Distributed Monitoring for Energy Infrastructures: A Two-Tier Analysis over Wireless Networks

Chi Zhang, Liangli Zhen, Joey Zhou, and Cen Chen

Abstract

Wireless networks (e.g., 5G networks) enable distributed energy infrastructures to be connected even when they are geometrically isolated. Intelligent monitoring from remote sites therefore becomes possible, allowing decision makers to examine the status of distributed energy infrastructures from a central location. The major challenge is when local devices cannot perform the monitoring independently; transmitting every signal back to the central server triggers enormous amounts of wireless communication. To address this, we propose a two-tier AI system by offloading computations to multiple devices. Specifically, we build lightweight AI models for deployment on edge clients (i.e., edge sensors) and a large-scale AI model for the central server. These two types of AI models are trained with different criteria: the models on the edges act as the filtering tools to detect abnormal events and maximally avoid making false negative predictions, whereas the server model is supposed to be an expert for accurate predictions. By validating on a power theft dataset, we show that such a cascading methodology could filter out sufficient negative examples on the edge side while still being able to provide precise predictions on the second-round analysis.

Introduction

Energy infrastructures in many scenarios are naturally distributed over multiple locations. For example, distributed energy resources (DERs) provide electricity consumers commercially competitive alternatives to the conventional centralized grid. Infrastructures on the consumption side tend to be even more distributed: devices are generally located in different households, factories, or even districts. This geometric isolation brings some practical difficulties when performing monitoring or maintaining tasks. Manual maintenance and anomaly detection often need engineers to check the status of distributed infrastructures, compare abnormal meter readings with normal ones, or examine a bypassed power transmission line on site. This would require enormous human resources to be deployed, along with other issues such as time consumption and costs.

With the development of wireless communication techniques (e.g., 5G networks and Bluetooth), many of the monitoring tasks can now be assisted or even replaced by artificial intelligence (AI) models, so limited human labor will be required [1–4]. Through a wireless network, energy devices can exchange their working status with their neighbors and/or report their sensor readings to the server. Moreover, these AI models could automatically perform 24/7 monitoring and support near-real-time warnings on site.

From the general view, the AI monitoring can be performed either locally, that is, purely relying on the local analysis, or with assistance from a remote center. The former is generally preferred in practice as it minimally triggers communication between edge devices and the central server, standing out as a communication-efficient approach. It can be applied to many situations where anomaly events can be classified easily by some simple AI models like linear regression, support vector machine, and shallow neural networks. Examples can be found in federated learning [5], where the edges process the data analysis first, and the server is only in charge of model aggregation.

In this article, we shall focus on the latter case, where local models cannot perform the monitoring task accurately alone, and assistance from the remote server is required. This applies to situations where precise AI models require sufficient computation and memory resources, but the resources on the distributed energy edges are limited. For example, to detect a power usage anomaly, the AI-based approach in [6] contains a hybrid attention model with 51 million parameters, which is clearly beyond the computation abilities of most edge energy devices. Methodologies mostly relying on local analysis, like federated learning, cannot provide precise predictions in this case. A conventional solution in this scenario would be transmitting all local readings to a central server that is assumed to have more computing resources and more powerful AI capabilities, and then conducting the data analysis on the server alone. However, considering the fact that a server may connect to thousands of potential edge devices and the monitoring task should be performed continuously, the major challenge

*Chi Zhang, Liang Zhen, and Joey Zhou are with the Institute of High Performance Computing, A*STAR; Cen Chen (corresponding author) is with the Institute for Infocomm Research, A*STAR.*

Digital Object Identifier: 10.1109/MWC.014.2100171

When edges cannot process the monitoring task alone, wireless communication between distributed infrastructures and a center is inevitable. Still, it should be utilized in a wiser way: edge devices should only trigger communication when necessary.

here is the massive bandwidth required for wireless communication between the edges and the server. Moreover, wireless communication is not utilized efficiently since the server would simply predict "normal" for edge requests in most cases.

The question now boils down to how to offload part of the computation to the distributed devices and use wireless communication more efficiently. To answer it, we need to review how humans conduct the monitoring task: an experienced engineer would only examine suspicious conditions and ignore those events that can be confirmed as normal cases. In other words, there is a simple "pre-classifier" that allows the person to perform some estimations of on which case he needs to focus. This inspires us to design a similar approach for AI-based monitoring: the edge devices should ignore those normal signals and only contact the central server when they detect a potential sign of anomalies.

This article transforms such a filter-and-then-classify blueprint into a deployable AI-assisted monitoring system that requires limited wireless communication. Specifically, we build multiple lightweight AI models for the distributed clients (i.e., the edge sensors) and a large-scale AI model deployed on the central server. These models are trained with different criteria. The model on the client side acts as a filtering tool to detect abnormal events of edge energy devices and maximally avoids committing false negative predictions. In contrast, the model on the server acts as an expert to provide the most accurate prediction it can. Demonstration through a power theft use case illustrates that such a collaborative scheme can filter out sufficient negative examples on the edge devices, while still providing precise analysis for anomalous power usage on the server side.

Challenges

Deploying AI models to distributed devices allows for sensing along the infrastructure deployment and reducing the human labor involved. Before moving forward, let us first highlight some challenges that an AI system could encounter in practical design.

Hardware Constraints and AI Capability

The success of modern AI is accompanied by the increasing demand for hardware capabilities [7]. AI models (e.g., deep neural networks) often consist of millions of parameters that require numerous memories and computations to provide accurate predictions for their designated task. In contrast, the hardware on most edge infrastructures is assumed to have very limited computing resources, and we cannot expect them to have the same AI capabilities as most workstations do.

Therefore, before any training or validation, the underlying model should first fit the hardware constraints. A direct consequence is that we have to utilize simple models (e.g., shallow neural networks) in many situations, even though their performance is inferior to their complex counterparts. Thus, the gap between expected good model performance and the constrained hardware capabilities becomes the first challenge when designing an AI system for edges.

Efficient Wireless Communication

In the sense that local AI capabilities are constrained, sending all the signals back to a "powerful" server would nevertheless be a straightforward approach. The main issue here is the frequent wireless communication between the edge infrastructures and the central server. Moreover, communication for this approach is not utilized efficiently since most of the signals tend to be normal, while abnormal events in general are rare [8]. Thus, most of the time, the central server would simply reply to each edge infrastructure with "boring health."

When edges cannot process the monitoring task alone, wireless communication between distributed infrastructures and a center is inevitable. Still, it should be utilized in a wiser way: edge devices should only trigger communication when necessary. As a canonical example, when the ratio of normal/abnormal signals is 100, the strategy of only reporting the abnormal events would require 99.01 percent less communication than the naive approach. Of course, such a number represents the ideal situation, and an AI model could commit mistakes when classifying signals from normal events and abnormal events. But the principle of efficiently utilizing wireless communication should be well addressed for practical AI deployment.

Data Imbalance

The different frequencies of normal/abnormal signals bring the challenges of data imbalance and the corresponding model training difficulties. When the number of normal cases dominates the abnormal cases, the machine learning model often tends to simply predict every sample to be a normal case to achieve the highest accuracy (see the example in [9]). Although there are some special techniques like training with the area under curve (AUC) maximization [10] or sampling techniques [11], they do not com for free: for example, emphasizing abnormal signals by extra sampling leads to more internal epochs and additional computation costs.

Instead, the data imbalance issue could be solved or alleviated from its origin: the system may only pass some of the normal samples to the server. Instead of burying itself with numerous negative examples, the model can now focus on positive samples and be well trained to detect anomalies. The challenge here is how to design a practical filtering strategy to control the portion of normal signals while allowing all (or most) abnormal signals to pass through at the same time.

Computation Offloading

As alluded to earlier, a center may need to monitor a large number of edge energy infrastructures, and purely relying on the center for AI computation leads to heavy burdens on one side. An efficient AI system should offload computations from one server to multiple end clients so that the analysis is performed in a decentralized way.

The major challenge lying ahead is that we assume edges cannot perform the analysis alone, and we cannot naively allocate monitoring tasks to them due to performance considerations. Instead, we need to design a proper distributing mechanism such that only some of the tasks are borne by these end clients.

Proposed Solution: Two-Tier AI Analysis

Instead of utilizing the edge models alone or purely relying on the central server, in this article, we aim to design a third approach by offloading some of the analysis tasks to edge devices. The solution we propose is to utilize a two-tier AI system: lightweight AI models to be deployed on the edge clients as "pre-classifiers" and a large-scale expert AI model designed for precise signal analysis on the central server.

General System Scheme

Specifically, the overall signal analysis contains the following steps.

Step I: An edge device generates and collects signals. These signals can be meter readings or sensor values.

Step II: Local models classify and filter the signals into normal and suspicious based on the features generated in Step I: $\hat{y}_f$ = Filter(x). These models are assumed to be simple and deployable on edge devices.

Step III: If local predictions are normal ($\hat{y}_f = 0$), the analysis is complete, and no communication is triggered.

Step IV: If the local filter model detects suspicious signals ($\hat{y}_f = 1$) that require further analysis, the edge device shall upload these suspicious signals to the server through a wireless network. The wireless network could be a radio-wave network if edges are distributed remotely, or a Bluetooth network if they are within certain ranges.

Step V: The expert model on the server makes the final decision for suspicious signals $\hat{y}_e$ = Expert(x). The model on the server is assumed to be free from memory and computation limits, hence acting as an expert to classify normal and abnormal events.

Step VI: Return results $\hat{y}_e$ to local devices.

In the above scheme, we design two circles for AI analysis, and the classified normal and suspicious signals of Step II go through different decision steps (Fig. 1}). If the signals are locally classified as "normal," the analysis is then complete, and the AI system does not trigger wireless communication; we call this "the local circle." But if local models observe signs of abnormality, these signals shall be further transmitted to the server on the cloud. Since the server is generally assumed to have more powerful AI capabilities (e.g., workstations with very deep neural networks), the second-round analysis is expected to be more accurate than the local counterparts. In Fig. 1, we conclude these wireless communication steps and analysis on the server as the "remote circle."

The goal of designing such a two-tier signal analysis is to address the challenges discussed previously. AI capabilities on the edge infrastructures are generally limited by hardware constraints, and therefore we can only deploy simple models with light analysis. These edge models are expected to filter out normal signals and leave the rest as suspicious ones. The underlying mechanism is that detecting normal signals is generally much more manageable than detecting abnormalities, and local models can frankly predict "I do not know" for unknown patterns. Communication is utilized more efficiently in this case since data analysis for normal signals stays local and minimally triggers wireless communication with the server. Such a first-round analysis also allows the AI system to well offload the computations to multiple devices and avoids the computation congestion on the server side if we were to transmit every signal to the cloud.

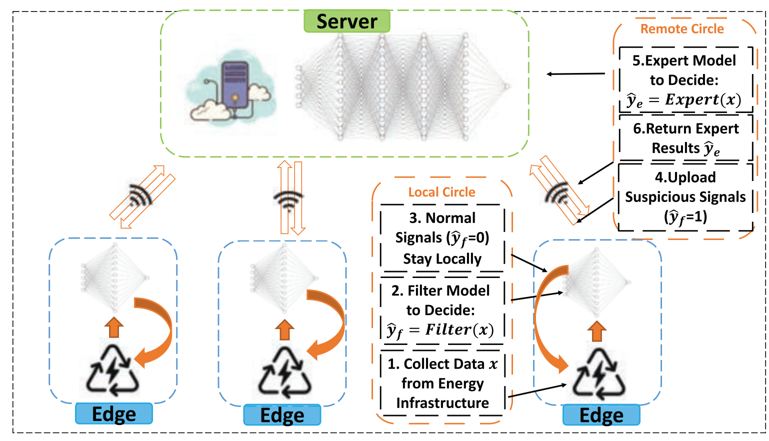

FIGURE 1. The general scheme for the proposed two-tier AI system. Data analysis is performed in two circles: edge models first perform some light analysis in the local circle; only suspicious signals shall be transmitted over the wireless network and further analyzed in the remote circle.

Edge Model Design

With the above general scheme, the next step would be designing some proper filter models for edge infrastructures. Specifically, the model on each local client acts as a "membrane" to filter out normal signals and let abnormal ones pass. Such a functionality makes it clearly different from the conventional machine learning approaches, where the goal is to achieve the highest accuracy or AUC score. Throughout this article, we assume memory and/or computation limits of edges; even the best local model may not fully classify the local signals. As such, instead of seeking its maximum classification ability (e.g., 60 or 70 percent), the light model now focuses on a simpler task: given a signal, it needs to classify whether the signal is definitely normal or just report it as suspicious.

To achieve this goal, these edge models require special designs. Note that machine learning often defines some loss functions to penalize the model's mistakes and forces it to move toward some directions with fewer errors. For a classification problem, the errors can be categorized into two classes:
- False positive (FP) prediction: incorrectly classifies normal events as suspicious signals
- False negative (FN) prediction: incorrectly classifies abnormal events as normal signals

For practical monitoring, the consequences of the above two mistakes are often different. The former error refers to the case where the filter model classifies normal events as abnormal signals (e.g., classify normal power usage as leakage). In Fig. 1, this would trigger communication with the server, and further analysis on the cloud would correct these mistakes. However, for the latter scenario, the filter classifies abnormal events (e.g., power leakage or power theft) as normal signals and does not inform either the local device or the remote server. The potential loss can be

The conventional idea on this dataset is to gather all the readings to one node (e.g., a server) and train a complex model to detect anomalies. However, power readings are naturally distributed over multiple locations, and transmitting signals to one place may face both communication and computation issues.

λ	False negative rate	Communication reduction	Label ratio
0	100 %	100 %	–
0.05	20.27 %	75.83 %	2.46
0.1	5.02 %	46.16 %	5.46
0.2	3.77 %	30.66 %	7.22

TABLE 1. Filter model results with various λ. As references, the SOTA model obtains $\approx$ 4.0% false negative rate, and the label ratio without filtering is 10.72.

disastrous: failure to predict power leakages could lead to fires of energy infrastructures; ignorance of power theft could result in continuous loss for power supply companies.

As such, instead of seeking the highest accuracy an edge model can achieve, we require it to obey the "Safety Principle" proposed in [12]:

Safety Principle: The filter models should eliminate or maximally avoid false negative predictions.

In machine learning, obeying the safety principle could be achieved by different approaches. In this article, we consider a simple method by adding an additional penalty to the false negative predictions. For example, in the binary classification problem, besides the prediction loss (e.g., mean-square loss $(y - \hat{y})^2$), an extra penalty term would be:

$$\lambda \cdot y \max(\lambda - \hat{y}, 0).$$

Here y refers to the true label, with y=0 denoting the normal events and $y = 1$ denoting the abnormal events. $\hat{y} \in [0, 1]$ (continuous, non-binary) refers to the predicted value from the local filter model, and ℓ refers to a threshold to perform binary classification: if $\hat{y} < \ell$, the final prediction would be 0; otherwise, the prediction would be 1. λ is a tuning parameter to exaggerate or shrink the penalty term. The goal of designing such an extra loss is to penalize FN predictions: the above formula is non-zero iff $y = 1$ and $\hat{y} < \ell$, namely the truth label y indicates an abnormal event, but our local AI model predicts it as a normal event.

Other penalty terms can be utilized similarly if the underlying problem is not binary classification. For example, in the regression problem, the penalty term could be $\lambda \cdot \max(y - \hat{y}, 0)$. In general, the design of these terms follows the same philosophy: we want the edge model to make as few FN predictions as possible, and hence be safer in filtering positive samples.

Expert Model Design

With the edge model settled, we can now proceed to the expert model design. Note that this model is to be deployed on the server side, where the server on the cloud often refers to a powerful workstation or a super-computer with abundant computation and memory resources. Hence, the choice of the expert model tends to be simple: we may choose the state-of-the-art (SOTA) AI model or use very deep neural networks (e.g., deep ResNet [13]). The principle that an expert should obey follows the conventional accuracy or AUC maximization principle: an expert should be able to precisely distinguish abnormal events from normal ones.

It is also worth mentioning that an expert model is not indispensable for many AI systems. In certain scenarios, its role can be safely replaced by humans instead of an AI model. For instance, as long as the edge models can filter out suspicious signals from potential tremendous data, a human engineer could focus on these tasks and perform the analysis and maintenance work. In this case, the human engineer acts as the "server model." Nevertheless, for a complete AI system or in the absence of human staff, we may still build an expert model as SOTA does.

Performance Evaluation

Having established the general scheme of the AI-assisted monitoring system, we can now proceed to experimental validations based on some real-world datasets.

Preliminary

The data we consider is from a power usage dataset released by State Grid Corporation of China (SGCC), which contains 42,372 electricity consumption records within 1035 days (https://github.com/henryRDlab/ElectricityTheftDetection). The goal is to detect power thieves by analyzing their power usage patterns. Since its first usage in [14], this dataset has become a public benchmark for power usage anomaly detection, especially for power theft detection. However, the current SOTA model is a hybrid attention model from [6], which consists of 51 million parameters and obtains a 0.92 AUC score. Such a huge model, though accurate, is clearly not suitable for edge infrastructures with limited memory and computation abilities. In the meantime, this dataset is highly imbalanced: 91.47 percent data are normal readings, whereas only 8.53 percent samples are power theft.

Edge Model Training

The conventional idea on this dataset is to gather all the readings to one node (e.g., a server) and train a complex model to detect anomalies. However, power readings are naturally distributed over multiple locations, and transmitting signals to one place may face both communication and computation issues. In this article, we propose a filter-and-then-classify paradigm that requires filter models to sort out suspicious signals first.

Specifically, the underlying model should fit the hardware constraints first and then maximize its AI capabilities. For simulation purposes, we adopt a convolutional neural network (CNN) model with three convolution layers and three linear layers, as has been validated in previous research [6]. To train such edge models, we use both the prediction loss $(1/2)(y - \hat{y})^2$ and the extra penalty term $\lambda \cdot y \max(\ell - \hat{y}, 0)$. Here λ acts as a hyper-parameter that decides how we shall penalize the FN predictions: a large λ would severely punish any FN predictions but at the same time lead to more FP predictions. This is because the algorithm tends to predict more instances to be positive (abnormal) to avoid the extra penalty term, some of which are actually negative (normal) samples. We split the overall dataset into a training set and a testing set with test_ratio = 0.3 and train the filter model with various λ.

Table 1 reports the final test performance on this dataset. Setting λ = 0 refers to the conventional approach by training edge models with standard mean squared error (MSE) loss. In this case, since we are using a relatively simple model and the negative samples dominate the whole dataset, the trained model predicts all instances as negative and does not trigger communication. Such a 100 percent false negative rate leads to an extreme case where all anomalies are neglected, and the filter model fails to sort out the suspicious signals. In contrast, by adding an extra penalty term (λ > 0), the false negative rate (FNR) can be significantly reduced. A larger λ refers to more severe penalties on the FN predictions; hence, the FNR decreases to maximally obey the safety principle.

Specifically, by setting λ = 0.2, we obtain an edge model with a 3.77 percent false negative rate, which is better than the SOTA complex model with ≈ 4.0 percent FNR. The limit of the hardware forces us to select a light model on edge energy infrastructures, but by training with the safety principle, this model minimally avoids predicting abnormal events as normal ones. This refers to the scenario where edge models filter out abnormal events to their maximum ability instead of seeking the best accuracy.

In the meanwhile, 30.66 percent of the signals are analyzed only locally and do not trigger wireless communication. By doing so, we are able to offload 30.66 percent computations to light models on edge devices instead of transmitting every signal and purely relying on the server. The data imbalance issue is also partially alleviated, reducing from 10.72 of the original dataset to 7.22 with local filtering.

Expert Model Training

The memory and computation ability on the server side is assumed to be sufficiently large. As such, we are able to deploy a more complex model on the server side, such as the SOTA Hybrid Attention model proposed in [6]. However, the safety principle should also be addressed for the expert model to penalize misclassifying abnormal signals as normal ones. To facilitate this idea, we add a similar penalty term with λ = 0.05 to force the expert model to address the false negative issue, which is not included in the original article.

Table 2 summarizes the overall performance of two previous research works and the two models proposed in this article. Note that the functionality of the filter model is not to seek the best AUC or F1 score; hence, its values are not as good as the rest. For the expert model, the obtained AUC is slightly higher than the original work, but the F1 score is lower. In general, though, the performance difference is rather minor since we use the same architecture as [6]. However, looking into the training process of these two models, as shown in Fig. 2, we observe that the underlying false prediction rate for the expert model is less by adding a penalty, and hence safer in terms of the safety principle.

Method	Parameters	Accuracy	AUC score	F1 score
Wide and deep CNN [14]	0.045	0.712	0.774	0.413
Hybrid attention [6]	1.000	0.918	0.919	0.597
Edge model	0.054	0.388	0.782	0.216
Expert model	1.000	0.923	0.921	0.596

TABLE 2. Model performance comparison. Note that the edge model is trained for low false negative rate, not for maximum accuracy.

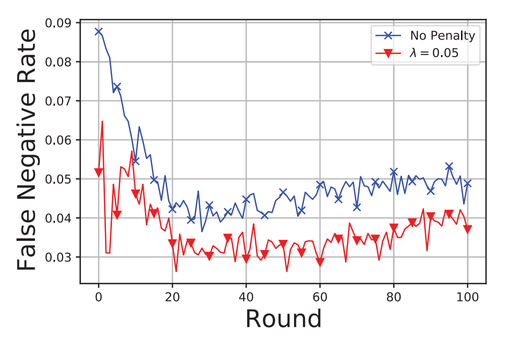

FIGURE 2. False negative rate for the expert model. "No penalty" refers to the reproduced work from [6]. λ =0.05 refers to training the expert model with penalty in this work.

Summary

This article considers a communication-efficient approach for energy infrastructure monitoring and maintenance enabled by wireless networks. Our goal is to build a two-tier AI-assisted management system by providing smart device status predictions, while at the same time addressing the communication overhead issues incurred by transmitting signals from edges to the server. We propose an AI system containing filter models that emphasize the false negative predictions and safety principle, as well as another expert model that can provide accurate analysis for suspicious signals. Validations are provided on a power theft detection dataset, where we show that the collaboration can distribute the computation burden to multiple edge devices and obey the safety principle simultaneously.

Regarding the edge models, we adopt some architectures from previous research studies but training with different criteria. For demonstration purposes, we adopt a simple penalty term to force the edge models to make as few false negative predictions as they can. The choice of the architecture and the penalty term may not be optimal, and we defer further explorations to future works.

References

[1] S. F. Bush, *Smart Grid: Communication-Enabled Intelligence for the Electric Power Grid*, Wiley, 2014.

[2] G. Dileep, "A Survey on Smart Grid Technologies and Applications," *Renewable Energy*, vol. 146, 2020, pp. 2589–2625.

[3] I. Al Ridhawi *et al.*, "Generalizing AI: Challenges and Opportunities for Plug and Play AI Solutions," *IEEE Network*, vol. 35, no. 1, Jan./Feb. 2020, pp. 372–79.

[4] H. B. Salameh *et al.*, "Energy-Efficient Cross-Layer Spectrum Sharing in Cr Green IoT Networks," *IEEE Trans. Green Commun. Networking*, 2021.

[5] J. Konečný *et al.*, "Federated Learning: Strategies for Improving Communication Efficiency," arXiv preprint arXiv:1610.05492, 2016.

[6] P. Finardi *et al.*, "Electricity Theft Detection with Self-Attention," CoRR, 2020; https://arxiv.org/abs/2002.06219.

[7] A. Canziani, A. Paszke, and E. Culurciello, "An Analysis of Deep Neural Network Models for Practical Applications," arXiv preprint arXiv:1605.07678, 2016.

[8] D. Pelleg and A. Moore, "Active Learning for Anomaly and Rare-Category Detection," *Advances in Neural Information Processing Systems*, vol. 17, 2004, pp. 1073–80.

[9] B. Rocca, "Handling Imbalanced Datasets in Machine Learning," Jan 2019; https://towardsdatascience.com/handling-imbalanced-datasets-inmachine-learning-7a0e84220f28.

[10] P. Zhao *et al.*, "Online AUC Maximization," *Proc. Int'l. Conf. Machine Learning*, 2011, pp. 233–40.
[11] J. L. Leevy *et al.*, "A Survey on Addressing High-Class Imbalance in Big Data," *J. Big Data*, vol. 5, no. 1, 2018, pp. 1–30.
[12] C. Zhang *et al.*, "Collaborative Inference for Efficient Remote Monitoring," CoRR, 2020; https://arxiv.org/abs/2002.04759.
[13] K. He *et al.*, "Deep Residual Learning for Image Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 770–78.
[14] Z. Zheng *et al.*, "Wide and Deep Convolutional Neural Networks for Electricity-Theft Detection to Secure Smart Grids," *IEEE Trans. Industrial Informatics*, vol. 14, no. 4, 2017, pp. 1606–15.

Biographies

Chi Zhang (zhang_chi@ihpc.a-star.edu.sg) is currently a research scientist with the Institute of High Performance Computing (IHPC) in the Agency for Science, Technology, and Research (A*STAR), Singapore. He obtained a Ph.D. degree in computer science from Nanyang Technological University (NTU), Singapore. His research interests include distributed computing, federated learning and efficient learning, aiming to design efficient AI algorithms for multi-agent systems.

Liangli Zhen (zhen_liangli@ihpc.a-star.edu.sg) received his Ph.D. degree in computer science from Sichuan University in 2018. Currently, he is a research scientist with IHPC, A*STAR. He leads a team of 10+ research staff members researching efficient machine learning, and safe and robust AI. Related research topics include representation learning, transfer learning, multimodal learning, model compression, adversarial attacks and defences, and more.

Joey Zhou (joey_zhou@ihpc.a-star.edu.sg) is currently a senior scientist, principal investigator, and group manager with IHPC, A*STAR. He also holds an adjunct faculty position at the National University of Singapore. He received a Ph.D. degree in computer science from NTU. He co-organized the ACML '16 Learning on Big Data Workshop, the IJCAI '19 Workshop on Multi-Output Learning, and the ICDCS '20 Workshop on Efficient AI; has served as an Associate Editor for *IEEE Transactions on Emerging Topics in Computational Intelligence*, *IEEE Access*, and *IET Image Processing*; was TPC Chair of Mobimedia 2020; and received the NeurIPS Best Reviewer Award in 2017.

Cen Chen (chen_cen@i2r.a-star.edu.sg) received his Ph.D. degree in computer science from Hunan University, China. Currently, he works as a scientist II with the Institute for Infocomm Research (I2R), A*STAR. His research interests include parallel and distributed computing, machine learning, and deep learning. He has published several research articles in international conference and journals on machine learning algorithms and parallel computing, including *IEEE-TC*, *IEEE-TPDS*, *AAAI*, ICDM, ICPP, and many more.