Journal of Computer and Communications, 2024, 12, 247-261
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

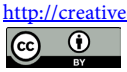
Decentralized Semi-Supervised Learning for Stochastic Configuration Networks Based on the Mean Teacher Method

Kaijing Li[1], Wu Ai[1,2*]

[1]School of Mathematics and Statistics, Guilin University of Technology, Guilin, China
[2]Guangxi Colleges and Universities Key Laboratory of Applied Statistics, Guilin, China
Email: *aiwu818@gmail.com

How to cite this paper: Kaijing Li, Wu Ai (2024) Decentralized Semi-Supervised Learning for Stochastic Configuration Networks Based on the Mean Teacher Method. *Journal of Computer and Communications*, **12**, 247-261.
https://doi.org/10.4236/jcc.2024.124017

Received: March 18, 2024
Accepted: April 27, 2024
Published: April 30, 2024



Abstract

The aim of this paper is to broaden the application of Stochastic Configuration Network (SCN) in the semi-supervised domain by utilizing common unlabeled data in daily life. It can enhance the classification accuracy of decentralized SCN algorithms while effectively protecting user privacy. To this end, we propose a decentralized semi-supervised learning algorithm for SCN, called DMT-SCN, which introduces teacher and student models by combining the idea of consistency regularization to improve the response speed of model iterations. In order to reduce the possible negative impact of unsupervised data on the model, we purposely change the way of adding noise to the unlabeled data. Simulation results show that the algorithm can effectively utilize unlabeled data to improve the classification accuracy of SCN training and is robust under different ground simulation environments.

Keywords

Stochastic Neural Network, Consistency Regularization, Semi-Supervised Learning, Decentralized Learning

1. Introduction

The popularity of smart devices and the rapid growth of the Internet have brought about an explosion of data. Typically, data is collected by devices and then transmitted to a fusion center for processing. Data explosion causes this approach to incur serious communication costs and data latency problems [1]. More importantly, for industries involving large amounts of private data (including financial and healthcare industries), a fusion center may collect data

without authorization, and the risk of data leakage is greatly increased if the fusion center is attacked during the data exchange process [2]. Therefore, it becomes particularly important to study machine learning systems where multiple interconnected agents are combined to complete a global inference model [3].

Distributed learning is mainly divided into two directions. The model distributed cuts the model horizontally or vertically into a number of sub-networks and divides it into each agent responsible for a part of the model's operation, reducing the amount of computation. This approach fails to solve the problem of private data. The other type of data distribution fits the current trend of data distribution more, where local data is stored on each agent, the model is trained based on the respective local data, and only the model parameters are uploaded to the server, which aggregates the calculations and then passes them back to the agents to update the parameters. But this method still can't avoid the problem that if the server is attacked, the whole system will be down. Therefore, peer-to-peer decentralized machine learning becomes very necessary [4]. Each agent only exchanges parameters with its neighboring agents and agrees collaboratively, so that even if some of the data is damaged, the overall aggregation is not affected. Currently, most decentralized learning systems focus on supervised learning. However, in areas such as computer-aided diagnosis [5], drug development [6] and speech tagging, labeled data is scarce and expensive, and at such times, under certain assumptions about the data distribution (e.g. the manifold assumption) unlabeled data points can provide additional information to support modeling and help obtain a better classifier [7].

The combination of semi-supervised theory and distributed algorithms has yielded some results so far. Semi-supervised learning has three basic assumptions: the low-density assumption, the streaming assumption, and the smoothing assumption [8]. These three assumptions derive corresponding semi-supervised methods that are applied in combination with distributed machine learning, such as maximum-margin methods relying on the low-density assumption applied to distributed SVM classification [9]. The popular regularization method based on the manifold assumption, on the other hand, completes the distributed adaptation of the method by means of a diffusion-adaptive framework for matrix complementarity via adjacency matrices [10]. Smoothness assumptions are often combined with neural networks to require that the predictive model be robust to local perturbations in the inputs. The currently distributed combination of neural networks and semi-supervised learning usually remains in the basic single-layer feed-forward neural network or the superposition of layers [11], and we believe that the modification of the network can effectively improve the classification accuracy of semi-supervised learning. Therefore, we turn our goal into stochastic configuration network (SCN) [12]. SCN is simple and easy to implement in terms of structure, has the advantages that deep neural networks lack, and because of their special training method, has better classification results than single-layer networks.

For the above reasons, we aim to develop a new distributed semi-supervised algorithm based on SCN. In this paper, the teacher model and student model are introduced in the SCN-based distributed learning algorithm to improve the distributed learning algorithm. On each agent, exponential moving average (EMA) aggregation is used for the model parameters instead of directly sharing the weights between the teacher model and the student model. Minimizing the consistency regularization loss of the teacher model and the student model so that both get consistent results for the same samples can further improve the robustness of the current parameters of the network. We call the newly proposed model CMT-SCN.

The contributions of this paper are summarized as follows.

- A decentralized learning algorithm is developed for the semi-supervised mean teacher method, which does not require a centralized data processing center and is suitable for handling massive data and protecting user privacy.
- Introducing a randomized configuration network to achieve better modeling results under the same network structure.
- The performance of the proposed DMT-SCN algorithm can effectively improve the data performance while utilizing unsupervised data.

The rest of the paper is organized as follows. In **Section 2**, the SCN formulation and the structure of the distribution required for the discussion are provided. In **Section 3**, we propose a decentralized semi-supervised SCN algorithm with a mean teacher method. Numerical experiments between DMT-SCN and decentralized supervised SCN are presented in **Section 4**. **Section 5** summarizes the findings.

2. Preliminaries

2.1. Stochastic Configuration Network (SCN)

SCNs are structured by first constructing a small network and then continuously adding hidden nodes until meet a predetermined tolerance error or reach the predetermined number of hidden nodes. Specifically, for the objective function $f:\mathbb{R}^d \to \mathbb{R}^C$, assuming the existence of a single-layer neural network with $L-1$ hidden nodes and a input value of x, the output of the network can be formulated as (1):

$$f_{L-1}(x)=\sum_{l=1}^{L-1}\theta_l h_l(x)=\sum_{l=1}^{L-1}\theta_l h_l(x,\beta_l,b_l),\quad L=1,2,\cdots, \tag{1}$$

where

$$h_l(x,\beta_l,b_l)=\phi_l\left(\beta_l^{\mathrm{T}}x+b_l\right). \tag{2}$$

Here, h_l refers to the output of node l in the hidden layer; $\theta_l=\left[\theta_{l,1},\theta_{l,2},\cdots,\theta_{l,C}\right]^{\mathrm{T}}$ denotes the output weights of node l and C refers to the number of categories of the label. Then the current residual can be expressed by

$$e_{L-1}=f(x)-f_{L-1}(x)=\left[e_{L-1,1},e_{L-1,2},\cdots,e_{L-1,C}\right]^{\mathrm{T}}. \tag{3}$$

If $\|e_{L-1}\|$ does not satisfy the pre-determined requirements for the tolerance limit, the network will continue to generate new hidden layer nodes with a new random basis function h_L (with input weights β_L and bias b_L) allowing the function to be represented as

$$f_L(x) = f_{L-1}(x) + \theta_L h_L(x, \beta_L, b_L), \tag{4}$$

to make the residual smaller until the tolerance limit is met. The supervision mechanism of SCN is realized by the following inequalities:

$$\begin{aligned}\xi_{L,q}(x) = &\frac{\left(e_{L-1,q}^{\mathrm{T}}(x)h_L(x)\right)^2}{h_L^{\mathrm{T}}(x)h_L(x)} - (1-r-\mu_L)e_{L-1,q}^{\mathrm{T}}(x)e_{L-1,q}(x) \\ &\geq 0, \quad q = 1, \cdots, C,\end{aligned} \tag{5}$$

where $\mu_L = (1-r)/(L+1)$, $0 < r < 1$. The optimization problem for the output weights can be obtained by the following equation:

$$\left[\theta_1^*, \theta_2^*, \cdots, \theta_L^*\right]^{\mathrm{T}} = \arg\min_{\boldsymbol{\theta}} \left\| f - \sum_{l=1}^{L} \theta_l h_l \right\|, \tag{6}$$

where $f_L^* = \sum_{l=1}^{L} \theta_l^* h_l$ and $\theta_l^* = \left[\theta_{l,1}^*, \theta_{l,2}^*, \cdots, \theta_{l,C}^*\right]^{\mathrm{T}}$.

2.2. Mean Teacher Method

Initially, the use of unsupervised data by Π model was achieved by applying noise with a mean of 0 to the data, which, according to the smoothness assumption, should not affect the prediction results of the data after applying disturbances to it. Therefore, the unlabeled data can be utilized by feeding the noisy and raw data together into the model and optimizing the loss of consistency in their predictions. The formula for its loss is as follows:

$$L(\theta) = \mathbb{E}_{x \in X}\left(g^L(\theta, x, \xi_1), g^L(\theta, \hat{x}, \xi_2)\right), \tag{7}$$

where $\hat{x}$ represents the input value after applying noise to the unlabeled x, and L shows the number of layers of Ladder network structure.

Based on Π model, emporal ensembling combines the forecasts of the previous round with the results of the current round by introducing exponential moving average (EMA), which gives it a higher stability compared to a single forecast. The approach leads to the need for each data to calculate its EMA in a single round of prediction, which greatly increases the computational cost.

Mean teacher method [13] uses EMA to update model weights instead of predictions. The mean teacher method for semi-supervised learning consists of two models, the teacher model and the student model. The teacher model is identical to the student model in terms of network structure, and the teacher model uses EMA to update model weights for the student model in each epoch. The consistency loss added between two models is shown in (8):

$$J(\theta) = \mathbb{E}_{x,\eta',\eta}\left[\left\| f(x, \theta', \eta') - f(x, \theta, \eta) \right\|^2\right]. \tag{8}$$

In general, the loss of consistency between predictions is obtained by calculating

RMSE.

3. Decentralized Semi-Supervised SCNs

In this section, we develop a semi-supervised decentralized algorithm called DMT-SCN. After summation it is mathematically equivalent to the learning problem of a centralized semi-supervised SCN. We use an undirected graph to model how agents communicate with each other, allowing unlabeled datasets scattered among different agents to be trained together and eventually output a unified model.

3.1. Communication Network Model

We use topology graphs to model the way different agents communicate with each other in real scenarios. A node represents a realistic agent that can store data and compute, while an edge represents nodes information interaction, if there is no edge between two nodes, it means that they cannot produce communication. Suppose a graph $\mathcal{G}=(\mathcal{J},\mathcal{E})$, where $\mathcal{J}=\{1,\cdots,J\}$ is the set of agents and $\mathcal{E}\subseteq\mathcal{J}\times\mathcal{J}$ is the set of edges. We use a weight matrix $\mathrm{G}_0\in\mathbb{R}^{|\mathcal{E}|\times|\mathcal{V}|}$ to represent the connectivity between agents. For agent m, $W_{mn}=1$ only if there is an edge connecting agent m to agent n, otherwise it is equal to 0.

3.2. Problem Formulation

In a centralized learning framework, even if the data are initially distributed across different agents, they all need to be transferred to a central server in order to build the complete dataset $\mathcal{D}$. In decentralized machine learning algorithms, we dispense with this data transfer step. The data maintains its original distribution among multiple agents, which are connected to each other through a communication network. Each agent holds only a portion of the total dataset, and we consider these J datasets to constitute an interconnected network in which each agent stores its unique dataset. Each agent j, all belonging to the set $\mathcal{J}=\{1,\cdots,J\}$, is able to access its corresponding natively labelled training dataset $\mathcal{D}_j=\{\mathbf{x}_j,\mathbf{t}_j\}_{j=1}^{N_j}$ of size N_j. Based on the above elements, we provide a clear definition of the decentralized cooperative learning problem. In a decentralized environment, there exist J agents that collectively pursue a unified goal: to minimise some joint objective function by means of a complete distribution. In other words, these agents are committed to solving the objective function described below.

$$\min_{\boldsymbol{\theta}} \left\{\frac{1}{2}\sum_{j=1}^{J}\left\|\mathrm{H}_j\theta-\mathrm{T}_j\right\|^2+\frac{1}{2}\sum_{j=1}^{J}\left\|\tilde{\mathrm{H}}_j\theta-\hat{\mathrm{H}}_j\omega\right\|^2+\frac{R}{2}\|\theta\|^2\right\}, \tag{9}$$

where H_j is the hidden layer output value of labeled data, while $\tilde{\mathrm{H}}_j$ and $\hat{\mathrm{H}}_j$ are the hidden layer output values of unlabeled data after adding different noises, respectively. Here θ is the parameter value of the overall teacher model and ω is the parameter of the student model. In order to transform it into a struc-

ture suitable for distributed problems, we add the following new constraints:

$$\min_{\boldsymbol{\theta}} \quad \left\{\frac{1}{2}\sum_{j=1}^{J}\left\|\mathrm{H}_j\theta_j-\mathrm{T}_j\right\|^2+\frac{1}{2}\sum_{j=1}^{J}\left\|\tilde{\mathrm{H}}_j\theta_j-\hat{\mathrm{H}}_j\omega_j\right\|^2+\frac{R}{2J}\sum_{j=1}^{J}\left\|\boldsymbol{\theta}_j\right\|^2\right\} \tag{10}$$

$$\text{s.t.} \quad \theta_j=\theta_i \quad \forall j\in\mathcal{J}, \quad \forall i\in\mathcal{N}_j. \tag{11}$$

When θ_j on each agent is equal and equal to θ, the function realizes the equivalence with (9).

In centralized learning, H contains the output of all the data in the hidden layer, while H_j contains only the data local to agent j. Specifically, the two possess the following relationship in their structure.

$$\mathrm{H}=\begin{bmatrix}\mathrm{H}_1\\ \vdots\\ \mathrm{H}_J\end{bmatrix}, \mathrm{T}=\begin{bmatrix}\mathrm{T}_1\\ \vdots\\ \mathrm{T}_J\end{bmatrix}, \tag{12}$$

$$\mathrm{H}_j=\begin{bmatrix}h_{(j,1)}\\ \vdots\\ h_{(j,N_j)}\end{bmatrix}, \mathrm{T}_j=\begin{bmatrix}t_{(j,1)}\\ \vdots\\ t_{(j,N_j)}\end{bmatrix}, \ j\in\mathcal{J}. \tag{13}$$

We note that in the realized form, H is obtained by splitting vertically into J copies of H_j. Each matrix H_j, $j\in\mathcal{J}$ has N_j rows from matrix H_j. The analogy can be made not only with labels T_j for labeled data, but also with unlabeled data $\tilde{\mathrm{H}}_j$ and $\hat{\mathrm{H}}_j$ with added noise.

In solving for θ_j, we transform it using graph structures. Since the essence of the restriction of θ is that the parameters of the neighboring agents are the same, the formula (10) can be represented as

$$\min_{\boldsymbol{\theta}} \quad \left\{\frac{1}{2}\sum_{j=1}^{J}\left\|\mathrm{H}_j\theta_j-\mathrm{T}_j\right\|^2+\frac{1}{2}\sum_{j=1}^{J}\left\|\tilde{\mathrm{H}}_j\theta_j-\hat{\mathrm{H}}_j\mathrm{w}_j\right\|^2+\frac{R}{2J}\sum_{j=1}^{J}\left\|\theta_j\right\|^2\right\} \tag{14}$$

$$\text{s.t.} \quad \left(\mathrm{G}\otimes\mathrm{I}_L\right)\mathcal{A}=0, \tag{15}$$

where $\mathcal{A}$ is equal to (14). Trying to solve the above problem by ADMM [14] is equivalent to transforming:

$$\mathcal{A}(t+1)=\arg\min_{\mathcal{A}} L_\rho\left(\mathcal{A},\boldsymbol{\mu}(t)\right), \tag{16}$$

$$\mu(t+1)=\mu(t)-\rho\left(\mathrm{G}\otimes\mathrm{I}_L\right)\mathcal{A}(t+1), \tag{17}$$

$$\omega(t+1)=\alpha\omega(t)+(1-\alpha)\theta(t+1), \tag{18}$$

where $\mathrm{G}=\left[\mathrm{G}_0^{\mathrm{T}},-\mathrm{G}_0^{\mathrm{T}}\right]^{\mathrm{T}}$, and $\mathrm{G}_0\in\mathbb{R}^{|\mathcal{E}|\times|\mathcal{J}|}$ is the edge-agent incidence matrix of the communication network $\mathcal{G}(\mathcal{J},\mathcal{E})$. In other words, iterations of θ are updated by

$$\begin{aligned}\theta_j(t+1)=\arg\min_{\boldsymbol{\theta}_j}\Bigg\{&\left\|\mathrm{H}_j\theta_j-\mathrm{T}_j\right\|^2+\left\|\tilde{\mathrm{H}}_j\theta_j-\hat{\mathrm{H}}_j\omega_j\right\|^2+\frac{R}{2J}\left\|\theta_j\right\|^2\\ &+\sum_{i\in N_j}\left(\mu_{ij}(t)-\mu_{ji}(t)\right)^{\mathrm{T}}\theta_j+\sum_{i\in N_j}\rho\left\|\theta_i(t)-\theta_j\right\|^2\Bigg\}.\end{aligned} \tag{19}$$

Algorithm 1: DMT-SCN

Input: $\mathcal{D}_j = \{(\boldsymbol{x}_{n,j}, \boldsymbol{t}_{n,j})\}_{n=1}^{N_j}, j \in \mathcal{J}$.
Output: θ_j.

Initialize $e_{0,j} = \left[\boldsymbol{t}_{1,j}, \boldsymbol{t}_{2,j}, \ldots, \boldsymbol{t}_{N_j,j}\right]^\top, L = 1, r \in (0,1), \boldsymbol{\Omega} = [\,], \boldsymbol{\Xi} = [\,]$.
while $L \le L_{\max}$ *and* $\|e_0\| \ge \epsilon$ **do**
  for $\lambda \in \Upsilon$ **do**
    for $t = 1$ *to* $T_{\max}$ **do**
      Randomly set $\boldsymbol{\beta} \in [-\lambda, \lambda]^D$ and $b \in [-\lambda, \lambda]$, respectively.
      Set Count=0; **for** *each agent* $j \in \mathcal{J}$ **do**
        Calculate $h_L(\mathbf{X}_j), \xi_{L,q}(\mathbf{X}_j)$ based on (2) and (5), and $\mu_L = (1-r)/(L+1)$.
        if $\min_{q\in\{1,\ldots,C\}}\{\xi_{L,q}(\mathbf{X}_j)\} \ge 0$ **then**
          Count = Count+1.
      if *Count*==J **then**
        Save $\boldsymbol{\beta}$ and b in $\boldsymbol{\Omega}$ and $\xi_L(\mathbf{X}_j) = \sum_{q=1}^{C} \xi_{L,q}(\mathbf{X}_j)$ in $\boldsymbol{\Xi}$.
      else
        go back to Step 3.
      if Ω *is not empty* **then**
        Find $[\boldsymbol{\beta}_L^*, b_L^*] = \underset{\boldsymbol{\beta}, b\in\boldsymbol{\Omega}}{\operatorname{argmax}}\{\xi_L \in \boldsymbol{\Xi}\}$ and Break (go to Step 18).
      else
        Arbitrarily select $\tau \in (0, 1-r)$, renew $r := r + \tau$, return to Step 3.
  /* Distributed evaluation of output weights */
  Calculate h_L^* according to (2) with $\boldsymbol{\beta}^*$ and b^*.
  Obtain $\mathbf{H}_j := \mathbf{H}_L(\mathbf{X}_j)$.
  Optimize $\boldsymbol{\theta}_j^*$.
  Calculate $e_{L,j} = (\mathbf{H}_j\boldsymbol{\theta}_j^*) - \mathbf{T}_j$
  Renew $e_{0,j} := e_{L,j}$.
  $L := L + 1$
return $\boldsymbol{\theta}_j$

We can obtain the solution to problems (10) in component form. The distributed iterations for solving problem (19) are

$$\begin{aligned}\boldsymbol{\theta}_j(t+1) = &\left(\mathrm{H}_j^{\mathrm{T}}\mathrm{H}_j + \tilde{\mathrm{H}}_j\hat{\mathrm{H}}_j + \left(\frac{R}{J} + 2\rho N_i\right)I_L\right)^{-1} \\ &\times\left(\mathrm{H}_j^{\mathrm{T}}\mathrm{T}_j + \tilde{\mathrm{H}}_j^{\mathrm{T}}\hat{\mathrm{H}}_j\omega_j + 2\rho\sum_{i\in N_j}\theta_i(t) - \sum_{i\in N_j}\left(\mu_{ij}(t) - \mu_{ji}(t)\right)\right)\end{aligned} \tag{20}$$

$$\mu_{ij}(t+1) = \mu_{ij}(t) - \rho\left(\theta_i(t) - \theta_j(t+1)\right), \tag{21}$$

$$\omega_j(t+1) = \alpha\omega_j(t) + (1-\alpha)\theta_j(t+1),\ j \in \mathcal{J}, i \in \mathcal{N}_i. \tag{22}$$

At the beginning of iteration, the model parameters are randomly selected. In the t-th round of iteration, the agent receives parameters from neighboring agents and also passes out its own t-th round parameters. The exchange process yields the parameters of the student model for the current round. The parameters of the teacher model for round t are obtained through (24). In this cycle, the parameters of the student model will eventually converge, allowing decentralized learning to achieve a unified model.

Figure 1 shows the steps in the process of generating the network structure of DMT-SCN more clearly and effectively in the form of a flowchart.

Algorithm 2: Decentralized solution of mean teacher method by ADMM.

Input: $\mathbf{H}_j, \mathbf{T}_j$ generated for the local dataset $\mathcal{D}_j^l = \left\{\left(\boldsymbol{x}_{n,j}^l, \boldsymbol{t}_{n,j}^l\right)\right\}_{n=1}^{N_j^l}$, $\mathcal{D}_j^{ua1} = \left\{\left(\boldsymbol{x}_{n,j}^{ua1}\right)\right\}_{n=1}^{N_j^{ua1}}$, $\mathcal{D}_j^{ua2} = \left\{\left(\boldsymbol{x}_{n,j}^{ua2}\right)\right\}_{n=1}^{N_j^{ua2}}$ using the cooperative configuration scheme.

Output: $\boldsymbol{\theta}_j(t)$

Each agent j chooses arbitrary initial values $\boldsymbol{\theta}_j^0, \boldsymbol{\mu}_{ji}^0, \boldsymbol{j} \in \mathcal{J}, i \in \mathcal{N}_j$.

repeat

  Receive $\boldsymbol{\theta_i}(t)$ from and transmit $\boldsymbol{\mu}_{ji}(t)$ to its neighbors $i \in \mathcal{N}_j$ simultaneously.

  Update $\boldsymbol{\theta}_j(t+1)$ as (20).

  For $\forall i \in \mathcal{N}_j$, update $\boldsymbol{\mu_{ji}}(t+1)$ as (21).

  Receive $\boldsymbol{\mu_{ji}}(t+1)$ from its neighbors $\boldsymbol{i} \in \mathcal{N}_j$.

  Update $\boldsymbol{\omega}_j(t+1)$ as (22).

until $t > T_{\max}$;

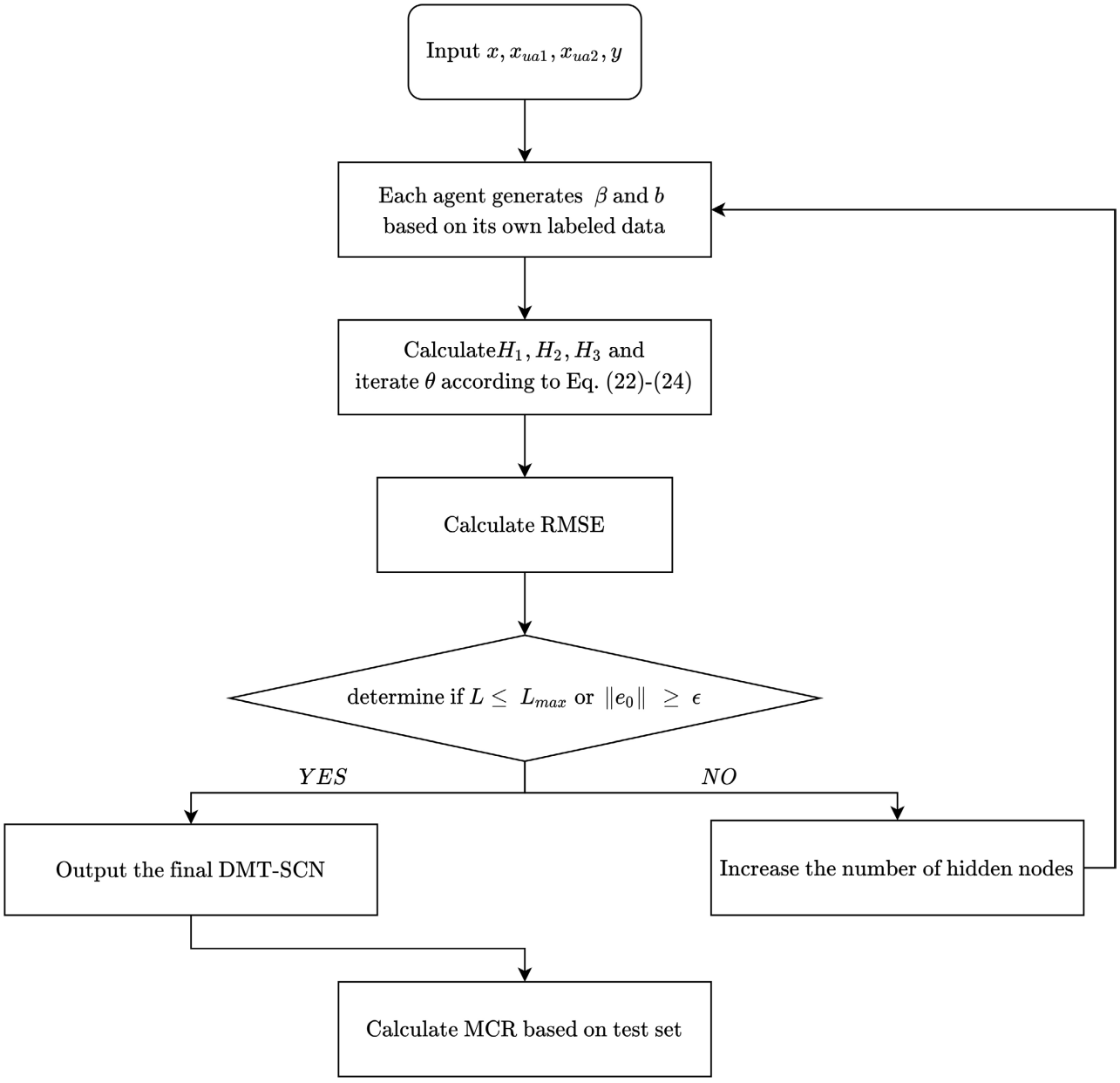


Figure 1. The flowchart of DMT-SCN.

4. Numerical Experiments

Based on the centralised supervised SCN, we transform in addition to the centralised semi-supervised SCN, whose value of beta is derived from both unlabelled and labelled data as follows:

$$\beta = \left(H_1^{-1} \cdot y + \left(H_2 - H_3\right)^{-1} \cdot 0\right) \Big/ 2. \tag{23}$$

Subsequently, we obtain the classification effects of fully supervised SCN and semi-supervised SCN on the three datasets as shown in Table 1. It is observed that incorporating unlabelled data into the iterative operation of SCN in a similar way to labelled data, directly by inverse, does not provide excellent results as expected, but on the contrary, it may even cause negative effects. Therefore, it is more important to develop semi-supervised loss functions that incorporate unlabelled data into the loss in a reasonable manner, which is exactly the work of this paper.

Obviously, DMT-SCN incorporates unlabeled data into model training, which is different from distributed SCN that only utilizes labeled data. Therefore, in this section, we will compare the test accuracies of the two algorithms with different number of agendas to demonstrate that DMT-SCN can effectively improve the data classification accuracy and is robust to different simulation environments. In the next simulation experiments, we consider the cases of 4, 8 and 12 number of agents for each dataset. The communication topologies are shown in Figure 2. We have chosen three datasets for our experiments, all of which are frequently used on single-layer neural networks. Considering that SCNs have been more often used in the past for processing and analysing structured data, such data are similarly chosen in this paper.

4.1. Classification on Hill-Valley Dataset

There are 600 data items in the dataset, each representing 100 points on a two-dimensional graph. When the data is plotted sequentially (from 1 to 100) in Y-coordinate, a pattern of hills ("bumps" in terrain) or valleys ("slopes" in terrain) is created. Since the data is inherently smooth, we applied two different types of noise to each piece of data, a Gaussian noise with $\mu_i^{noi} = \mu\left(x_i\right)$ and $\delta_i^{noi} = \delta\left(x_i\right)/10$. The other is impulse noise with 1/100th of the number of elements, appearing at random locations and with intensity $\mu\left(x_i\right)$. In the other two datasets, we similarly used this data enhancement method to ensure the applicability of this noise on all types of datasets.

Table 1. Classification accuracy of centralised SCN with different datasets.

Model	Hill-Valley	Cancer	Vehicle
Supervised SCN	0.93536	0.87431	0.74675
Semi-Supervised SCN	0.85439	0.90617	0.74327

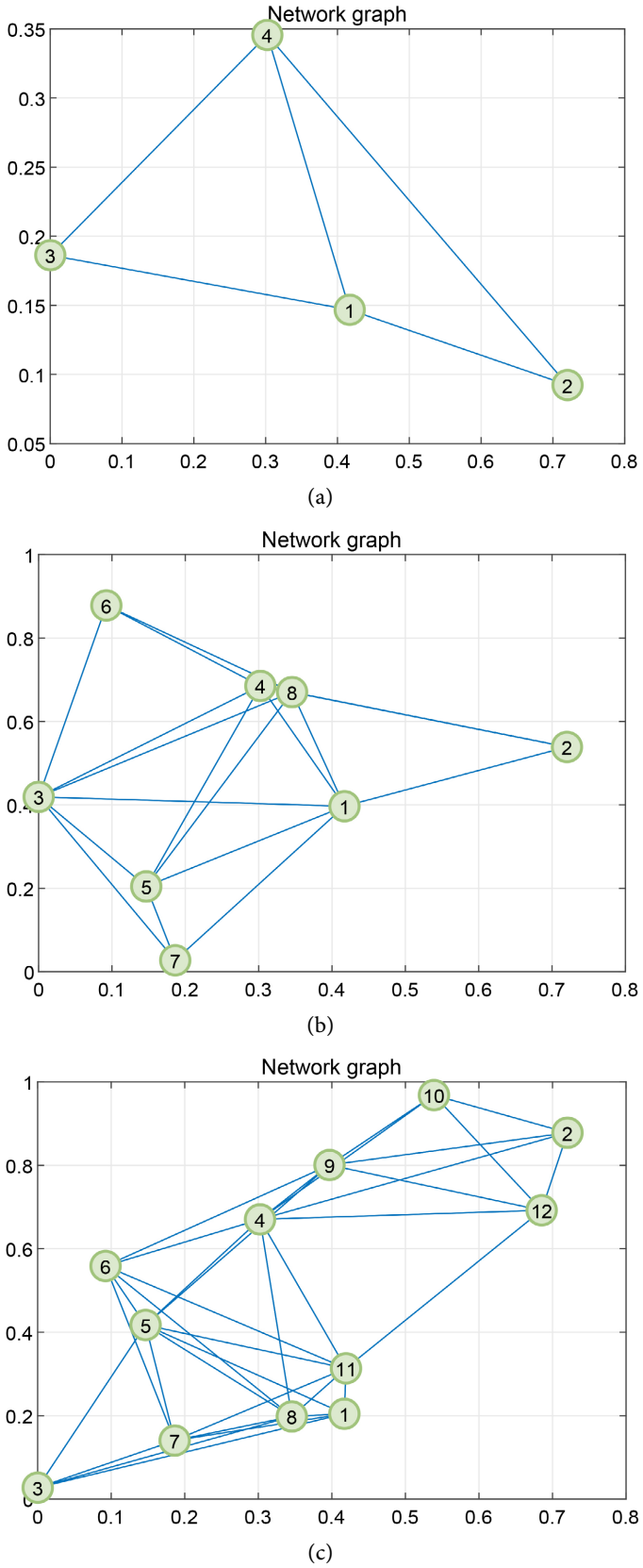


Figure 2. The graph structure composed of agents. (a) 4 agents; (b) 8 agents; (c) 12 agents.

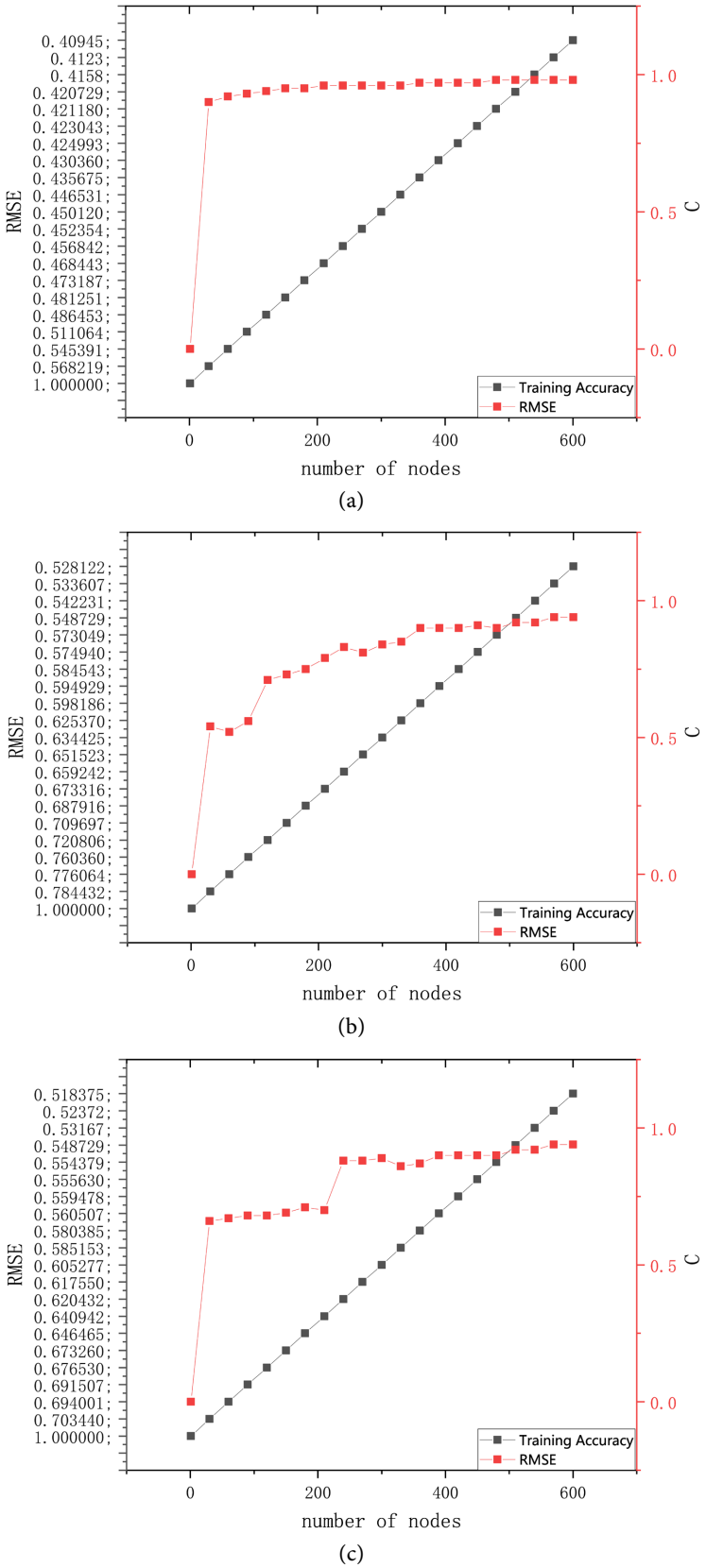
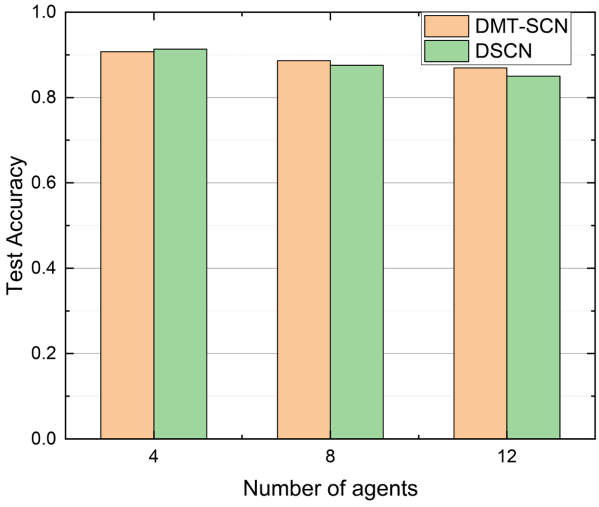
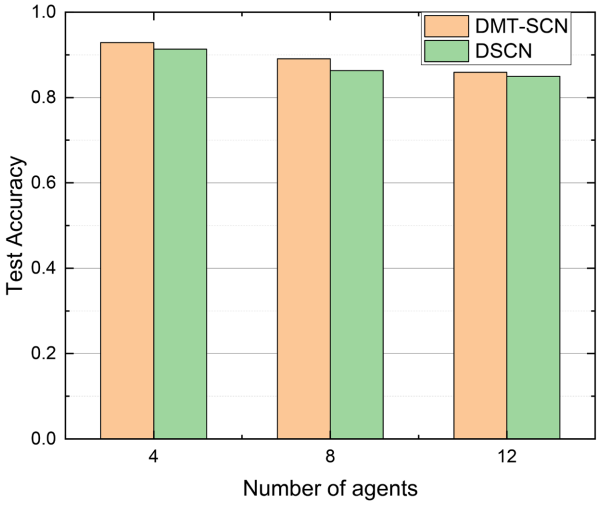


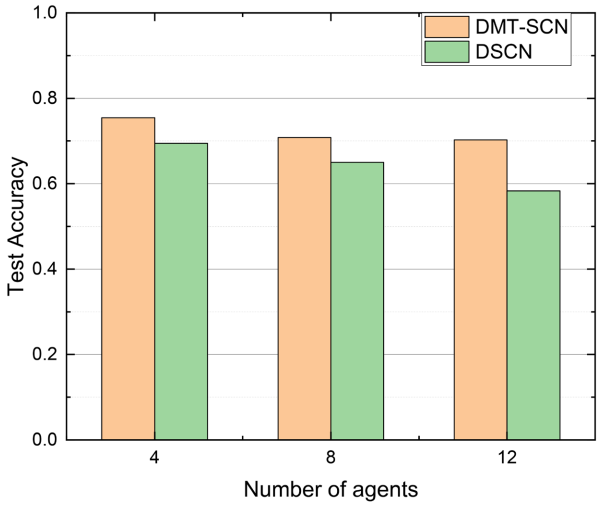
Figure 3. RMSE and test accuracy for Hill-Valley dataset with different number of agents. (a) Hill-Valley (4 agents); (b) Hill-Valley (8 agents); (c) Hill-Valley (12 agents).



(a)



(b)



(c)

Figure 4. Classification performance of three datasets on two models. (a) Hill-Valley; (b) Cancer; (c) Vehicle.

Figure 3 shows the RMSE and classification correctness of the data under the number of 4, 8, and 12 agents, and it can be seen that the RMSE of the data shows a continuous decline, and eventually stabilizes without repeated fluctuations. While the classification correct rate rises rapidly when the number of agents is small, it takes more time to achieve high accuracy as the number of agents increases. Figure 4(a) shows the performance of DMT-SCN and DSCN on the Hill-Valley dataset. As the number of agents increases, the classification advantage of DMT-SCN is gradually highlighted. The classification accuracy of DSCN is 0.5% higher than that of DMT-SCN when the number of agents is 4, but the accuracy of DMT-SCN is higher than that of DSCN when the number of agents is both 8 and 12.

4.2. Classification on Cancer Dataset

This dataset covers tumour characteristic measurements taken from breast cancer patients with associated labels of whether the tumour is benign or malignant. Specifically, this dataset contains nine digitised features describing different measurement dimensions of breast tumours, such as tumour radius size, surface texture, and shape symmetry. Overall, this dataset contains 683 samples, of which 444 were labelled as benign, while the remaining 239 were classified as malignant.

Here, we applied Gaussian noise and impulse noise to the unlabeled data. Figure 4(b) demonstrates the classification performance of our proposed DMT-SCN model and DSCN on this sample, and it can be seen that DMT-SCN does not show significant performance degradation due to the imbalanced class distribution of the data, on the contrary, it outperforms DSCN for different number of agents.

4.3. Classification on Vehicle Dataset

The Vehicle data is the most complex, which is processed from image data, and the 19 features extracted include data on scale variance, skewness, and kurtosis about the primary/secondary axes, and the classification includes four types of vehicles. The dataset has a total of 846 data sets. The performance of this dataset can be seen in Figure 4(c), where we find that the advantage of the classification accuracy of DMT-SCN is more obvious as the complexity of the dataset increases. In the simplest Hill-Valley dataset, there is at most a 2% difference in the correctness of DMT-SCN classification, while in Vehicle, the difference in accuracy between the two reaches 11%.

5. Conclusions

In this paper, we design a fully decentralized algorithm DMT-SCN based on the mean teacher. Combining with ADMM, the global problem is effectively disassembled, and a semi-supervised decentralized learning algorithm is proposed. From this, the problem of the final optimal output weights of DMT-SCN is

solved. The algorithm combines the idea of consistency regularization, introduces teacher and student models to improve the accuracy of prediction, and effectively protects the data privacy of each agent. Finally, the effectiveness of the algorithm is verified using three datasets. The algorithm proposed in this paper shows significant results in improving the classification performance of decentralized supervised learning algorithms, and at the same time, validation against the benchmark model on different numbers of agents and different datasets confirms that the model has a good generalisation performance.

We have observed that DMT-SCN requires a longer time to accomplish its objectives when contrasted with centralized models and fully supervised decentralized algorithms. Therefore, we are considering the implementation of an event-driven mechanism, wherein information is transmitted solely upon meeting specific conditions. The intention behind this approach is to enhance efficiency by conserving time and reducing communication overheads more effectively. Furthermore, in the simulation experiments, although each intelligence operates in a safe and robust environment from external attacks, we must recognise that in real application scenarios, intelligence may be subject to sudden attacks. Therefore, how to ensure the robustness of the intelligence in such situations will be an important topic for our future research.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (No. 62166013), the Natural Science Foundation of Guangxi (No. 2022GXNSFAA035499) and the Foundation of Guilin University of Technology (No. GLUTQD2007029).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Gupta, D. and Rani, R. (2019) A Study of Big Data Evolution and Research Challenges. *Journal of Information Science*, **45**, 322-340. https://doi.org/10.1177/0165551518789880

[2] Phong, L.T., Aono, Y., Hayashi, T., Wang, L. and Moriai, S. (2018) Privacy Preserving Deep Learning via Additively Homomorphic Encryption. *IEEE Transactions on Information Forensics and Security*, **13**, 1333-1345. https://doi.org/10.1109/TIFS.2017.2787987

[3] Wolff, R., Bhaduri, K. and Kargupta, H. (2008) A Generic Local Algorithm for Mining Data Streams in Large Distributed Systems. *IEEE Transactions on Knowledge and Data Engineering*, **21**, 465-478. https://doi.org/10.1109/TKDE.2008.169

[4] Qin, J., Fu, W., Gao, H. and Zheng, W.X. (2016) Distributed k-Means Algorithm and Fuzzy C-Means Algorithm for Sensor Networks Based on Multiagent Consensus Theory. IEEE Transactions on Cybernetics, **47**, 772-783. https://doi.org/10.1109/TCYB.2016.2526683

[5] Zhang, E., Seiler, S., Chen, M., Lu, W. and Gu, X. (2020) Birads Features Oriented Semi-Supervised Deep Learning for Breast Ultrasound Computer-Aided Diagnosis. *Physics in Medicine & Biology*, **65**, Article 125005. https://doi.org/10.1088/1361-6560/ab7e7d

[6] Chen, X., Ren, B., Chen, M., Wang, Q., Zhang, L. and Yan, G. (2016) Nllss: Predicting Synergistic Drug Combinations Based on Semi-Supervised Learning. *PLoS Computational Biology*, **12**, e1004975. https://doi.org/10.1371/journal.pcbi.1004975

[7] Zhao, H., Zheng, J., Deng, W., *et al.* (2020) Semi-Supervised Broad Learning System Based on Manifold Regularization and Broad Network. *IEEE Transactions on Circuits and Systems I: Regular Papers*, **67**, 983-994. https://doi.org/10.1109/TCSI.2019.2959886

[8] Van Engelen, J.E. and Hoos, H.H. (2020) A Survey on Semi-Supervised Learning. *Machine Learning*, **109**, 373-440. https://doi.org/10.1007/s10994-019-05855-6

[9] Scardapane, S., Fierimonte, R., Di Lorenzo, P., Panella, M. and Uncini, A. (2016) Distributed Semi-Supervised Support Vector Machines. *Neural Networks*, **80**, 43-52. https://doi.org/10.1016/j.neunet.2016.04.007

[10] Fierimonte, R., Scardapane, S., Uncini, A. and Panella, M. (2016) Fully Decentralized Semi-Supervised Learning via Privacy-Preserving Matrix Completion. *IEEE Transactions on Neural Networks and Learning Systems*, **28**, 2699-2711. https://doi.org/10.1109/TNNLS.2016.2597444

[11] Xie, J., Liu, S.-Y. and Chen, J.-X. (2022) A Framework for Distributed Semi-Supervised Learning Using Single-Layer Feed forward Networks. *Machine Intelligence Research*, **19**, 63-74. https://doi.org/10.1007/s11633-022-1315-6

[12] Wang, D. and Li, M. (2017) Stochastic Configuration Networks: Fundamentals and Algorithms. *IEEE Transactions on Cybernetics*, **47**, 3466-3479. https://doi.org/10.1109/TCYB.2017.2734043

[13] Tarvainen, A. and Valpola, H. (2017) Mean Teachers Are Better Role Models: Weight-Averaged Consistency Targets Improve Semi-Supervised Deep Learning Results. *Advances in Neural Information Processing Systems*, **30**, 1195-1204. https://doi.org/10.48550/arXiv.1703.01780

[14] Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., *et al.* (2011) Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends® in Machine learning*, **3**, 1-122. https://doi.org/10.1561/2200000016