

Fine-Grained Bandwidth Estimation for Smart Grid Communication Network

Jingtang Luo¹, Jingru Liao², Chenlin Zhang³, Ziqi Wang⁴, Yuhang Zhang², Jie Xu^{2,*} and Zhengwen Huang⁵

¹State Grid Sichuan Economic Research Institute, Chengdu, 610041, China

²School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China

³Science and Technology on Security Communication Laboratory, Institute of Southwestern Communication, Chengdu, 610093, China

⁴State Grid Economic and Technological Research Institute CO., LTD, Beijing, 100055, China

⁵Department of Electronic and Computer Engineering, Brunel University, Uxbridge, UB8 3PH, United Kingdom

*Corresponding Author: Jie Xu. Email: xuj@uestc.edu.cn

Received: 19 August 2021; Accepted: 11 October 2021

Abstract: Accurate estimation of communication bandwidth is critical for the sensing and controlling applications of smart grid. Different from public network, the bandwidth requirements of smart grid communication network must be accurately estimated in prior to the deployment of applications or even the building of communication network. However, existing methods for smart grid usually model communication nodes in coarse-grained ways, so their estimations become inaccurate in scenarios where the same type of nodes have very different bandwidth requirements. To solve this issue, we propose a fine-grained estimation method based on multivariate nonlinear fitting. Firstly, we use linear fitting to calculate the convergence weights of each node. Then, we use correlation to select the important characteristics. Finally, we use multivariate nonlinear fitting to learn the nonlinear relationship between characteristics and convergence weight, and complete the fine-grained bandwidth estimation. Our method exploits multiple node characteristics to reveal how different nodes affect bandwidth requirements differently, and it can learn multivariate estimation parameters from present network without human interference. We use NS2 to simulate a real-world regional smart grid. Simulation shows that our method outperforms existing works by up to 56.5% higher estimation accuracy.

Keywords: Bandwidth estimation; fine-grained; multivariate nonlinear fitting; smart grid communication network

1 Introduction

Smart grid empowers modern society by creating the foundation necessary for electric transportation, energy efficiency, emissions reductions, and new energy technologies. Private communication networks are widely used by smart grid to deliver massive sensing and controlling data for critical applications like power metering, environment monitoring, and power dispatching. Different from the applications in



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

public networks (e.g., social media), the applications in smart grid usually have very stringent communication QoS (Quality of Service) requirements. For instance, dispatching application demands that transmission delay must be lower than 100 ms and transmission error rate must be lower than 10^{-8} . As a result, to meet these applications' QoS demands, the bandwidth requirements of each communication node must be accurately estimated in prior to the deployment of applications or even the building of communication network.

The most reasonable idea for bandwidth estimation is using present networks' bandwidth consumption information to estimate new networks' bandwidth requirements. Based on this idea, [1] and [2] propose elastic coefficient method, which has been widely used in practice due its ease of use. However, because elastic coefficient method assumes that all data are uploaded to a few core nodes (namely, the dispatching centers of smart grid), it often results in significant overestimation of bandwidth demands.

To solve the above problem, some works like [3–7] exploit importance recognition methods to reveal the influences of different nodes on bandwidth estimation. However, they mainly identify important nodes based on the physical topology of the network, such as node centrality, K-shell, structure hole, PageRank. But to accurately analyze node importance, the applications on each node should also be considered explicitly.

There are some other ways to improve bandwidth estimation. For instance, [8] proposes a new method of optimizing bandwidth calculation. In [9], the bandwidth of each application is estimated and accumulated to obtain the bandwidth of a single node. Although these works increase bandwidth estimation accuracy to some extent, they still have some shortcomings such as ignoring the characteristics of different nodes and relying on human experience for parameter selection.

In this paper, we propose a novel fine-grained bandwidth estimation method for smart grid. Compared to present works, our method achieves up to 56.5% higher estimation accuracy. Such performance is mainly due to the following two novelties:

- 1) Our method divides the characteristics and studying the influence of different characteristics of each node. Our method explicitly reveals how data converge from outer nodes to core nodes in smart grid, and how such convergence is affected by each node's characteristics (e.g., number and type of applications). As a result, our method can provide fine-grained bandwidth estimation for different nodes;
- 2) The parameter setting in our method requires no human interference. The parameters are learned through multiple iterations. Our method exploits multivariate nonlinear fitting to learn parameter settings from present network. Since our method can learn multiple node characteristics as well as the nonlinear relationships among these characteristics without human interference, it achieves higher estimation accuracy especially in heterogeneous networks where nodes have highly diverse bandwidth requirements.

The rest of this paper is organized as follows. Section 2 introduces the existing researches related to our work. Section 3 introduces two important features of smart grid communication network. Section 4 proposes a fine-grained bandwidth estimation method. Section 5 proposes a multivariate nonlinear fitting scheme to learn estimation parameters. Section 6 exploits simulations to validate the accuracy of our estimation method. Section 7 concludes this paper.

2 Related Works

The most popular methods estimate bandwidth according to node's voltage level [1–2,10,11]. In general, these methods assume that the grid nodes with the same voltage level (e.g., 220 kV substations) have very close bandwidth demands for uploading data to their upper nodes. While these methods have very simple estimation process, they often make significant overestimation of bandwidth demands especially for the core nodes like dispatching centers, due to the fact that nodes with the same voltage level probably have very different bandwidth requirements.

Recognizing the importance of different nodes based on network topology [12] is an effective way to improve bandwidth estimation. Reference [13] introduces several types of node centralities, such as degree centrality, close centrality, intermediate centrality, and eigenvector centrality. Furthermore, several centrality indicators may be used together to comprehensively analyze the importance of a node. In [14], a K-shell algorithm is used to calculate the influence of nodes in the network. In [15], an E-Burt algorithm based on structural holes is proposed, which sets the weight of the edge as the edge connection. Reference [16] uses PageRank algorithm to obtain the node weight to replace the node degree matrix in the centrality, and determines the importance of nodes in the network through the improved centrality. Reference [4] improves the traditional calculation method by evaluating the importance of power communication network nodes based on node strength and node tightness.

Some works analyze how different applications affect bandwidth requirements. Reference [8] improves estimation accuracy in tree-structured networks through selecting concurrent proportions for different applications. In reference [9], the bandwidth of each service is estimated and accumulated to obtain the bandwidth of each node. This work uses no machine learning technologies, and it focuses on estimating bandwidth of single node rather than whole network. Reference [17] proposes a passive capacity and available bandwidth measurement method for the data plane, employing packet dispersion and autocorrelation.

However, as far as we know, the existing works only consider one or two node characteristics (e.g., voltage level, topology, applications, etc.), which makes their estimations coarse-grained and thereby inaccurate especially in heterogeneous networks with highly diverse nodes. Moreover, since many of the existing works rely on expert experience to select and configure estimation parameters, they are less adaptive to rapidly developing smart grids with more advanced applications like demand response [18], integrating renewable energy [19], and cyber security [20].

Machine learning is one of today's most rapidly growing technical fields [21–23]. Traditional machine learning models, such as logistic regression [24], support vector machine [25], and decision tree [26], are based on statistical learning theories. These models have high interpretability (i.e., a human can easily understand the models' behaviors) [27,28] and are relatively simple to train. In recent years, deep learning models based on artificial neural networks have achieved outstanding performances for many difficult tasks like computer vision [29–32], medical diagnosis [33], translation [34], path planning [35] and semantic understanding [36–39]. However, deep learning models still lack sufficient interpretability until now [27]. In this paper, we exploit traditional logistic regression model (namely, nonlinear fitting) to estimate bandwidth requirements, because power grid is a highly regulated domain where the interpretability of decisions is mandatory. In fact, our nonlinear fitting method is able to provide rather accurate estimations in complex smart grid scenarios, as will be proved by simulations later.

3 Features of Smart Grid Communication Network

Unlike public network, communication network in smart grid is built according to the structure and the applications of smart grid, thus it has the following two distinct features, as Fig. 1 illustrates.

Firstly, communication nodes of smart grid are usually built on electricity substations, and communication links are usually built along electricity cables. As a result, smart grid communication network has a hierarchal tree structure, where lower-voltage nodes connect to higher-voltage nodes, and the latter connect to dispatching centers.

Secondly, as lower-voltage substations generate application data, some of these data are aggregated to higher-voltage substations (these higher-voltage substations may also generate some data to upload), and eventually aggregated to dispatching centers. Hence, bandwidth requirements hierarchically converge from lower-voltage substations to dispatching centers.

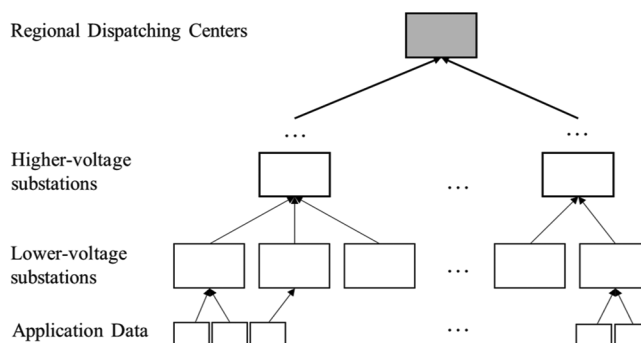


Figure 1: Hierarchical structure and converged data flow of smart grid communication network

For a new smart grid communication network, we often just know the number and the bandwidth demands of the applications on each node. The bandwidth requirements from lower-voltage nodes to higher-voltage node are unknown and need to be estimated, as discussed in the next section.

4 Fine-Grained Bandwidth Estimation Method

Based on the aforementioned features, we propose a fine-grained method for estimating bandwidth requirements of communication nodes in smart grid. We divide and study the characteristics of each node to obtain a more accurate bandwidth estimation method. [Tab. 1](#) summarizes the variables used in this paper.

Table 1: Summary of variables

Variable notation	Definition	Type
\bar{y}	Estimated converged bandwidth of higher-voltage node.	Estimation result
y	Converged bandwidth of higher-voltage node.	Known parameter (only for learning)
N	Number of lower-voltage nodes.	Known parameter
B_i	Bandwidth requirement of lower-voltage node.	Known parameter
ω_i	Convergence weight of lower-voltage node.	Coefficient to learn
$\bar{\omega}_i$	Estimated Convergence weight of lower-voltage node.	Coefficient to learn
X_j	Node characteristic that may affect bandwidth convergence.	Coefficient to learn
β_j	Coefficient relating node characteristics to node convergence weight.	Coefficient to learn
k	Number of key node characteristics	Hyperparameter
n	Power of nonlinear fitting from key characteristics to convergence weight.	Hyperparameter
α_1	Learning rate of gradient descent of linear fitting.	Hyperparameter
α_2	Learning rate of gradient descent of nonlinear fitting.	Hyperparameter

As shown by Fig. 2, our basic idea is using lower-voltage nodes' bandwidth requirements (which are derived from application requirements) to estimate the bandwidth requirements of higher-voltage nodes, and then use these bandwidth estimations to further estimate the bandwidth requirements of even higher-voltage nodes, and eventually estimate the bandwidth requirements of dispatching centers.

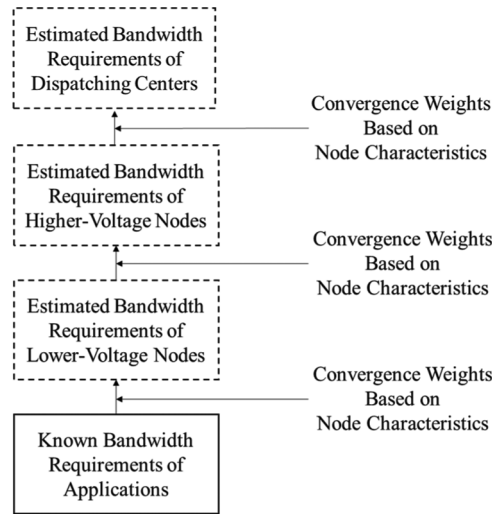


Figure 2: Fine-grained bandwidth estimation method

Specifically, the bandwidth of an upper node (a higher-voltage node or a dispatching center) can be estimated as follows:

$$\bar{y} = \sum_{i=1}^N w_i \cdot B_i, \quad w_i \in [0, 1] \tag{1}$$

where \bar{y} is the estimated converged bandwidth of the upper node, B_1, B_2, \dots, B_N are the total bandwidth requirements of N lower nodes (i.e., data uploaded by even lower nodes plus data generated by the node itself), w_1, w_2, \dots, w_N are the convergence weights of the N nodes (i.e., the ratio of data to upload).

The convergence weight w_i of node i must lie in $[0,1]$ because a lower-voltage node can never transmit data more than its own bandwidth. The value of w_i can be derived from k characteristics of node i :

$$w_i = \sum_{j=1}^k \beta_j \cdot f(X_j) + \beta_{k+1} \cdot g(X_1, X_2 \dots X_k) \tag{2}$$

where X_1, X_2, \dots, X_k are node characteristics, $f(\cdot)$ is a multinomial function of single characteristic, $g(\cdot)$ is a multinomial function of multiple characteristics, β_j and β_{k+1} are coefficients relating node characteristics to node convergence weight.

Eqs. (1) and (2) reveal the bandwidth convergence of lower-voltage nodes to higher-voltage nodes (and dispatching centers), and how such convergence is affected by multiple node characteristics. This way, we achieve fine-grained estimations corresponding to the differences among nodes.

Notably, we have not determined which node characteristics should be considered, and how these characteristics are related to node convergence weight. These two issues will be solved by multivariate nonlinear fitting in the next section.

5 Multivariate Nonlinear Fitting Scheme

In this section, we propose a multivariate nonlinear fitting scheme to relating node characteristics to node bandwidth. In other words, our scheme will learn all the undetermined coefficients in Eqs. (1) and (2) from measured node bandwidth and characteristics.

As Fig. 3 illustrates, our fitting scheme has three major steps. The first step is linear fitting, which takes converged bandwidth as target value to construct the loss function. Linear fitting uses the gradient descent method to derive convergence weight for each node. The second step is correlation coefficient calculation, which is using correlation coefficient to find key characteristics with more significant impacts on convergence weight across network. The last step is multivariate nonlinear fitting. This step reveals the nonlinear relationship among multiple characteristics and the convergence weight of each node, and therefore relates node characteristics to node bandwidth.

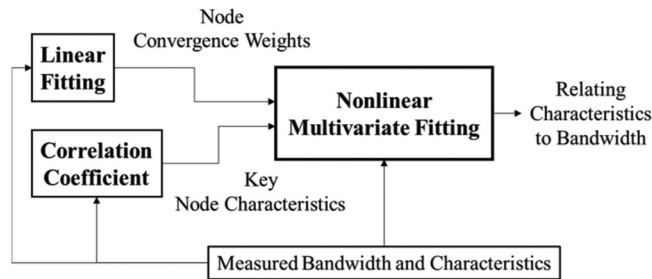


Figure 3: Multivariate nonlinear fitting scheme relating node characteristics to node bandwidth

5.1 Linear Learning

Here we utilize gradient descent to learn node convergence weight from real bandwidth data. In brief, we iteratively calculate the cost between the estimated bandwidth (which is derived from node convergence weight) and the actual bandwidth, and update convergence weight with gradient descent [40], until the cost becomes minimal.

First, we initialize node convergence weights as small positive numbers that are randomly generated within (0, 1), and substitute these weights and the known bandwidth requirements of lower nodes into Eq. (1) to derive the estimated bandwidth of the upper node \bar{y} .

Then we calculate the *cost* for gradient descent as follows:

$$cost = (y - \bar{y})^2 \quad (3)$$

where y is the actual bandwidth of the converged node.

The idea of gradient descent is to minimize *cost* by gradually adjusting node convergence weights. Specifically, the gradient of the cost function can be computed as:

$$\frac{\partial cost}{\partial w_i} = 2 \times (y - \bar{y}) \times B_i \quad (4)$$

where w_i and B_i are the convergence weight and the bandwidth of the node i , respectively.

We gradually adjust the convergence weight of each node as follows:

$$w_i \leftarrow w_i - \alpha_1 \times \frac{\partial cost}{\partial w_i} \quad (5)$$

where α_1 is the learning rate of gradient descent. Its value will be determined by experiments later.

Eqs. (4) and (5) will be executed iteratively until the cost function converges to its minimum. The obtained convergence weight w_i essentially reflects the ratio of node i 's bandwidth that converges to its upper node.

5.2 Correlation Coefficient Calculation

We use correlation coefficient calculation to decide which node characteristics (e.g., number or type of application) have more significant impacts on convergence weight. Only these key node characteristics will be concerned for bandwidth estimation. This is to simplify the acquirement process of node characteristics as well as the subsequent non-linear multivariate learning.

Specifically, for all N nodes, we calculate correlation coefficient between each candidate characteristic and the set of convergence weights as follows:

$$corr(\mathbf{w}, \mathbf{X}_j) = \frac{\text{cov}(\mathbf{w}, \mathbf{X}_j)}{\sqrt{\text{var}(\mathbf{w}) \cdot \text{var}(\mathbf{X}_j)}} \quad (6)$$

where \mathbf{w} is the set of the convergence weights of all nodes (obtained in Section 5.1), \mathbf{X}_j is the set of the values of a candidate characteristic for all nodes, $\text{cov}(\mathbf{w}, \mathbf{X}_j)$ is the covariance of \mathbf{w} and \mathbf{X}_j , and $\text{var}(\mathbf{w})$ and $\text{var}(\mathbf{X}_j)$ are the variance of \mathbf{w} and \mathbf{X}_j , respectively.

After calculating the correlation coefficients for all candidate characteristics, we choose the key characteristics with higher correlation coefficients for the non-linear multivariate learning in the next subsection. The number of key characteristics should be carefully decided to balance between implementation complexity and bandwidth estimation accuracy. This will be further discussed in Section 6.

5.3 Nonlinear Multivariate Fitting

At last, we exploit nonlinear multivariate fitting to reveal how a node's convergence weight is affected by its characteristics. Here, "nonlinear" is to reflect the complexity of such relationship, and "multivariate" is to reflect the combined impacts of multiple node characteristics. Multivariate nonlinear fitting means using mathematical model to express the nonlinear relationship between different characteristics and the convergence weight.

Since we have derived the convergence weight in Section 5.1 and the key characteristics in Section 5.2, we only need to determine the rest unknown parameters in Eq. (2), namely, the multinomial functions $f(\cdot)$ and $g(\cdot)$, and the coefficients β_j and β_{k+1} . This is accomplished through fitting Eq. (2) to the convergence weight obtained in Eq. (5).

First, suppose that we have decided there are k key characteristics and the power of $f(\cdot)$ is n (we will discuss how to derive them later). The nonlinear influence of each characteristic X_j on the convergence weight can be expressed by:

$$f(X_i) = a_1 X_i + a_2 X_i^2 + \dots + a_n X_i^n \quad (7)$$

and the nonlinearly correlated influence of the k key characteristics on the convergence weight can be expressed by:

$$g(X_1, X_2 \dots, X_k) = b_{(1,2)}X_1X_2 + b_{(1,3)}X_1X_3 \dots + b_{(1,2,\dots,k)}X_1X_2 \dots X_k \quad (8)$$

where the coefficients $a_1, \dots, a_n, b_{(1,2)}, \dots, b_{(1,2,\dots,k)}$ will be learned later. It is noteworthy that the complexity of $g(\cdot)$ is $o(2^k)$, which is why we should keep the number of node characteristics k as small as possible.

Second, we substitute Eqs. (7) and (8) into Eq. (2) to express the fitted convergence weight \bar{w}_i for node i as follows:

$$\bar{w}_i = \sum_{j=1}^k \beta_j \cdot f(X_j) + \beta_{k+1} \cdot g(X_1, X_2 \dots X_k) \quad (9)$$

Third, we compare the fitted convergence weight \bar{w}_i in Eq. (9) to the convergence weight w_i derived by Eq. (5), i.e.,

$$loss = \sum_{i=1}^n (w_i - \bar{w}_i)^2 \quad (10)$$

Now we can learn the undetermined parameters via multivariate gradient descent [10]:

$$\theta = \theta \leftarrow \alpha_2 \cdot \frac{\partial loss}{\partial \theta} \quad (11)$$

where α_2 is the learning rate, and its value will be determined by experiments later. θ represents multiple variables $a_1, \dots, a_n, b_{(1,2)}, \dots, b_{(1,2,\dots,k)}, \beta_1, \dots, \beta_{k+1}$.

The above Eqs. (10) and (11) will be executed iteratively until the loss function Eq. (10) converges to its minimum. By repeating this process on all node convergence weights, the values of $a_1, \dots, a_n, b_{(1,2)}, \dots, b_{(1,2,\dots,k)}, \beta_1, \dots, \beta_{k+1}$ are learned.

5.4 Determining Hyperparameters

Finally, we determine the hyperparameters that should be set before learning, namely, the learning rates α_1 in Eq. (5) and α_2 in Eq. (11), the number of the key node characteristics n , and the power of nonlinear fitting k in Eq. (7).

We begin with the learning rates α_1 and α_2 . We initially set them to very small values, e.g., 10^{-8} , and observe that whether the cost in Eq. (3) or the lost in Eq. (10) steadily decreases as we perform linear fitting in Eq. (4) or nonlinear fitting in Eq. (11), respectively. If the decrement is too slow, we gradually increase α_1 or α_2 to learn the unknown parameters more drastically. On the other hand, if the decrement is unstable, we gradually decrease α_1 or α_2 to learn more cautiously. We keep adjusting α_1 and α_2 until the decrement is stable and notable. The resulting α_1 and α_2 will be used for learning later.

Afterwards, we determine the number of the key node characteristics n , and the power of nonlinear fitting k . For practical considerations, we should set them as small as possible (otherwise, there are too many variables to learn in Eqs. (7) and (8)). Therefore, we gradually increase them from $n = 1$ and $k = 1$. For each pair of (n, k) , we use Eq. (6) to find the k key node characteristics, use Eqs. (9)–(11) to derive the other unknown parameters in Eqs. (7) and (8), and use Eqs. (1) and (2) to obtain bandwidth estimations for all upper nodes. This increment process stops as the bandwidth estimation accuracy has become reasonably high and the accuracy increment has become marginal. The pair of (n, k) that has the highest bandwidth estimation accuracy will be used for learning later.

After determining the hyperparameters α_1, α_2, n , and k , we can learn all parameters' values in Eqs. (1) and (2) from a present network with Eqs. (3)–(11). Then we can use Eqs. (1) and (2) to estimate the bandwidth requirements of new networks.

6 Simulation Results

We perform NS2 and TCL simulations to test our estimation method. Among them, NS2 is an open-source simulation platform for network technology. TCL is the script language on NS2. The simulated networks are based on a real-world regional smart grid in China. The network topologies are set as Figs. 4 and 6, and the applications are configured and deployed according to [1].

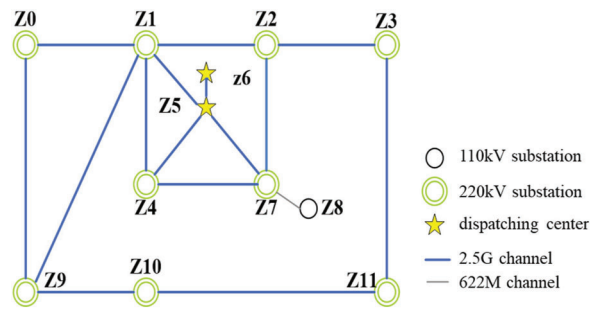


Figure 4: Network for learning estimation parameters

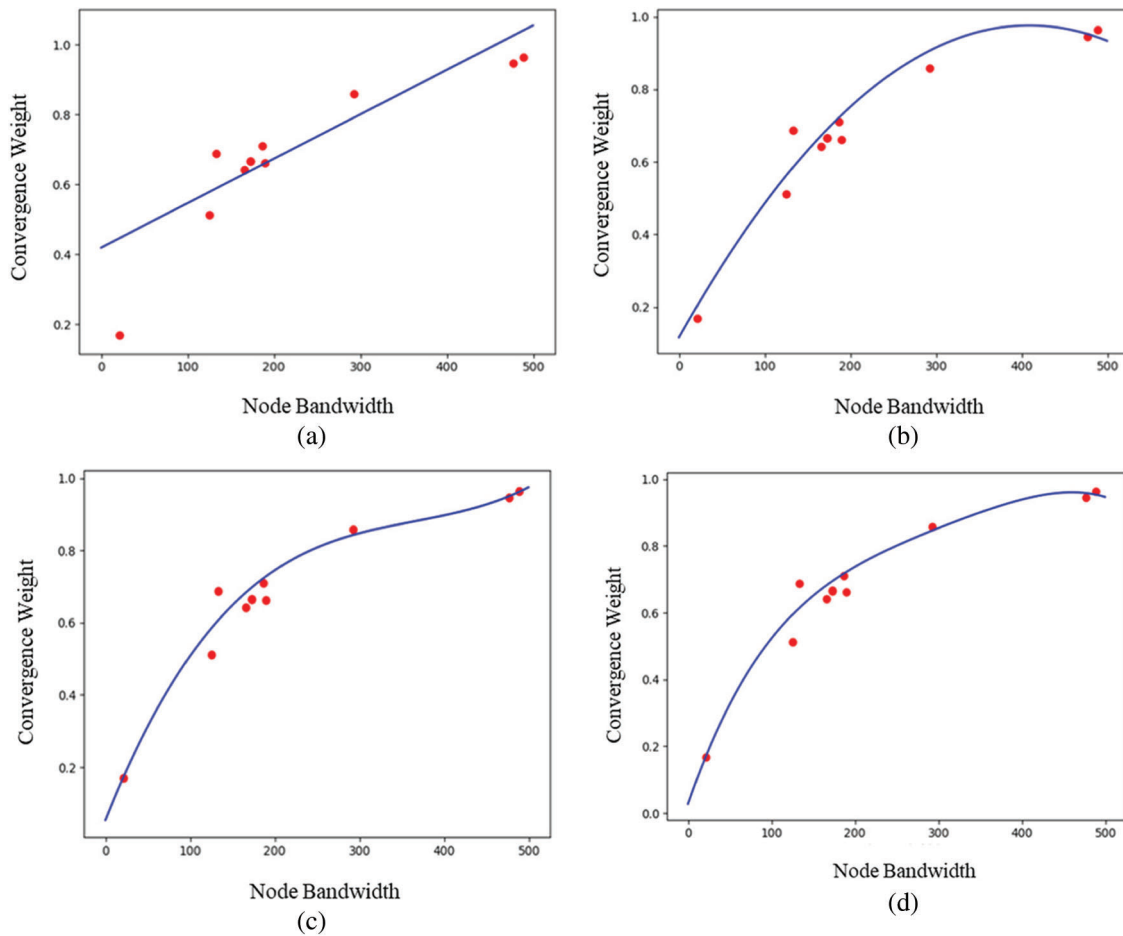


Figure 5: The fitting curves relating node characteristics to convergence weight for different values of n . Note that Z2 is excluded here for it is used for validating (a) $n = 1$ (b) $n = 2$ (c) $n = 3$ (d) $n = 4$

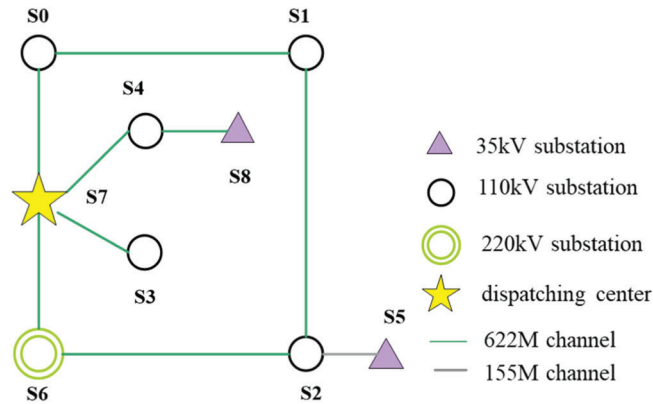


Figure 6: Network for estimating bandwidth

6.1 Learning from Present Network

We learn the estimation parameters from the network in Fig. 4. This network is based on the regional power grid of a moderate-sized city in China, which has 2 regional dispatching centers, 9 220 kV substations, and 1 110 kV substation. It represents a “present network” where the nodes’ bandwidth consumptions have been known. During learning, Z2 node is used for validating, and the rest nodes are used for fitting.

We first determine the linear learning rate α_1 and the nonlinear learning rate α_2 . We increase α_1 and α_2 from 10^{-8} to 10^{-4} , respectively, and find that the fitting costs of Eqs. (4) and (11) stably decrease only when $\alpha_1 \leq 10^{-6}$ and $\alpha_2 \leq 10^{-6}$. Since larger learning rates lead to quicker learning, we choose $\alpha_1 = 10^{-6}$ and $\alpha_2 = 10^{-6}$.

Next, we investigate 8 node characteristics that maybe related to convergence weight, as listed in Tab. 2.

We calculate correlation coefficient as Eq. (6) to find that there are $k = 3$ node characteristics closely related to convergence weight, which are: node’s total bandwidth > number of real-time applications > node strength.

Table 2: Definitions of node characteristics

Node characteristics	Definition
Node’s total bandwidth	The total bandwidth of the data uploaded from the lower nodes and the data generated by the node itself
Number of real-time applications	The number of real-time applications (e.g., controlling signals of dispatching system) carried by the node.
Node strength	The number of other nodes connected to the node.
Number of normal applications	The number of non-real-time applications (e.g., consumer information of energy meters) carried by the node.
Bandwidth of real-time applications	The bandwidth requirements of the real-time applications on the node.
Bandwidth of normal applications	The bandwidth requirements of the non-real-time applications on the node.
Node’s voltage level	The node’s voltage level in power grid (e.g., a 220 kV substation).
Node capacity	The node’s transmission channel capacity to its upper nodes.
Node distance	The length of the shortest path between the node and the dispatching center
Node centrality	The average length of the shortest path between the node and all other nodes

We show 4 node characteristics with the highest correlation coefficient values in Tab. 3, which are node’s total bandwidth, number of real-time applications, node strength, and number of normal applications. It can be seen that while the first 3 characteristics have correlation coefficients larger than 0.6, the fourth characteristic (i.e., number of normal applications) drastically drops to 0.36. Such result indicates that number of normal applications (and the characteristics after it) has statistically ignorable impacts on convergence weight.

Table 3: The first 4 node characteristics most related to convergence weight

	Node bandwidth	Number of real-time applications	Node strength	Number of normal applications
Convergence weight	0.78	0.73	0.63	0.36

The above result is reasonable. Firstly, a node’s total bandwidth basically reflects how important it is for smart grid (as complex applications tend to be deployed in critical grid sites), which means that a node with higher bandwidth requirement often has proportionally more data to upload. Secondly, most real-time applications need to communicate with dispatching center, so a node with more real-time applications implies that it has more data to upload. Thirdly, a node with higher node strength means that it is connected by many other nodes, so it tends to have more data to upload. Later we will show that our estimation based on these three characteristics indeed achieves rather high accuracy.

Afterwards, we determine the power of Eq. (7), n . According to Eq. (9), the value of n directly affects the fitting performance from node characteristics to convergence weight. To demonstrate this, we draw the fitting curves relating the 3 node characteristics (for conciseness, we only show each node’s total bandwidth on the x-axis) to the 11 nodes’ convergence weights in Fig. 5. Observe that the fitting curve can barely match itself to all the points when $n \leq 2$, which means that the relationship between the node characteristics and the convergence weight is too complicated for these values of n to capture. When $n \geq 3$, the fitting curve is able to reach most of the points, and thereby the convergence weight is well related to the node characteristics.

Nevertheless, according to machine learning theory, n being too large will lead to overfitting, that is, our method can achieve high accuracy during learning, but its accuracy will drop if it is applied to nodes excluded by learning process (i.e., Z2 in this case). This statement is verified by Tab. 4, where we compute the average bandwidth estimation accuracy across the network for different values of n . Observe that although the fitting accuracy keeps increasing as n grows, the validating accuracy for Z2 reaches the maximum at $n=3$, and decreases as n becomes larger. This clearly indicates that overfitting occurs for $n > 3$. Combing this result with Fig. 5, we let $n = 3$.

Finally, with $k=3$ and $n=3$, we can further derive the values of $a_1, \dots, a_n, b_{(1,2)}, \dots, b_{(1,2,\dots,k)}, \beta_1, \dots, \beta_{k+1}$. The deriving process and the final results are omitted here.

Table 4: Average accuracy for different values of n

n	1 (%)	2 (%)	3 (%)	4 (%)
Fitting accuracy	78.5	91.9	93.7	93.8
Validating accuracy	71.5	89.3	90.6	89.6

6.2 Estimating on New Network

Now we use the results in Section 6.1 to estimate the network bandwidth in Fig. 6. This network is based on a small city’s power grid, which has 1 regional dispatching centers, 1 220 kV substations, 5 110 kV substation, and 1 35 kV substation. This represents a “new network” where only applications’ bandwidth requirements are known. Note that the network is highly heterogeneous with 4 different types of nodes, which is difficult for bandwidth estimation.

We are mostly interested in the estimation accuracy of the dispatching center, because it is the most important node in smart grid’s control system, and its estimation accuracy basically depends on the estimation accuracies of all the other nodes. Tab. 5 compares our method with 3 recent works in [2,4,6], where t , a , and b mean that network topology, applications, and node’s total bandwidth are considered in estimation, respectively.

Table 5: Estimated bandwidth of dispatching center

Method	b	t	a	Estimation (Mbps)	Accuracy (%)
PCCM [2]	–	–	√	1190.10	43.1
CA [4]	–	√	–	662.27	87.3
CBT [6]	–	√	√	660.81	87.0
Our method (t)	–	√	–	807.53	93.6
Our method ($t + a$)	–	√	√	802.58	94.2
Our method ($t + a + b$)	√	√	√	761.59	99.6
Actual bandwidth	–	–	–	759.56	–

Observe that our method ($t + a + b$) is just slightly higher than the actual bandwidth requirement of 759.56 Mbps, which outperforms the alternative methods (PCCM [2], CA [4], and CBT [6]) by 12.6% to 56.5% higher accuracy. The major reason is that our method accomplishes finer-grained estimation via considering network topology (node strength), applications (number of real-time applications), and node bandwidth as key node characteristics (see Section 6.1). In contrast, the alternative methods only consider one or two of these characteristics, thus they can hardly differentiate the convergence impacts of different lower-voltage nodes on the dispatching center.

In fact, Tab. 5 also shows that our method achieves higher accuracy as it takes more characteristics into consideration, i.e., $t < (t + a) < (t + a + b)$. Such result further proves that finer-grained estimation leads to higher accuracy.

We further investigate how the selection of node characteristics affects convergence weight learning, and eventually affects bandwidth estimation. This can be clearly illustrated by the ranking of learned convergence weights of different methods in Tab. 6.

By considering node strength (t), number of real-time applications (a), and node bandwidth (b), our method derives the same ranking as the actual network does. As the matter of fact, for the dispatching center (S7), the ranking of the convergence weights of all nodes should be: itself (S7), the nodes linked by lower-voltage nodes (S2, S6, and S4), the nodes without lower-voltage nodes (S0, S1, S3), and the lowest-voltage nodes (S5, S8). Our method’s fine-grained recognition of different nodes is the key to accurate bandwidth estimation.

Table 6: Ranking of learned convergence weights

PCCM	CA	CBT	Our method ($t + a + b$)	Actual
–	S7	S7	S7	S7
–	S2	S2	S2	S2
–	S6	S6	S6	S6
–	S4	S0	S4	S4
–	S0	S1	S0	S0
–	S1	S4	S1	S1
–	S8	S5	S3	S3
–	S5	S3	S5	S5
–	S3	S8	S8	S8

On the other hand, because PCCM ignores network topology and node bandwidth, it cannot fully recognize the differences among various nodes to learn convergence weights in a heterogeneous network like Fig. 6. This explains why PCCM has rather low estimation accuracy in Tab. 5.

Both CA and CBT have considered topology for estimation, so they can roughly infer how the nodes converges the dispatching center, and hence make partially correct rankings in Tab. 6. However, note that both CA and CBT incorrectly rank S3 after S5 due to their neglect of node bandwidth. This explains their relatively low estimation accuracy in Tab. 5.

7 Conclusion

In this paper, we propose a novel fine-grained bandwidth estimation method for smart grid communication network. The method achieves fine-grained estimations through explicitly considering how bandwidth requirements of different nodes converges to upper nodes, and it exploits multivariate nonlinear learning to derive multiple convergence parameters from present network without needing human experience. Due to these two novelties, our method outperforms existing methods by up to 56.5% higher estimation accuracy. Through the comparison of different characteristics, we find that the fitting accuracy of the three characteristics selected in this paper is higher, which can reach 99.6%. In future, we will collect transmission data from other industrial Internet to train this model. So that this method can be applied to other Industrial Internets, such as the communication networks of railway or oil pipeline. Furthermore, we will study how to directly estimate bandwidth requirement and predict long-term development based on current network information using deep learning and big data technologies.

Funding Statement: This work was supported by Natural Science Foundation of China (Grant No.62071098); Sichuan Application and Basic Research Funds (Grant No. 2021YJ0313); Sichuan Science and Technology Program (Grant No. 2021YFG0307).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] Z. Zhao, J. Fan, X. Jin, D. Zhang, Y. Liu *et al.*, “Information flow model and service flow calculation of electric power backbone communication network,” *Telecommunications Science*, vol. 33, no. 5, pp. 153–163, 2017.

- [2] Z. Peng, B. Li and C. Xu, "A study on the change law and trend of the elasticity coefficient of jiangsu electric power," *Statistical Science and Practice*, vol. 10, pp. 13–16, 2017.
- [3] S. Mankad and G. Michailidis, "Discovery of path-important nodes using structured semi-nonnegative matrix factorization," in *IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Saint Martin, France, pp. 288–291, 2013.
- [4] S. Ji, X. Wang, J. Meng, W. Bao and Y. Qi, "A node importance evaluation method for electric power communication networks with weighted bandwidth," *Power Information and Communication Technology*, vol. 12, no. 5, pp. 25–29, 2014.
- [5] Y. Zeng, "Evaluation of node importance and invulnerability simulation analysis in complex load-network," *Neurocomputing*, vol. 416, no. 1, pp. 158–164, 2019.
- [6] B. Fan, C. Zheng and L. Tang, "Risk assessment of power communication network based on node importance," in *Conf. IEEE Advanced Information Management, Communicates, Electronic and Automation Control (IMCEC)*, Chongqing, China, pp. 818–821, 2019.
- [7] J. Liu, Q. Xiong, W. Shi, X. Shi and K. Wang, "Evaluating the importance of nodes in complex networks," *Physica a: Statistical Mechanics and its Applications*, vol. 452, pp. 209–219, 2016.
- [8] G. Gao, Y. Sun, J. Zhou and H. Chen, "Study on the prediction and analysis method of electric power communication service," *Electric Power Information and Communication Technology*, vol. 14, no. 7, pp. 108–112, 2016.
- [9] Z. Zhu, X. You, C. Zheng, L. Li and B. Fan, "Power communication website bandwidth estimation method based on position weighting," *Telecommunication Science*, vol. 33, no. 10, pp. 150–156, 2018.
- [10] F. Phillipson, D. Worm, N. Neumann, A. Sangers and S. Wiarda, "Estimating bandwidth coverage using geometric models," in *European Conf. Networks and Optical Communications (NOC)*, Lisbon, Portugal, pp. 152–156, 2016.
- [11] Y. Takano, R. Mutoh, N. Oguchi and S. Abe, "Estimating available bandwidth in mobile networks by correlation coefficient," in *Conf. Asia-Pacific Network Operations and Management Symposium (APNOMS)*, Kanazawa, Japan, pp. 1–4, 2016.
- [12] Q. Huang, "A novel important node discovery algorithm based on local community aggregation and recognition in complex networks," *International Journal of Wireless Information Networks*, vol. 27, pp. 253–260, 2020.
- [13] F. Liu, B. Xiao, H. Li and J. Xue, "Complex network node centrality measurement based on multiple attributes," in *Int. Conf. on Modelling, Identification and Control (ICMIC)*, Guiyang, China, pp. 1–5, 2018.
- [14] Z. Liang and J. Li, "Identifying and ranking influential spreaders in complex networks," in *Int. Computer Conf. on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Chengdu, China, pp. 393–396, 2014.
- [15] P. Hu and T. Mei, "Ranking influential nodes in complex networks with structural holes," *Physica a: Statistical Mechanics and Its Applications*, vol. 490, pp. 624–631, 2018.
- [16] T. Agryzkov, J. Oliver and L. Tortosa, "A new betweenness centrality measure based on an algorithm for ranking the nodes of a network," *Applied Mathematics and Computation*, vol. 244, pp. 467–478, 2014.
- [17] N. S. Kagami, R. I. T. da Costa Filho and L. P. Gaspar, "CAPEST: Offloading network capacity and available bandwidth estimation to programmable data planes," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 175–189, 2020.
- [18] X. Yan, Y. Ozturk, Z. Hu and Y. Song, "A review on price-driven residential demand response," *Renewable and Sustainable Energy Reviews*, vol. 96, pp. 411–419, 2018.
- [19] M. Ourahou, W. Ayrir, B. E. Hassouni and A. Haddi, "Review on smart grid control and reliability in presence of renewable energies: Challenges and prospects," *Mathematics and Computers in Simulation*, vol. 167, pp. 19–31, 2020.
- [20] M. Z. Gunduz and R. Das, "Cyber-security on smart grid: Threats and potential solutions," *Computer Networks*, vol. 169, pp. 107094, 2020.
- [21] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

- [22] J. Xu, W. Wang, H. Y. Wang and J. H. Guo, "Multi-model ensemble with rich spatial information for object detection," *Pattern Recognition*, vol. 99, pp. 107098, 2020.
- [23] Z. Li, W. Li, F. Lin, Y. Sun, M. Yang *et al.*, "Hybrid malware detection approach with feedback-directed machine learning," *Science China Information Sciences*, vol. 63, no. 3, pp. 139103, 2020.
- [24] S. Sperandei, "Understanding logistic regression analysis," *Biochemia Medica*, vol. 24, no. 1, pp. 12–18, 2014.
- [25] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, 2020.
- [26] Y. Song and L. Ying, "Decision tree methods: Applications for classification and prediction," *Shanghai Archives of Psychiatry*, vol. 27, no. 2, pp. 130–135, 2015.
- [27] D. V. Carvalho, E. M. Pereira and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, pp. 832, 2019.
- [28] K. Ma, X. Liu, G. Li, S. Hu, J. Yang *et al.*, "Resource allocation for smart grid communication based on a multi-swarm artificial bee colony algorithm with cooperative learning," *Engineering Applications of Artificial Intelligence*, vol. 81, pp. 29–36, 2019.
- [29] M. Luo, J. Cao, X. Ma, X. Zhang and R. He, "FA-Gan: Face augmentation GAN for deformation-invariant face recognition," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2341–2355, 2021.
- [30] H. Song, W. Yang, H. Yuan and H. Bufford, "Deep 3d-multiscale densenet for hyperspectral image classification based on spatial-spectral information," *Intelligent Automation & Soft Computing*, vol. 26, no. 6, pp. 1441–1458, 2020.
- [31] J. Xu, R. Song, H. L. Wei, J. H. Guo, Y. F. Zhou *et al.*, "A fast human action recognition network based on spatio-temporal features," *Neurocomputing*, vol. 441, pp. 350–358, 2020.
- [32] Z. Li, W. Wei, T. Zhang, M. Wang, S. Hou *et al.*, "Online multi-expert learning for visual tracking," *IEEE Transactions on Image Processing*, vol. 29, pp. 934–946, 2020.
- [33] S. H. Park and K. Han, "Methodologic guide for evaluating clinical performance and effect of artificial intelligence technology for medical diagnosis and prediction," *Radiology*, vol. 286, no. 3, pp. 800–809, 2018.
- [34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.*, "Attention is all you need," in *Proc. of Int. Conf. on Neural Information Processing Systems*, NY, USA, pp. 6000–6010, 2017.
- [35] X. Chen, Z. Wan and J. Wang, "A study of unmanned path planning based on a double-twin rbm-bp deep neural network," *Intelligent Automation & Soft Computing*, vol. 26, no. 6, pp. 1531–1548, 2020.
- [36] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan *et al.*, "Language models are few-shot learner," in *Proc. of Int. Conf. on Neural Information Processing Systems*, Virtual-only Conference, pp. 1877–1901, 2020.
- [37] M. Wang, S. Niu and Z. Gao, "A novel scene text recognition method based on deep learning," *Computers, Materials & Continua*, vol. 60, no. 2, pp. 781–794, 2019.
- [38] A. Zhang, B. Li, W. Wang, S. Wan and W. Chen, "Mii: A novel text classification model combining deep active learning with bert," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1499–1514, 2020.
- [39] J. Huh, S. Otgonchimeg and K. Seo, "Advanced metering infrastructure design and test bed experiment using intelligent agents: Focusing on the PLC network base technology for smart grid system," *Supercomput*, vol. 72, no. 5, pp. 1862–1877, 2016.
- [40] I. Goodfellow, Y. Bengio and A. Courville, "in Deep learning," Cambridge, MA, USA, MIT Press, 2016.