Ensemble Learning Methods for Dirty Data: A Keynote at CIKM 2022

Ling Liu
School of Computer Science, Georgia Institute of Technology
USA
ling.liu@cc.gatech.edu

Abstract

Neural network ensemble is a collaborative learning paradigm that utilizes multiple neural networks to solve a complex learning problem. Constructing predictive models with high generalization performance is an important and yet most challenging goal for robust AI systems in the presence of dirty data. Given a target learning task, popular approaches have been dedicated to designing and finding the top performing model. However, it is difficult in general to estimate the best model when available data is finite, possibly dirty, or insufficient for the problem. The problem of dirty data in machine learning (ML) can be characterized by the out of distribution data and the digital or physical deception of data. Such dirty data may cause unintended or harmful behavior for well trained ML models. In this paper, a curated version of my keynote at ACM CIKM 2022, I will first give a brief overview of ensemble learning methodology. Then I will review different types of dirty data that could deceive well-trained ML models. Finally, I will describe a focal diversity optimized ensemble learning framework, developed at Georgia Tech, for measuring, enforcing, and combining multiple neural networks, delivering high generalization performance of ensemble learner, while maximizing ensemble utility and resilience to dirty data.

Date: 17 October 2022.

1 Introduction to Ensemble Learning

Deep neural networks (DNNs) have enjoyed notable success in various mission critical applications. However, empirical observations and specific neural network applications to date both acknowledge that it is common to find a learning algorithm to outperform others for a specific problem, but it is rare to find a single predictor that can achieve the best generalization performance for the overall problem domain. Even with the growing trends of more complex DNN algorithms, including the recent success of the large language models (LLMs), individual DNN algorithms are unlikely to generalize perfectly to all possible test cases, especially in the presence of dirty data. Ensemble methods present an appealing solution and can address several scaling challenges: (i) Ensembles present an efficient approach to tackle the problem of inductive learning over large and growing datasets. (ii) Ensembles can solve the problem of learning over a population of geographically distributed datasets, which cannot be collected and stored in a central location due to privacy

concerns, data streaming velocity concerns, or data generation volume concerns. (iii) Ensemble learners can be an effective solution for learning evolving data streams. By training a team of diverse base models from different chunks of the data streams, ensembles can adaptively combine the individual models to unravel the problem of concept drift.

An ensemble machine is a type of meta learning systems [Huang et al., 2017], which consists of multiple individual learners as a committee. An ensemble learner exploits the different local behavior of the individual learners and combine their independent decisions to enhance the performance of the overall inductive learning system in accuracy and reliability. There are different ways to create base learners that compose an ensemble, and there are different ways to learn and determine how to combine the predictions of multiple individual models [Liu et al., 2019]. The tie that binds all the techniques is that an ensemble and its member models all attempt to solve the same problem, e.g., driving a car. Although each predictor may make error in one situation or another, the base learners within a good ensemble should exhibit "diverse errors" in diverse situations: driving a car on a highway, a street, a dirt track, or under foggy weather, and so on. The challenges are how to recognize and complement the limitations of individual models, and how to compose an ensemble of individual learners so that they do not make the same bad decision at the same time.

How an ensemble learner outperforms individual member models. *Statistically*, DNN algorithms attempt to discover a hypothesis in the space H of hypotheses for a specified problem. When given sufficient data, the optimal hypothesis may be found. However, in real scenarios where only limited data is available and some available data may be dirty, different algorithms may find different hypotheses, with different generalization performance. An ensemble in these cases may combine the individual algorithms to provide a sound statistical approximation of the unknown true hypothesis [Ronen et al., 2018]. *Representationally*, diverse DNN algorithms may have diverse representational capabilities, especially when the space of hypotheses explored by each individual algorithm, trained on limited data (insufficient for the problem), is much smaller compared to H. An ensemble can tackle the representational limitations of individual learners by taking different trajectories to traverse over H and combining the set of hypotheses found. Hence, by combining multiple individual predictors, an ensemble magnifies the space of representable functions, and increases the probability of embracing the true optimal hypothesis [Dietterich, 2000]. *Computationally*, DNN algorithms perform minimization on an error function for the given training data by applying local optimization techniques, e.g., gradient decent, and may get stuck in suboptimal results when multiple local minima exist for the underlying error function. An ensemble can avoid the worst local minima and achieve a better approximation by merging different local suboptimal solutions [Huang et al., 2017].

Background and Research Challenges. Ensemble machines of multiple learners have been one of the main ML research directions, and empirically applied to a range of real problems, offering improved predictive performance over single models, including adversarial learning [Eykholt et al., 2018b; Wei and Liu, 2020]. The developments of ensemble methods evolve mainly along three threads. The first category of efforts designs the algorithms for ensemble learning by combining a set of weak learners using serial, parallel, and hierarchical parallel ensemble architecture, e.g., bagging [Breiman, 1996], boosting [Breiman, 1998] and random forests [Breiman, 2001]. The second category of work centers on voting methods to obtain ensemble prediction by combining all member predictors of an ensemble committee (team) using the sum rule [Kittler, 1998; Liu

Figure 1. Visualization of adverse effect of dirty data by 11 different adversarial perturbation attacks [Wei and Liu, 2020].

et al., 2019], such as simple averaging, majority voting, plurality voting, or weighted kernels. The third category of work attempts to provide theoretical explanation of why multiple learner systems improve single learner [Geman et al., 1992].

For example, Dietterich [2000] shows a theoretical reason of why ensembles can improve the performance of single learner. Consider N classifiers, each with a probability of > 0.5 being correct (i.e., error $\epsilon \leq 0.5$), the ensemble with majority voting will result in an error (ϵ_{ens}) less than that of any component classifier, if the N individual learners make uncorrelated errors. The overall error ϵ_{ens} of the ensemble is given by $\epsilon_{ens} = \sum_{i=\lceil N/2 \rceil}^{N} C_i^N \epsilon^i (1-\epsilon)^{(N-i)}$, where C_i^N denotes the number of i combinations out of the set of N models. If $N = 21$, $\epsilon = 0.3$, the errors of the N models are independent, we have $\epsilon_{ens} = 0.0026 < \epsilon = 0.3$.

This simple example indicates that the effectiveness of an ensemble machine of N base learners depends on two factors: (i) the accuracy of the component learners and (ii) their diversity of responding to negative examples. The best scenario is when all member predictors of an ensemble committee of size N can learn and predict with uncorrelated errors. Then a simple averaging method can effectively reduce the average error of a member model by a factor of N. The worst scenario represents another end of the spectrum: all N member models are N perfect duplicates so that they are identical in positive and negative predictions, resulting in zero utility in combining their outputs. However, when the errors are correlated to some extent, which is typical in practice, it is realistic to expect that the overall error reduction will be smaller and yet the expected ensemble committee error will not exceed the average of the expected error of its member models.

2 Introduction to Dirty Data

Well-trained deep neural networks (DNNs) are known to be vulnerable against adversarial examples and out-of-distribution examples.

Adversarial examples [Goodfellow et al., 2014] are the input artifacts that are created from benign inputs by adding adversarial distortions, aiming to fool the victim model to misbehave with high confidence and without being perceived by the human. There are two categories of deceptions due to adversarial distortions: digital distortion based deception and physical distortion based deception.

Figure 1 provides a visualization of dirty data by digital deception from 11 different adversarial perturbation attacks [Wei and Liu, 2020]. The victim is a well trained DenseNet model on CIFAR-10. Its prediction on the benign input is a horse image with 0.983 confidence. Each adversarial

Figure 2. Visualization of adverse effect of dirty data by 3 different adversarial physical distortions: Left [Simen Thys, 2019], right top [Eykholt et al., 2018a], and right bottom [Sharif et al.].

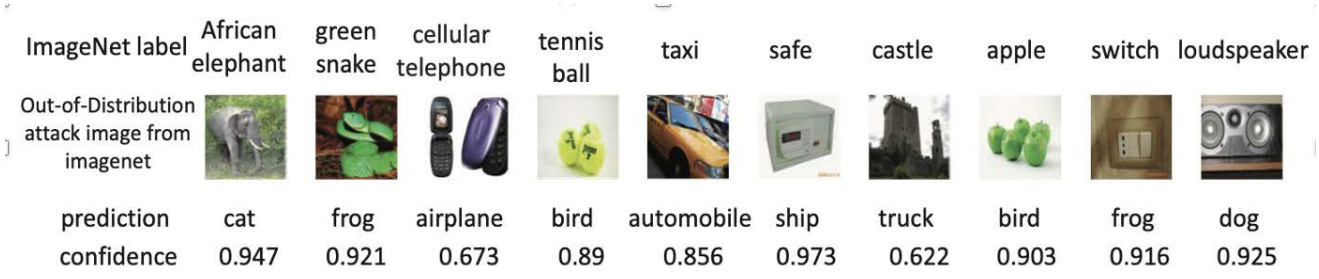


Figure 3. Visualization of the OOD attack, in which TinyImageNet is fed to a CIFAR-10 classifier [Wei and Liu, 2020].

example is generated by one of the 11 adversarial perturbation algorithms and it fools the victim model to mis-detect the horse image as a bird or an airplane with high confidence.

Figure 2 provides a visualization of dirty data deception by three different types of physical distortions. The left physical attack is to place a high density colored paper in the center of the object to fool a well trained DNN object detector to fail to detect the person holding the colored paper in front center of himself [Simen Thys, 2019]. The top right shows the stop sign was physically disturbed by a few black or white scotch tapes, which fools a well trained DNN object detector to detect speed limit 45 instead of stop sign [Eykholt et al., 2018a]. The bottom right shows the actress Reese Witherspoon (left) and when wearing a pair of glasses with dense colored frame, a well trained DNN face recognition model is fooled badly and detects Reese Witherspoon wearing a pair of glasses with colored frame (middle) as Russell Crowe, a male actor [Sharif et al.].

Out-of-distribution (OOD) examples are the artifacts of dirty data due to abnormal inputs drawn from a completely different distribution than the data generating distribution on which the model is trained [Wei and Liu, 2020]. OOD examples can also be utilized by an adversary to launch malicious attacks because when deploying neural networks in real-world applications, there is often very little control over the test data distribution.

Figure 3 shows eleven OOD examples, which are taken from TinyImageNet and fed into a well trained CIFAR-10 DNN model. We observe that the cell phone image is mis-classified as an airplane and the apple image is mis-classified as a bird, because the DNN classifier trained on the CIFAR-10 dataset has no knowledge of additional classes other than the 10 classes in CIFAR-10. With the classification prediction API of the victim DNN model trained on CIFAR-10 is limited

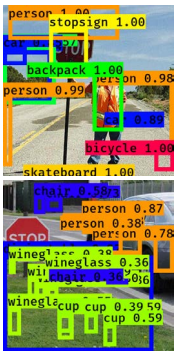
Benign (No Attack)	Adversarial Attacks with Different Types of Attack Specificity			
	Untargeted Random	Object-vanishing	Object-fabrication	Object-mislabeling stopsign→umbrella
				

Table 1. Detection on two examples by the TOG family of vision attacks [Chow et al., 2020a].

to only output the prediction probability vector of the 10 classes, an adversary can easily utilize the out-of-distribution examples to launch successful untargeted attacks.

Complex Adversarial Input Corruption to Vision Learners. The deception-induced dirty data attack allows an adversary to control the detection capability of well-trained DNN models by generating deceptive query inputs. Unlike irregular out-of-distribution input and physical distortion based deception input, a representative approach to creating deception-induced digital distortion to the input examples is to use the gradients of the DNN models to find tiny perturbations to input [Chow et al., 2020c]. Such adversarial distortion patterns can effectively mislead the DNN object detectors to amplify the perturbation progressively, which will become large enough to interfere with the final decision in the model output layer and yet not too large to be perceived by human. Although the goal of adversarial examples is similar for different learning tasks, the concrete adversarial perturbation techniques tend to vary substantially. For example, real time object detection task requires to learn object existence, object bounding-box and object classification for every candidate object in an input video frame or a multi-object image. Hence, an adversary can succeed in generating dirty input to fool the model by simply attacking one of these subtasks.

Table 1 illustrates the four types of adversarial attacks to object detection in a computer vision system. With no attack, the well trained object detector can accurately identify the person, the car, and the stop sign on the two benign images (1st column). However, the *same* detector is fooled blindly by for different attack strategies (2nd-5th columns) that strategically and malignantly perturb the input image to fool the DNN object detector by either launching a random attack, or an object vanishing attack, an object fabrication attack or an object mislabeling attack. These attacks are stealthy as the adversarial examples generated are indistinguishable by human-perception compared to the original benign input images.

Figure 4 shows the comparison of four representative adversarial vision attack algorithms to a well trained Faster RCNN (FRCNN) model [Ren et al., 2015]. They are TOG [Chow et al., 2020d], DAG [Xie et al., 2017], RAP [Li et al., 2018], and UEA [Wei et al., 2018]. As a result, the mAP of FRCNN model drops drastically from 67.37% to 2.64 (TOG), 3.56 (DAG), 4.78 (RAP) and 18.07% (UEA) respectively. This deception-induced malfunctioning can lead to severe con-

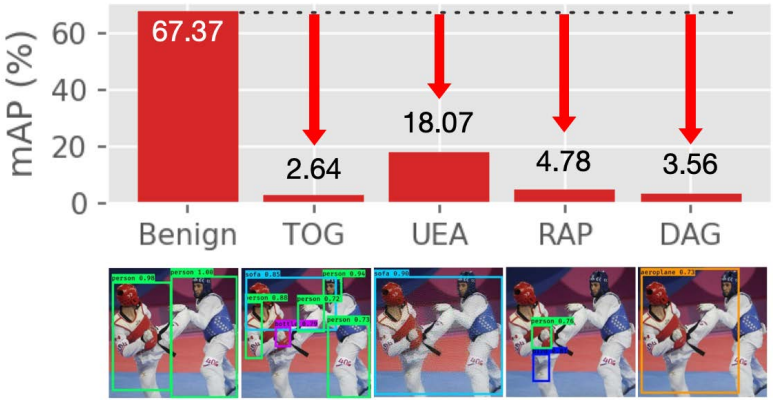


Figure 4. Visualization of the adversarial vision attacks on FRCNN trained on VOC [Chow et al., 2020b].

sequences in safety-critical edge AI applications such as autonomous vehicles [Feng et al., 2021] and intelligent surveillance [Teixidó et al., 2021].

In short, the dirty data problems such as those mentioned above can pose detrimental threats on many mission-critical AI applications, such as object recognition, self-driving cars, voice command recognition, to name a few. Different defense methods to date tend to have different robustness under different attack algorithms or different settings of attack parameters for the same attack algorithm. We argue that a robust defense solution should be independent of concrete adversarial attack algorithms and can generalize well across different datasets, different DNN algorithms, and different types of dirty data.

Existing defense approaches against adversarial examples do not generalize over different attack algorithms. We argue that neural network ensemble methods hold the potential to mitigate dirty data disruption against both out-of-distribution examples and adversarial examples. First, NN ensemble learners can leverage the complimentary wisdom of its member models to generate the consensus based committee recommendation by resolving the inherent inconsistency. Second, although both categories of dirty data are transferable from one model to another, such adversarial transferability across different DNN models is often not consistent. However, not all the ensemble learners are robust against dirty data deceptions. We conjecture that only those ensemble learners that have high failure independence can maximize the generalization performance of the ensembles in the presence of both OOD examples and adversarial examples.

3 Focal Diversity Optimized Ensemble Learning

Effective ensemble methods comprise three main tasks: (i) the creation of multiple independently trained predictive models, (ii) the ensemble pruning (selection) through measuring and enforcing error diversity, and (iii) the combination of the base models through ensemble consensus, which produces the output of the ensemble by combining the predictions made by its member models.

Creating a Base Model Pool. Conceptually, preferred base model candidates are the neural networks trained independently on the same dataset using different backbone algorithms (e.g.,

LeNet, VGG, ResNet) with good predictive performance because they make different assumptions about the data as they extract and learn different sets of hidden features of data [Liu et al., 2019]. Models trained by using different configurations of the same algorithm are good candidates, e.g., different ways of bagging data, different initial weights filters, or different settings of hyperparameters (e.g., batch size, # epochs, #iterations, learning rate functions, optimization algorithms). Figure 1 shows 10 base models for CIFAR-10 and 10 based models for ImageNet.

Ensemble Assessment. Predictive accuracy, diversity and size are the three key measures for assessment of Ensembles. Accuracy can be estimated using cross validation. There are different ways to assess diversity, some are conceptual, such as the structural NN diversity used for creating a pool of base models, and others are quantitative. We introduce focal ensemble diversity to measure the negative correlation among ensemble member models. Low negative correlation indicates that the member predictors are more failure independent and thus complement one another well. High negative correlation indicates that the ensemble composition is less efficient, and member models tend to fail on the same inputs. Size refers to the number of base models required to make up an efficient ensemble. Having a large size ensemble adds high computational overhead and large memory demands. Minimizing runtime overhead is crucial for many real applications like stream mining. More importantly, ensembles of larger size may not improve predictive performance and may even reduce ensemble accuracy if there is high negative correlation and thus low disagreement diversity among its member models.

Ensemble Pruning (a.k.a. ensemble selection) aims to increase efficiency by reducing the number of base models without sacrificing accuracy and preferably enhancing predictive performance. A baseline approach to pruning an ensemble of N models is to conduct an exhaustive search over the space of $2^N - 1$ candidate ensemble subsets, guided by some metrics, e.g., some measure of ensemble diversity [Martinez-Munoz and Suarez, 2006]. Kuncheva and Whitaker [2003] compared ten different disagreement measures and pointed out that none of them is effective as they all fall to capture an inherent correlation between ensemble accuracy and ensemble diversity. To date, how to effectively measure ensemble diversity of member networks and how to effectively prune ensembles to boost the generalization performance (accuracy) of ensemble system remains an open challenge.

Focal Ensemble Diversity and Focal Negative Correlation Measures. We describe a new approach for measuring the negative correlation among member models of an ensemble, and then computing the focal ensemble diversity [Wu et al., 2021b; Wu and Liu, 2021b]. Unlike existing metrics [Kuncheva and Whitaker, 2003], our focal diversity approach has two novel features: (i) Given the base model pool $BasePool(N)$, for each ensemble subset T_{ens} of size S, where $T_{ens} \subset BasePool(N)$, we compute S focal negative correlation scores λ_{focal} for every ensemble subset under consideration, each focal negative correlation score λ_{focal} corresponds to one of the S models, which is used as the focal model, say M_i^f $(i = 1, \ldots, S)$, and λ_{focal} is obtained by using the negative examples randomly sampled from the validation set of the corresponding focal model M_i^f. (ii) For each sub-ensemble of size S, where $S = 2, 3, \ldots, N-1$, we compute its focal ensemble diversity score, d_{ens}, by averaging the S focal negative correlation scores. Figure 5 provides an illustration of how the focal diversity for an ensemble of size $S = 4$ is computed based on the four focal negative correlation scores. We also propose to optimize the focal diversity computation by leveraging hierarchical focal diversity pruning algorithm [Wu and Liu, 2021a].

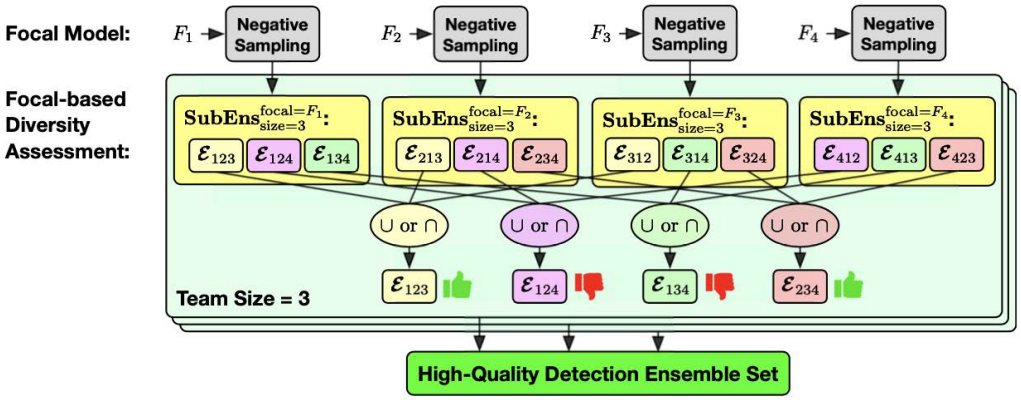


Figure 5. Visualization of the focal diversity of an ensemble of size $S = 4$.

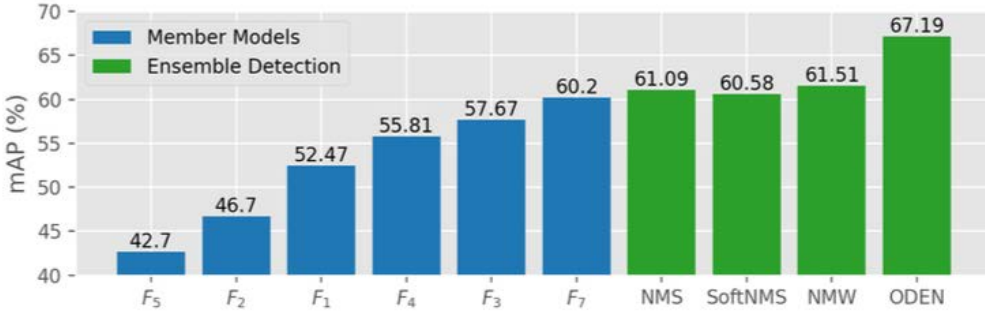


Figure 6. Comparing the focal diversity optimized ensemble with the seven member models and three conventional fusion methods to combine object detection results from multiple object detection models [Chow and Liu, 2022].

Combining Individual Predictions to Generate Ensemble Fusion Prediction. For a single regression or classification learner, there are two broad categories of techniques for combining multiple predictive models: consensus by voting or learn to combine. For voting ensemble, each model outputs a class label and/or probability distribution. The ensemble recommends the class received the most votes. The majority voting chooses the class that have more than half of the votes as the winner. The plurality voting chooses the class with the maximum number of votes as the winner. Instead of treating the base models equally, the weighed voting associates each model with a weight reflecting proportionally the classification accuracy and/or confidence of the model. Instead of using the constant weights over the input space, the learn to combine methods use an iterative weight learning algorithm, e.g., a gating network [Jacobs et al., 1991], to learn the weights for the N models for each input.

For multi-task learners, e.g., object detection models, the ensemble fusion requires to design a complex inconsistency solver, which learns to address the inconsistency in predictions for each of the three learning tasks: object existence, object bounding box (location), and object classification, from all member models of an ensemble [Chow and Liu, 2022]. Figure 6 shows the results produced by our focal diversity optimized ensemble, compared to other representative conventional object

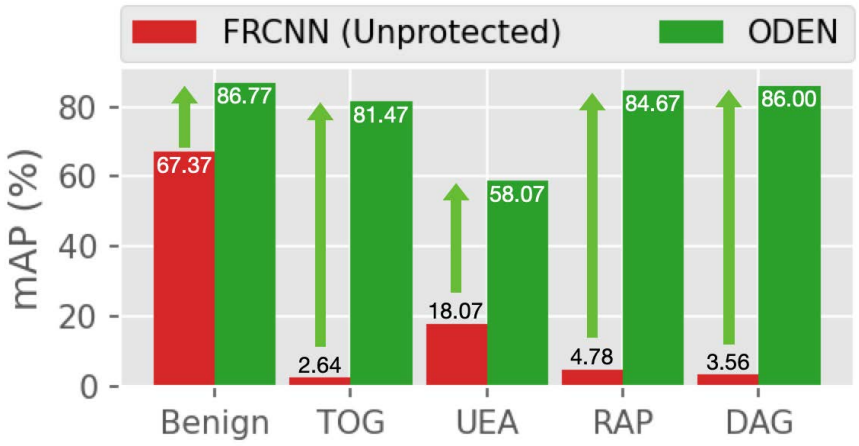


Figure 7. Visualization of the focal diversity optimized ensemble defense against the four adversarial vision attacks to the victim FRCNN on VOC.

detection fusion methods, such as NMS [Neubeck and Van Gool, 2006], SoftNMS [Bodla et al., 2017] and NMW [Zhou et al., 2017].

Figure 7 shows the effectiveness of using our focal diversity optimized ensemble method in both benign scenario (no attack) and under the four representative vision attacks (recall Figure 4). Our focal diversity optimized ensemble method offers a proactive defense methodology with built-in auto-verification and auto-repairing capability through two synergetic functional components. First, the inconsistency solver for producing robust ensemble detection results by attesting and restoring inconsistent detection results from meber models of an ensemble team. Second, our focal diversity optimized ensemble selection method can select the ensemble of high focal diversity (high failure independence) and smaller ensemble team size, hence strengthening the effectiveness of our ensemble fusion method at low computation cost. As a result, our focal diversity optimized ensemble method can provide high generalization performance under both benign scenario and vision attack scenarios.

Efficient Ensemble Prediction on Edge Devices. An efficient way to execute ensemble learner on edge device is to leverage in-expensive and light weight AI hardware, such as Intel Neural Compute Stick, NVIDIA Jetson, or Google Coral. Figure 8 shows our edge AI experiments by utilizing parallel model execution to support ensemble of N models with N Intel NCS2 sticks [Wu et al., 2021a]. The execution run-time performance of our focal diversity ensemble of seven models is on par with the execution performance of the victim model FRCNN.

4 Concluding Remarks

Neural network ensemble learning holds great potential for improving generalization performance of ML models in the presence of dirty data and concept drift. This keynote calls for research methodologies on ensemble learning methods for different types of learning tasks, ranging from classification, object detection, image segmentation, to natural language understanding, generative

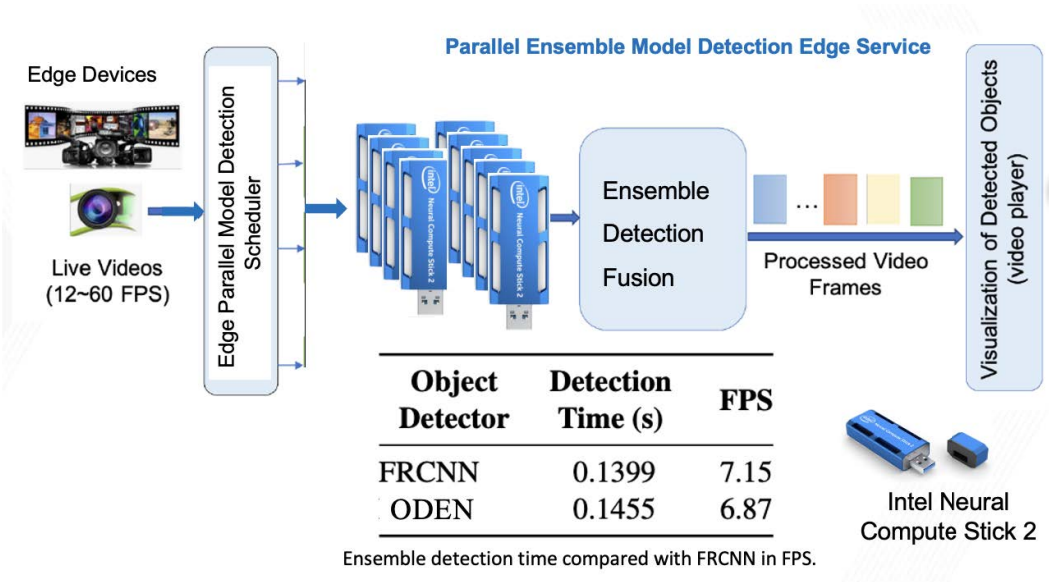


Figure 8. Parallel model execution for focal diversity optimized ensemble on edge device with multiple Intel NCS2-sticks attached.

AI, and reinforcement learning, and ensemble learning for mixed data modalities, ranging from image, video, audio to text.

Acknowledgments

The author thanks Ka Ho Chow, Wenqi Wei, Yanzhao Wu, Stacy Truex, Fatih Ilhan and Selim Tekin for their discussions and research collaborations on ensemble learning methods as well as AI privacy, security and trust against dirty data. This research is partially sponsored by NSF 1564097, NSF 2038029, NSF 2026945, an IBM faculty award and a CISCO grant. Any opinions, findings, and conclusions or recommendations ex- pressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or other funding agencies and companies mentioned above.

References

Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *ICCV*, 2017.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

L. Breiman. Arcing classifiers. *The Annals of Statistics*, 26(3):801–849, 1998.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Ka Ho Chow and Ling Liu. Protecting dnn from evasion attacks using ensemble of high focal diversity. *Technical Report, Georgia Institute of Technology*, 2022.

Ka-Ho Chow, Ling Liu, Emre Gursoy, Stacey Truex, Wenqi Wei, and Yanzhao Wu. Tog: Targeted adversarial objectness gradient attacks on real-time object detection systems. *arXiv preprint arXiv:2004.04320*, 2020a.

Ka Ho Chow, Ling Liu, Emre Gursoy, Stacey Truex, Wenqi Wei, and Yanzhao Wu. Understanding object detection through an adversarial lens. *ESORICS 2020*, 2020b.

Ka-Ho Chow, Ling Liu, Mehmet Emre Gursoy, Stacey Truex, Wenqi Wei, and Yanzhao Wu. Understanding object detection through an adversarial lens. In *Springer ESORICS*, 2020c.

Ka-Ho Chow, Ling Liu, Margaret Loper, Juhyun Bae, Mehmet Emre Gursoy, Stacey Truex, Wenqi Wei, and Yanzhao Wu. Adversarial objectness gradient attacks in real-time object detection systems. In *IEEE TPS-ISA*, 2020d.

T.G. Dietterich. Ensemble methods in machine learning. *J. Kittler and F. Roli, editors, Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy, volume 1857 of Lecture Notes in Computer Science, Springer-Verlag*, pages 1–15, 2000.

K. Eykholt, I. Evtimov, E. Fernandes, A. Rahmati B. Li, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust physical-world attacks on deep learning models. *CVPR*, 2018a.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Physical adversarial examples for object detectors. *arXiv preprint arXiv:1807.07769*, 2018b.

Di Feng, Ali Harakeh, Steven L Waslander, and Klaus Dietmayer. A review and comparative study on probabilistic object detection in autonomous driving. *IEEE TITS*, 2021.

S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias-variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *ICLR*, 2017.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.

J. Kittler. Combining classifiers: a theoretical framework. *Pattern Analysis and Applications*, (1): 18–27, 1998.

L.I. Kuncheva and C.J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51:181–207, 2003.

Yuezun Li, Daniel Tian, Xiao Bian, Siwei Lyu, et al. Robust adversarial perturbation on deep proposal-based models. *arXiv preprint arXiv:1809.05962*, 2018.

Ling Liu, Wenqi Wei, Ka-Ho Chow, Margaret Loper, Emre Gursoy, Stacey Truex, and Yanzhao Wu. Deep neural network ensembles against deception: Ensemble diversity, accuracy and robustness. In *MASS*, 2019.

G. Martinez-Munoz and A. Suarez. Pruning in ordered bagging ensembles. *23rd International Conference in Machine Learning (ICML), ACM Press*, 2006.

Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *IEEE ICPR*, 2006.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

Royi Ronen, Marian Radu, Corina Feuerstein, Elad Yom-Tov, and Mansour Ahmadi. Microsoft malware classification challenge, 2018.

Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. *ACM CCS*.

Toon Goedemé Simen Thys, Wiebe Van Ranst. Fooling automated surveillance cameras: adversarial patches to attack person detection. *CVPR COPS Workshop*, 2019.

Pedro Teixidó, Juan Antonio Gómez-Galán, Rafael Caballero, Francisco J Pérez-Grau, José M Hinojo-Montero, Fernando Muñoz-Chavero, and Juan Aponte. Secured perimeter with electromagnetic detection and tracking with drone embedded and static cameras. *Sensors*, 21(21): 7379, 2021.

Wenqi Wei and Ling Liu. Robust deep learning ensemble against deception. *IEEE Transaction on Dependable and Secure Computing (TDSC), Special Issue on Secure and Emerging Collaborative Computing and Intelligent Systems*, 2020.

Xingxing Wei, Siyuan Liang, Ning Chen, and Xiaochun Cao. Transferable adversarial attacks for image and video object detection. *arXiv preprint arXiv:1811.12641*, 2018.

Yanzhao Wu and Ling Liu. Boosting deep ensemble performance with hierarchical pruning. *ICDM*, 2021a.

Yanzhao Wu and Ling Liu. Boosting deep ensemble performance with hierarchical pruning. In *IEEE ICDM*, 2021b.

Yanzhao Wu, Ling Liu, and Ramana Kompella. Parallel detection for efficient video analytics at the edge. In *IEEE CogMI*, 2021a.

Yanzhao Wu, Ling Liu, Zhongwei Xie, Ka-Ho Chow, and Wenqi Wei. Boosting ensemble accuracy by revisiting ensemble diversity metrics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16469–16477, June 2021b.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *ICCV*, 2017.

Huajun Zhou, Zechao Li, Chengcheng Ning, and Jinhui Tang. Cad: Scale invariant framework for real-time object detection. In *ICCV Workshops*, 2017.