by an encoder-decoder network structure that models a stochastic process. NP encoder consists of two paths: the deterministic path and the latent path. For the deterministic path, the encoder produces representations r_i for each of the context pairs C using a multi-layer perceptron (MLP_θ) based on the formula: $r_i = \mathrm{MLP}_\theta(C)$. The individual representations r_i are combined into a unified representation r_C using a permutation invariant operator (e.g., addition). On the other hand, the latent path of the NP encoder calculates a representation $s_i = \mathrm{MLP}_\gamma(C)$ similar to the deterministic path. Unlike the deterministic path, however, the latent path uses stochastic layers to obtain a distribution of the latent variable Z, which is calculated using s_i. Z is assumed to be a normal distribution, so Z is calculated on the basis of the formula: $Z = N(\mu(s), \sigma^2(s))$, where μ and σ are the mean and standard deviation, and s is a single representation obtained by combining r_i. Finally, by concatenating these aggregated r_C, Z with the target inputs X_T, the decoder MLP_φ produces the predictions $\hat{y}_T$ based on the formula: $\hat{y}_T = \mathrm{MLP}_\varphi(X_T, r_C, Z)$.

C. Attentive Neural Process

Attention mechanisms in deep learning are designed to extract crucial information from data. This is especially important in computer vision and natural language processing [12]. At their core, attention mechanisms operate on the premise that outputs are influenced by particular input segments that hold relevance. By identifying these relevant parts, attention allows the model to focus on critical aspects and make more informed decisions. Recently, NP has been enhanced with attention mechanisms, resulting in ANP [7]. ANP incorporates attention mechanisms to address the underfitting issues observed in standard NP. Unlike NP, which produces an aggregated variable r_C from the deterministic path, ANP utilizes an attention mechanism to summarize the information in the context set most relevant to the target sets.

D. Wavelet Decomposition

Wavelet transform is a method used to decompose signals using a system of wavelets, where each wavelet is a function that represents a shifted and scaled copy of a base function. Specifically, Empirical Wavelet Transform (EWT) is an adaptive wavelet transform technique [6]. The EWT has been recognized as a valuable tool for dealing with non-stationary

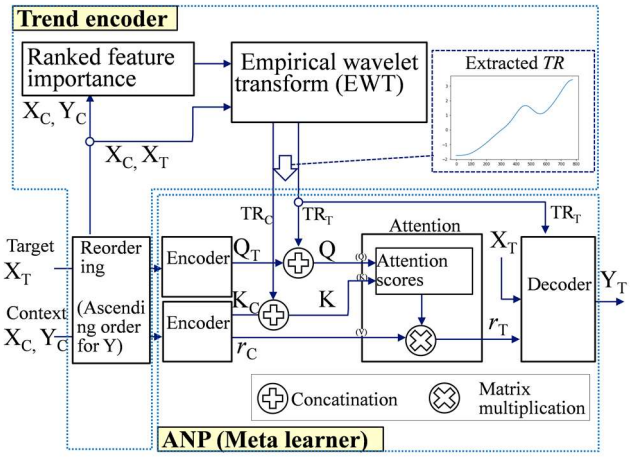

Fig. 2. Procedure of proposed trend encoder with ANP.

signals [13]. The EWT enables the adjustment of the decomposition layers of the signal, which is represented as *N*. The output of the EWT undergoes filtration using both the scaling function and *N* different wavelet functions. Specifically, any signal *f(t)* is decomposed into different frequency bands based on the empirical scale function ϕ and the empirical wavelet function ψ, and then the appropriate filter bank is selected. The formula for the reconstructed original signal is $f(t) = W_f(0,t)\phi_1(t) + \sum_{n=1}^{N} W_f(n,t)\psi_n(t)$, where $W_f(0,t)$ is the scaling coefficients and $W_f(n,t)$ is the wavelet coefficients. At each stage of the process, the high-frequency component is derived from the wavelet coefficients, while the low-frequency component is obtained from the scaling coefficients.

IV. Proposed Node Modeling

A. Concept of Network Digital Replica

An NDR digitally evaluates the performance of a network node for unknown external network conditions without touching the actual network for network designing [3]. To achieve this, the NDR creates a network node model to digitally evaluate how well the current node configuration can ensure performance against future traffic increases. In this work, we primarily focus on studying router modeling for fixed networks, which is a crucial component of carrier networks as an initial investigation.

B. Proposed Node Modeling Algorithm for Extrapolation

Our research aims to improve the inference accuracy of router performance (specifically, router throughput) for extrapolation to support unknown traffic conditions. To address this issue, the basic idea is to capture the trend of changing router performance in extrapolation for non-time-series datasets. Therefore, the proposed method extracts and learns the relationship between both router configurations and input traffic influencing router performance; we call this relationship "trend information".

Fig.1 shows the architecture of the proposed method. The proposed method consists of two elements: (1) a trend encoder for non-time-series datasets and (2) ANP combined with trend information to extrapolate router performance. The proposed trend encoder extracts rough relationships between the targeted router performance and the other datasets. It employs feature importance [5] to prioritize data for extraction. The feature importance is a technique to estimate the relationship between input features and model outcomes. The proposed trend encoder assumes that the increasing trend of router performance in extrapolation correlates with a similar trend of the training dataset with high-ranked feature importance. Meanwhile, ANP, equipped with an attention mechanism, focuses on the trend information between router configurations and input traffic volume on router performance, enhancing the extrapolation of router performance.

Fig. 2 shows the procedure of the proposed trend encoder with ANP. This procedure consists of four steps:

Step 1: Acquisition of training datasets from routers

The initial step involves obtaining training datasets for router performance from a router. To acquire the measured router throughput P_{th}, a traffic generator is employed. This traffic generator measures the router throughput while changing the input traffic conditions and the router configurations.

Step 2: Extracting the trend information from non-time-series datasets for router performance via trend encoder

To acquire the trend of the relationship between training datasets and router performance, the proposed method deploys a trend encoder, which consists of dataset reordering, feature importance extraction, and a wavelet filter. First, the proposed method reorders the datasets with "router throughput" as the key in ascending order. The proposed algorithm calculates the feature importance [5] to extract the main trends contributing to router throughput. For example, the virtual router throughput tends to increase as the router configuration, such as the number of allocated virtual CPUs and memory allocation size, increases. Consequently, the feature importance of these two factors attains a higher value. Considering these data trends with high-ranked feature importance, the proposed method extrapolates the router throughput. We apply the EWT [6] to capture the trend information for the data selected by the feature importance. The proposed algorithm uses the scaling coefficients extracted by the EWT, which is a low-frequency feature that reflects the overall trend of the data. Here, we define $S_0(X)$ as the lowest frequency decomposed from data X by the EWT. The extracted trend information (TR) is calculated by multiplying this extracted low-frequency information by each feature importance to consider the contribution to router throughput, and each extracted TR is added in descending order of ranked feature importance as follows;

$$\mathrm{TR}_i = \sum_j^L S_0\ (X_i^j) \cdot \mathrm{FI}^j,$$

where i is the position index of datasets, FI^j is j-th ranked feature importance normalized such that they sum up to 1, X^j is datasets X with j-th ranked FI, and L is the number of datasets considering the trends. L is a hyperparameter.

Step 3: Training the model with the trend information based on ANP

To incorporate the router throughput with the trend information TR, we add the extracted TR to the attention layer of ANP. The attention mechanism can explicitly express the distance between context and target points via attention weights. The attention mechanism uses three vectors: Query (Q), Key (K), and Value (V). Q and K are used to compute a score that indicates the importance of a given V. The higher the score, the more important V is considered to be. V is then

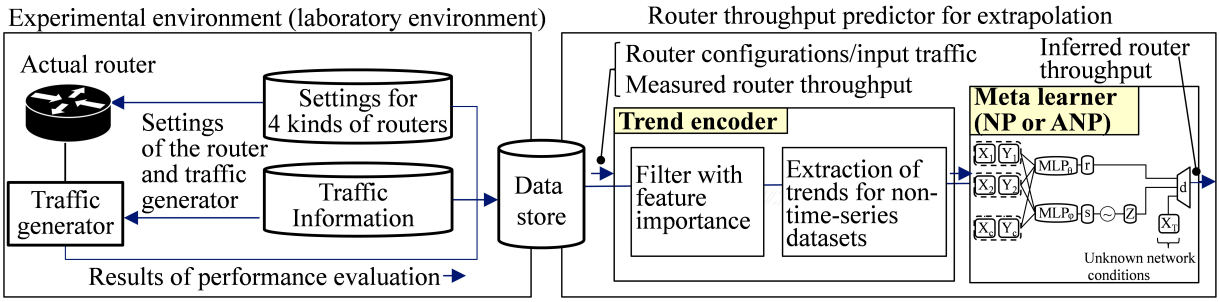

Fig. 3 Experimental configurations.

TABLE I. EXPERIMENTAL CONDITIONS

Router configurations (R)	Number of physical ports (N_{port})	2 to 8
	Number of flow entries ($N_{entries}$)	0.4 k to 770 k
	Number of allocated virtual CPU cores (N_{CPUs})	4 to 36
	Size of memory allocation (S_{Mem}) (GB)	12 to 64
Input traffic (T)	Ethernet frame size (S_{Eth}) (Bytes)	64 to 1518
	Number of traffic flows (N_{Flow})	0.4 k to 770 k
	Input traffic rate per flow (R_{Input}) (bps)	64 k to 80 M

used to construct the output of the attention mechanism. The representations through the attention layer in the proposed algorithm, denoted as r_T, can be defined like [12] as follows:

$$r_T = \text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$
$$Q = [Q_T, \text{TR}_T], K = [K_C, \text{TR}_C], V = r_C,$$

where $[\cdot,\cdot]$ represents the concatenation of two vectors, d_k is the dimension of the input features, and r_C is the encoded representation of X_C. Q_T and K_C are vectors of X_T and X_C, respectively. Note that this trend information is not exclusive to ANP and can also be integrated with NP. For NP, the representations r_i produced by the encoder are added as follows: $r_i \leftarrow [r_i, \text{TR}_i]$. By adding TR to each feature, the model can learn rough relationships between the targeted router throughput and the other datasets in extrapolation.

Step 4: Inference for the accuracy of the router performance with ANP

The router throughput is inferred with the trend information for the datasets, including router configurations R and input traffic conditions T using ANP. The accuracy of the router throughput model was evaluated using the coefficient of determination (R^2). R^2 represents a statistical measure used to examine the correlation between the actual observed values and the predicted output by a model. R^2 is described in R^2 =1-$\sum(y_i - \hat{y})^2/\sum(y_i - \bar{y})^2$, where $\hat{y}$ and $\bar{y}$ are the predicted value and the average measured value of the router throughput y, respectively. When R^2 = 1, it signifies that the model provides an ideal fit to the datasets, achieving a perfect match between the predicted and actual values. If the predicted values deviate too much from the measured values, then R^2 could be negative.

V. EVALUATION & RESULTS

We evaluated the accuracy of the inferred router throughput by applying the proposed method. In this study, we evaluated how router performance could be extrapolated, assuming an increase in traffic volume. Fig. 3 presents the experimental configurations. We applied the proposed trend encoder, combined with either ANP or NP, to the measured results of the performance on four kinds of virtual routers, including Cisco CSR 1000V [14] and Juniper vMX [15] using a x86-based server: two Xeon E5-2697 18-cores 2.30 GHz CPUs and 192 GB RAM with 8 SFP+ ports. First, we measured the performance of these routers in the packet forwarding process in a laboratory environment to acquire training datasets under the conditions shown in Table I. These conditions were established to assess the impact of the relationship between router configurations and traffic conditions on router throughput, including conditions of overloaded traffic. The Keysight Ixia platform with 8 SFP+ ports generated the traffic in the experiments. In total, 930 samples were acquired from these routers. In this scenario, we measured P_{th} (bits per second) for each T and R. The training datasets consist of R (number of physical ports: N_{port}, number of flow entries: $N_{entries}$, number of allocated virtual CPU cores: N_{CPUs}, and size of memory allocation: S_{Mem}) and T (Ethernet frame size: S_{Eth}, number of traffic flows: N_{Flow}, and average rate of input traffic: R_{Input}). To evaluate the accuracy of the extrapolation, we define the splitting ratio ($\gamma \in [0,1]$) to categorize our datasets. The smallest $100 \times \gamma$ % of datasets, sorted by "router throughput" in ascending order, are used for training. The remaining $100 \times (1-\gamma)$ % of the datasets are reserved for testing in the extrapolation. This indicates that a smaller γ value results in a narrower training domain, making extrapolation predictions more challenging due to limited training datasets. On the other hand, for interpolation, 100×γ % of datasets are used for training with random sampling, while the remaining is used for testing. We utilized the Python package 'ewtpy' [16] to implement the EWT. We chose a length of 10 for the EWT filter (: N). The detailed implementation of the EWT used in this paper is given in [13]. We carried out our work using PyTorch [17], which is widely used as an open-source deep-learning library. Both the encoder and decoder of NP and ANP are constructed with 10 hidden layers, and each layer consists of 128 hidden units. The dimensionality of Z and the context points C were set to 128 and 50, respectively. Q_T and K_C are each generated by 2 layers of multi-layer perceptron. The Adam optimizer [18] is used with a learning rate of $1e^{-4}$. To build the model, 10,000 epochs were run. Then, we evaluated R^2 for interpolation (R^2_{in})and that for extrapolation (R^2_{ex}) to clarify how well the proposed model inferred P_{th}.

Fig. 4 shows the feature importance with the top 5 features for P_{th}. In this evaluation condition, N_{port} and N_{CPUs} were the major contributions to the P_{th} model. As the N_{port} increases, the amount of traffic to transfer the packets also increases, and as a result, the chances of an increase in the P_{th} itself also increase. In addition, the N_{CPUs} were the main factor for the performance

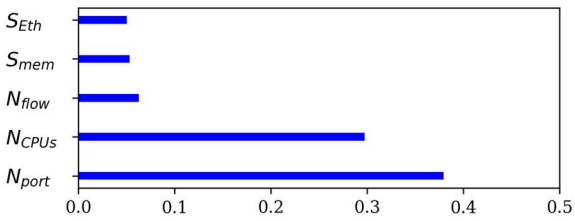

Fig. 4. Feature importance with the top 5 features.

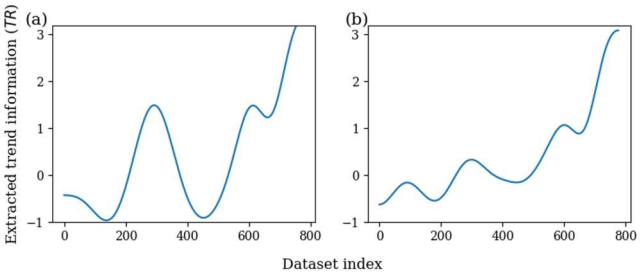

Fig. 5. TR for (a) N_{port} and (b) N_{port} and N_{CPUs}.

to store and process the packets in the virtual routers. This indicates a potential for enhanced accuracy in extrapolating P_{th} by extracting the TR of N_{port} and N_{CPUs}, and incorporating it into the datasets.

Fig.5 shows the results of TR, which represent the low-frequency signals extracted by EWT from the datasets with the higher ranked feature importance for P_{th}: (a) N_{port} (L=1) and (b) N_{port} and N_{CPUs} (L=2). Since P_{th} are sorted in ascending order, TR exhibits an upward trend. Fig. 5(b) shows that the increase in P_{th} is proportional to N_{port} and N_{CPUs}. This result suggests that the proposed method has successfully extracted the trend for the relationship between router configurations and P_{th}. By using the superposition of TR in the feature importance order for the targeted router throughput, the proposed method captures the trend information related to the router throughput.

We then evaluated the robustness of the proposed method for extrapolating P_{th} by changing the splitting ratio γ using the TR in Fig.5(b). Fig. 6 shows the results of R^2_{in} and R^2_{ex} for the inferred P_{th} against γ with/without applying the proposed method to ANP and NP. The proposed method improved R^2 compared to the conventional method, including vanilla ANP and NP, even for smaller values of γ. In particular, R^2_{ex} with the proposed method using ANP improved and remained close to 0.5, while that with the conventional method fell to negative values for γ between 0.5 and 0.7. R^2_{ex} with the proposed method using NP also improved, although not as significantly as with ANP, reaching about 0.15. This indicates that the proposed method has the capability to extrapolate the router throughput until the training and test data ratio is about half (γ = 0.5). Furthermore, although it is not the main focus of this study, for R^2_{in}, which demonstrates the performance of interpolation, the proposed methods achieved 0.95 or higher (with an average improvement of 3%) for the measured range of γ. These improvements are attributed to the successful extraction of TR using the proposed trend encoder for non-time series datasets and the attention mechanism's focus on these trends. As a result, we found that the proposed method provides superior accuracy in extrapolating router throughput compared to the conventional ANP or NP for the given scenario.

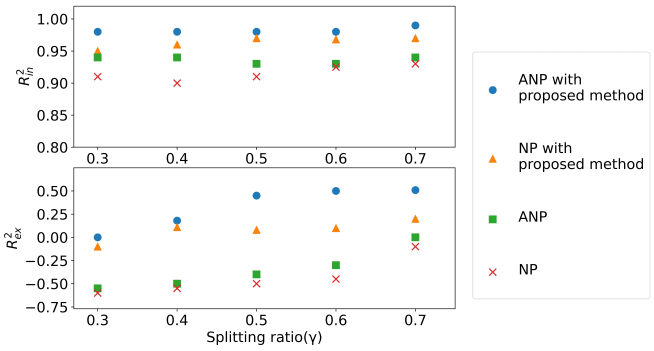

Fig. 6. Evaluation results of R^2_{in} and R^2_{ex} for P_{th}.

VI. Conclusion

We proposed a trend encoder for non-time-series datasets with an Attentive Neural Process (ANP) to improve the accuracy of extrapolating router performance. The proposed method improved the coefficient of determination for extrapolating router throughput from negative values to about 0.5 compared to the vanilla ANP until the training and test data ratio is about half on the given dataset.

References

[1] M. Giordani et al., "Toward 6G Networks: Use Cases and Technologies," IEEE Commun. Mag., vol. 58, no. 3, pp. 55-61, Mar. 2020.

[2] K. Ishii, R. Matsumoto, T. Inoue, and S. Namiki, "Disaggregated optical-layer switching for optically composable disaggregated computing [Invited]," J. Opt. Commun. Netw. 15, A11-A25, 2023.

[3] K. Hattori et al., "Recursive Router Metrics Prediction Using ML-based Node Modeling for Network Digital Replica," in IEEE GLOBECOM, 2022.

[4] K. Xu et al., "How neural networks extrapolate: from feedforward to graph neural networks," ICLR, 2021.

[5] J. Heaton, "Feature Importance in Supervised Training," Predictive Analytics and Futurism Newsletter, No. 17, April 2018.

[6] J. Gilles, "Empirical wavelet transform," IEEE Trans. Signal Process., vol. 61, no. 16, pp. 3999–4010, Aug. 2013

[7] H. Kim et al., "Attentive neural processes," arXiv:1901.05761, 2019.

[8] J. Decugis et al., "On the Abilities of Mathematical Extrapolation with Implicit Models," NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications.

[9] A. G. Wilson, E. Gilboa, A. Nehorai, and J. P. Cunningham, "GPatt: Fast multidimensional pattern extrapolation with Gaussian processes," arXiv preprint arXiv:1310.5288. 148, 2013.

[10] M. Garnelo et al., "Neural processes," arXiv:1807.01622, 2018.

[11] K. A. Althelaya, S. A. Mohammed, and E.-S. M. El-Alfy, "Combining deep learning and multiresolution analysis for stock market forecasting," IEEE Access, vol. 9, pp. 13 099–13 111, 2021.

[12] A. Vaswani et al., "Attention is all you need," in Proc. Adv. Neural Inf. Process. Syst., pp. 5998–6008, 2017.

[13] Q. Wang et al., "Future Trend Forecast by Empirical Wavelet Transform and Autoregressive Moving Average," Sensors, 2621, 2018.

[14] Cisco, "Cisco Cloud Services Router 1000v Series," https://www.cisco.com/c/en/us/products/routers/cloud-services-router-1000v-series/index.html (accessed Aug. 30, 2023).

[15] Juniper, "vMX Virtual Router," https://www.juniper.net/gb/en/products/routers/mx-series/vmx-virtual-router-datasheet.html (accessed Aug. 30, 2023).

[16] V. R.Carvalho et al., "Evaluating five different adaptive decomposition methods for EEG signal seizure detection and classification," Biomed. Signal Process. Control, vol. 62, p. 102073, 2020.

[17] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," Adv. Neural Inf. Process. Syst., vol. 32, pp. 8024–8035, 2019.

[18] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," ICLR, 2015.