

## Overmind: A Collaborative Decentralized Machine Learning Framework

Puttakul Sakul-Ung<sup>1,\*</sup>, Amornvit Vatcharaphruksadee<sup>2</sup>, Pitiporn Ruchanawet<sup>3</sup>, Kanin Kearpimy<sup>1</sup>, Hathairat Ketmaneechairat<sup>1</sup>, Maleerat Maliyaem<sup>1</sup>

<sup>1</sup>King Mongkut's University of Technology North Bangkok Bangkok, 10800, Thailand

<sup>2</sup>North Bangkok University Bangkok, 12130, Thailand

<sup>3</sup>King Mongkut's University of Technology Thonburi Bangkok, 10140, Thailand

### ARTICLE INFO

Article history:

Received: 29 August, 2020

Accepted: 01 November, 2020

Online: 10 November, 2020

Keywords:

Decentralization

Machine Learning

Distributed Artificial Intelligence

Feature Transfer

PM2.5 Prediction

Collaborative Machine Learning

### ABSTRACT

*This paper is an extension of work originally presented in PM2.5 Prediction-based Weather Forecast Information and Missingness Challenges: A Case Study Industrial and Metropolis Areas, which focused on imputation algorithm to solve missingness challenge and demonstrated a basic prediction system to prove the proposed algorithm, II-MPA. Distributed and decentralized systems, recently, have been proven for their effectiveness in multiple perspectives. This paper introduces "Overmind", the solution that governs and builds the network of decentralized machine learning as a prediction framework named after its functionality: it aims to discover a set of data and associated attributes for assigning machine learnings in the collaborative decentralized manner. Overmind also empowers feature transfer learning with data preservation. It demonstrates how discovered features are transferred and shared among synergic agents in the network. This model is tested and evaluated the accuracy against the traditional single machine learning prediction model in the original work. The results are satisfactory in both prediction performance and transfer learning.*

## 1. Introduction

This paper is an extension of work in [1]. Traditionally, a single machine learning is built to process upon a dataset; this can be served as a recommendation, prediction, classification, etc. Unfortunately, the drawback of the traditional model is realized where machine learning problems and algorithms are far enough that all scenarios can be resolved using a single model [2]. A traditional approach relies on the training process over a whole dataset; data exploration, perhaps, consumes time and resources. Also, the experience in data analytics does not guarantee the effectiveness of the built machine learning systems to specific business scenarios. Moreover, discovering new attributes in real-world scenarios may require a brand-new dataset and retraining process. As business prefers to preserve data as much as possible, this is supported by system design which deals with a new data pattern and the least trade-off and penalty in performance.

\*Corresponding Author: Puttakul Sakul-Ung, King Mongkut's University of Technology North Bangkok Bangkok, Thailand, [puttakul.s@gmail.com](mailto:puttakul.s@gmail.com)

This paper innovates a framework under the "Data Blind" concept – it indicates the knowledge over dataset and scenarios, which are unknown and undiscovered. Hence, the proposed framework performs data analytics for building machine learning models that serve multiple purposes. Overmind, a system conceptual model in distributed artificial intelligence, is named after its theoretical design. Overmind aims for managing dataset, discovering data pattern, and creating collaborative agents, as a complete framework. This paper experiments a demonstration of how it works, and steps, which are taken to gather results and contributions. An interpretation of behaviours and outcomes will be studied and presented in future works.

## 2. Previous works

The concept of Overmind has been forged base on previous machine learning algorithms and the decentralized systems. We also explore related works on feature transfer and selection. Thus, this section illustrates in two aspects, which are Multi-agent and Decentralized System, and Feature Transfer.

### *2.1. Multi-Agent and Decentralized System*

A term of Distributed Artificial Intelligence (DAI) has been defined based on the concept of distributed intelligent agents. Many projects and methodologies have been reported and summarized into categories proposed in [3] as system conceptual and agent conceptual models. The fine examples of DAI contributions were presented in [4–6] in the area of Intelligent Manufacturing System (IMS) since 1990. Then, a decentralized and distributed machine learning works were published in 1993 [7, 8]. The Cooperative Distributed Problem Solving (CDPS) that involves active agents sharing cooperative behaviours and protocols was invented in 1995 [9]. However, the cooperation is working based on broadcasting negotiation between a system and active agents by receiving the message to perform solving collaboratively.

Undeniably, this paper is highly influenced by the concept proposed in years ago. Although, machine learning has been continuously improved its capability and intelligence to the level that can resolve complex problems in the real-world scenario. Today, a decentralized reinforcement learning [10], which is again inspired by the multiagent system, describes problem statements that require large-scale real-world machine learning whereas the completed work focuses on two dimensions - Networked Evolution Strategies (NES), and Bayesian Optimality in the Wisdom of the Crowd (BOWC). NES describes network structure of communication between parallel-processing agents, that speed up learning process. This network structure of communication inspires Overmind's concept to integrate network analysis components and allow communication among decentralized machine learnings. Lastly, BOWC draws attention to the modelling on how humans learn from the influence of each other; this also inspires the Overmind concept for incorporating the knowledge management framework to mimic the collaboration in humans.

Decentralized Clustering in Large Multi-Agent Systems [11] draws a concept of using multi-agents in decentralized clustering, which is one of the major machine learning objectives. To overcome data clustering real-world problems such as data is widely distributed, data sets are volatile, and data instances also cannot be compactly processed, decentralization of multi-agent has proven its better performance against the traditional k-mean algorithm. This reviewed work hints decentralized clustering of a single dataset, perhaps, gives significant contributions in pre-data processing.

Recently in 2019, Feature Distributed Machine Learning (FDML) is studied collaboratively called FDML: A Collaborative Machine Learning Framework for Distributed Features [12]. This work inspires privacy preservation in feature engineering when one machine learning model can acquire prediction power by using additional features from other machine learning without sharing actual features. This works fully introduces the collaborative machine learning framework with high capability in transfer learning. The experiments are conducted using a different algorithm, and the results yield significance in real-world application development.

Nowadays, a concept of distributed and decentralized artificial intelligence has been widely reported, which shows [www.astesj.com](http://www.astesj.com)

upcoming era in machine learning integrated fields of research such as artificial intelligence in distributed ledger technology [13], artificial intelligence as IoT [14, 15], distributed security solution [16, 17], etc. Surprisingly, these integrations are all based on a distributed and decentralized concept.

### *2.2. Transfer Learning*

In this sense, transfer learning refers to a process where discovered knowledge is fed to existing machine learning for reinforcement and adaptation over modelling that intends to re-train for better performance. Furthermore, the communication among models in a network is established for transferring particular knowledge from one machine learning to others [18]

Similarly, work in [19] introduces the transferring knowledge over different classes. This work also emphasizes transfer learning in feature selection for machine learning models using a proposed framework. The results of this work influences possibility of transfer learning in real-world dataset across classes and domains.

In sum, this paper draws conceptual designs based on previous works in two dimensions, which are distributed and collaborative systems, and transfer learning. However, the algorithm and process behind our work are different from the origins due to its' concepts and purposes. Ultimate research questions under our contributions are following: (1) how to establish an end-to-end platform for demonstrating collaborative machine learning models in a dynamic network in order to power up the prediction and (2) how to create a flexible system for transfer learning between models in a network that reflects the reinforcement.

## **3. Methodology**

This work follows a quantitative research method. Firstly, the literature review is performed by limiting the scope of previous works and using keyword-searching. The results of the review are compounded into the conceptual design of the proposed framework, Overmind, which will be described in the next section of this paper. The reviewing of the proposed framework is conducted by researchers and authors. Experiment and testing are conducted on our previous work dataset, Thailand Air Quality Index (AQI) and weather forecast information. The result of applying this proposed framework is studied by comparison against the traditional framework on the same dataset, as presented in sections 5 and 6 of this paper.

## **4. Overmind: A Framework for Decentralized Machine Learnings**

Overmind, a collaborative decentralized machine learning framework, is proposed considering dealing with anonymous dataset before building a machine learning model to serve purposes such as prediction, recommendation and classification. Figure 1 shows the main conceptual design of deploying multiple collaborative machine learning models ("agents") each of which is responsible for a specific cluster of data. Collaboration means the agents can communicate and exchange knowledge, and share the workload for data processing.

This paper proposes Overmind as a collection of functional processes which are:

1. Determine Data Clustering

2. Assign Agent
3. Establish Network
4. Transfer Learning

In the following section, the aforementioned functional processes are explained in the respective order.

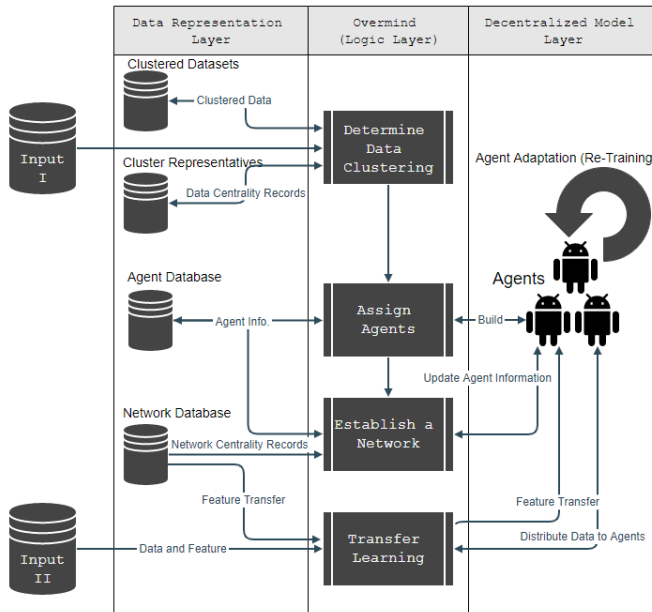


Figure 1: Overmind Architecture

#### 4.1. Determine Data Clustering

Firstly, just the traditional approach, Overmind mimics the data cleansing process by using unsupervised learning algorithms to divide the dataset into clusters. This can be demonstrated as follows:

1. Let  $D$  be the training dataset where  $D = \{d_1, d_2, d_3, \dots, d_n\}$ , given that  $n$  is the number of instances.
2. Let  $F$  be the feature set, where  $F = \{f_1, f_2, f_3, \dots, f_z\}$ , given that  $z$  is the number of features in  $D$ .
3. Overmind deploys unsupervised learning method to divide  $D$  into clusters, given  $C$  is a set of clusters where  $C = \{c_1, c_2, c_3, \dots, c_k\}$  and members in  $D$  are allocated and distributed over members of  $C$ , and  $k$  is the number of clusters.
4. The framework proposes  $s$  as centrality node of similarity graph of data under  $c_i$ . Hence, each cluster contains centrality node of data which is represented as  $s_i$ , where  $i \leq k$

Figure 2 illustrates the output of determining data clustering and assigning agent.

#### 4.2. Assign Agent

Agents are the supervised machine learning models trained using data in each cluster. Overmind determines the number of agents and builds a supervised machine learning algorithm to create agents, which can be explained as follows:

1. Let  $A$  be a set of agents trained by using  $C$ , given that  $a_i$  is trained by using data under  $c_i$  where  $i \leq k$
2. Hence,  $a_i$  will be agent based on  $z$  features, given each feature contains weight/coefficient for particular  $a_i$ . Let  $W_i$

be a set of weight/coefficient of features for particular agent  $a_i$

#### 4.3. Establish Network

To build a collaborative framework, there shall be linkages between agents and rules for cooperation. This paper introduces a similarity network of agents based on the graph as below:

1. Let  $G$  be a graph of pairs  $(V(G), E(G))$  where  $V(G) = A$ .
2. Similarity of agents in  $A$  is measured by  $W_i$  of particular  $a_i$ , given that  $y$  is a pre-determined threshold of similarity.
3.  $E(G)$  will be created and linked between agent in  $V(G)$  where the similarity measurement of a pair of agents satisfies  $y$ . This results in a network of agents.
4. Denote the centrality of  $G$  by  $H_G$ . If the centrality of  $G$  corresponds to multiple agents, Overmind selects one.

Eventually, Overmind creates decentralized machine learning as a collection of nodes in a network by using similarity measurement. Each node represents a supervised machine learning model.

assessed in term of similarity, which can be advanced for future works when one would like to measure affinity among trained models.

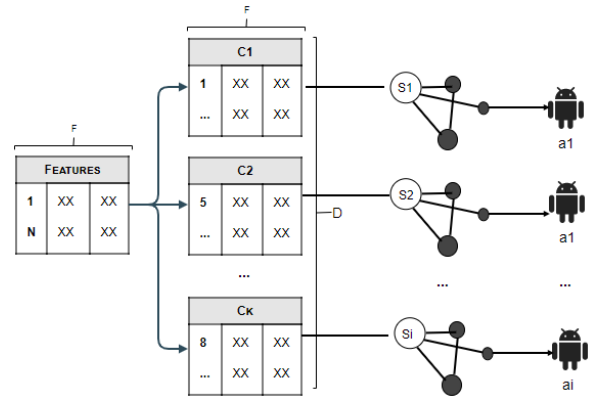


Figure 2: Illustration of Determining Data Clustering and Assigning Agent

#### 4.4. Transfer Learning

This section describes two major transfer learnings of Overmind – data and feature transfers.

##### 4.4.1 Data Transfer

Data transfer is involved when new data arrive for processing. Assumption on testing scenario when  $d_{n+1}$  is discovered and fed to framework, Overmind determines a suitable  $a_i$  for  $d_{n+1}$  by using  $s_i$ . This process implies similarity measurement on the centrality of clusters and new data instances. The steps are taken as below:

1. Let  $D'$  be a test dataset of this framework. Hence,  $\{d'_1, d'_2, d'_3, \dots, d'_q\} = D'$ .
2. Overmind determines each  $d'_x$  in  $D'$  and measures the similarity of  $d'_x$  to  $s_i$ .
3. Overmind assigns  $d'_x$  to  $a_i$  which has the highest similarity measured between  $s_i$  and  $d'_x$ . This step can be considered as

a batch process where  $D'$  is divided into subsets for feeding to particular  $a_i$ .

4. The accuracy of framework is measured as average accuracy of  $d'_x$  or average accuracy of subsets in  $D'$ .

#### 4.4.2 Feature Transfer

Features and Feature Transfer are challenging components in designing the machine learning model. A traditional approach may require retraining of model when dealing with newly discovered features. A retaining process takes resources and time consuming for a big dataset. Yet, in the real-world scenario, some business may obtain a set of new information, both data and features, whether it contributes significances or not, but they have no willing to dispose of an existing dataset or pre-trained model.

In this sense, feature transfer refers to the ability that the existing trained model determines and deploy the discovered new features without retraining model; this means the transfer process will occur in every agent in the network. To elaborate flexibility in a dynamic machine learning model which accepts changes in both dataset and features, Overmind is conceptually designed to overcome the aforementioned challenges by described steps below:

1. When  $f_{z+1}$  is discovered within a new dataset  $D_{new}$ , Overmind predetermines a need of  $f_{z+1}$  by mimicking the supervised machine learning model of  $H_G$  and process upon  $D_{new}$  along with  $f_{z+1}$ . However, to avoid data volume bias, Overmind will proportionally random data in each  $c_i$  to form a dataset as equal as  $D_{new}$ .
2. Let accuracy of forming dataset is denoted by  $acc_{ext}$  and accuracy of model upon  $D_{new}$  is denoted by  $acc_{new}$ .
3. If  $acc_{ext} > acc_{new}$ , Overmind drops  $f_{z+1}$  from  $D_{new}$  and takes as data transfer process. This indicates that  $f_{z+1}$  cannot improve performance of existing system.
4. If  $acc_{ext} \leq acc_{new}$ , Overmind considers  $f_{z+1}$  as significant feature and process further steps for feature transfer process which is described below:
  - 4.1. Where the existing dataset lacks of  $f_{z+1}$  true values, Overmind creates imputation model by changing target label to  $f_{z+1}$  using forming dataset. This imputation model is denoted by  $a_{imp}$ .
  - 4.2. Overmind inserts  $f_{z+1}$  for each  $c_i$  in  $C$ , denoted by  $c'_i$ . This may result in extreme missing values in  $f_{z+1}$ . Hence,  $a_{imp}$  is used for imputation of missingness.
  - 4.3. Starting from  $H_G$ , agent  $a_i$  retrains its model along with  $f_{z+1}$ , the retrained agents are denoted by  $a'_i$ .
  - 4.4. Retrained agents  $a'_i$  perform cross-validation and compare a result against  $a_i$ . There are two possible outcomes
    - $a'_i$  gives a better performance, then  $f_{z+1}$  is accepted.  $a'_i$  adjusts  $W_i$ , denoted by  $W'_i$ , or else
    - $a_i$  gives a better performance, then  $f_{z+1}$  is rejected.  $a'_i$  flags weight/coefficient of  $f_{z+1}$  as zero in  $W'_i$
5. Overmind reshapes a network of agents, this creates  $G' = (V(G), E(G'))$ . Perhaps, centrality of network may be

different from  $G$ , Overmind updates the new centrality, denoted by  $H_{G'}$

Figure 3 shows the establishing of network of agents and transferring learning that impacts changes over a network.

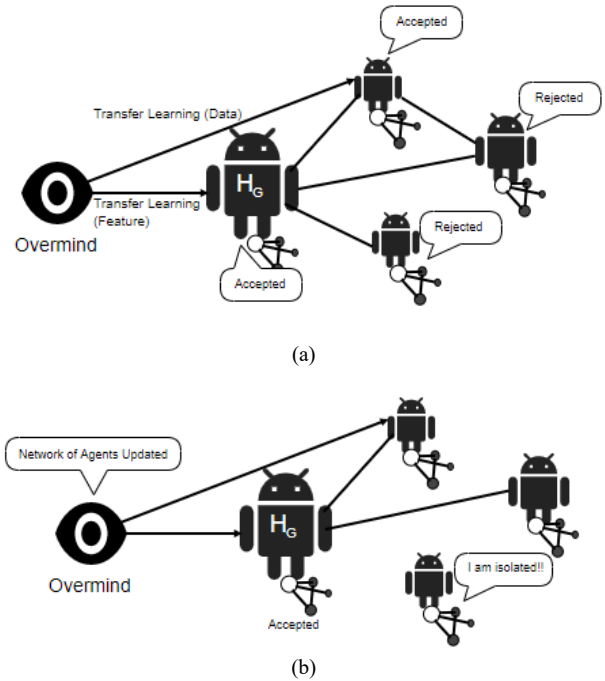


Figure 3: (a) Illustrating an established network with transfer learnings for both data and feature (b) Demonstrating a reshaped network resulted from transfer learning process.

Until now, a proposed framework creates a dynamic network which illustrates the transfer process among decentralized agents. Hence, this is called the collaborative decentralized machine learning framework.

#### 5. Performance evaluation: a case study Thailand PM2.5 and weather forecast information database

To evaluate framework performance, this paper simulates the framework over the testing dataset, which is Thailand PM2.5 and weather forecast dataset presented in [1]. This dataset contains 2,383 instances with 21 features (including target). Overmind's simulation takes place in two functions, Framework Performance and Demonstration on Transfer Learning for Feature Transfer, to predict PM2.5 values from a given set of features.

##### 5.1. Framework Performance

This paper deploys 80% of dataset for training and 20% for testing, for both traditional machining learning model –a single model –and Overmind.

In the training process, five uncorrelated features are removed from the dataset. Two algorithms, Gradient Boosted Trees (GBT) and Neural Net (NN), are used in this paper because the previous work has described Thailand PM.5 prediction using these algorithms. The set of tuning parameters is left by default using RapidMiner 9.6. For GBT, number of trees = 50 with learning rate = 0.01. For NN, training cycle = 200 with learning rate = 0.01. In addition, 10 folds cross-validation with automatic sampling type is used.



Throughout this paper, gradient color codes are used to indicate how good the results are. Green-colored fields indicate the higher accuracy and correlated values and red-colored fields indicate the opposite.

The clustering and agents training results show in Table 1

Table 1: Distributed Agents in Training Process

Agents	Train Dataset (0.8, 1906, F=15)		
	Data Instances (N)	GBT (RMSE)	NN (RMSE)
$D: a_s$	1906	13.50	7.21
$c_0: a_0$	1162	6.56	5.25
$c_1: a_1$	256	17.14	9.62
$c_2: a_2$	488	12.28	7.00

s: single traditional model  
 $C = \{c_0, c_1, c_2\}$   
 Distributed Agents,  $A = \{a_0, a_1, a_2\}$

Table 1 proves a better performance, measured by Root Mean Square Error (RMSE), on agents,  $a_0$  and  $a_2$  than on a traditional single model,  $a_s$ , in both algorithms. It may indicate the abnormal data pattern in  $c_1$  which causes variation of performance for a traditional model and assumption of removing these instances are not acceptable by business scenario. In other words, 13.43% of the instances drop performance of agent for from RMSE 6.56 to 13.50, and it seems to be unacceptable tradeoffs.

Establishing a network of agents, we explore the weights of features based on GBT variable importance percentage as shown in Table 2

Table 2: Variable Importance in agents based GBT

Variable Importance	$D: a_s$	$c_0: a_0$	$c_1: a_1$	$c_2: a_2$
AQI	0.762	0.689	0.123	0.282
CO	0.114	0.004	0.683	0.202
PM10	0.075	0.139	0.069	0.112
O3	0.017	0.003	0.041	0.018
PRESSURE	0.009	0.095	0.035	0.002
PRECIP	0.007	0.000	0.013	0.000
HUMIDITY	0.006	0.020	0.019	0.008
CLOUD	0.005	0.018	0.007	0.008
SO2	0.005	0.011	0.000	0.357
NO2	0.000	0.001	0.003	0.004
WIND	0.000	0.011	0.001	0.001
GUST	0.000	0.001	0.001	0.003
TEMP	0.000	0.006	0.005	0.003
TEMP_FEEL	0.000	0.002	0.000	0.000
TIME_POINT	0.000	0.000	0.000	0.000

By examining variable importance in Table 2, both of three highest importance features and lowest importance features were examined giving a clue that the criteria as feature weight on the

prediction of PM2.5 differ for each agent. The data in each cluster should require a specifically trained agent.

Recall, an establishing network of agents based GBT algorithm, Overmind determines the similarity measurement of agents using variable importance. A Euclidean Distance Measurement is used to find the distance among agents. In Table 3, the similarity is measured by using the equation:

$$Similarity(s, d) = 1 - Distance(s, d) \tag{1}$$

where s is source node, d is destination node.

Table 3: Agent Similarity Measurement

Source	Destination	Distance	Similarity
$D: a_s$	$c_0: a_0$	0.17	0.83
$D: a_s$	$c_1: a_1$	0.86	0.14
$D: a_s$	$c_2: a_2$	0.60	0.40
$c_0: a_0$	$c_1: a_1$	0.89	0.11
$c_0: a_0$	$c_2: a_2$	0.58	0.42
$c_1: a_1$	$c_2: a_2$	0.62	0.38

Setting a threshold  $\gamma = 0.3$  means the edges will be created when the similarity between nodes is higher or equal to 0.3. However, this threshold is an adjustable parameter, the lower the threshold the higher chance for connected agents, the similarity of agents affects and shapes the transfer process. Where connection exists, a pair of the agent can communicate and transfer both data and feature, in another hand, isolated agent requires specially and specifically consideration. In this study, 0.3 is represented for the demonstration that creates linkages between agents unless the transfer process cannot be illustrated. Considering a tradition model, an agent in a network will results in Figure 4, however, this network will be examined again in the feature transfer section.

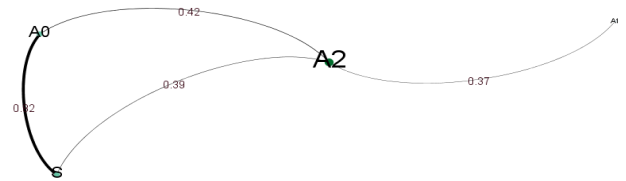


Figure 4: Network of Agents (F=15,  $\gamma=0.3$ )

In the testing process, 477 instances of the dataset were distributed to specific agents based on the data similarity measurement. Moreover, the prediction assessments on both Gradient Boost Tree (GBT) and Neural Network (NN) were shown in Table 4.

The results show high accuracy when distributed agents perform a prediction over the associated clustered dataset. Undeniably, the traditional model is flexible and stable for input data. Perhaps, it should be considered as a tradeoff. We calculate average performance for distributed agents as:

$$Acc = \frac{\sum_i^I RMSE(a_i, c_i)}{i} \tag{2}$$

where  $RMSE(a_i, c_i)$  is an accuracy of prediction by  $a_i$  on dataset  $c_i$

Table 4: Test Results

Agent	Trained By	Test Dataset (0.2, n=477)					
		GBT (RMSE)			NN (RMSE)		
		$c_0$	$c_1$	$c_2$	$c_0$	$c_1$	$c_2$
		294	55	128	294	55	128
$a_s$	477	8.28	22.87	11.25	4.29	8.69	6.99
$a_0$	294	5.80	36.46	18.51	3.99	22.41	12.34
$a_1$	55	38.64	17.04	21.40	12.30	7.80	12.04
$a_2$	128	19.32	22.34	11.00	13.31	15.36	7.23

The GBT has achieved average performance RMSE of 11.28 whereas the traditional model has obtained RMSE of 14.13. On the other hand, the NN achieve the RMSE of 6.34, while the traditional model achieves at 6.65. These performances described in Figure 5.

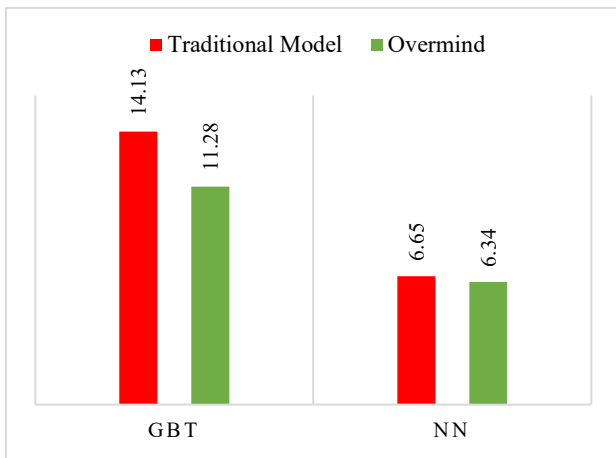


Figure 5: Average Performance, measured in RMSE.

### 5.2. Demonstration on Transfer Learning for Feature Transfer

In this demonstration, one correlated feature from a training dataset is removed, but a testing dataset remains with full features. Based on variable importance in Table 2, AQI is assumedly removed from the training dataset. The results of newly trained agents are shown in Table 5.

Table 5: Distributed Agents in Training Process, |F|=14

Agents	Train Dataset (0.8, 1906,  F =14)		
	Data Instances (n)	GBT (RMSE)	NN (RMSE)
$D: a_s$	1906	14.28	9.97
$c_0: a_0$	553	15.72	11.96
$c_1: a_1$	234	16.20	12.45
$c_2: a_2$	1119	8.17	7.05

The performance of two minor clustered ( $n=553$ , and  $n=234$ ) significantly drops. However, it is not yet comparable results where  $|F|=15$  because the clustering process may yield a different output. Base on this experiment, a same threshold value,  $\gamma = 0.3$  will result isolated  $a_2$  which indicates that the data in  $c_0$  and  $c_1$  should be removed from the demonstration and only one agent required in this scenario, it is  $a_2$ ; the demonstration will be then

bias, hence, we lower a threshold to  $\gamma = 0.2$  and the result of networks of agents shows in Figure 6.

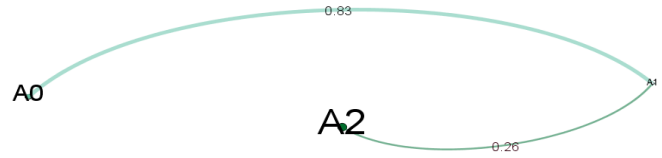


Figure 6: Network of Agents ( $|F|=14, \gamma=0.2$ )

Figure 6 shows that a centrality of network is  $a_1$ , this means centrality of the network is determined neither by agent performance nor numbers of instances. The centrality of the network is responsible for sharing and transferring the discovered both data and feature.

Demonstration on a discovered new dataset and feature ( $n=477, |F|=14+1$ ), assumedly, AQI is a new feature. Overmind firstly determines the need for AQI data if it can improve the performance of the overall agent or not. Overmind takes sampling proportional data from clusters in a network, then building a model to evaluate performance against a new dataset.

Table 6: Comparing Results for a Transfer Learning

Agent	No. of instance (n)	No. of feature ( F )	GBT (RMSE)	NN (RMSE)
Existing Dataset: $acc_{ext}$	477	14	14.56	10.12
New Dataset: $acc_{new}$	477	14+1	12.43	7.90

Table 6 shows a better performance with a new feature, hence, a transfer learning should be accepted for the new feature, AQI.

An existing dataset in a network contains no value of AQI, and the business scenario does preserve existing dataset. Hence, an imputation model is built by using a new dataset for training process ( $n=477, |F|=14+1$ ). Eventually, an existing dataset is imputed for a new feature, AQI; it is deployed to the centrality of the network, all agents accept AQI as a new feature because the comparison identifies AQI can improve performance as shown in Table 7.

Table 7: Distributed Agents in Training Process, |F|=14+1 (imputed AQI)

Agents	Train Dataset (0.8, 2383,  F =14+1 imputed AQI)		
	No. of instances (n)	GBT (RMSE)	NN (RMSE)
$D': a'_s$	1906	13.39	6.85
$c'_0: a'_0$	553	14.19	7.10
$c'_1: a'_1$	234	14.55	9.55
$c'_2: a'_2$	1119	7.03	4.71

Overmind update a network of agents based on changing in similarity measurement among agents.

A re-connected nodes between  $a'_2$  and  $a'_0$  is formed, this results in a fully connected network, and the centrality of the network is shifted to  $a'_2$  because it has the highest performance and controlling a majority of data in the network ( $N=1119$ ). Overmind then continues data transfer process as described in (A). The data prediction results are shown in Figure 7.

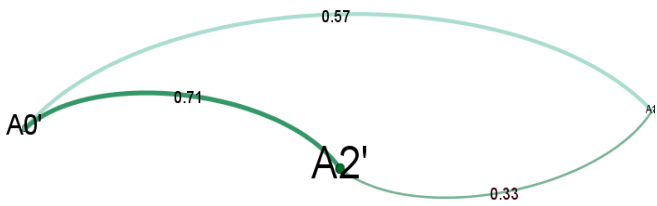


Figure 7: Network of Agents (F=14+1, Y=0.2)

Table 8: Test Result after feature transfer process

Agent	Test Dataset (0.2, 477,  F =14+1)						
	No. of instance	GBT (RMSE)			NN (RMSE)		
		$c_0$	$c_1$	$c_2$	$c_0$	$c_1$	$c_2$
$a'_s$	477	14.44	17.02	11.92	7.11	13.11	6.36
$a'_0$	162	13.25	18.16	17.23	7.07	16.9	11.79
$a'_1$	37	19.73	13.61	25.73	19.81	10.38	15.64
$a'_2$	278	18.39	21.18	13.22	11.64	16.21	8.62

The comparison shows average performance in RMSE using GBT with 13.36 in this testing dataset, while a traditional model would give an accuracy, 14.46. For NN, the average performance in RMSE is 8.69, and 8.86 for a traditional model. This comparison is described as shown in Figure 8.

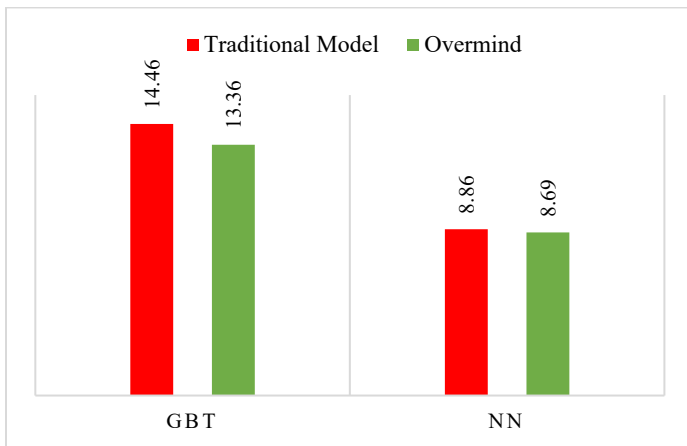


Figure 8: Average Performance, measured in RMSE after feature transfer process

Figure 8 shows penalty in transfer process caused by the imputation model that increases the RMSE after transfer process as shown in Figure 9. However, it should be reminded that demonstrating feature transfer is taken place on the important variable which is the AQI.

To verify the consistency of this approach, the progression of the framework functional processes, Determine Data Clustering, Assign Agent, and Establish Network has been performed. However, in the second round, the dataset is slightly different because (1) 1,906 instances (80%) of training dataset have imputed AQI values, and (2) 477 instances (20%) of the testing dataset have true values of AQI. To avoid confusing, the agents in the second execution is denoted by  $a_{x:2}$  where x is a number of clusters produced by this execution.

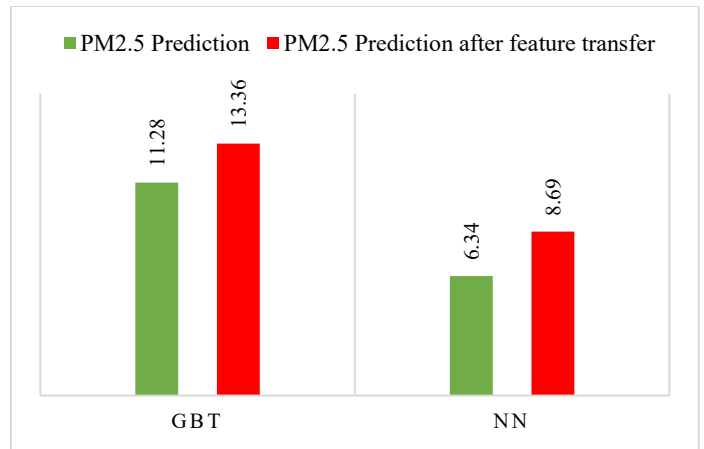


Figure 9: Penalty after feature transfer process

Table 9: Distributed Agents in Training Process by Second Execution

Agents	Train Dataset (1, 2383,  F =15)		
	No. of instances (n)	GBT (RMSE)	NN (RMSE)
$D: a_{s:2}$	2383	13.05	6.95
$c_0: a_{0:2}$	1458	6.24	4.22
$c_1: a_{1:2}$	316	17.11	11.27
$c_2: a_{2:2}$	609	11.90	6.44

s: single traditional model  
 $C = \{c_0, c_1, c_2\}$   
 Distributed Agents,  $A = \{a_{0:2}, a_{1:2}, a_{2:2}\}$

The result in Table 9 shows consistency over the clustering process which still produce three clusters with similar proportions. Moreover, the overall performance of agents is consistency and even slightly improved. Examining agent similarity using variable importance between the first and second executions are shown in Table 10 and Figure 10. The comparison in Figure 10 shows changes in variable importance, but the agents are keeping their characteristics over the same set of features.

Eventually, a network of agents is re-established as shown in Figure 11; a network of agents also preserves its structure where the centrality of network remains unchanged. However, if the similarity threshold is recalled in the very first demonstration of establishing a network where  $y = 0.3$ , the isolated agent ( $a_{1:2}$ ) may occur which is assigned to the minority of data (n=316) that contribute poor performance.

Table 10: Variable Importance in agents based GBT by Second Execution

Variable Importance	$D: a_{s:2}$	$c_0: a_{0:2}$	$c_1: a_{1:2}$	$c_2: a_{2:2}$
AQI	0.762	0.689	0.123	0.282
CO	0.114	0.004	0.683	0.202
PM10	0.075	0.139	0.069	0.112
O3	0.017	0.003	0.041	0.018
PRESSURE	0.009	0.095	0.035	0.002
PRECIP	0.007	0.000	0.013	0.000
HUMIDITY	0.006	0.020	0.019	0.008
CLOUD	0.005	0.018	0.007	0.008

Variable Importance	$D: a_{s:2}$	$c_0: a_{0:2}$	$c_1: a_{1:2}$	$c_2: a_{2:2}$
SO2	0.005	0.011	0.000	0.357
NO2	0.000	0.001	0.003	0.004
WIND	0.000	0.011	0.001	0.001
GUST	0.000	0.001	0.001	0.003
TEMP	0.000	0.006	0.005	0.003
TEMP_FEEL	0.000	0.002	0.000	0.000
TIME_POINT	0.000	0.000	0.000	0.000

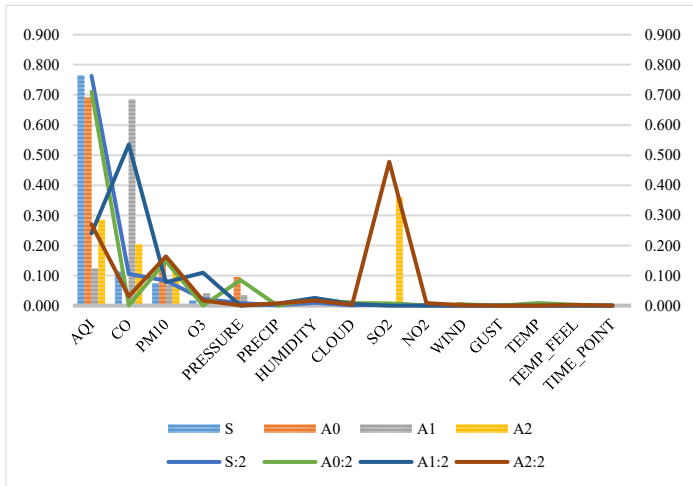


Figure 10: Comparison of Agent Variable Importance based GBT by first and second executions

This scenario implies that the business may consider, (1) a separate studying of agents in order to build a specific algorithm for prediction over the associated dataset, (2) removing associated data instances out of dataset in order to reduce inconsistency and improve prediction performance, (3) reducing a threshold ( $y$ ) in order to keep particular agent connected to a network because this agent can be a specific dataset where a business foresees this data will be increasing over time in soon future.

## 6. Results and Discussion

Till now, demonstrations of both framework performance, and the transfer process are explained. The summary of the performance of Overmind with four comparing scenarios in cross-validation agents' average performance can be concluded as follows:

1. Scenario I: Overmind assigns distributed agents on a fully completed dataset.
2. Scenario II: Overmind assigns distributed agents on a dataset where importance attributes is removed.
3. Scenario III: Overmind assigns distributed agents to demonstrate a situation where a new feature is discovered and transferred, and
4. Scenario IV: Overmind assigns retrained distributed agents and restates its network

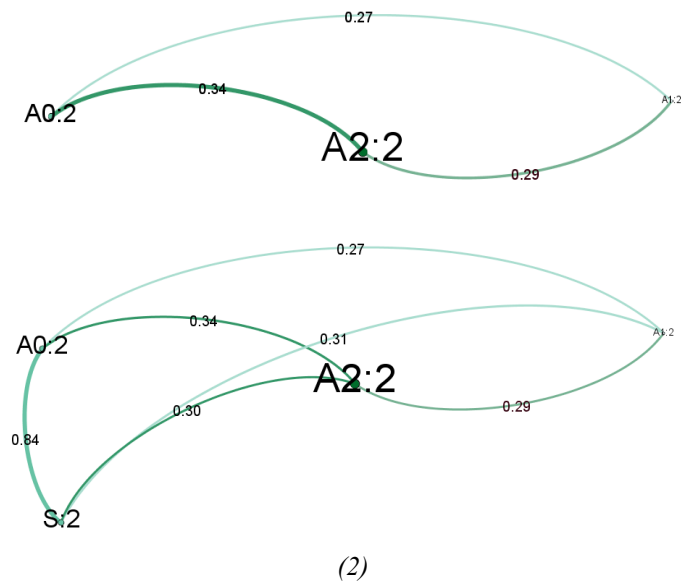


Figure 11: (1) Network of Agents in Second Execution ( $F=15, Y=0.2$ ) without Traditional Model, (2) Network of Agents in Second Execution ( $F=15, Y=0.2$ ) with Traditional Model,

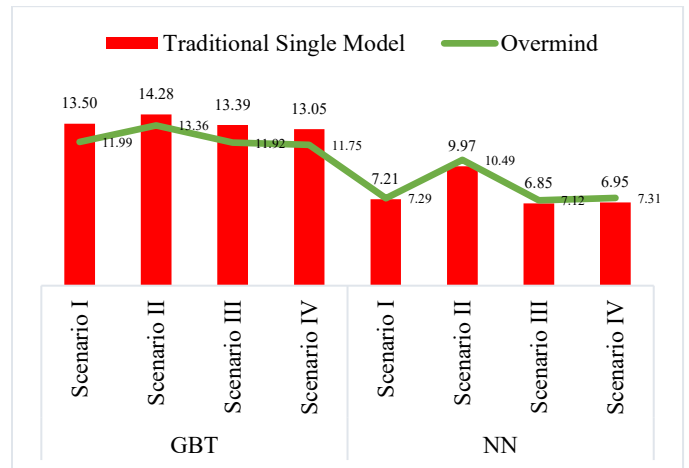


Figure 12: Comparison of Average Performance in Cross-Validation Agent Generated Models measured in RMSE

Figure 12 explains, for all scenarios, the cross-validation performance in Overmind generated models is impressive compared to a traditional single model under GBT algorithm. In the other side, the performance is not satisfied when compared to the traditional model under NN algorithm.

To summarize testing performance in testing data, we divide into two scenarios, (1) Scenario I: Overmind assigns distributed agents on a fully completed dataset, (2) Scenario II: Overmind assigns distributed agents to demonstrate a situation where a new feature is discovered and transferred. The results are shown in Figure 13.

Figure 13 indicates a better testing performance in all scenarios, but the performance is slightly better in the NN algorithm. In sum, Overmind can achieve its primary intended design and concept, this achievement can be demonstrated through our previous dataset for PM2.5 prediction in Thailand.



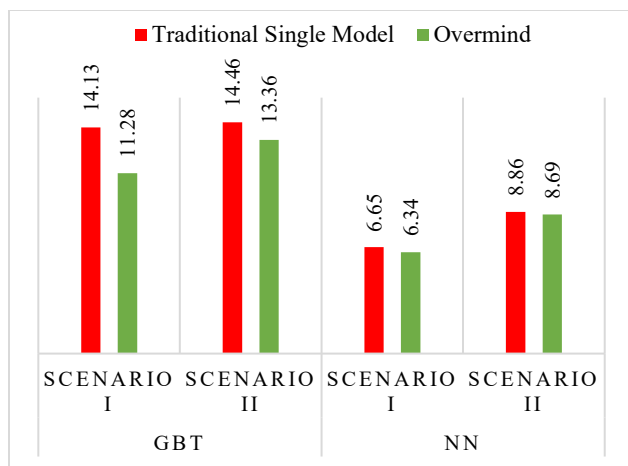


Figure 13: Summary and Comparison on Testing Performance measured in RMSE

## 7. Conclusion

Overmind is a framework aiming to demonstrate and facilitate a decentralized machine learning-based network analysis. This paper proves the concept and its performance through the mixed dataset between Thailand PM2.5 dataset and weather forecast dataset.

In data prediction, this framework can significantly reduce the data bias scenario, where input data is uncontrollable, variant, and clustered. Moreover, Overmind has proven its performance by comparing an average accuracy measured in RMSE on both traditional models (single machine learning) and distributed agents (multiple machine learning, agents). Overmind performs better performance than the traditional method in selected dataset using the GBT algorithm, reducing the RMSE from 14.13 to 11.28 (20.16% decrease). Furthermore, the RMSE is slightly reduced from 6.65 to 6.34 (4.6% decrease) in the NN algorithm.

A feature transfer process is demonstrated as a conceptual design for data preservation. However, the evaluated performance yields penalty due to imputation method.

In sum, Overmind has successfully formed a collaborative decentralized machine learning framework serving as an alternative in the design of a machine learning system. Possible contributions of Overmind are:

### 7.1. Higher Performance and Unaffected by Data Bias.

Overmind proves its effectiveness as a prediction system in the selected testing dataset. The design of Overmind is not affected by data bias because distributed agents are designed to function over associated sub-dataset.

### 7.2. Distributed Resource Utilization

Overmind mimics the concept of a distributed system so that it inherits the advantage where resources are allocated in a decentralized manner. Each agent utilizes computational resources for processing associated input instead of processing the whole dataset. This contribution also involves parallel processing and queue system where particular agents may transfer their queued inputs, as a bottleneck, to similar connected agents in a network for parallel processing where penalty inaccuracy can be traded off.

This parallel processing for Overmind is already considered and planned for future work.

### 7.3. Dynamic Models in Collaborative Manner

Overmind demonstrates dynamic machine learning models for feature discovery and transferring across collaborative network analysis with data preservation. The concept of similarity measurement and network of agents has innovated methods in a dynamic framework. There are still opportunities in exploring the aforementioned methods to study optimization parameters and techniques.

## 8. Future Work

A study on optimization is recommended on Overmind's parameters. An interpretation over a network of agents formed by Overmind is an interesting topic to be explored and described how it contributes significances to business viewpoints.

In addition, the enterprise system architecture of Overmind can be improved and enhanced for higher performance and resilience such as queueing system and redundancy in agent level. For example, when one agent is in abnormal stage, data can be continuously processed by neighbors or connected agents in the network; this will reduce the impact of disruption and error in artificial intelligence system.

## Acknowledgment

We would like to thank the members of Faculty of Information Technology and Digital Innovation, King Mongkut's University of Technology North Bangkok, for their guidance and suggestions. Importantly, we would also like to thank Bangkok Air Quality for supporting data.

## References

- [1] P. Sakul-Ung, P. Ruchanawet, N. Thammabunwarit, A. Vatcharaphruksadee, C. Triperm, M. Sodanil, "PM2.5 Prediction based Weather Forecast Information and Missingness Challenges: A Case Study Industrial and Metropolis Areas," in 2019 Research, Invention, and Innovation Congress (RI2C), 1–5, 2019, doi:10.1109/RI2C48728.2019.8999941.
- [2] K.F. Li, "Engineering Distributed Intelligence: Issues and Challenges," in 2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems, 404–408, 2014, doi:10.1109/CISIS.2014.56.
- [3] L. Eduardo, C. Hern, "On distributed artificial intelligence," *The Knowledge Engineering Review*, **3**(1), 21–57, 1988, doi:10.1017/S026988900004367.
- [4] G.M.P. O'Hare, A.W.J. Chisholm, "Distributed Artificial Intelligence: An Invaluable Technique for the Development of Intelligent Manufacturing Systems," *CIRP Annals*, **39**(1), 485–488, 1990, doi:https://doi.org/10.1016/S0007-8506(07)61102-0.
- [5] B. Chaib-draa, "Industrial applications of distributed AI," **38**(11 %J Commun. ACM), 49–53, 1995, doi:10.1145/219717.219761.
- [6] S. Kim, "Organisational intelligence and distributed AI," No. 40. *Arbeitsberichte des Instituts für Wirtschaftsinformatik, Westfälische Wilhelms-Universität Münster*, 1995.
- [7] Y. Feng-Chao, H. Yu-Kuen, "Design of a distributed artificial intelligence system shell," in Proceedings of TENCON '93. IEEE Region 10 International Conference on Computers, Communications and Automation, 192–195 vol.1, 1993, doi:10.1109/TENCON.1993.319961.
- [8] S. Sen, "Multiagent systems: Milestones and new horizons," *Trends in Cognitive Sciences*, **1**(9), 334–340, 1997, doi:10.1016/S1364-6613(97)01100-5.
- [9] G. Weiß, L. Steels, "Distributed Reinforcement Learning," Springer Berlin Heidelberg, Berlin, Heidelberg: 415–428, 1995.
- [10] D. Adjudah, "Decentralized Reinforcement Learning Inspired by Multiagent Systems," Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, 1729–1730, 2018.

- [11] E. Ogston, B. Overeinder, M. van Steen, F. Brazier, "A method for decentralized clustering in large multi-agent systems," in Proceedings of the second international joint conference on Autonomous agents and multiagent systems - AAMAS '03, ACM Press, New York, New York, USA: 789, 2003, doi:10.1145/860575.860702.
- [12] Y. Hu, D. Niu, J. Yang, S. Zhou, FDML: A Collaborative Machine Learning Framework for Distributed Features, Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2232–2240, 2019, doi:10.1145/3292500.3330765.
- [13] K.D. Pandl, S. Thiebes, M. Schmidt-Kraepelin, A. Sunyaev, "On the Convergence of Artificial Intelligence and Distributed Ledger Technology: A Scoping Review and Future Research Agenda," IEEE Access, **8**, 57075–57095, 2020, doi:10.1109/ACCESS.2020.2981447.
- [14] Y. Lin, Y. Lin, C. Liu, "Altalk: a tutorial to implement AI as IoT devices," IET Networks, **8**(3), 195–202, 2019, doi:10.1049/iet-net.2018.5182.
- [15] X. Lin, J. Li, J. Wu, H. Liang, W. Yang, "Making Knowledge Tradable in Edge-AI Enabled IoT: A Consortium Blockchain-Based Efficient and Incentive Approach," IEEE Transactions on Industrial Informatics, **15**(12), 6367–6378, 2019, doi:10.1109/TII.2019.2917307.
- [16] H.F. Rashvand, K. Salah, J.M.A. Calero, L. Ham, "Distributed security for multi-agent systems - review and applications," IET Information Security, **4**(4), 188–201, 2010, doi:10.1049/iet-ifs.2010.0041.
- [17] X. Zhao, R. Jiang, "Distributed Machine Learning Oriented Data Integrity Verification Scheme in Cloud Computing Environment," IEEE Access, **8**, 26372–26384, 2020, doi:10.1109/ACCESS.2020.2971519.
- [18] S.J. Pan, Q. Yang, "A Survey on Transfer Learning," IEEE Transactions on Knowledge and Data Engineering, **22**(10), 1345–1359, 2010, doi:10.1109/TKDE.2009.191.
- [19] Z. Wang, X. Ye, C. Wang, P.S. Yu, "Feature Selection via Transferring Knowledge Across Different Classes," ACM Trans. Knowl. Discov. Data, **13**(2), 1–29, 2019, doi:10.1145/3314202.