THEMIS: A Fair Evaluation Platform for Computer Vision Competitions

Zinuo Cai[1*], **Jianyong Yuan**[1*], **Yang Hua**[2], **Tao Song**[1], **Hao Wang**[3], **Zhengui Xue**[1], **Ningxin Hu**[4], **Jonathan Ding**[4], **Ruhui Ma**[1], **Mohammad Reza Haghighat**[4], **Haibing Guan**[1]
[1]Shanghai Jiao Tong University
[2]Queen's University Belfast
[3]Louisiana State University
[4]Intel
{kingczn1314, sjtu2017yjy, songt333, zhenguixue, ruhuima, hbguan}@sjtu.edu.cn, Y.Hua@qub.ac.uk, haowang@lsu.edu, {ningxin.hu, jonathan.ding, mohammad.r.haghighat}@intel.com

Abstract

It has become increasingly thorny for computer vision competitions to preserve fairness when participants intentionally fine-tune their models against the test datasets to improve their performance. To mitigate such unfairness, competition organizers restrict the training and evaluation process of participants' models. However, such restrictions introduce massive computation overheads for organizers and potential intellectual property leakage for participants. Thus, we propose **THEMIS**, a framework that trains a noise generator jointly with organizers and participants to prevent intentional fine-tuning by protecting test datasets from surreptitious manual labeling. Specifically, with the carefully designed noise generator, THEMIS adds noise to perturb test sets without twisting the performance ranking of participants' models. We evaluate the validity of THEMIS with a wide spectrum of real-world models and datasets. Our experimental results show that THEMIS effectively enforces competition fairness by precluding manual labeling of test sets and preserving the performance ranking of participants' models.

1 Introduction

The rapid advancement of machine learning in academia and industry has sprung numerous online competitions, especially in the computer vision area. Large-scale competitions have motivated researchers to push forward the performance of machine learning algorithms continuously. Many key algorithms are firstly proposed at competitions, such as AlexNet [Krizhevsky *et al.*, 2012], GoogleNet [Szegedy *et al.*, 2015], and ResNet [He *et al.*, 2016] appeared in ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [Russakovsky *et al.*, 2015]. The substantial amount of rewards attracts global talents to design machine learning models for particular problems and chase the championship. However, some dishonest participants intentionally fine-tune their models with the hand-labeled test sets to achieve a higher testing accuracy in the leaderboard, which violates the competition ethics and ruins the healthy competition ecosystem. Therefore, it is imperative for competition platforms to evaluate participants' models with *fairness* preserved.

We classify the mainstream platforms of computer vision competitions into three categories in Figure 1: Platform A, B, and C. The major difference between the three platforms lies in the way that they distribute the test data and labels. In Platform A, the organizer of competitions releases test data and labels without any further maintenance. It relieves the burden of both the organizer and participants, but releasing test labels allows participants to fine-tune their models with the test set. Platform A is more prevalent in machine learning communities rather than competitions, such as the handwritten digit recognition task on MNIST [LeCun *et al.*, 1998b]. In Platform B, the organizer only releases test data and keeps test labels private to avoid the above situation. Participants are required to submit their predictions to the platform for evaluation. However, Platform B can hardly prevent artificial tagging of test data, resulting in potential unfairness for honest participants. Kaggle[1] is the most famous machine learning competition platform falling into the Platform B category. A few similar vision competition platforms include ILSVRC, PASCAL VOC [Everingham *et al.*, 2010], MOT Challenge [Leal-Taixé *et al.*, 2015], and DAVIS Challenge on Video Object Segmentation [Perazzi *et al.*, 2016]. In Platform C, the organizer releases neither test data nor test labels. Participants are required to upload machine learning models or source code for evaluation. Thus, Platform C can successfully prohibit artificial tagging, but due to the heavy maintenance and configuration cost of model evaluation environments, Platform C only fits for small-scale competitions with a limited number of participants, such as CodaLab[2], the Visual Object Tracking (VOT) Challenge [Kristan *et al.*, 2016]. Besides, participants are usually reluctant to give up their intellectual property when uploading their models.

In this paper, we design **THEMIS**, a novel competition platform that prevents participants from fine-tuning their models with test sets and does not collect participants' models, which combines the advantages of the three platforms and avoids

[*]Equal contribution

[1]https://www.kaggle.com/

[2]https://competitions.codalab.org/

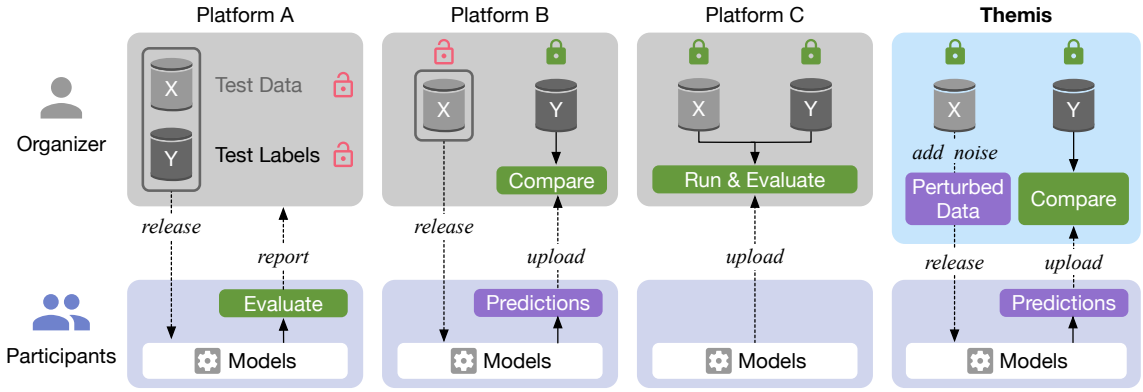

Figure 1: Comparison between current mainstream frameworks and THEMIS. Platform A releases both test data and test labels to participants and participants only report their final results to the platform. Platform B only releases test data and it compares the results uploaded by participants with ground-truth labels. Platform C does not release either test data or labels. Our platform, THEMIS, releases noisy test data and compares participants' predictions with test labels. *Best viewed in color.*

their drawbacks. Unlike Platform B, THEMIS prevents participants from hand-labeling the test set by releasing noisy test data to participants. The noise is generated from a series of independent Gaussian distributions with parameters trained by the organizer and participants collaboratively. Besides, THEMIS is much more scalable and trusted than Platform C because the organizer is free of maintaining a model evaluation environment, and participants can keep their models private. By comparing the participants' predictions over the noisy test data with the ground-truth test labels, THEMIS can estimate performance ranking of the participants' models.

We evaluate the effectiveness of THEMIS on popular models including LeNet [Lecun *et al.*, 1998a], ResNet [He *et al.*, 2016] and VGG [Simonyan and Zisserman, 2014] etc., and public datasets including UTKFace [Zhang *et al.*, 2017], CIFAR-10 [Krizhevsky, 2012], and CIFAR-100 [Krizhevsky, 2012]. Our extensive experiments demonstrate that THEMIS effectively guarantees the competition fairness by disturbing test data with random noise and precisely preserves the performance rankings of participants' models predicting the noisy test data, compared to their performance on plain test data with no noise added.

Our main contributions are as follows:

- To promote fairness in computer vision competitions, we propose a new evaluation platform, THEMIS, to avert participants from fine-tuning their models on test sets.
- We design a noise generator to protect test sets, derive constraints on its parameters theoretically, and prove its feasibility to ensure fairness with extensive experiments.
- Our experiments on public datasets, including UTKFace, CIFAR-10, and CIFAR-100, demonstrate that THEMIS can guarantee competition fairness by protecting the test set from human visual recognition and withstanding dishonest participants.

2 Related Work

Existing Evaluation Platforms. A few platforms host computer vision competitions. Kaggle is one of the most popular platforms based on Platform B. It promotes fairness by only revealing the accuracy of participants' models on partial test set before the competition deadline. Platforms based on Platform C—such as CodaLab—require participants to submit their source code. Few studies have focused on enhancing the fairness of competition platforms, and we only find that Blum and Hardt designed "The Ladder" [Blum and Hardt, 2015]—a reliable leaderboard for machine learning competitions—to solve the problem of overfitting and make competitions' leaderboards more reliable.

Methods to Protect Datasets. However, "The Ladder" still ignores the situation where participants may manually label the test set. Unlike "The Ladder", THEMIS focuses on how to process test sets to protect them from being hand-labeled. There are mainly two types of methods to protect datasets: cryptographic approaches and perturbation approaches [Al-Rubaie and Chang, 2019].

(1) Cryptographic Approaches. Homomorphic Encryption (HE) [Gentry, 2009] is one of the most prevalent encryption forms. Feasible homomorphic encryption schemes such as Leveled Homomorphic Encryption [Brakerski *et al.*, 2012] can support both addition and multiplication. Dowlin *et al.* proposed a method to convert learned neural networks to CryptoNets [Dowlin *et al.*, 2016] for encrypted data, while Hesamifard *et al.* extended the framework to deep neural networks in [Hesamifard *et al.*, 2017]. They also proposed to approximate nonlinear functions, e.g., sigmoid, ReLU [Nair and Hinton, 2010] with polynomials. Phong *et al.* built a deep learning system via additively HE to prevent privacy leakage to an honest-but-curious server in [Phong *et al.*, 2018]. Although HE's features are attractive for privacy-preserving, it requires adjusting the architectures of participants' models between training and inference, which introduces extra workloads for participants.

(2) Perturbation Approaches. Differential Privacy (DP) [Dwork, 2006]—as a typical perturbation approach—has been widely applied to enhance the dataset privacy in machine learning. Papernot *et al.* proposed Private Aggregation of Teacher Ensembles (PATE) [Papernot *et al.*, 2016] to protect private information via noise voting among models. [Abadi *et al.*, 2016] achieved the same goal within the frame-

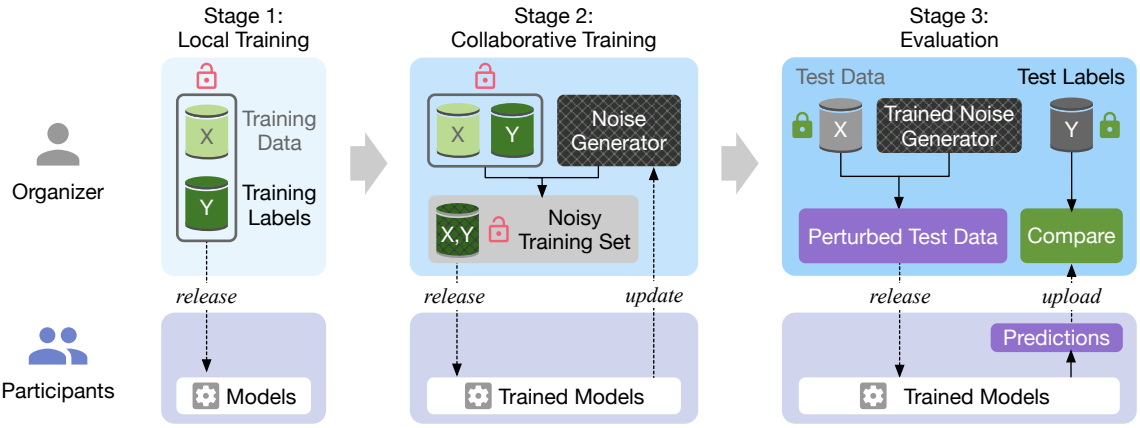

Figure 2: The workflow of THEMIS. The workflow is split into three stages: local training, collaborative training, and evaluation. Participants firstly train their models on the training set in the first stage. In the second stage, they jointly train a noise generator with the organizer. Finally, they submit the results of their models on the noisy test data, based on which the organizer announce their rankings after comparing their predictions with test labels. *Best viewed in color.*

work of DP. However, differential privacy is commonly utilized in the training stage rather than in the inference stage because it leads to accuracy degradation. Recently, some research has shifted attention to the inference stage: Wang *et al.* introduced random noise addition to their framework, ARDEN [Wang *et al.*, 2018], contributing to privacy protection and performance improvements simultaneously; Mireshghallah *et al.* proposed Cloak [Mireshghallah *et al.*, 2020], distinguishing pertinent features from uncorrelated features for specific tasks in the meantime of training a noise generator. Our framework, THEMIS, based on Cloak, enhances the privacy preservation of test sets and prevents human visual recognition.

3 The Design of THEMIS

3.1 Workflow

To overcome those drawbacks exposed by current platforms of computer vision competitions, we design THEMIS to promote the fairness of computer vision competitions. Figure 2 presents the workflow of THEMIS, including three stages: local training, collaborative training, and evaluation.

In the local training stage, the competition organizer releases the training set, including plain training data and labels. The participants download the training set, create models, and train their models on it. In the collaborative training stage, the organizer first initializes a noise generator and then invites all participants to jointly train the noise generator for several epochs. An epoch of the collaborative training process has the following steps: 1) The noise generator generates noise and adds it to the training set; 2) It sends the noisy training set to participants, and then participants feed the noisy training set to their models and compute the noise generator's gradients; 3) Participants send back the gradients to the organizer to update the noise generator's parameters. In the evaluation stage, the organizer uses the trained noise generator to add noise to the test data, and then release the noisy test data to participants. Participants are required to submit the predictions of their models on the noisy test data. After comparing the participants' results with the ground-truth test labels, the organizer estimates participants' scores and rankings.

3.2 The Noise Generator

The noise generator is the critical part of our framework. It is adopted to generate noise to perturb test data and prevent participants from identifying the raw test data and labeling them manually. With more noise added to the test data, the accuracies of participants' models will inevitably decrease. However, we can keep the rankings of participants' models unchanged by adding some constraints to the noise generator's parameters. Particularly, by training the noise generator with some participants' models collaboratively, we can recover the accuracy of most models and relax the constraints of the noise generator's parameters, which means we can increase the noise scale to further protect the test data.

The design of our noise generator follows that in Cloak [Mireshghallah *et al.*, 2020]. It retains a series of μ and σ, which respectively represent the mean and variance of a series of independent Gaussian distributions. The number of Gaussian distributions is equal to the size of the input data. For instance, if the size of input images is $3 \times 32 \times 32$, the noise generator will have 3072 independent Gaussian distributions. The scale of σ represents the amount of noise we add, and thus describes the protection level of test data. That means the larger the scale of σ is, the higher level of protection for test data we provide.

3.3 Constraints for Rank Preservation

As a competition platform, it is fundamental to ensure fairness for all the participants. Specifically, the ranking of participants' models must keep consistent after applying noise to the test data. We apply mathematical analysis to explore the relationship between noise and accuracy degradation.

For a computer vision problem, we consider that the distribution of the dataset follows a normal distribution with mean μ_x and variance σ_x^2. We make this assumption since normalizing input data with a normal distribution is a widely-applied data preprocessing method in computer vision. For a

Algorithm 1 Training the noise generator

Require: $D_{train}, y_{train}, F, total\_iteration$
1: Initialize μ, ρ, $iteration = 0$
2: **repeat**
3: Select training batch x from D_{train}
4: Sample $e \sim N(0, 1)$
5: Let $\sigma = \frac{1+\tanh(\rho)}{2} \times M$
6: Let $r = \sigma \cdot e + \mu$
7: Let $X = x + r$
8: Compute $loss$ from Eq. (2)
9: Gradient descend on μ, ρ from $loss$.
10: Let $iteration = iteration + 1$
11: **until** $iteration == total\_iteration$
12: **return** μ, ρ

user model m that is trained on the dataset $X \sim N(\mu_x, \sigma_x^2)$, we consider that it is trying to find a distribution $\widetilde{X} \sim N(\mu_m, \sigma_m^2)$ to fit the distribution of the dataset X. Under this premise, the model's accuracy α is correlated to the similarity of X and $\widetilde{X}$, and we use Kullback-Leibler Divergence [Joyce, 2011] to quantify the similarity of two Gaussian distributions. In our framework, the noise distribution follows $N(\mu_n, \sigma_n^2)$. So after adding noise to the input data, the new distribution of the noisy input data is $X' \sim N(\mu_x + \mu_n, \sigma_x^2 + \sigma_n^2)$.
For any two user models m_1 and m_2, suppose that $\alpha_{m_1} > \alpha_{m_2}$. Targeted to preserve the rankings of distinct models regardless of model structures, we need to solve the Inequation $\alpha'_{m_1} > \alpha'_{m_2}$. We simplify the inequation and finally get

$$\frac{\sigma_n^2}{2\mu_n} \leq \mu_x - \mu_m \leq upperbound. \tag{1}$$

3.4 Training the Noise Generator

We discover that collaboratively training the noise generator with a variety of models not only improves model accuracies on the noisy test data, but also broadens Eq. (1) for ranking preservation and enhance fairness. That's why we design the second stage—collaborative training stage—in our framework, in which all the participants are required to train the noise generator with the organizer using the loss function defined as:

$$\mathcal{L} = -\log \frac{1}{n} \sum_{i=0}^{n} \sigma_i^2 + \lambda \sum_{f_i \sim F}^{F} E_{r \sim \mathcal{N}(\mu, \sigma^2), x \sim \mathcal{D}} \left[-\sum_{k=1}^{K} y_k \log \left(f_i(x + r) \right)_k \right], \tag{2}$$

where the first term tempts to increase the scale of noise to further protect test data, while the second term optimizes the accuracies of all participants' models F and improves ranking preservation. λ is a hyperparameter that keeps a balance between these two terms. Algorithm 1 shows the detailed process of the collaborative training stage. Here F denotes a set of participants' models and we use another parameter ρ to replace σ to restrict the scale of σ to the upper bound M.

4 Evaluation

4.1 Experiment Settings

Datasets and Tasks. We select three datasets to evaluate our framework: the UTKFace dataset, the CIFAR-10 and CIFAR-100 datasets. On the UTKFace dataset, we carry out a gender-classification task, and on the CIFAR-10 and CIFAR-100 datasets, we conduct 10-class and 100-class classification, respectively. In all experiments, we split them into three parts—training sets, validation sets, and test sets—with the ration 4:1:1. Training sets and validation sets are available for both the organizer and participants throughout the contest, while test sets are not accessible until the organizer and participants collaboratively train the noise generator. Besides, the organizer only releases the noisy test set processed by noise generators.
Models. We evaluate THEMIS in real-world production scenarios by training various numbers of models for each competition task. The number of models varies according to the difficulty of the task. For the gender-classification task, there are three models participating in the training, including LeNet [Lecun *et al.*, 1998a], ResNet [He *et al.*, 2016] and VGG [Simonyan and Zisserman, 2014]. For CIFAR-10 classification, we introduce AlexNet [Krizhevsky *et al.*, 2012], DenseNet [Huang *et al.*, 2017], GoogleNet [Szegedy *et al.*, 2015], and other models into our framework. For the classification task on CIFAR-100, we do not only adopt different architectures of models but also models of different complexity for the same architecture. For instance, in terms of ResNet, we adopt ResNet-20, ResNet-32, and ResNet-56.
Implementation Details. For simplicity, we simulate the second stage instead of implementing end-to-end interactions between the organizer and participants. For one task, we first train several models on the training set. And then, we feed the noisy training sets to those models and only update the noise generator's parameters. The time to update the noise generator is related to the number of models and their complexity. Finally, we evaluate the models' performance on the noisy test sets. We implement the code in PyTorch and run the experiment on an NVIDIA virtual machine with 4 Tesla K80 GPU cores. THEMIS is open-sourced at https://github.com/AISIGSJTU/Themis.

4.2 THEMIS's Effectiveness

Figure 3 demonstrates the effects of noise generators on three datasets. For each group, the original images are on the left while the noisy images are on the right. The first row describes how THEMIS works on the UTKFace dataset. It is impressive that ResNet-56 and VGG16 can respectively have an accuracy of 82.92% and 83.01% with noise added while it is challenging enough for humans to gain a high accuracy for the gender classification. For more complex tasks like 10-category classification on the CIFAR-10 dataset, some visual information still left in the noisy test set to keep rankings consistent. However, we have a test with human annotator, which also shows THEMIS's effectiveness in protecting test sets.

To show the significance of training the noise generator and its effects on recovering accuracies of different participants' models, we display the relationship between the accuracies of

Task	Model	Plain Test Sets		Untrained Noisy Sets		Trained Noisy Sets	
		Accuracy	Rank	Accuracy	Δ_{rank}	Accuracy	Δ_{rank}
Gender Classification	LeNet	88.57%	3	64.46%	+1	82.75%	0
	ResNet-56	89.31%	2	65.01%	+1	82.91%	0
	VGG16	90.70%	1	57.06%	-2	83.06%	0
CIFAR-10 Classification	AlexNet	71.68%	9	65.38%	+8	70.24%	0
	DenseNet-121	91.45%	1	22.04%	-7	86.38%	-1
	GoogleNet	91.43%	2	16.35%	-8	87.02%	+1
	LeNet	68.28%	10	44.40%	+8	67.67%	0
	PreResNet-20	78.81%	8	27.14%	+4	75.24%	0
	PreResNet-56	80.18%	7	25.35%	+1	76.53%	0
	ResNet-20	82.83%	6	21.27%	-3	78.96%	0
	ResNet-56	83.70%	5	25.86%	0	80.60%	+1
	VGG11	86.19%	4	27.72%	+1	80.58%	-1
	VGG16	88.20%	3	24.53%	-4	84.07%	0

Table 1: Accuracy comparison of models evaluated with different test sets. The differences between plain test sets, untrained noisy sets, and trained noisy sets are whether the test sets are disturbed by noisy generators, or their generators are collaboratively trained. Compared with the untrained noisy generator, the noise generator after training can alleviate accuracy degradation and simultaneously make the rankings of models have minute fluctuations.

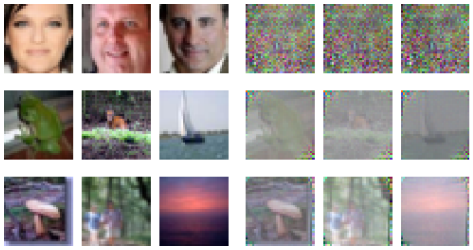

Figure 3: Effects of noise generators on the three datasets. For each group, the original images are on the left while the images disturbed by the noise generator are on the right. The effect of noise on the UTKFace dataset is more significant than that on the CIFAR-10 and CIFAR-100 datasets. *Best viewed in color.*

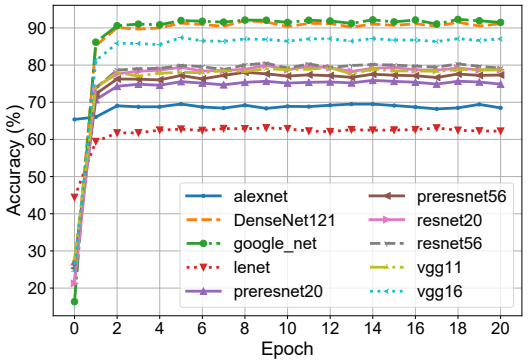

Figure 4: Model accuracy on validation sets *v.s.* noise training epoch. Ten models participate in the training of the noise generator on the CIFAR-10 dataset. In the beginning, all models have a dramatical degradation in performance but they gradually recover their performances as the training goes on.

the models on the noisy validation sets and training epochs of the noise generator during the CIFAR-10 classification task in Figure 4. At the beginning, there are two explicit problems with the noise generator, resulting in that we cannot utilize the noise generator without training in our framework. One is that the accuracy of different models deteriorates so dramatically that it cannot reflect their actual performance on the test sets. The other one is that it cannot be guaranteed that the rankings between models remain unchanged. We can conclude from Figure 4 that models with deeper architectures usually suffer more than those lightweight models. With the training going on, the models gradually recover their capability on the validation sets. We adopt validation sets instead of test sets to evaluate models' capacity when training the noise generator since test sets are not accessible until evaluation.

4.3 THEMIS's Fairness

Serving as a novel platform for computer vision competitions that aims at protecting the test data and improving the competition atmosphere, we can not ignore the intrinsic fairness of the competition. It means that suppose every participant trains his models without the trick of fine-tuning directly on the test data, the rankings of different models should reflect where they really are, just like other platforms.

To verify our framework's fairness, we simulate three classification competitions. For the gender classification and the 10-category classification, we display details and results of models and their performance in Table 1. In the left columns, we first list the results of different models' performance on the test sets without noise, similar to other platforms. The middle columns show the results on the test with random noise. The initial parameters of random noise are the same as our framework for the same task, but there is no further training with the noise generator. The right columns display our framework's final results. For the UTKFace dataset, although the discrimination regarding accuracy on the test set is minute, the ranking of the models does not change after applying noise. For the CIFAR-10 dataset, we assume that

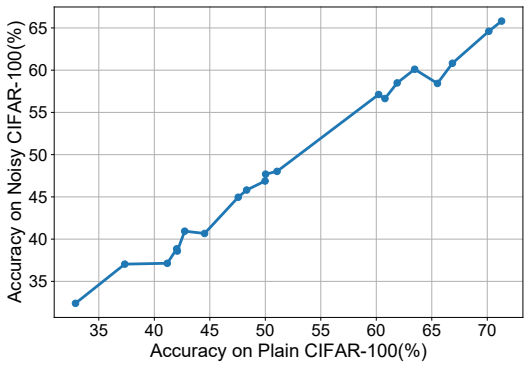

Figure 5: Accuracy on the plain CIFAR-100 dataset *v.s.* on the noisy CIFAR-100 dataset. The X-axis represents the accuracy of the model on the plain test set while the Y-axis on the noisy test set. There is an approximately linear correlation between the X-axis and Y-axis variables.

Model Name	Original Rankings	Case 0	Case 1	Case 2
AlexNet	9	9	9	9
DenseNet-121	1	2	2	2
GoogleNet	2	1	1	1
LeNet	10	10	10	10
PreResNet-20	8	8	8	8
PreResNet-56	7	7	7	7
ResNet-20	6	6	6	6
ResNet-56	5	4	4	4
VGG11	4	5	5	5
VGG16	3	3	3	3

Table 2: THEMIS's guarantee of fairness with unexpected situations. In Case 0, participants' models update the noise generator in their names' alphabetical order, while they update the noise generator in the reverse order in Case 1. In Case 2, we train the noise generator 20 epochs, but two models randomly drop out of each epoch's collaborative training.

ten participants take part in the contest. On the plain test sets, DenseNet-121 and GoogleNet win the first and second place respectively, while on the test sets with trained noise, the opposite is true. Although their rankings have a trivial variation, it can be tolerated since no matter on the plain or noisy test sets, the difference between these two models in terms of accuracy is below 0.1%. For the CIFAR-100 dataset, we adopt twenty models to train the noise generator and describe the relationship between each model's accuracy on the plain and noisy test sets in Figure 5. The overall trend reflected in the figure is that there is a linear relationship between the accuracy values for different models, which is up to the requirements of our framework. Despite there may be minor changes among rankings, such changes usually occur between two models with similar rankings.

4.4 THEMIS's Robustness

Besides effectiveness and fairness, robustness is also crucial for a computer vision competition platform. THEMIS's robustness refers to that it can handle unexpected situations. Specifically, we consider whether the following two scenarios influence the final rankings: (1) the joining order of the participants' models is different; (2) some participants miss some epochs of the collaborative training. We experiment on the CIFAR-10 dataset with ten models listed in Table 1. Our results in Table 2 show that THEMIS can handle these two scenarios without hurt to fairness. We only show the models' rankings in Table 2 because of limited space.

In the first scenario, THEMIS's fairness is unrelated to the training order in which participants' models join the collaborative training stage. We verify it by changing the training order and evaluate participants' models with noisy test data perturbed by the new noise generator. Case 0 and Case 1 in Table 2 demonstrates two different training orders. In Case 0, participants' models update the noise generator in alphabetical order of their names, i.e., AlexNet is the first to update the noise generator while VGG16 is the last. Although the order of Case 1 is contrary to that of Case 0, the rankings of models remain unchanged. We also try more random orders, and the results remain the same.

The second scenario demonstrates that when participants miss some epochs in the collaborative training, there will be little influence on the final evaluation results. In the ideal workflow displayed in §3.1, all the participants' models should join the second stage. However, some participants may miss some epochs by accident or on purpose. Case 2 in Table 2 simulates this scenario, where two of the ten models randomly drop out of the collaborative training in each training epoch. In the evaluation stage, we evaluate all the models' performance and compare it with the normal situation. We can conclude that THEMIS can ensure fairness even when models miss some training epochs in the second stage.

5 Conclusion

Fine-tuning models with test sets to get higher accuracy is demoralizing participants and debasing the significance of competition platforms that promotes machine learning development. Yet, there is no effective strategies to preclude such unfair practices in current platforms. Therefore, we propose THEMIS, a new evaluation platform that fills this gap and guarantee the fairness in computer vision competitions. THEMIS prevent participants from fine-tuning models by adding noise to test data with a noise generator collaboratively trained across participants and the organizer. We implement THEMIS and evaluate its effectiveness, fairness, and robustness with theoretical analysis and real-world experiments. Our experiments show that THEMIS effectively prevents model fine-tuning on test sets and preserves fairness in a wide spectrum of computer vision tasks. In our future study, we plan to extend THEMIS to support domains such as natural language processing.

Acknowledgments

This work is partially funded by National Natural Science Foundation of China (NO. 61872234, 61732010, 61525204), Shanghai Key Laboratory of Scalable Computing and Systems and Intel Corporation. Tao Song (songt333@sjtu.edu.cn) is the corresponding author.

References

[Abadi *et al.*, 2016] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *CCS*, 2016.

[Al-Rubaie and Chang, 2019] Mohammad Al-Rubaie and J Morris Chang. Privacy-preserving machine learning: Threats and solutions. *IEEE Security & Privacy*, 17(2):49–58, 2019.

[Blum and Hardt, 2015] Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. In *ICML*, 2015.

[Brakerski *et al.*, 2012] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, 2012.

[Dowlin *et al.*, 2016] Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *ICML*, 2016.

[Dwork, 2006] Cynthia Dwork. Differential privacy. In *ICALP*, 2006.

[Everingham *et al.*, 2010] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[Gentry, 2009] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[Hesamifard *et al.*, 2017] Ehsan Hesamifard, Hassan Takabi, and Mehdi Ghasemi. Cryptodl: Deep neural networks over encrypted data. *arXiv preprint arXiv:1711.05189*, 2017.

[Huang *et al.*, 2017] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[Joyce, 2011] J. M. Joyce. Kullback-leibler divergence. *International Encyclopedia of Statistical Science*, pages 720–722, 2011.

[Kristan *et al.*, 2016] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomas Vojir, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:2137–2155, Nov 2016.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[Krizhevsky, 2012] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Tech Report*, 05 2012.

[Leal-Taixé *et al.*, 2015] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*, 2015.

[Lecun *et al.*, 1998a] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LeCun *et al.*, 1998b] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Mireshghallah *et al.*, 2020] Fatemehsadat Mireshghallah, Mohammadkazem Taram, Ali Jalali, Ahmed Taha Elthakeb, Dean Tullsen, and Hadi Esmaeilzadeh. Interpretable privacy for deep learning inference. *Tech Report*, 2020.

[Nair and Hinton, 2010] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[Papernot *et al.*, 2016] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

[Perazzi *et al.*, 2016] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

[Phong *et al.*, 2018] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345, 2018.

[Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Szegedy *et al.*, 2015] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[Wang *et al.*, 2018] Ji Wang, Jianguo Zhang, Weidong Bao, Xiaomin Zhu, Bokai Cao, and Philip S. Yu. Not just privacy: Improving performance of private deep learning in mobile cloud. In *KDD*, 2018.

[Zhang *et al.*, 2017] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *CVPR*, 2017.