On Forecasting-Oriented Time Series Transmission: A Federated Semantic Communication System

Journal:	*Transactions on Mobile Computing*
Manuscript ID	TMC-2023-12-1386
Manuscript Type:	Regular
Keywords:	semantic communication, federated learning, time series forecasting, collaborative inference, mobile edge computing

SCHOLARONE™
Manuscripts

On Forecasting-Oriented Time Series Transmission: A Federated Semantic Communication System

Bowen Zhao, Huanlai Xing, Lexi Xu, Yang Li, Li Feng, Jincheng Peng, Zhiwen Xiao

***Abstract*—Time series data widely exist in public services, industrial environments and military applications. Traditionally, transmission of a huge volume of data for analytic tasks poses challenges, particularly in mobile environments with limited computing and communication resources. Semantic communication emerges as a solution for intelligently extracting various features from source data and efficiently transmitting task-related information to receivers, thereby reducing bandwidth consumption significantly. In this paper, we introduce a novel federated semantic communication system tailored for forecasting-oriented time series transmission tasks. The correlation of source data collected from terminal devices is mined and the corresponding semantic information is transmitted to an edge server for collaborative inference. To optimize the semantic analysis process, we devise a deep decomposition block at the transmitter side, decomposing time series into trend and multiple period components. This reduces noise interference from wireless channels, enhancing the overall transmission quality. For effective training and collaborative inference, we propose a FEDerated Mixture of period Router (FedMoR) architecture. Within each channel encoder, period routers are divided into private and public ones. Private routers extract specialized features from individually collected data, mitigating accuracy degradation. Public routers share knowledge across all transmitters, enhancing temporal analysis robustness. Simulation results demonstrate that the FedMoR system outperforms two traditional technique-based and two semantic communication-based baselines under three common channels. The system achieves low mean square errors on five widely-used real-world time series forecasting datasets, particularly in the low signal-to-noise ratio regime.**

***Index Terms*—semantic communication, federated learning, time series forecasting, collaborative inference, mobile edge computing**

I. Introduction

THE development of the internet of things (IoT) and the proliferation of smart devices across various domains have brought comprehensive and intelligent services in the past few years [1], [2]. As these services run continuously, a large volume of time series data is naturally generated documenting their histories. As one of the most important and widely used technologies, time series forecasting (TSF) has underlain plenty of real-world scenarios in fields such as personal services, environmental monitoring, and smart cities [3]. For instance, electrocardiograph and blood pressure detection in smart health, temperature and humidity forecasting in weather monitoring, traffic flow and vehicle speed analysis in intelligent transportation are essentially TSF-based applications as shown in Fig. 1. Traditionally, almost all of these data are transmitted to data centers or central servers for further scrutiny. However, continuous transmission of large amount of data brings heavy pressure to traditional communication systems [4], resulting in slow real-time response and significant bandwidth consumption [5].

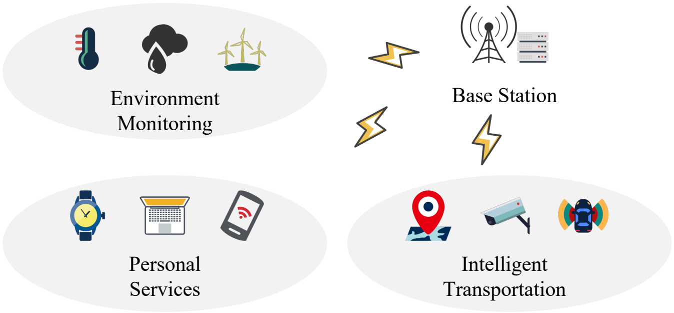


Fig. 1. Examples of TSF-based real-world applications.

In reality, a substantial portion of the time series data collected doesn't necessitate complete transmission to a data center. For instance, environmental indicators like temperature, humidity, and wind speed, remain stable and stay within normal ranges most of the time. Consequently, monitors are more interested in deducing and transmitting exceptional records during abnormal periods rather than the entire data collected.

Based on deep learning, semantic communication (SC) emerges as one of the most promising techniques in 6G networks [6]. Different from traditional communication methods striving for bit-level accuracy, SC allows a transmitter to send the extracted semantic information to a receiver rather than the original source data. Through analyzing the relationship between downstream requirements and the associated data, this paradigm efficiently compresses the information to be transmitted over a wireless channel, significantly reducing the bandwidth consumption. Studies on SC are usually classified into two categories: reconstruction-oriented [7], [8] and task-oriented [9], [10]. As the emergence of increasing number of intelligent services, task-oriented SC holds substantial research

Bowen Zhao, Huanlai Xing, Yang Li, Li Feng, Jincheng Peng, Zhiwen Xiao are with the School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 610031, China, also with the Tangshan Institute of Southwest Jiaotong University, Tangshan 063000, China, and also with the Engineering Research Center of Sustainable Urban Intelligent Transportation, Ministry of Education, China.(e-mail: hxx@home.swjtu.edu.cn)

Lexi Xu is with the Research Institute, China United Network Communications Corporation, 100048 Beijing, China, and also with the Queen Mary University of London, London, U.K. (e-mail: davidlexi@hotmail.com).

This work was partially supported by the National Natural Science Foundation of China (No. 62172342 and No.62202392), the Natural Science Foundation of Hebei Province (No. F2022105027), the Natural Science Foundation of Sichuan Province (No. 2022NSFSC0568, No. 2022NSFSC0944, and No. 2023NSFSC0459), and the Fundamental Research Funds for the Central Universities, P. R. China.

and application value [11], bringing benefits to efficient time series transmission in various IoT applications.

Deficient Studies on TSF-oriented SC: Transmission of forecasting-oriented temporal information in mobile environments is an inevitable yet challenging problem, widely witnessed in real-world public services [12], such as, industrial anomaly detection, traffic flow prediction, and disaster warning. Traditional communication techniques aim at accurate transmission of bits, leading to nontrivial bandwidth consumption. To address the issue above, TSF-oriented semantic communication becomes meaningful and urgent as this approach holds the potential to alleviate communication pressure and to enable real-time response and high-accuracy analysis for many critical services such as autonomous driving. However, to the best of our knowledge, the TSF-oriented multi-user SC system has received little research attention.

Robust Temporal Feature Extraction Needed by SC: For robust semantic information extraction, it is necessary to consider the primary characteristics of time series data. These data generally exhibit characteristics, like strong periodicity, locality, and temporal redundancy, posing challenges to analysis and forecasting [13], [14]. However, most of the conventional coding techniques such as turbo coding [15] and SC methods like DeepSC-S [7], neglect these intrinsic characteristics. In addition, state-of-the-art TSF algorithms like Informer [16], DLinear [17], and PatchTST [14] lack specific adaptation to wireless environments. Both of the situations above result in unnecessary bandwidth wastage, compromised noise resistance and even transmission failure. To enable effective TSF within SC systems, communication-specific temporal coding for mobile devices becomes highly desirable.

Higher Accuracy Expected for Distributed SC Systems: In contrast to model inference on a central server, analyzing time series data within a distributed system is much harder since it suffers from knowledge scarcity and necessitates cooperation among local models on all mobile devices. Therefore, the capability of collaborative inference of the distributed system plays a crucial role. Most studies on SC [10], [11] do not consider the mutual connections among multivariate time series (MTS) collected by different devices, failing to share common knowledge among them. This limitation hampers the semantic information extraction capability of the whole system, which leads to substantial accuracy degradation.

In this paper, we focus on a forecasting-oriented multi-user semantic communication system designed for time series transmission. The proposed system comprises a number of mobile devices (as transmitters) and an edge server (as the receiver). Mobile devices collect MTS with different distributions. Semantic features are extracted from the MTS according to different forecasting tasks and then transmitted to the edge server over a wireless channel. It is quite a challenge to extract multi-scale temporal features to ensure accurate forecasting because this needs efficient knowledge-sharing among mobile devices. The main contributions of this paper are summarized as follows:

- We design a many-to-one semantic communication system for MTS forecasting. Semantic features are extracted from the MTS generated at each mobile device and transmitted to an edge server over a wireless channel for collaborative inference.
- To guarantee efficient time series transmission, we introduce a novel deep decomposition block for semantic encoders. It extracts temporal features by decomposing input time series into a trend component and multiple period components of different scales. Each component is compressed individually for robust and low-redundancy transmission.
- For efficient training of the semantic and channel encoders, and knowledge sharing among all devices, we propose a federated mixture of period routers (FedMoR) architecture. This architecture effectively compresses features of different scales, maintaining a number of private routers in each device to analyze device-specific features and a number of public routers to share common knowledge among transmitters.
- Simulation results demonstrate that the proposed system exhibits excellent collaborative inference performance on five widely-used real-world time series datasets and achieves significantly lower prediction loss than two traditional and two SC systems under additive white Gaussian noise (AWGN), Rayleigh, and Rician channels.

The rest of the paper is organized as follows. In Section II, we review the recent studies on multi-user semantic communication, federated time series analysis and federated learning algorithms. The system model is formulated in Section III and the proposed forecasting-oriented federated semantic communication system is detailed in Section IV. In Section V, we conduct ablation study and overall performance comparison. Section VI concludes the whole work.

II. Related Work

In this section, we first review distributed semantic communication systems. Then, the recent studies on federated time series analysis are presented. Finally, the well-known federated learning (FL) algorithms are introduced.

A. Distributed Semantic Communication System

Distributed semantic communication systems have attracted increasing more research attention in many areas, e.g., computer vision (CV), natural language processing (NLP), and multi-modal problems.

In the context of CV, Shao *et al.* [18] applied the information bottleneck (IB) principle to eliminate redundant information at edge devices and designed a distributed IB framework for multi-view image classification. Zhang *et al.* [10] highlighted the impact of data-stream interference from other users, significantly deteriorating the system's overall performance. These authors devised a multi-user semantic communication system for cooperative object identification (DeepSC-COI), a convolutional neural network (CNN)-based framework, fusing semantic features from several cameras for cooperative object identification. Based on federated learning, Nan *et al.* [19] developed a unified distributed learning framework of semantic communications (UDSem) to share knowledge among multiple devices and speed up the training process for image classification. In [20], Xie *et al.* presented a FL-based semantic communication (FLSC) framework for multi-task image transmission between IoT devices. A hierarchical vision Transformer model and a task-adaptive translator were embedded into a semantic-aware federated architecture to facilitate multi-scale task-specific semantic information analysis.

In the context of NLP, Hu *et al.* [21] studied a one-to-many SC system for broadcasting scenarios, named MR_DeepSC, where transfer learning was used to speed up the convergence process of new receivers. Xie *et al.* [22] emphasized the constrained resources on numerous IoT devices and proposed a lite distributed SC system for text transmission, called L-DeepSC. Through strategic pruning of unnecessary computation and reducing the weight resolution, L-DeepSC was an affordable solution to various mobile devices. In [23], a device-edge collaborative inference framework was designed for classification tasks. It ensured privacy-preserving collaboration among all devices by incorporating a frozen network at the server side and a learnable network at the device side.

For multi-modal problems, Xie *et al.* [11] proposed DeepSC-VQA, a solution to fusing text and image data from various users for visual question answering (VQA). The Transformer encoders and decoders were adopted at the receiver side. Wang *et al.* [24] introduced a multi-modal distributed semantic communication system with a bidirectional caching model for augmented/virtual reality applications, where the caching process was modeled as a Markov decision process and a content popularity-based DQN was designed to generate promising decisions. Moreover, there have been studies [25]–[28] delved into resource management in distributed SC systems.

Currently, most studies on SC focus on the semantic information extraction and transmission of images and texts. However, efficiently transmitting time series data for downstream tasks is also important. As one of the widely-adopted techniques in data mining, TSF has underlain an increasing number of real-world applications. The rapid development of IoT is generating unprecedented amount of time series data every second, posing great challenges to the transmission and analysis of these data. Transmitting all the data to a data center could bring tremendous pressure to the traditional communication system. SC could greatly relieve such pressure. However, to the best of our knowledge, TSF-oriented SC has received little research attention.

B. Distributed Time Series Analysis

Distributed time series analysis plays a crucial role in industrial and mobile scenarios. For the industrial applications, to address the handover prediction in resource management, Kim *et al.* [29] designed a three-phase clustering algorithm based on FL. They achieved privacy preserving during data clustering and mitigated accuracy reduction associated with non-independent data through cluster-specific model design. In [30], Liu *et al.* proposed a federated anomaly detection model to prevent device failures in industrial IoT networks. They hybridized attention mechanisms, CNNs and long short-term memory (LSTM) networks to capture features of multiple scales. Moreover, they devised a gradient compression mechanism to enhance the overall communication efficiency of the FL system. Zhou *et al.* [31] developed a Siamese neural network based few-shot learning (FSL-SCNN) algorithm to identify cyber-physical attacks in industrial cyber-physical systems. They assumed there were limited labeled data and incorporated few-shot learning together with a hybridized loss function composed of a transforming loss, an encoding loss and a prediction loss in the training stage. Truong *et al.* [32] cascaded an autoencoder block and a novel Transformer-Fourier block to detect potential threats in industrial control systems. This distributed anomaly detection system was based on FL and trained in an unsupervised manner. Wang *et al.* [33] designed an automatic time series feature extraction model for recommender systems. The proposed system realized federated table join and aggregation operations for tabular data without privacy issues compromised.

Besides, there have been numerous applications in mobile environments. Xing *et al.* [12] handled various time series classification tasks via federated distillation, with multiple mobile users considered. The hidden-layer knowledge was transferred from the teacher network at the central server to student networks on mobile devices, where the least square distance was employed to match similar models to facilitate knowledge sharing. Gkillas *et al.* [34] presented a lightweight autoencoder-based time series imputation (TSI) method to generate missing data caused by sensing failure in IoT networks. With resource-constrained IoT devices taken into account, the authors integrated the TSI model into a federated architecture to alleviate the impact of knowledge deficiencies. Zhao *et al.* [35] proposed a federated SC architecture based on dynamic neural networks for time series classification. They used an auxiliary self-supervised task to coordinate features at different transmitters for data redundancy reduction.

Many existing studies assume that time series analysis tasks are executed on just one mobile device. These studies overlook specific challenges within wireless environments, such as strong noise interference and heavy data compression. Besides, they often ignore collaborative inference scenarios. The limitations above make the existing studies unsuitable for practical applications in SC systems. Distributed time series forecasting is a valuable technique applicable across various domains. Motivated by this, we aim to develop a forecasting-oriented multi-user SC system to offer efficient time series transmission and analysis.

C. Federated Learning

FL is a promising solution to addressing privacy concerns in distributed systems and proves highly applicable in real-world applications. FL algorithms can roughly be classified into two categories in terms of the challenges they face: statistical heterogeneity and systems heterogeneity.

In the context of statistical heterogeneity, FederatedAveraging (FedAvg) [36] was one of the earliest attempts to achieve collaborative training in a privacy-sensitive distributed system. It randomly sampled devices to collect a set of local model parameters and aggregated them through weights averaging. Through global model distribution, performance

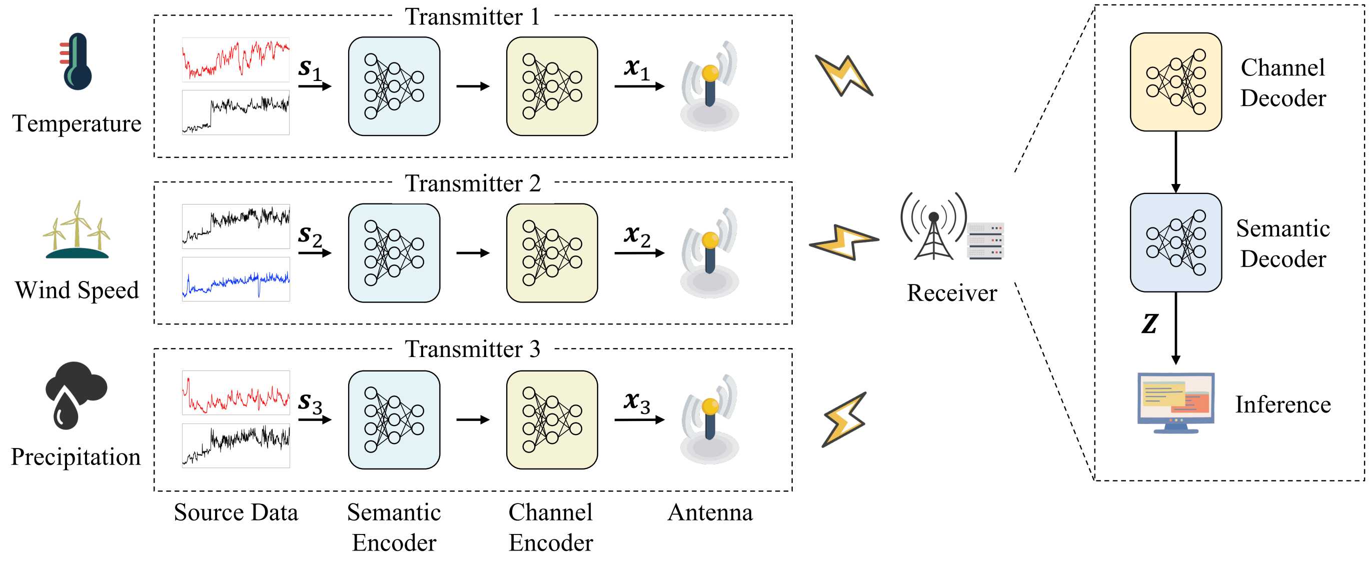


Fig. 2. Scenario of the proposed semantic communication system.

of all local models could be enhanced without risks of data leakage. Stochastic controlled averaging for federated learning (SCAFFOLD) [37] used control variates to rectify the convergence direction of local models, significantly accelerating the convergence. In order to tackle the feature shift problem caused by different distributions among users, Li *et al.* [38] proposed FedBN to freeze the batch normalization layers while updating local models through the global model. FedBN was more stable in convergence, compared with traditional FL algorithms. Targeting at unseen data from new users after the distributed training process, FedSR [39] achieved domain generalization. It adopted L2-norm and conditional mutual information to regularize training models and limit unnecessary knowledge acquisition.

In the context of systems heterogeneity, FedProx [40] considered local models deployed on devices with different computing power. The imbalanced model training could sometimes significantly deteriorate the overall FL performance. Therefore, FedProx added a proximal term to the training loss, with the imbalance among models considered during federated optimization process. Data from different clients may not follow the independent identical distribution, i.e. non-IID. The differences among local models may hamper the global model's convergence. Federated heterogeneous neural networks (FedHeNN) [41] was concerned with heterogeneous device architectures in a federated system. It employed the centered kernel alignment technique to measure the cross-architecture distance and facilitate knowledge transfer among different devices. Ruan *et al.* [42] designed FedSoft, a soft clustered FL algorithm, to improve the performance of locally personalized models and cluster models together. Different from many traditional FL algorithms that have too strong and unrealistic assumptions [43], Zhang *et al.* devised an algorithm design strategy through primal-dual optimization, named FedPD, where the designed algorithms were optimized with low communication complexity.

Existing studies mainly focus on the differences among data distributions or model architectures. However, most of them lack specific design consideration for time series transmission over fading channels and ignore critical characteristics of MTS, such as periodicity and redundancy. Additionally, these algorithms predominantly concentrate on individual tasks at each client rather than collaborative inference. Deploying these algorithms directly to distributed SC systems may result in substantial performance degradation. Therefore, it is necessary to develop a dedicated FL algorithm for TSF-oriented SC systems.

III. System Model and Problem Formulation

As illustrated in Fig. 2, we formulate a many-to-one SC system for MTS collaborative forecasting. This system consists of a number of mobile devices and an edge server. Each mobile device is equipped with a single-antenna transmitter and the server has a multi-antenna receiver. In an arbitrary mobile device, semantic information is extracted from the time series by a semantic encoder and adapted to the associated physical channel by a channel encoder. The signals received by the receiver are processed by a channel decoder and a semantic decoder for collaborative MTS inference.

A. Transmitter

Assume there are N transmitters in the proposed SC system. We denote the MTS data collected from transmitter i by $\boldsymbol{s_i} \in \mathbb{R}^{L_{in} \times N_v}$, $i = 1, 2, ..., N_{tr}$, where $\boldsymbol{s}_i$ consists of N_v dimensions and is of length L_{in}. Let the length and dimension of the semantic features before transmission be l_{ex} and d, respectively. The semantic features to be transmitted, $\boldsymbol{x}_i \in \mathbb{R}^{l_{ex} \times d}$, are defined as

$$\boldsymbol{x}_i = \mathcal{T}^C(\mathcal{T}^S(\boldsymbol{s}_i; \alpha_i); \beta_i), \tag{1}$$

where $\mathcal{T}^S(\cdot; \alpha_i)$ is the semantic encoding model for temporal analysis and $\mathcal{T}^C(\cdot; \beta_i)$ denotes the channel encoding model.

α_i and β_i are the trainable parameters associated with the semantic and channel encoders, respectively.

After joint source-channel encoding, the shape of extracted features is transformed to $\mathbb{C}^{l_{tr}\times 1}$, a complex signal of length l_{tr}. To transmit the complex signal over the wireless channel, the transmission power is constrained as

$$\mathbb{E}\,||\boldsymbol{x}_i||^2 = 1, \tag{2}$$

where $\mathbb{E}$ is mathematical expectation and $||\cdot||$ is used to normalize the signal to be transmitted.

After each round of local training, parameters of all local models are aggregated and updated within the system. Let $(\hat{\alpha}^{t_{gl}}, \hat{\beta}^{t_{gl}})$ and t_{gl} denote the aggregated parameters at the server and the number of model aggregation rounds undergone, respectively. The global-wise parameter aggregation can be expressed as

$$(\hat{\alpha}^{t_{gl}}, \hat{\beta}^{t_{gl}}) = \text{Agg}((\alpha_1^{t_{gl}}, \beta_1^{t_{gl}}), (\alpha_2^{t_{gl}}, \beta_2^{t_{gl}}), ..., (\alpha_{N_{tr}}^{t_{gl}}, \beta_{N_{tr}}^{t_{gl}})), \tag{3}$$

where $\text{Agg}(\cdot)$ denotes the aggregation function for merging parameters from models across all mobile devices. $(\alpha_i^{t_{gl}}, \beta_i^{t_{gl}})$ stands for the local model parameters of transmitter i at round t_{gl}. After model aggregation, the parameters of the global model are sent to each local model for local parameter update, as described below.

$$(\alpha_i^{t_{gl}+1}, \beta_i^{t_{gl}+1}) = \text{Update}((\alpha_i^{t_{gl}}, \beta_i^{t_{gl}}), (\hat{\alpha}^{t_{gl}}, \hat{\beta}^{t_{gl}})), \tag{4}$$

where $\text{Update}(\cdot)$ signifies the local update function that fuses the previous local model and the received global model. A common way is to completely or partially replace the old parameters $(\alpha_i^{t_{gl}}, \beta_i^{t_{gl}})$ with the most recent ones $(\hat{\alpha}^{t_{gl}}, \hat{\beta}^{t_{gl}})$.

B. Channel

After power normalization, the semantic features extracted by all transmitters are injected into a physical channel for transmission. Assume it is a wireless fading channel. Let $\boldsymbol{Y} \in \mathbb{C}^{M\times l_{tr}}$ denote the signal received by the receiver. The fading process under the wireless channel is simulated as

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{X} + \boldsymbol{N}, \tag{5}$$

where $\boldsymbol{X} \in \mathbb{C}^{N_{tr}\times l_{tr}}$ consists of N_{tr} signals containing the semantic information from N_{tr} transmitters. Assume the receiver is equipped with M antennas. $\boldsymbol{N} \in \mathbb{C}^{M\times l_{tr}}$ represents the circular symmetric Gaussian noise following $\mathcal{CN}(0, \sigma^2)$ where covariance σ depends on the signal-to-noise ratio (SNR). Matrix $\boldsymbol{H} \in \mathbb{R}^{M\times N_{tr}}$ is the channel gain and expressed as $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_M]^{\mathrm{T}}$. For the additive white Gaussian noise (AWGN) channel, $\boldsymbol{H}$ equals to the identity matrix $\boldsymbol{I}$. For the Rayleigh fading channel, $\boldsymbol{H}$ follows $\mathcal{CN}(0, \boldsymbol{I})$ with a mean of 0 and a covariance of $\boldsymbol{I}$. For the Rician fading channel with coefficient k_r, $\boldsymbol{H}$ follows $\mathcal{CN}(\mu, \sigma^2)$ where $\mu = \sqrt{k_r/(k_r+1)}$ and $\sigma = \sqrt{1/(k_r+1)}$.

TABLE I
SUMMARY OF MAIN NOTATIONS

Notation	Definition
	Mathematical Notation
$\mathbb{R}^{m\times n}$	Real matrix of size $m \times n$
$\mathbb{C}^{m\times n}$	Complex matrix of size $m \times n$
$\boldsymbol{N}$	Circular symmetric Gaussian noise
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and covariance σ^2
$\mathcal{CN}(\mu, \sigma^2)$	Circular complex Gaussian distribution
$\boldsymbol{H}^{\mathrm{H}}$	Hermitian matrix of $\boldsymbol{H}$
$\|\boldsymbol{x}\|$	Euclidean normalization
	Notation used in system model
d	Dimension of the extracted features
$\boldsymbol{H}$	Channel gain
k_r	Rician coefficient
l_{ex}	Length of the extracted features
L_{in}	Length of the collected time series
M	Number of antennas at the receiver
N_{tr}	Number of transmitters in the SC system
N_v	Dimension of MTS collected at each transmitter
Q	Length of the target time series
$\mathcal{R}^C(\cdot;\phi)$	Semantic decoder with parameters ϕ
$\mathcal{R}^S(\cdot;\psi)$	Channel decoder with parameters ψ
$\boldsymbol{s_i}$	MTS collected by the i-th devices
t_{gl}	Round of global federated aggregation
t_{lo}	Round of local backward propagation
$\mathcal{T}^S(\cdot;\alpha_i)$	Semantic encoder with parameters α_i
$\mathcal{T}^C(\cdot;\beta_i)$	Channel encoder with parameters β_i
$\boldsymbol{x_i}$	Semantic features extracted by device i
$\boldsymbol{X}$	Complex signals transmitted over the wireless channel
$\hat{\boldsymbol{X}}$	Recovered signal at the receiver according to $\boldsymbol{Y}$
$\boldsymbol{Y}$	Signal received by the receiver
$\boldsymbol{Z}$	Predicted MTS at the receiver
$(\hat{\alpha}^{t_{gl}}, \hat{\beta}^{t_{gl}})$	Parameters of the aggregated global model in FL
$(\alpha_i^{t_{gl}}, \beta_i^{t_{gl}})$	Parameters of the local global model in transmitter i
	Notation used in methodology
L_r	Length of each router
N_r	Number of routers at each transmitter
$\boldsymbol{M}_{i,r}$	Router r in transmitter i
$\hat{\boldsymbol{M}}_r^{t_{gl}}$	Aggregated router r in the server
N_{lo}	Number of samples at each transmitter
$\boldsymbol{p}_{i,j}$	Patch j in transmitter i after the patching process
$\hat{\boldsymbol{p}}_{i,j}$	Patch j in transmitter i after PwLocal attention
$\boldsymbol{p}_{i,j}^G$	Patch j in transmitter i after PwGlobal attention
$\boldsymbol{p}_{i,1:P}^{pe}$	Period-based patch sequence at transmitter i
$\boldsymbol{p}_{i,1:P}^{tr}$	Trend-based patch sequence at transmitter i
$\boldsymbol{p}_{i,k}^{ro}$	Patch k in the patch sequence to be transmitted at transmitter i
$\boldsymbol{W}_i$	Weights generated by the gating mechanism at transmitter i

C. Receiver

At the receiver side, the signals are recovered with known channel state information (CSI) immediately after their arrivals. Let $\hat{\boldsymbol{X}}$ be the recovered signals. The recovery process is expressed as

$$\hat{\boldsymbol{X}} = (\boldsymbol{H}^{\mathrm{H}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{H}}\boldsymbol{Y}, \tag{6}$$

where $\boldsymbol{H}^{\mathrm{H}}$ represents the Hermitian matrix of $\boldsymbol{H}$.

Once recovered, the semantic features go through channel and semantic decoders for downstream tasks. In the proposed forecasting-oriented SC system, the downstream task is to predict MTS after the predefined number of time steps. Denote the predicted MTS by $\boldsymbol{Z} \in \mathbb{R}^{Q\times(N_v\times N_{tr})}$ where Q is the length of the time series. The joint channel-source decoding process can be formulated as

$$\boldsymbol{Z} = \mathcal{R}^S(\mathcal{R}^C(\hat{\boldsymbol{X}};\phi);\psi), \tag{7}$$

where $\mathcal{R}^C(\cdot;\phi)$ and $\mathcal{R}^S(\cdot;\psi)$ represent the channel and semantic decoding models, respectively. $\mathcal{R}^S(\cdot;\psi)$ is for the final inference of the forecasting task. ϕ and ψ are the trainable parameters of the channel and semantic decoders, respectively.

Different from image or text processing, it is difficult to use conventional techniques like CNNs or recurrent neural networks (RNNs) to extract semantic features coming from

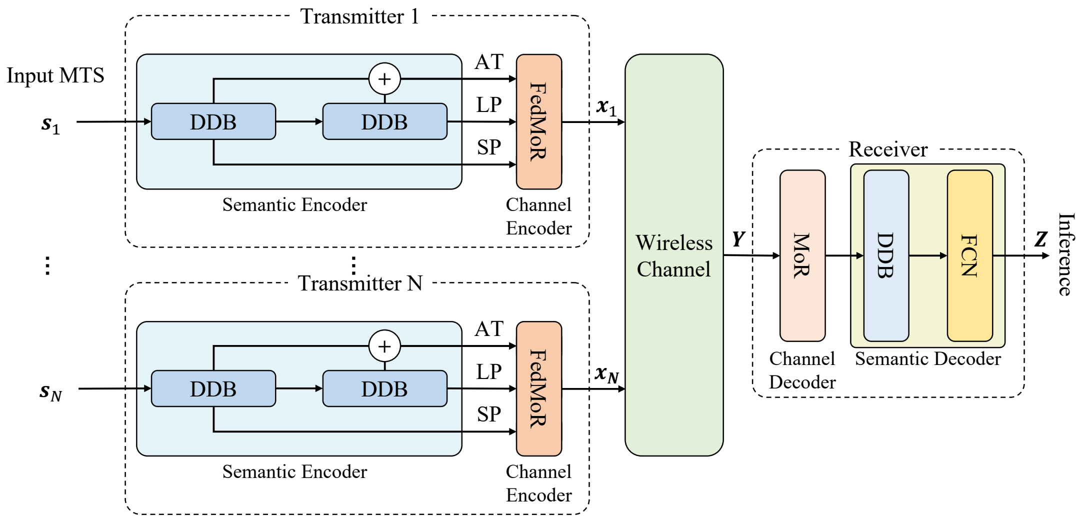


Fig. 3. Architecture of the forecasting-oriented semantic communication system. Note that LP, SP, and AT are abbreviations of the long-period, short-period and accumulated trend features, respectively.

various MTS. Existing TSF models are usually not directly applicable to mobile environments, e.g., compression loss and noise interference pose challenges to efficient data transmission and distributed model training. It is hence worth exploring effective methods for MTS transmission and forecasting in the context of distributed SC.

IV. Proposed TSF-Oriented Federated Semantic Communication System

We devise a forecasting-oriented federated SC system for efficient MTS transmission and analysis as depicted in Fig. 3. Specifically, to offer promising capability for semantic feature extraction, we design deep decomposition blocks for semantic encoding and combine deep decomposition blocks and a fully connected network (FCN) for semantic decoding. To offer high-quality channel coding, we present a mixture of period route (MoR) for robust multi-period transmission under noise interference. To update the MoR-based channel encoders efficiently, we design a FL-based instance-level knowledge sharing architecture called FedMoR.

A. Decomposition-based Semantic Coding

To realize high-quality MTS forecasting in distributed SC environments, transmitters are required to efficiently compress the semantic features extracted and resist noise interference. State-of-the-art TSF methods focus either on pixel-wise fine-grained analysis [16] [17] or on patch-wise coarse-grained prediction. Neither of them is suitable for a distributed system with (sometimes severe) information compression and noise interference since the two factors make a distributed learning model hard to train. Unfortunately, the SC scenario concerned in this paper considers such a system. Most predictable time series normally exhibit evident characteristics, like locality, periodicity, and redundancy [3]. To make full use of these characteristics, we propose a deep decomposition block (DDB) based on patch-wise local and global attention [44] for semantic encoding. By identifying semantic information of different periods, DDB selects the most appropriate periods for subsequent transmission and mitigates cross-scale feature interference, significantly compressing the semantic information to be transmitted. The overview of DDB is illustrated in Fig. 4.

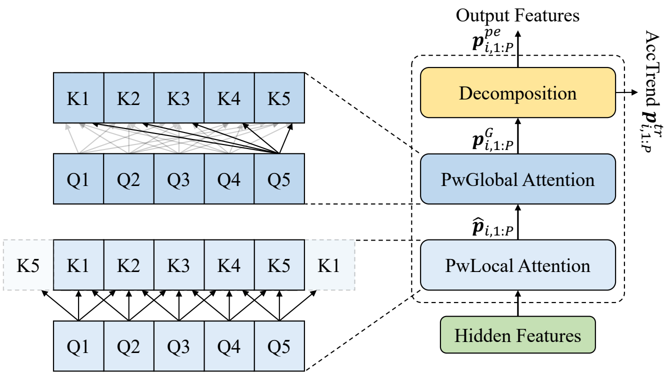


Fig. 4. Pipeline of the deep decomposition block. Note that PwGlobal Attention and PwLocal Attention represent patch-wise global attention and patch-wise local attention, respectively.

1) PwLocal Attention: The quality of local feature extraction helps generate the details of the predicted time series and is one of the key factors that influence the accuracy of forecasting. To fully utilize the local information from MTS, the input time series are firstly transformed into patches for semantic feature extraction. In DDB, patches are the projections of overlapping or non-overlapping segments within the time series and naturally contain local information. To enhance local awareness and explore the relationship among neighboring patches for the small-scale and short-period feature extraction, hidden features are processed by the patch-wise local (PwLocal) attention mechanism first. Let d and $\hat{\boldsymbol{p}}_{i,j} \in \mathbb{R}^d$ be the dimension of each patch and the aggregated neighboring information of the j-th patch in transmitter i, respectively. Let k^{at} be the radius of PwLocal attention. The PwLocal attention of scope $2 \times k^{at} + 1$ is defined as

$$\hat{\boldsymbol{p}}_{i,j} = \mathrm{MSA}(\boldsymbol{p}_{i,j}, \boldsymbol{p}_{i,(j-k^{at}):(j+k^{at})}, \boldsymbol{p}_{i,(j-k^{at}):(j+k^{at})}), \quad (8)$$

where $\mathrm{MSA}(Q, K, V)$ denotes the multi-head self attention mechanism. Q, K, and V stand for matrices of query, key and value, respectively. $\boldsymbol{p}_{i,j} \in \mathbb{R}^d$ represents the j-th patch generated from the MTS collected by transmitter i, and $\boldsymbol{p}_{i,(j-k^{at}):(j+k^{at})} \in \mathbb{R}^{(2\times k^{at}+1)\times d}$ is the group of patches spanning from $j - k^{at}$ to $j + k^{at}$.

2) PwGlobal Attention: To facilitate the decomposition process and realize robust compression, the dependencies among the extracted short-period semantic features are crucial. They decide which features to compress and influence the quality of restored features at the receiver after noise distortion, greatly affecting the performance of the downstream forecasting task. For instance, features with periods of length 11 and 27 are dominant in the target time series. Therefore, they must be restored after transmission and the other features could be discarded according to the compression ratio. Patch-wise global (PwGlobal) attention calculates the pair-wise similarity among all patches and is seen as an efficient tool to achieve global awareness [45]. Let $\boldsymbol{p}^G_{i,j}$ denote the aggregation of global features at position j, and the PwGlobal attention is defined as

$$\boldsymbol{p}^G_{i,j} = \mathrm{MSA}(\hat{\boldsymbol{p}}_{i,j}, \hat{\boldsymbol{p}}_{i,1:P}, \hat{\boldsymbol{p}}_{i,1:P}), \quad (9)$$

where $\hat{\boldsymbol{p}}_{i,1:P}$ is the complete patch sequence after PwLocal attention and P is the number of patches generated from the given time series.

3) Decomposition: Most of the predictable time series can be decomposed into a trend part, several period (or seasonal) parts and random disturbance [13]. It is imperative to identify task-oriented components for compression and leverage the connections among them to mitigate noise interference. For example, in the study on climate change, long period information and the overall trend are task-oriented components while high frequency components are usually regarded as noise caused by the environment. In contrast, when studying traffic flow data, researchers are more likely to discard longer periods, e.g., a month or a year, since the information freshness is quite an important factor. Apart from noise resistance, mining the relationships among trend and period components and distinguishing random disturbances reasonably are both critical for accurate TSF.

To capture promising task-oriented components from the global and local features extracted, we adopt a decomposition method. Denote the period and trend components by $\boldsymbol{p}^{pe}_{i,1:P}$ and $\boldsymbol{p}^{tr}_{i,1:P}$, respectively. The decomposition process is formulated as

$$\boldsymbol{p}^{tr}_{i,1:P} = \mathrm{AvgPooling}(\boldsymbol{p}^G_{i,1:P}), \quad (10)$$

$$\boldsymbol{p}^{pe}_{i,1:P} = \boldsymbol{p}^G_{i,1:P} - \boldsymbol{p}^{tr}_{i,1:P}, \quad (11)$$

where $\boldsymbol{p}^G_{i,1:P}$ represents the sequence of patches in transmitter i output by the PwGlobal attention. $\mathrm{AvgPooling}(\cdot)$ signifies the moving average operation in a given sliding window of size N_{AP}. As shown in Fig. 3, the extracted period components are directly forwarded to the channel encoder for scale-adaptive processing. The trend components output by all deep decomposition blocks are firstly combined as accumulated trend (AT) features and then forwarded to channel encoder.

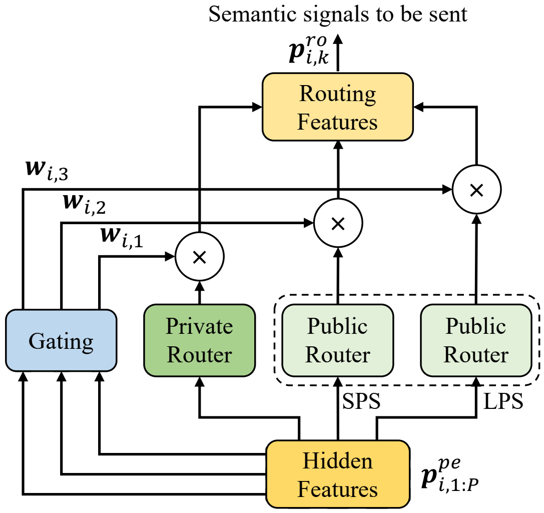


Fig. 5. Structure of the channel encoder based on MoR in transmitter i. Note that LPS and SPS are the abbreviations of long period sensitive and short period sensitive.

B. Channel Coding Based on MoR

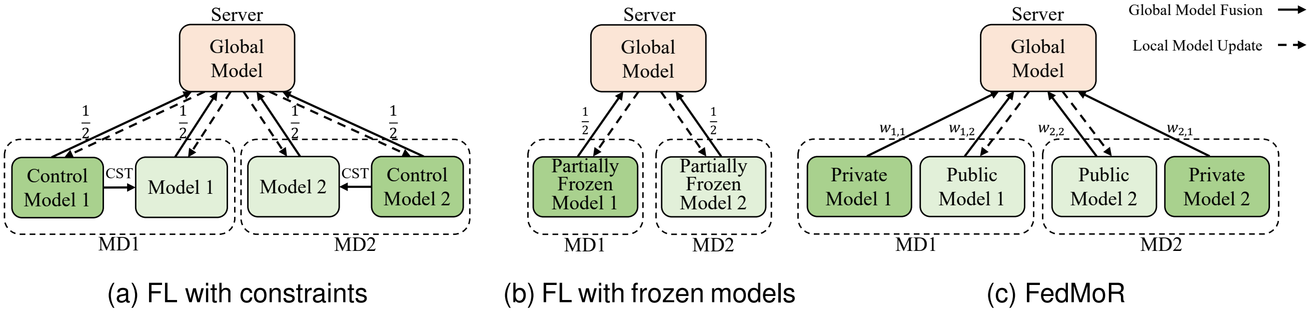
Channel encoder aims at efficiently transmitting the extracted semantic information from the given time series with a certain level of noise resistance. The scales of these extracted features may vary significantly across different scenarios. Most traditional algorithms use a single set of parameters for task-oriented encoding, unable to analyze features with diverse scales. Therefore, a robust coding scheme capable of recognizing the scales of different features to be transmitted is desirable. To this end, inspired by the idea of the mixture of experts [46], we introduce the mixture of routers (MoR), as shown in Fig. 5. We define a series of learnable vectors as experts that are sensitive to temporal features of different scales.

The routers utilize PwGlobal attention to aggregate the global information of the features output by the decomposition process into the routing features. Let $\hat{\boldsymbol{p}}^{pe}_r$ be the routing features of $\boldsymbol{p}^{pe}_{i,1:P}$, $r = 1, 2, ..., N_r$, where N_r stands for the number of routers. The routing process is formulated as

$$\hat{\boldsymbol{p}}^{pe}_r = \mathrm{MHA}(\boldsymbol{M}_{i,r}, \boldsymbol{p}^{pe}_{i,1:P}, \boldsymbol{p}^{pe}_{i,1:P}), \quad (12)$$

where $\boldsymbol{M}_{i,r} \in \mathbb{R}^{L_r \times d}$ is a learnable routing matrix and L_r represents the length of routers indicating the noise resistance level of the signals to be transmitted. In order to adapt to temporal semantic features of various scales, a number of routers are adopted, where a gating mechanism is used to distinguish them in terms of suitability. This gating mechanism generates a weight vector $\boldsymbol{W}_i = [\boldsymbol{w}_{i,1}, \boldsymbol{w}_{i,2}, ..., \boldsymbol{w}_{i,N_r}]^{\mathrm{T}} \in \mathbb{R}^{N_r}$ to mix different routing features output by period routers based on the input features $\boldsymbol{p}^{pe}_{i,1:P}$. The gating process is defined as

$$\boldsymbol{W}_i = \mathrm{SoftMax}(\mathrm{Gating}(\boldsymbol{p}^{pe}_{i,1:P})), \quad (13)$$



(a) FL with constraints (b) FL with frozen models (c) FedMoR

Fig. 6. (a) FL with control models is adopted by FedProx and SCAFFOLD. (b) Partial layers, like batch normalization layers, are frozen in FedBN. (c) FedMoR comprises advantages of (a) and (b). Parameters on the private model participate in the global model fusion according to their contribution to the downstream tasks and are frozen during local model update process. Note that MD and CST are abbreviations of mobile device and constraint.

$$\boldsymbol{p}_{i,k}^{ro} = \sum_{r=1}^{N_r} \boldsymbol{w}_{i,r} \hat{\boldsymbol{p}}_r^{pe}, \tag{14}$$

where Gating$(\cdot)$ is a fully connected network and SoftMax$(\cdot)$ is used to normalize the weights for different routers. $\boldsymbol{p}_{i,k}^{ro}$ denotes the k-th routing features in transmitter i. $\boldsymbol{w}_{i,r}$ indexes the weight of the output of r-th router. The MoR of the channel decoder at the receiver side adopts a similar process.

C. Federated Mixture of Routers Architecture

Existing FL algorithms either adjust convergence directions of local models based on regularization methods or simultaneously optimize local models with similar characteristics based on alignment methods. Neither of the methods easily strikes a balance between the overall performance of the whole system and the individual behavior at each client. Given the same application scenario, time series collected from different mobile devices are likely to have similar fluctuations. For instance, in the meteorological monitoring case, various indicators, such as illumination intensity, air humidity and wind speed, may change simultaneously on a rainy day. However, the same kind of indicators at different locations could differ significantly. Therefore, it is quite meaningful to share knowledge among transmitters and maintain preference of each mobile device so that all local models are aware of the common task-oriented knowledge and achieve effective collaborative inference.

In the context of FL, a common method for model aggregation is to upload all local model parameters to the server and fuse them as depicted in Fig. 6. Fig. 6a presents the update principle of FedProx [40] and SCAFFOLD [37] and Fig. 6b presents the principle of FedBN [38]. Most of these algorithms are based on model-wise fusion. However, different models may have different focuses and characteristics even on resource-sufficient devices. In this case, model-wise fusion could not meet the needs from diverse situations. In this paper, we propose a federated mixture of routers (FedMoR) architecture to decompose the FL process into instance-level sub-processes and separate the FL processes for the semantic encoders and the channel encoders. The noise resistance ability of channel encoding heavily influences the TSF performance. For the channel encoders, the server fuses the model parameters uploaded according to their contributions to the final forecasting accuracy. The entire process of FedMoR is given in Algorithm 1.

Algorithm 1 Federated Mixture of Routers

```
**Input:** input time series, $s_i$, total number of federated optimization rounds, $T_{gl}$, learning rate, $\eta$, parameters on transmitter $i$, $\omega_i$.
**for** transmitter $i = 1, 2, ..., N_{tr}$ **in parallel do**
    Initialize parameters on transmitter $i$, $\omega_i$;
**end for**
**for** round $t_{gl}$ = 1, 2, ..., $T_{gl}$ **do**
    **for** transmitter $i = 1, 2, ..., N_{tr}$ **in parallel do**
        $\omega_i^{t_{gl}+1} \leftarrow \text{LocalUpdate}(i, t_{gl}, \omega_i^{t_{gl}})$;
        Aggregate parameters $M_{i,1:N_r}$ by Eq. (16);
        Aggregate other parameters in $\omega_i^{t_{gl}+1}$ by Eq. (18);
    **end for**
**end for**
```

Algorithm 2 Local Update of Transmitter i

```
**Input:** index of a transmitter, $i$, parameters of transmitter $i$, $\omega_i$, total number of local optimization rounds, $T_{lo}$, local dataset, $\mathcal{B}_i$.
Download parameters of the global model $\hat{\omega}_i$;
**for** router $r = 1, 2, ..., N_r$ **do**
    **if** $\boldsymbol{M}_{i,r}$ **is** public **then**
        Update $\boldsymbol{M}_{i,r}$ by Eq. (17);
    **end if**
**end for**
Replace other model parameters except $\boldsymbol{M}_{i,1:N_r}$ with $\hat{\omega}_i$;
**for** local round $t_{lo} = 1, 2, ..., T_{lo}$ **do**
    Sample mini-batch $B_i \subset \mathcal{B}_i$;
    Extract semantic features from $B_i$;
    Transmit the semantic features extracted to the receiver;
    Implementation backward propagation by Eq. (15);
**end for**
**Output** $\omega_i^{T_{lo}}$;
```

After the initialization, all the local models are updated by the global model as shown in Algorithm 2. After that, each local model randomly samples a batch of time series B_i from its local dataset $\mathcal{B}_i$ and extracts the semantic features from B_i for wireless transmission. The receiver restores the semantic information obtained from all transmitters to execute TSF tasks. In the t_{lo}-th iteration of backward propagation, the local model parameters $\omega_i^{t_{lo}}$ (including α_i and β_i) are updated through

$$\omega_i^{t_{lo}+1} = \omega_i^{t_{lo}} - \eta \nabla \mathrm{Loss}(\omega_i^{t_{lo}}; B) \tag{15}$$

where η is the learning rate. The loss function for the TSF problem is the mean square error (MSE). The backward propagation process starts from the forecasting loss at the receiver. After a number of iterations, all local model parameters are uploaded to the server for FL.

To avoid local performance degradation caused by introduction of knowledge from other transmitters, we divide the local routers into private and public parts as illustrated in Fig. 6c. Both of them upload their parameters for global aggregation to share common knowledge, where the parameters of routers are fused according to their relative contributions quantified by the weights generated by the gating mechanism. Let $\hat{\boldsymbol{M}}_r^{t_{gl}}$ be the aggregated routing matrix at the t^{gl}-th round of FL optimization. The aggregating process is formulated as:

$$\hat{\boldsymbol{M}}_r^{t_{gl}} = \frac{1}{N_{tr}} \sum_{b=1}^{N_{lo}} \sum_{i=1}^{N_{tr}} \boldsymbol{w}_{i,r} \boldsymbol{M}_{i,r}^{t_{gl}}, \tag{16}$$

where N_{tr} represents the number of transmitters in the SC system. As the semantic signals transmitted from different devices are correlated, routers contribute more to a sample should share more knowledge in the corresponding round of FL. Hence, more attention should be paid on the shared knowledge during global model fusion.

After the global model is received by a mobile device, only the public part of local routers updates their parameters based on the global model in a momentum-based moving average manner according to their contributions to the accuracy of TSF as illustrated in Fig. 5. Let $\boldsymbol{M}_{i,r}^{t_{gl}+1}$ be the r-th local routing matrix at transmitter i already updated by the global routing matrix received. The local model update is formulated as

$$\boldsymbol{M}_{i,r}^{t_{gl}+1} = \frac{1}{N_{lo}} (\boldsymbol{w}_{i,r} \hat{\boldsymbol{M}}_r^{t_{gl}} + (1 - \boldsymbol{w}_{i,r}) \boldsymbol{M}_{i,r}^{t_{gl}}), \tag{17}$$

where N_{lo} represents the number of samples transmitted by each transmitter. $\boldsymbol{w}_{i,r}$ is the weight from the gating mechanism in Eq. (13) and $\boldsymbol{M}_{i,r}^{t_{gl}}$ is the local routing matrix before the t_{gl}-th round of FL optimization.

Let $\hat{\theta}^{t_{lo}+1}$ be the global model parameters except $\hat{\boldsymbol{M}}_r^{t_{gl}}$. We simply update the corresponding local parameters $\theta_i^{t_{lo}}$ through the FedAvg algorithm as shown in

$$\hat{\theta}^{t_{lo}+1} = \sum_{i=1}^{N_{tr}} \frac{N_{lo}}{N_{to}} \theta_i^{t_{lo}}, \tag{18}$$

where N_{to} denotes the total number of samples from all transmitters.

The MoR block in the channel decoder does not participate in the FL process because different downstream tasks usually have different problem-specific requirements.

TABLE II
SUMMARY OF FIVE REAL-WORLD MTS DATASETS AND THEIR SETTINGS IN EXPERIMENTS.

Datasets	Weather	Traffic	Electricity	ILI	Exchange
Dimensions	16	128	64	6	8
Prediction	336	336	336	336	60
Devices	8	32	16	2	4

V. SIMULATION RESULTS

In this section, we first introduce the experimental setup, including the datasets, parameter setting, and implementation details. We then conduct ablation studies to verify the effectiveness of the novel components in FedMoR. Finally, we compare FedMoR with two SC-based and two traditional coding-based TSF systems, against prediction loss with different SNRs under three types of channels.

A. Experimental Setup

1) Datasets: We conduct experiments on five widely-used real-world TSF datasets, namely Weather, Traffic, Electricity, ILI, and Exchange [14]. The ILI dataset documents the proportion of patients with influenza-like illness in the total number of patients weekly. The Exchange dataset records the fluctuations in the exchange rates of eight countries. The Weather dataset comprises indicators recorded every 10 minutes from 21 devices, including temperature, air pressure, wind velocity and more. The Electricity dataset contains hourly records for the electricity consumption from hundreds of users. The Traffic dataset collects hourly data from a number of sensors on a freeway, offering a comprehensive description of the variation on road occupancy rates.

To evaluate the performance of the proposed FedMoR system, we carefully select partial data from each dataset and evenly distribute the dimensions of these data across devices, as detailed in Table II. In order to assess the transmission quality and predictive accuracy, we set the input length of all models and the prediction length both to 60 time steps, on the ILI dataset. We choose 60 to generate enough number of training samples since this dataset is relatively small. We set the input length of all models and the prediction length both to 336 time steps on the other four datasets.

2) Parameter Settings: AWGN, Rayleigh and Rician channels are used to simulate the wireless environments and test all systems for comparison. The SNR ranges from 0 dB to 20 dB. For both Rayleigh and Rician channels, the channel gain matrices $\boldsymbol{H}$ follow a complex Gaussian distribution denoted as $\mathcal{CN}(0, \boldsymbol{I})$. Besides, the Rician factor k_r is set to 2.

In the ablation studies and overall performance evaluation, we set the dimension of hidden features in most neural networks, d, to 8. The number of routers in each transmitter, N_r, is set to 4, where we have 1 private router and 3 public ones. In the patching process, we set the patch size to 16 and the sampling stride to 8. We set the radius of attention scope in PwLocal attention, k^{at}, to 1 and the sliding window size in the decomposition process of DDB, N_{AP}, to 25. Each semantic encoder consists of 2 DDBs and the semantic decoder is composed of 1 DDB and a one-layer FCN.

TABLE III
MSE loss on the ILI dataset under AWGN, Rayleigh, and Rician channels.

Channels	Algorithms	SNR=0	SNR=2	SNR=4	SNR=6	SNR=8	SNR=10	SNR=12	SNR=14	SNR=16	SNR=18	SNR=20	Average
AWGN	FedMoR	**2.403388**	**2.387067**	2.369835	**2.286397**	2.323536	**2.301039**	**2.277403**	**2.250452**	2.298627	**2.265750**	**2.249300**	**2.310254**
	FedAvg	2.530626	2.528200	2.506309	2.439269	2.406851	2.418643	2.440927	2.458740	2.466445	2.447788	2.395803	2.458146
	FedProx	2.434067	2.422343	**2.346978**	2.293580	2.307406	2.268497	2.321927	2.324640	**2.285161**	2.267783	2.294770	2.324287
	SCAFFOLD	2.517177	2.522330	2.537547	2.517222	2.475587	2.505083	2.500256	2.517623	2.494620	2.459456	2.440759	2.498878
	FedBN	2.409585	2.397620	2.367617	2.287005	**2.284196**	2.309604	2.287458	2.275212	2.286950	2.287413	2.251550	2.313110
Rayleigh	FedMoR	2.526801	2.523064	**2.491775**	**2.420286**	**2.504173**	**2.511260**	**2.461657**	**2.475630**	2.514866	2.537995	**2.491890**	**2.496309**
	FedAvg	**2.519940**	2.548145	2.819399	2.514556	2.525603	2.621534	2.543105	2.594178	2.516377	2.550551	2.549181	2.572961
	FedProx	2.701267	2.600499	2.521823	2.548349	2.587877	2.519103	2.696669	2.475951	**2.488650**	2.533028	2.541333	2.564959
	SCAFFOLD	2.608217	2.613726	2.581725	2.624805	2.579424	2.557392	2.566432	2.624310	2.573966	2.582242	2.544687	2.586993
	FedBN	2.607757	**2.494104**	2.534664	2.566944	2.551242	2.619488	2.536697	2.503545	2.587749	**2.530100**	2.544956	2.552477
Rician	FedMoR	**2.657426**	**2.614338**	**2.595560**	**2.601093**	**2.577967**	**2.568704**	2.623723	**2.611620**	**2.565020**	**2.525506**	**2.523084**	**2.587640**
	FedAvg	2.739882	2.728664	2.648258	2.804940	2.699233	2.711329	2.898276	2.673030	2.639194	2.681159	2.657476	2.716495
	FedProx	2.751341	3.421074	3.013827	2.664812	2.733325	2.678901	2.735427	2.702575	2.636701	2.676615	2.661311	2.788719
	SCAFFOLD	2.748898	2.669413	2.663242	2.832956	2.664030	2.588184	2.620201	2.667152	2.584048	2.571969	2.623252	2.657577
	FedBN	2.787830	2.778337	2.657797	2.637839	2.593143	2.637514	**2.590276**	2.635468	2.631398	2.616697	2.580644	2.649722

TABLE IV
MSE Loss on the Weather dataset under AWGN, Rayleigh, and Rician channels.

Channels	Algorithms	SNR=0	SNR=2	SNR=4	SNR=6	SNR=8	SNR=10	SNR=12	SNR=14	SNR=16	SNR=18	SNR=20	Average
AWGN	FedMoR	0.906300	0.903813	**0.900205**	**0.900484**	**0.900884**	0.901298	0.900066	**0.898749**	**0.899370**	**0.898348**	**0.897261**	**0.900616**
	FedAvg	**0.904414**	**0.902486**	0.902429	0.901004	0.900997	**0.900631**	**0.899999**	0.900292	0.900455	0.900343	0.899917	0.901179
	FedProx	0.943040	0.938203	0.936558	0.935365	0.935160	0.934936	0.935179	0.934470	0.934441	0.934327	0.934495	0.936016
	SCAFFOLD	0.929568	0.930630	0.931851	0.932529	0.933268	0.933989	0.934124	0.934473	0.934355	0.934543	0.934540	0.933079
	FedBN	0.960401	0.955215	0.950888	0.948091	0.946971	0.946276	0.945701	0.944927	0.944601	0.944053	0.944288	0.948310
Rayleigh	FedMoR	**0.916807**	0.909121	**0.901971**	**0.893459**	0.911746	0.904895	**0.900405**	**0.896895**	**0.894584**	**0.895062**	**0.897653**	**0.902054**
	FedAvg	0.918555	**0.904084**	0.905267	0.908662	**0.909075**	**0.902674**	0.910164	0.900545	0.900936	0.900142	0.900360	0.905497
	FedProx	0.935070	0.937330	0.929549	0.929455	0.923777	0.922129	0.920875	0.919960	0.920093	0.920148	0.919737	0.925284
	SCAFFOLD	0.923050	0.917451	0.918380	0.914883	0.916532	0.914325	0.914505	0.914453	0.914977	0.914469	0.914387	0.916129
	FedBN	0.974500	0.984213	0.982397	0.980906	0.982038	0.980472	0.970935	0.978040	0.977223	0.977176	0.978071	0.978725
Rician	FedMoR	0.948543	0.933796	**0.912321**	**0.914696**	**0.902319**	**0.902087**	**0.906010**	**0.902498**	**0.900145**	**0.902371**	**0.906272**	**0.911914**
	FedAvg	0.961410	0.949519	0.965826	0.949983	0.949659	0.944825	0.946840	0.945010	0.950569	0.945096	0.944972	0.950337
	FedProx	0.945951	0.933694	0.930991	0.930821	0.928024	0.928559	0.928716	0.928204	0.926405	0.927222	0.926359	0.930450
	SCAFFOLD	**0.934355**	**0.919492**	0.919758	0.922518	0.921210	0.917835	0.920240	0.919634	0.919158	0.919921	0.919636	0.921251
	FedBN	0.998320	1.005272	0.997123	0.998821	0.999760	0.996085	0.998437	0.999506	0.997919	0.996712	0.996900	0.998623

3) Implementation Details: Neural networks are implemented with Pytorch 1.13. Adam optimizer with L2 loss is used to train models in SC and traditional systems. For traditional wireless transmission, we normalize the input time series before source coding to reduce the number of bits to be transmitted, particularly when dealing with time series fluctuated in quite different ranges. All experiments are run on a server with Ubuntu 20.04.2 OS, an Intel Xeon(R) CPU E5-2667 v4 3.2 GHz, 128GB RAM, and four Nvidia Titan V GPUs with 12G of VRAM.

B. Ablation Study

To evaluate the effectiveness of FedMoR, we compare it against four widely-used FL algorithms, i.e., FedAvg [36], FedProx [40], SCAFFOLD [37] and FedBN [38]. We conduct experiments on three datasets, i.e., ILI, Weather, and Electricity, with identical number of epochs and learning rate for training. The five algorithms are listed below.

- **FedAvg**: it aggregates all the model parameters uploaded to the server without preference, which has been proven to converge to a global optimum when samples in the distributed system follow the IID assumption.
- **FedProx**: it defines a proximal term to address the problem of system heterogeneity and punishes local models that fail to converge to a global optimum.
- **SCAFFOLD**: it introduces server and client control variates to avoid the client-drift problem observed in local iterations and corrects the direction of global convergence.
- **FedBN**: it addresses the feature shift problem when clients follow different distributions by fixing parameters at the normalization layers in each client.
- **FedMoR**: the proposed algorithm concerned in this paper.

Table III shows the MSE loss results with five algorithms on the ILI dataset under AWGN, Rayleigh, and Rician channels. Note that the ILI dataset is relatively small, with the number of samples less than those of the other datasets. It is easily seen that FedMoR and FedBN are the best and second best algorithms, with three types of channels considered. Specifically, under the AWGN channel, FedMoR exhibits an average loss of 2.310 and is slightly better than FedBN. Similar advantage of FedMoR over FedBN is observed in cases of Rayleigh and Rician channels. The findings above show that FedMoR achieves excellent performance in sample-insufficient situation. On the other hand, FedAvg leads to poor performance under three channels due to the limited number of samples used.

Table IV shows the MSE loss results with five algorithms on the Weather dataset under three different channels. Clearly, FedMoR outperforms the other four algorithms for comparison. Compared with the ILI dataset, the Weather dataset provides sufficient samples for training, meaning that the

TABLE V
MSE LOSS ON THE ELECTRICITY DATASET UNDER AWGN, RAYLEIGH, AND RICIAN CHANNELS.

Channels	Algorithms	SNR=0	SNR=2	SNR=4	SNR=6	SNR=8	SNR=10	SNR=12	SNR=14	SNR=16	SNR=18	SNR=20	Average
AWGN	FedMoR	**0.920827**	**0.914652**	**0.912941**	**0.912854**	**0.912021**	**0.916209**	**0.915398**	**0.913811**	**0.916194**	**0.915880**	**0.914704**	**0.915044**
	FedAvg	0.955272	0.955377	0.955535	0.963093	0.957101	0.957353	0.958075	0.959442	0.959498	0.960597	0.960752	0.958372
	FedProx	0.942862	0.946085	0.954140	0.954920	0.960062	0.965489	0.968339	0.967412	0.968342	0.968083	0.967429	0.960288
	SCAFFOLD	0.929977	0.928994	0.929700	0.930093	0.932082	0.930941	0.934831	0.934784	0.931259	0.938284	0.932335	0.932116
	FedBN	1.037327	1.033775	1.030836	1.039328	1.034827	1.036201	1.037757	1.032477	1.036287	1.036835	1.038383	1.035821
Rayleigh	FedMoR	0.946592	0.949540	0.928260	0.942725	0.932354	0.932820	0.930642	0.934941	0.924786	0.928934	0.929363	0.934633
	FedAvg	0.963679	0.952700	0.939038	0.935330	0.932545	0.930019	0.932390	0.935106	0.930320	0.926634	0.935356	0.937556
	FedProx	**0.904728**	0.913411	0.895164	0.892778	0.892970	0.892796	0.894794	0.893426	**0.894047**	0.892398	0.891362	0.896170
	SCAFFOLD	0.923370	**0.892964**	**0.894802**	**0.890226**	**0.888000**	**0.889520**	**0.891909**	**0.886459**	0.897055	**0.886363**	**0.888726**	**0.893581**
	FedBN	0.962645	0.959499	0.921566	0.933480	0.924726	0.926157	0.924903	0.921333	0.919447	0.918242	0.919516	0.930138
Rician	FedMoR	**0.920568**	**0.928200**	**0.923757**	**0.924163**	**0.918021**	**0.920106**	**0.920068**	**0.922800**	0.921257	**0.914100**	**0.920342**	**0.921217**
	FedAvg	1.007948	0.940764	0.925650	0.948219	0.935363	0.933994	0.945762	0.946755	**0.907662**	0.918980	0.925203	0.939664
	FedProx	0.973217	0.966170	0.975832	0.974187	0.970298	0.966540	0.965199	0.967634	0.966741	0.971481	0.967849	0.969559
	SCAFFOLD	0.948065	0.939858	0.929927	0.927583	0.935144	0.927421	0.929020	0.925954	0.925823	0.924896	0.922566	0.930569
	FedBN	1.010772	1.011469	1.002445	1.007418	1.004958	1.002925	0.996710	1.000642	1.000159	1.000919	0.999260	1.003425

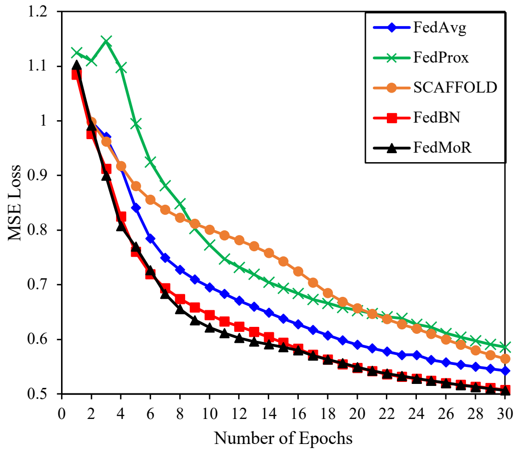


Fig. 7. Convergence curves with five algorithms regarding the MSE loss.

distributions of the training and test sets could be more similar. FedAvg's performance on Weather is substantially improved compared with that on ILI. Under AWGN and Rayleigh channels, FedAvg's MSE loss is very close to FedMoR. Yet, FedAvg cannot achieve promising TSF accuracy under the Rician channel where the channel interference is severe. In such a challenging condition, SCAFFOLD and FedProx obtain relatively good MSE loss results since both of them strategically guide the convergence directions of local models. By preventing local models from drifting far away from the global one, the strategical guidance helps SCAFFOLD and FedProx adapt to severe channel interference.

Table V shows the MSE loss results with five algorithms on the Electricity dataset under AWGN, Rayleigh, and Rician channels. FedMoR and SCAFFOLD rank the first and second under AWGN and Rician. Under the Rayleigh channel, both SCAFFOLD and FedProx exhibit promising performance as they well manipulate control variates and training loss to drive the update of local models towards global optima. Notably, the performance of FedBN results in relatively weak MSE loss values under three channels, mainly because some local models deviate severely from the global one. This misleads FedBN to a local optimum.

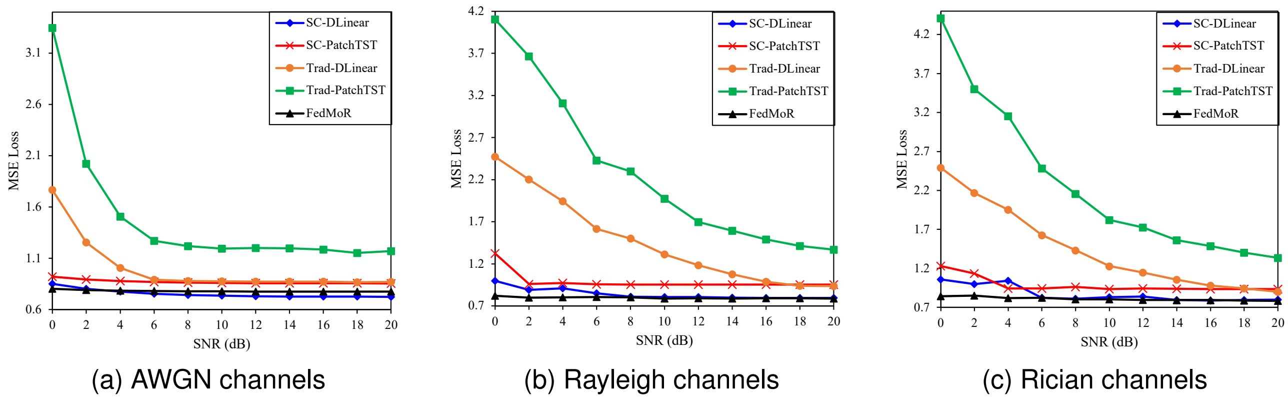
In addition, we compare the five algorithms in terms of convergence speed. We use the Traffic dataset for performance comparison as we set the number of mobile devices to 32, much larger than that on any other dataset. Fig. 7 shows the results of the convergence test conducted on the Traffic dataset under AWGN channel where $SNR = 10$ dB. Obviously, FedMoR is featured with the fastest convergence speed within the predefined number of epochs and it results in the lowest MSE loss in the last epoch. FedBN is a close competitor to FedMoR. FedBN is based on FedAvg and they differ only in the update of batch normalization layers. FedAvg ranks the third, which reaffirms its training effectiveness in sample-sufficient situations. FedProx and SCAFFOLD are based on similar concepts, i.e. constraining the differences between parameters of local and global models. The experimental results reveal that FedProx and SCAFFOLD are not robust enough, with all the epochs considered.

In summary, FedMoR outperforms four state-of-the-art FL algorithms with respect to MSE loss in almost all experiments. This reflects, to a certain extent, FedMoR achieves excellent performance in training and validates its effectiveness for collaborative inference in the distributed SC system, particularly in severe training conditions.

C. Overall Performance Evaluation

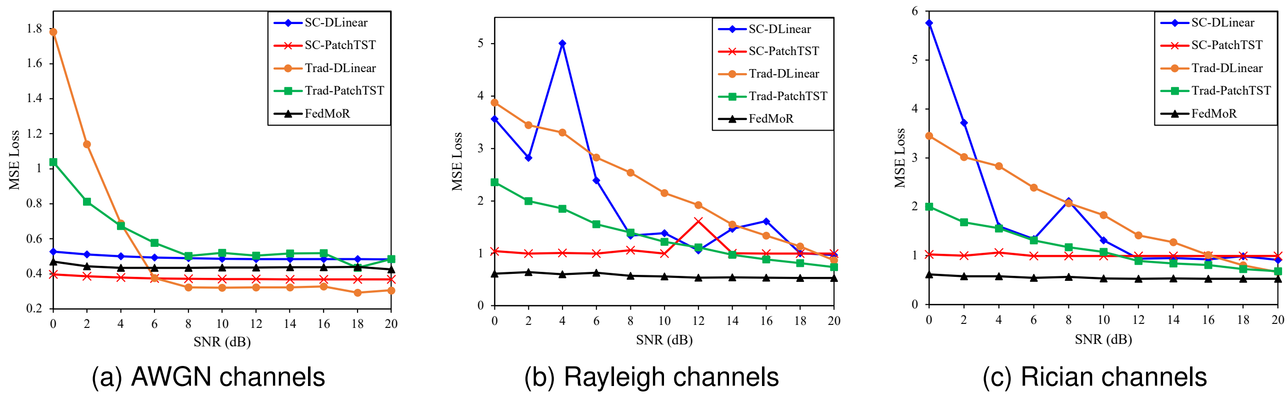
As the TSF-oriented SC problem concerned in this paper is brand-new and no existing algorithm is immediately available. In order to thoroughly evaluate the overall performance of the proposed SC-FedMoR, we compare it with two SC-based and two traditional technique-based baselines. The five systems are listed below.

- **SC-DLinear**: a TSF-oriented SC system based on an advanced pixel-wise TSF algorithm, DLinear [17], for semantic feature extraction with CNNs for noise resistance in channel coding.
- **SC-PatchTST**: a TSF-oriented SC system based on the latest patch-wise TSF algorithm, PatchTST [14], for semantic feature extraction with feed forward networks for noise resistance in channel coding.
- **Trad-DLinear**: a traditional distributed TSF system employing Turbo for channel coding and 64-QAM for mod-



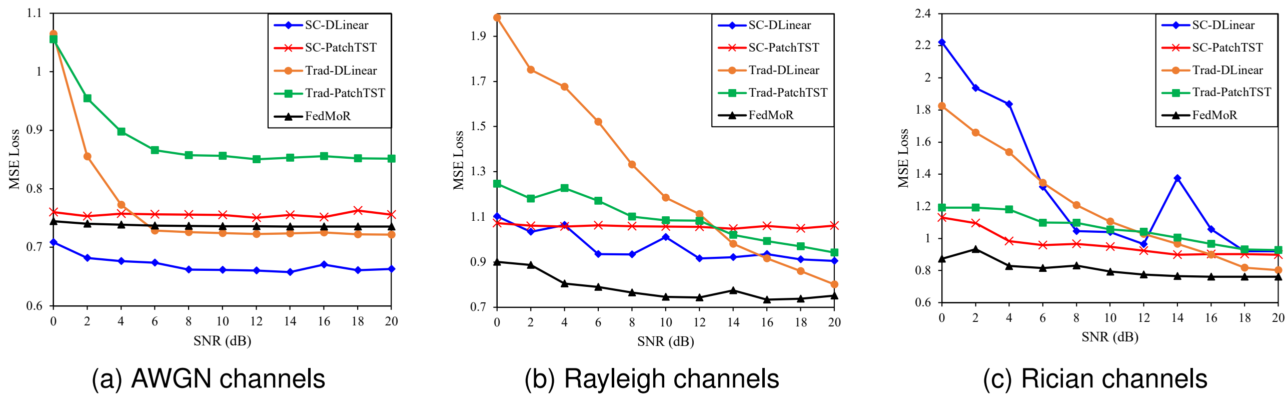
(a) AWGN channels (b) Rayleigh channels (c) Rician channels

Fig. 8. MSE loss with five systems on the Weather dataset tested under different channels.



(a) AWGN channels (b) Rayleigh channels (c) Rician channels

Fig. 9. MSE loss with five systems on the Traffic dataset tested under different channels.



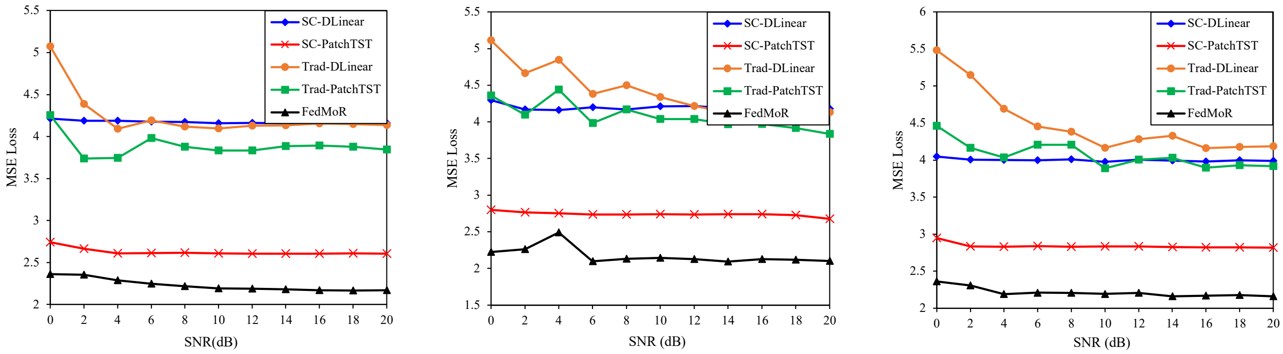
(a) AWGN channels (b) Rayleigh channels (c) Rician channels

Fig. 10. MSE loss with five systems on the Electricity dataset tested under different channels.

ulation in the transmission process and adopting DLinear for prediction at the server side.

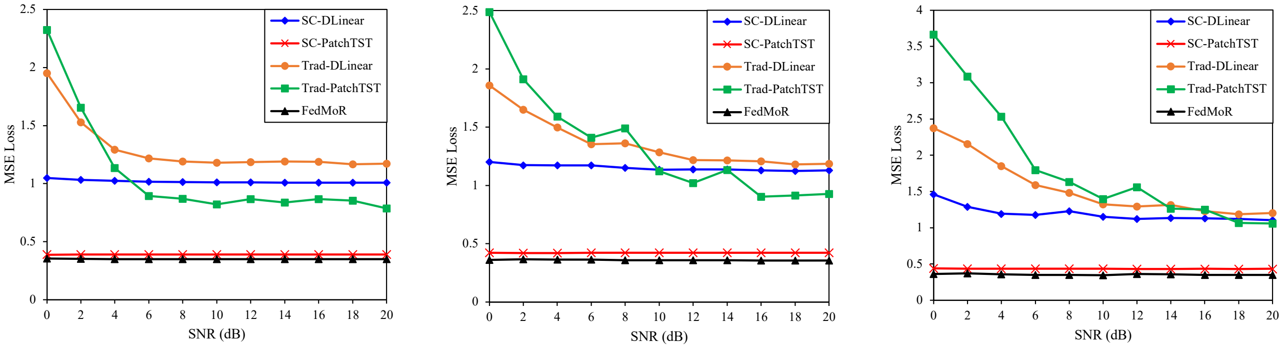
- **Trad-PatchTST**: a traditional distributed TSF system employing Turbo for channel coding and 64-QAM for modulation in the transmission process and adopting PatchTST for prediction at the server side.
- **SC-FedMoR**: the proposed many-to-one TSF-oriented SC system concerned in this paper.

Fig. 8 depicts the MSE loss results with five systems on the Weather dataset. Across all scenarios except those under AWGN channels, SC-FedMoR always achieves the best performance thanks to the excellent local and global semantic feature aggregation and the effective decomposition process in the DDB. As the data of this dataset contain severe outliers, traditional channel coding algorithms cannot handle this situation properly. That is why Trad-DLinear and Trad-PatchTST lead to poor performance in low SNR areas no matter which channel is considered. Outliers exert a substantial influence on the patch-wise similarity comparison in the attention process of SC-PatchTST. But in pixel-based systems, the outliers have limited influence on the predicted MTS. Therefore, pixel-based SC systems are more robust in resistance to the outliers than patch-based SC systems. That is why SC-DLinear outperforms SC-PatchTST in most cases regarding the prediction loss.



(a) AWGN channels (b) Rayleigh channels (c) Rician channels

Fig. 11. MSE loss with five systems on the ILI dataset tested under different channels.



(a) AWGN channels (b) Rayleigh channels (c) Rician channels

Fig. 12. MSE loss with five systems on the Exchange dataset tested under different channels.

Fig. 9 shows the MSE loss results on the Traffic dataset. Trad-DLinear is susceptible to noise under AWGN channels, particularly in low SNR areas ranging from 0 dB to 6 dB. Nevertheless, it outperforms the other four systems in the rest areas. SC-PatchTST and SC-FedMoR are the second and third best systems, respectively. In Rayleigh and Rician channels, the advantages of SC-based systems become more evident. SC-FedMoR always obtains the lowest MSE loss values. SC-PatchTST performs better than SC-DLinear, Trad-DLinear, and Trad-PatchTST in low SNR areas. On the other hand, Trad-DLinear and Trad-PatchTST, gain acceptable performance when SNR is higher than 14 dB.

Fig. 10 shows the MSE loss results on the Electricity dataset. Across all scenarios except those under AWGN channels, SC-FedMoR always achieves the best performance among the five algorithms for comparison. Under the AWGN channels, SC-DLinear outperforms SC-PatchTST while Trad-DLinear surpasses Trad-PatchTST in terms of MSE loss, reflecting that pixel-based schemes are more effective in extracting semantic features from input MTS than patch-based ones. However, pixel-based systems cannot deal with severe noise interference according to its performance under Rician channels. For example, SC-DLinear and Trad-DLinear are worse than SC-FedMoR under Rayleigh channels, especially in low SNR areas. The decomposition process and patch-based calculation help SC-FedMoR resist noise interference. Additionally, SC-FedMoR remains robust even in severe SNR situations under Rician channels, indicating its excellent noise resistance potential.

Fig. 11 shows the MSE loss results on the ILI dataset. Under three channels, SC-FedMoR and SC-PatchTST are the best and second best among the systems for comparison. The number of samples associated with the ILI dataset is relatively small. The pixel-based algorithms could not fully understand the semantic information in sample-insufficient situations and might overfit to some task-irrelevant features, making the received semantic information easily distorted and hard to recover. Therefore, SC-DLinear and Trad-DLinear are beaten by SC-FedMoR and SC-PatchTST.

Fig. 12 shows the MSE loss results on the Exchange dataset. Evidently, SC-FedMoR always achieves the best MSE results under three channels and SC-PatchTST ranks the second. In the high SNR areas of AWGN and Rayleigh channels, Trad-PatchTST outperforms SC-DLinear and Trad-DLinear. The above observations, to a certain extent, reflect the superiority of patch-based systems over pixel-based systems. On the other hand, SC-DLinear and Trad-DLinear are losers, due to the inherent noise in the data and the corresponding distortion amplified during wireless communication. This, to some extent, unveils the weakness of pixel-based algorithms under severe noise interference.

VI. Conclusion

This paper presents a novel time-series-forecasting-oriented federated semantic communication (SC) system for collaborative inference. The proposed deep decomposition block can identify the importance of trend and periods of various scales, offering strong resistance to noise. The proposed federated mixture of routers (FedMoR) architecture is able to share forecasting-oriented knowledge for collaborative inference, which significantly improves the convergence speed and forecasting accuracy of the whole SC system. Experimental results demonstrate the superior performance of FedMoR over four well-known federated learning algorithms in terms of MSE loss. In addition, the proposed federated SC system is more robust in forecasting-oriented time series transmission compared with four advanced baselines, particularly in severe wireless environments with low signal-to-noise ratio.

References

[1] R. Karmakar, G. Kaddoum, and O. Akhrif, "A novel federated learning-based smart power and 3D trajectory control for fairness optimization in secure UAV-assisted MEC services," *IEEE Transactions on Mobile Computing*, pp. 1–17, 2023.

[2] F. Song, H. Xing, X. Wang, S. Luo, P. Dai, Z. Xiao, and B. Zhao, "Evolutionary multi-objective reinforcement learning based trajectory control and task offloading in UAV-assisted mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 22, no. 12, pp. 7387–7405, 2023.

[3] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2D-variation modeling for general time series analysis," in *Proceedings of the International Conference on Learning Representations*, 2022.

[4] Q. Yang, M. Mohammadi Amiri, and D. Gündüz, "Audience-retention-rate-aware caching and coded video delivery with asynchronous demands," *IEEE Transactions on Communications*, vol. 67, no. 10, pp. 7088–7102, 2019.

[5] Z. Yang, M. Chen, W. Saad, W. Xu, and M. Shikh-Bahaei, "Sum-rate maximization of uplink rate splitting multiple access (rsma) communication," *IEEE Transactions on Mobile Computing*, vol. 21, no. 7, pp. 2596–2609, 2022.

[6] D. Gündüz, Z. Qin, I. E. Aguerri, H. S. Dhillon, Z. Yang, A. Yener, K. K. Wong, and C.-B. Chae, "Beyond transmitting bits: Context, semantics, and task-oriented communications," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 5–41, 2022.

[7] Z. Weng and Z. Qin, "Semantic communication systems for speech transmission," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2434–2444, 2021.

[8] K. Lu, R. Li, X. Chen, Z. Zhao, and H. Zhang, "Reinforcement learning-powered semantic communication via semantic similarity," 2022.

[9] G. Zhang, Q. Hu, Z. Qin, Y. Cai, and G. Yu, "A unified multi-task semantic communication system with domain adaptation," in *Proceedings of the IEEE Global Communications Conference*. IEEE, 2022, pp. 3971–3976.

[10] Y. Zhang, W. Xu, H. Gao, and F. Wang, "Multi-user semantic communications for cooperative object identification," in *Proceedings of the IEEE International Conference on Communications Workshops*. Seoul, Korea, Republic of: IEEE, 2022, pp. 157–162.

[11] H. Xie, Z. Qin, X. Tao, and K. B. Letaief, "Task-oriented multi-user semantic communications," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 9, pp. 2584–2597, 2022.

[12] H. Xing, Z. Xiao, R. Qu, Z. Zhu, and B. Zhao, "An efficient federated distillation learning system for multitask time series classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022.

[13] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in Neural Information Processing Systems*, vol. 34, pp. 22 419–22 430, 2021.

[14] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *Proceedings of the International Conference on Learning Representations*, 2023.

[15] L. Hanzo, T. H. Liew, and B. L. Yeap, *Turbo coding, turbo equalisation and space-time coding*. John Wiley & Sons, 2002.

[16] 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*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[17] 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*, vol. 37, no. 9, 2023, pp. 11 121–11 128.

[18] J. Shao, Y. Mao, and J. Zhang, "Task-oriented communication for multi-device cooperative edge inference," *IEEE Transactions on Wireless Communications*, pp. 73–87, 2022.

[19] G. Nan, X. Liu, X. Lyu, Q. Cui, X. Xu, and P. Zhang, "Udsem: A unified distributed learning framework for semantic communications over wireless networks," *IEEE Network*, 2023.

[20] B. Xie, Y. Wu, Y. Shi, D. W. K. Ng, and W. Zhang, "Communication-efficient framework for distributed image semantic wireless transmission," *IEEE Internet of Things Journal*, 2023.

[21] H. Hu, X. Zhu, F. Zhou, W. Wu, R. Q. Hu, and H. Zhu, "One-to-many semantic communication systems: Design, implementation, performance evaluation," *IEEE Communications Letters*, pp. 2959–2963, 2022.

[22] H. Xie and Z. Qin, "A lite distributed semantic communication system for internet of things," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 142–153, 2021.

[23] J. Li, G. Liao, L. Chen, and X. Chen, "Roulette: A semantic privacy-preserving device-edge collaborative inference framework for deep learning classification tasks," *IEEE Transactions on Mobile Computing*, 2023.

[24] C. Wang, X. Yu, L. Xu, Z. Wang, and W. Wang, "Multimodal semantic communication accelerated bidirectional caching for 6G MEC," *Future Generation Computer Systems*, vol. 140, pp. 225–237, 2023.

[25] W. Zhang, Y. Wang, M. Chen, T. Luo, and D. Niyato, "Optimization of image transmission in a cooperative semantic communication networks," *IEEE Transactions on Wireless Communications*, 2023.

[26] Y. Wang, M. Chen, T. Luo, W. Saad, D. Niyato, H. V. Poor, and S. Cui, "Performance optimization for semantic communications: An attention-based reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 9, pp. 2598–2613, 2022.

[27] L. Xia, Y. Sun, X. Li, G. Feng, and M. A. Imran, "Wireless resource management in intelligent semantic communication networks," in *Proceedings of the IEEE Conference on Computer Communications Workshops*, 2022.

[28] Z. Yang, M. Chen, Z. Zhang, and C. Huang, "Energy efficient semantic communication over wireless networks with rate splitting," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 5, pp. 1484–1495, 2023.

[29] Y. Kim, E. A. Hakim, J. Haraldson, H. Eriksson, J. M. B. da Silva, and C. Fischione, "Dynamic clustering in federated learning," in *Proceedings of the IEEE International Conference on Communications*, 2021.

[30] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, and M. S. Hossain, "Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6348–6358, 2021.

[31] X. Zhou, W. Liang, S. Shimizu, J. Ma, and Q. Jin, "Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5790–5798, 2021.

[32] H. T. Truong, B. P. Ta, Q. A. Le, D. M. Nguyen, C. T. Le, H. X. Nguyen, H. T. Do, H. T. Nguyen, and K. P. Tran, "Light-weight federated learning-based anomaly detection for time-series data in industrial control systems," *Computers in Industry*, vol. 140, p. 103692, 2022.

[33] S. Wang, J. Li, M. Lu, Z. Zheng, Y. Chen, and B. He, "A system for time series feature extraction in federated learning," in *Proceedings of the ACM International Conference on Information & Knowledge Management*, ser. CIKM '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 5024–5028.

[34] A. Gkillas and A. S. Lalos, "Missing data imputation for multivariate time series in industrial IoT: A federated learning approach," in *Proceedings of the International Conference on Industrial Informatics*, 2022, pp. 87–94.

[35] B. Zhao, H. Xing, X. Wang, Z. Xiao, and L. Xu, "Classification-oriented distributed semantic communication for multivariate time series," *IEEE Signal Processing Letters*, vol. 30, pp. 369–373, 2023.

[36] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. PMLR, 2017, pp. 1273–1282.

[37] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proceedings of the International conference on machine learning*. PMLR, 2020, pp. 5132–5143.

[38] X. Li, M. JIANG, X. Zhang, M. Kamp, and Q. Dou, "FedBN: Federated learning on non-IID features via local batch normalization," in *Proceedings of the International Conference on Learning Representations*, 2021.

[39] A. T. Nguyen, P. Torr, and S. N. Lim, "Fedsr: A simple and effective domain generalization method for federated learning," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 38 831–38 843.

[40] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems*, I. Dhillon, D. Papailiopoulos, and V. Sze, Eds., vol. 2, 2020, pp. 429–450.

[41] D. Makhija, X. Han, N. Ho, and J. Ghosh, "Architecture agnostic federated learning for neural networks," in *Proceedings of International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 2022, pp. 14 860–14 870.

[42] Y. Ruan and C. Joe-Wong, "Fedsoft: Soft clustered federated learning with proximal local updating," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, pp. 8124–8131, 2022.

[43] X. Zhang, M. Hong, S. Dhople, W. Yin, and Y. Liu, "FedPD: A federated learning framework with adaptivity to Non-IID data," *IEEE Transactions on Signal Processing*, vol. 69, pp. 6055–6070, 2021.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

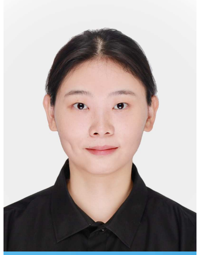
[45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations*, 2021.

[46] F. Xue, Z. Shi, F. Wei, Y. Lou, Y. Liu, and Y. You, "Go wider instead of deeper," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, pp. 8779–8787, 2022.

Bowen Zhao received his B. Eng. degree in Computer Science and Technology from Southwest Jiaotong University, Chengdu, China, in 2020. He is currently pursuing the Ph.D. degree in the School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, China. His research interests include deep reinforcement learning, time series classification and mobile edge computing.

Huanlai Xing (Member, IEEE) received Ph.D. degree in computer science from University of Nottingham, Nottingham, U.K., in 2013. He was a Visiting Scholar in Computer Science, The University of Rhode Island, USA.Supervisor: Dr.Haibo He (Robert Haas EndowedChair Professor, IEEE Fellow, https://www.ele.uri.edu/faculty/he/) in 2020-2021. Huanlai Xing is an Associate Professor and PhD Supervisor with the School of Computing and Artificial Intelligence, Southwest Jiaotong University. He is on Editorial Board (Young Scientists Committee) of SCIENCE CHINA INFORMATIONSCIENCES. He was a member of several international conference program and senior program committees, such as ECML-PKDD, MobiMedia, ISCIT, ICCC, IJCNN, TrustCom, and ICSINC. His research interests include semantic communication, representation learning, data mining, reinforcement learning, machine learning, network function virtualization, and software defined networking.

Lexi Xu received PhD degree from Queen Mary University of London, London, United Kingdom in 2013. He is now a senior engineer at Research Institute, China United Network Communications Corporation (China Unicom). He is also a China Unicom delegate in ITU, ETSI, 3GPP, CCSA. His research interests include big data, self-organizing networks, satellite system, radio resource management in wireless system, etc.



Yang Li received the PhD degree in Faculty of Science and Technology from the University of Macau in 2023. She is currently an Assistant Professor with the School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, China. Her research interests include the edge intelligence, federated learning, and semantic communication.

Li Feng received his PhD degree from Xi'an Jiaotong University under the supervision of Prof. Xiaohong Guan (Academian of CAS, IEEE Fellow). He is a Research Professor and PhD supervisor with the School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu. His research interests include artificial intelligence, cyber security and its applications.

Jincheng Peng received his M.S. degree in Computer Science and Technology from Guizhou University, Guiyang, China, in 2022. He is currently pursuing the Ph.D. degree in the School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, China. His research interests include software-defined networking, deep reinforcement learning and semantic communication.

Zhiwen Xiao received his B. Eng. degree in network engineering from Chengdu University of Information Technology in 2019. He is currently studying at Southwest Jiaotong University, Chengdu, China. He is going to pursue a Ph. D. degree in computer science. His research interests are deep learning, representation learning, data mining, and computer vision.