Robustness Analysis of Gaussian Process Convolutional Neural Network with Uncertainty Quantification

Mahed Javed, Lyudmila Mihaylova, and Nidhal Bouaynaya

Abstract—**This paper presents a novel framework for image classification which comprises a convolutional neural network (CNN) feature map extractor combined with a Gaussian process (GP) classifier. Learning within the CNN-GP involves forward propagating the predicted class labels, then followed by backpropagation of the maximum likelihood function of the GP with a regularization term added. The regularization term takes the form of one of the three loss functions: the Kullback-Leibler divergence, Wasserstein distance, and maximum correntropy. The training and testing are performed in mini batches of images. The forward step (before the regularization) involves replacing the original images in the mini batch with their close neighboring images and then providing these to the CNN-GP to get the new predictive labels. The network performance is evaluated on MNIST, Fashion-MNIST, CIFAR10, and CIFAR100 datasets. Precision-recall and receiver operating characteristics curves are used to evaluate the performance of the GP classifier. The proposed CNN-GP performance is validated with different levels of noise, motion blur, and adversarial attacks. Results are explained using uncertainty analysis and further tests on quantifying the impact on uncertainty with attack strength are carried out. The results show that the testing accuracy improves for networks that backpropagate the maximum likelihood with regularized losses when compared with methods that do not. Moreover, a comparison with a state-of-art CNN Monte Carlo dropout method is presented. The outperformance of the CNN-GP framework with respect to reliability and computational efficiency is demonstrated.**

Index Terms—**Adversarial robustness, artificial intelligence, convolutional neural networks, machine learning.**

I. Introduction

Robustness in artificial intelligence (AI) is related to reliability and explainability, especially when deep neural networks (DNNs) are applied in uncertain environments [1]. DNNs operate by sequentially learning complex representations through layers of linear computations followed by non-linear transformations. This form of hierarchical learning has, since the previous decade of AI, witnessed a giant leap in accuracy, with systems achieving near human-level performance on tasks, such as image classification [2]. Recently, there is a surge of machine learning algorithms that not only predict but also quantify the impact of uncertainties over their predictions [3]. Although it is difficult to foresee what the next big leap of AI is going to be, there is now a growing motivation towards developing AI systems that are robust to adversarial attacks [4].

Developing robust AI systems entails plenty of challenges. These include tackling human user errors, mis-specified goals, incorrect models and unmodeled phenomena [5]. Adversarial attacks can be of two types: black box or white box [6]. These attacks challenge the network's learned capabilities. Black- box attacks only have access to the inputs of the network. White-box attacks [6] on the other hand, have full access to the DNN architecture, the inputs, outputs and the gradient information in each of the nodes. Mis-specified goals often arise because the original intended AI system design goals do not meet the end-user goals. The reverse of this situation results in incorrect models. Another reason for incorrect model occurrence is also the lack of representation of model uncertainty. If a model is more uncertain at solving the problem, likely it is not suitable for the task. Model uncertainty is also referred to as *epistemic uncertainty* [7].

Finally, unmolded phenomenon challenges arise because not all AI systems can incorporate prior knowledge of everything in the environment. This phenomenon is also known as *aleatoric uncertainty* and is present within the inputs of the AI system [7]. Accounting for uncertainty in AI systems will also improve its explainability since it allows the model to explain its predictions. This is also essential for critical decision-making systems. Previous approaches to building robust AI systems rarely considered such aspects. This is the research challenge that this paper focuses on.

This paper explores the possibility of building a robust AI system with only two convolutional layers and validates it on noisy and blurred images and white-box attacks. The tests are carried on relatively simple datasets MNIST [8] and FMNIST [9], as well as on complex datasets CIFAR10 [10] and large dataset CIFAR100 [10]. The main idea is to regularize the maximum likelihood with a similarity cost function while the input images are perturbed to motivate weights that can tolerate noisy, uncertain conditions. The proposed framework trains a combined convolutional neural network (CNN) [11] feature extractor with a Gaussian process (GP) classifier [12]. The GP is introduced for two purposes, one to characterize uncertainty and the second to use the features from the CNN feature extractor for classifying the input images. Further tests on sensitivity to attack strength with uncertainty information are carried out. The uncertainty is characterized by the variance of the post-softmax sample variance sampled from the GP classifier. The CNN model transforms large complex input spaces to simple, low dimensional features for the GP to classify. The CNN-GP training is carried out based on the regularized maximum likelihood function in noisy, uncertain conditions.

Manuscript received May 11, 2021; revised March 1, 2022.

Mahed Javed and Lyudmila Mihaylova are with the Department of Automatic Control and Systems Engineering, University of Sheffield, UK (e-mail: mjaved1@sheffield.ac.uk, l.s.mihaylova@sheffield.ac.uk).

Nidhal Bouaynaya is with the Department of Electrical and Computer Engineering, Rowan University, USA (e-mail: bouaynaya@rowan.edu).

doi: 10.18178/ijmlc.2022.12.5.1097

Before the regularization the weight of the CNN-GP are updated based on this likelihood.

The main contributions of this work are highlighted below.

1) A CNN-GP framework is proposed for classification with uncertainty quantification. The framework is trained to be robust to noisy, blurred images and adversarial attacks. The performance is compared with a state-of-the-art approach with Monte Carlo dropout
2) The framework is extensively tested on four types of datasets with increasing complexity; MNIST, FMNIST, CIFAR10 and CIFAR100. The framework demonstrates that backpropagation of regularized maximum likelihood loss with similarity losses is vital for developing CNNs and DNNs with strong robustness against noisy, blurred images and white-box adversarial attacks
3) The uncertainty quantification is based on the post-softmax sample variance. The analysis charts show reduced uncertainty in predictions on noisy images. Precision-recall and ROC curves characterize the accuracy of the results
4) The proposed framework provides reliable uncertainty estimates and has an increased computational efficiency compared with the state-of-art CNN Monte Carlo dropout approach [13]. The validation is performed with increasing strength of the noisy images and white-box attacks

The rest of the paper is organized as follows. Section II gives a brief overview of recent methods from the fields of meta-learning and adversarial learning. Section III presents the proposed framework and Section IV outlines the training algorithm. This is followed by Section V which presents the robustness analysis and tests on the accuracy of the framework based on attacks on four different datasets varying in complexity and size. The uncertainty is analyzed with the post-softmax variance obtained after sampling from GP classifier. The variance information is used to sense the increase of attack strength. Section VI presents the discussion of the results and finally ends with the section on future works in Section VII.

II. Related Works

Learning to estimate prediction uncertainty is an actively developing field in Bayesian deep learning. It is practiced in many forms and under several learning monikers, of the most popular ones being meta-learning [14] and adversarial learning [15]. Meta-learning treatment of uncertainty-based learning consists of recognizing the fact that learning from uncertainty is a “meta” step operating in addition to the main learning step. On the other hand, adversarial learning treats uncertainty as means for generating attacks that may be black or white-box. There is a plethora of techniques in both regimes [16] as well as defense strategies.

However, there are a few that leverage uncertainty. Amongst these are the works of [1], which focus on the detection of attacks, while [17] and [18] focus more on their mitigation. Some methods even merge the two fields. For example, in [19], a generative adversarial network (GAN) based discrimination is used to reduce the epistemic uncertainty. In the next section, we study the literature and compare baseline CNN techniques to the proposed framework.

A. Comparison of Current Approaches

Uncertainty related research is adopted in semi-supervised tasks. These tasks entail learning from a dataset with limited labels. This is carried out in noisy conditions. Examples of this in literature can be seen practiced in [20] and [21]. The main difference in the individual approaches is that [20] adopts a global averaging scheme on DNN weights as a means of modelling noise in the labels, while [21] generates an external noise model and a student-teacher learning scheme to teach their network to be consistent in predictions under noisy conditions. Methods that involve external noise generation do not require alteration of their training architecture and are easy to scale.

Research in the field of adversarial learning, [17] and [18], aim to reduce the effects of adversarial attacks. Major differences between the approaches are that [17] uses a GAN to train their main network to resist attacks while [18] and [22] uses Bayesian methods. Specifically, [18] uses softmax variance to account for uncertainty while [22] uses Monte Carlo (MC) dropout [13]. MC dropout quantifies uncertainty by sampling via multiple forward passes and then computing the variance of these samples. GAN methods, on the other hand, do not discriminate between black-box or white-box attacks. Therefore, such methods are flexible and applicable to any form of classifier. MC dropout, on the other hand, can scale well with network architecture but at the price of computational cost. Additionally, [22] shows that softmax variance is an approximation to the measure of mutual information. Comparing this with predictive entropy obtained from MC dropout, it is proved by [22] that the mutual information is informative at detecting attacks. Here, information criteria characterize how well the uncertainty is represented and its sensitivity to adversarial attacks.

The drawbacks of the approaches [17], [18] are that GAN based methods are difficult to train since they involve optimizing two DNN models (discriminator and generator). The MC dropout is relatively slow at uncertainty computation and the quality of the uncertainty measure is dependent on the sampling rate. Another important factor is the issue of calibration. Both GAN and MC dropout methods have insufficient calibrated representation of uncertainty as opposed to the better-calibrated softmax variance in [22].

B. Comparison with Proposed Methods

The aforementioned techniques [17], [18] and [22] provide solutions in uncertainty-based robustness. However, they only consider test-time estimation of uncertainty. In this work, we confirm the theory posed by [1] and improve the methods by both [17] and [19]. The framework, proposed in this paper, is faster than [17] and [19] and less computationally expensive than [17] and [19]. This is because GANs are hard to train, and the Monte Carlo dropout methods require a long sampling time. The proposed framework uses a Gaussian process classifier that allows fast quantification of uncertainties. By backpropagating regularized maximum likelihood with similarity losses (KLD, Wasserstein and maximum correntropy) under noisy conditions, it is possible to reduce the uncertainty in the predictions.

III. Proposed Framework

A. Notations

This subsection describes the main notations (in Table I) used in this paper and especially in the CNN-GP framework, described in Section IV. The next subsections introduce both the CNN and the GP parts of the proposed framework. It presents the formal definitions and the needed concepts.

TABLE I: Notations and Definitions

Notation	Meaning
M	Total number of episodes
γ	Learning rate of the base CNN feature extractor with GP classifier
K	Number of neighbors sampled for synthetic image generation
N	Batch size
β	Kullback-Leibler divergence scaling factor
m	Episode number
λ	Lengthscale parameters of the GP classifier
A	The amplitude for the squared exponential kernel
u_i	Variational free parameters for the i^{th} batch of data
$q(u_i)$	Variational likelihood based on the i^{th} batch of data
$p(u_i)$	Expected real likelihood based on the i^{th} batch of data
θ_{CNN}^{m}	Weights of the CNN feature extractor for the m^{th}episode
θ_{GP}^{m}	Weights of the GP classifier for the m^{th}episode
x_i	Data sample from the i^{th}batch of data
y_i	Label sample from the $\mathrm{i^{th}}$ batch of data
X	4D-data tensor holding the data samples
Y	4D-data tensor holding the labels samples
D	Dataset ordered pair holding X and Y
Z	Number of units passed as features from the final layer of the CNN
σ_i^2	Epistemic variance / uncertainty for the $\mathrm{i^{th}}$ batch
$\hat{\sigma}_i^2$	Aleatoric variance / uncertainty for the $\mathrm{i^{th}}$ batch
δx_i	The difference between the $\mathrm{i^{th}}$ data point and the GP prediction
f^{GP}	The Gaussian process function
f^{CNN}	The convolutional neural network function
$\hat{y}_i^{CNN}$	Softmax prediction from the CNN base feature extractor
$\hat{y}_z^{CNN}$	Prediction from the $\mathrm{z^{th}}$ node from the CNN base feature extractor
$\mathcal{L}_{max}$	Maximum likelihood loss
$\mathcal{L}^{GP}$	Similarity loss penalizing output from the GP classifier and labels

B. Convolutional Neural Networks

CNNs are a specific type of neural networks that learn features from images in a hierarchical fashion [11]. The main idea is to use convolutional kernels that adapt to the input image. Given a loss function, learning in CNNs is performed by differentiating the outputs with respect to the loss function and updating the weights of each kernel by adding on the scaled value, via the learning rate γ, of this gradient.

The proposed framework combines a CNN feature extractor and a GP placed after it, in one architecture (see Fig. 1). The CNN has two convolution layers of 32 and 64 filters of 3×3 kernel size. The padding size of convolutional layers varies. This is because MNIST and FMNIST datasets share the same input size of 28×28×1 as opposed to CIFAR10 and CIFAR100 i.e., 32×32×3. For MNIST and FMNIST padding size is set to 2 and 1 for CIFAR10 and CIFAR100. A max-pooling layer is introduced between the second convolutional and the first dropout layer. Pooling layers downsample the features and dropout layers are used as regularizers. The fully connected layer, on the other hand, flattens the features to a 128×10 (for MNIST and FMNIST, 128×16 for CIFAR10, 128×100 for CIFAR100) feature vector. These features are then fed to the GP half of the framework discussed in the next subsection.

C. Gaussian Process

A Gaussian Process is a Bayesian nonparametric approach [12] that can represent highly nonlinear phenomena. The GP approach models a distribution over functions. Learning a GP is similar to learning in CNNs, in the sense that it involves a kernel learning process. However, the choice of the kernel and the likelihood function is problem-dependent. In the proposed framework, we use a squared exponential kernel for the kernel choice and a softmax likelihood for squashing the posterior mean of the output distribution to probabilities. For the choice of the GP model, we use Massively Scalable Gaussian Processes (MSGP) introduced in [23]. MSGPs are the preferred methods for many applications, thanks to their scalability and celebrated achievements in sparse GP models with inducing points. The computational load of computing the inverse of the covariance matrix is reduced by using an eigendecomposition of the covariance matrix to a series of Toeplitz matrices.

Within the architecture, the output from the GP is a categorical distribution, from which a $1 \times N$ vector (N is the batch size) is then estimated via maximum likelihood.

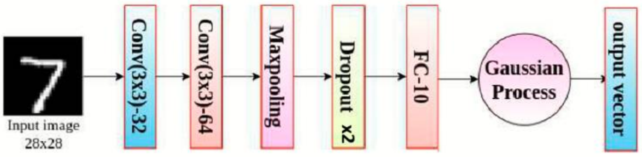


Fig. 1. The GP-CNN framework at test time. It consists of a CNN base feature extractor with a GP after it.

IV. A Convolutional Neural Network Combined with a Gaussian Process for Uncertainty Quantification

A. Training Algorithm for the Proposed Framework

Learning within the CNN-GP involves forward propagating of the predicted class labels, then followed by backpropagation of the maximum likelihood function of the GP with a regularization term added. The forward step before the regularization involves replacing the original images in the mini-batch with their close neighbouring images and then providing these to the CNN-GP to get the new predictive labels. This step is inspired by the work of [21]. The main difference is that they use this step for neighbouring labels while our work focuses on input images. Then, looping through the entire mini batch.

The regularizes loss $\mathcal{L}^{MLE} + \mathcal{L}^{SIM}$ is backpropagated. These losses allow the development of noise-tolerant weights. Three functions characterize the similarity losses $\mathcal{L}^{SIM}$: a) the Kullback-Leibler divergence (KLD), b) the Wasserstein distance and c) the maximum correntropy (MC) loss function. We formulate the losses in the next sub-section and provide the full algorithm description below. The notations that are used in the algorithm section are also provided in Section III. Algorithm 1 presented below summarizes the implemented CNN and GP framework. The similarity loss functions are described in Section IV.

ALGORITHM 1: THE CNN-GP TRAINING ALGORITHM

All experiments in this paper use the following default arguments; batch size=16, episodes=100, learning rate of GP=0.1, neighbors sampling number=11, KLD scaling factor = 1

```
Require: $M$: episodes, $\gamma$: learning rate (GP), $k$: neighbors sampling number, $N$: batch size, $\beta$: KLD scaling factor
DO: initialization of weights: $\theta^m_{CNN}$, $\theta^m_{GP}$
for $m$=0,…,$M$ do
    Sample mini batch $(x_i, y_i)$, of length $N$ from dataset $D = \{X, Y\}$ where $X$ and $Y$ are 4-D tensors holding images and labels from the entire dataset, where $x_i \in \mathbb{R}^{h\,x\,w\,x\,c}$ (image height, width and channel) and $y_i \in \mathbb{R}^{1\text{xC}}$ ($C$ is total number of classes).

    BEGIN  Update of the weights of the CNN-GP based on the maximum likelihood loss $\mathcal{L}^{MLE}$
        do → forward pass of CNN base represented as a function $f^{CNN} : x_i \rightarrow z_i$, where $z_i \in \mathbb{R}^{Z\text{xC}}$ and $Z$ is the number of hidden units' feature outputs passed from final fully-connected layer of CNN base feature extractor
        do → forward pass of GP $f^{GP}(z_i)$ to obtain the posterior likelihood $p(y_i | f^{GP}(z_i) ; \mu_i, \sigma_i^2) = \mathcal{N}(\mu_i, K_i)$
        where $\mu_i$ represents the mean of the GP and $K_i$ is the kernel (i.e., squared exponential $K_i = A \exp\left[-\frac{1}{2}\left(\frac{\delta x_i}{\lambda}\right)\right]$ and $\mathcal{N}$ represents the Gaussian distribution
        Compute the expected log likelihood to obtain max likelihood loss: $\mathcal{L}_{max} \approx \sum_{i=1}^{N} \mathbb{E}_q [\log(p(y_i | f^{GP}(z_i); \mu_i, \sigma_i^2) - \beta D_{KL}(q(u_i) \,||\, p(u_i)))]$
        Compute gradients of loss with respect to weights of CNN base feature extractor and GP : $\frac{\partial \mathcal{L}_{max}}{\partial \theta_{GP}}, \frac{\partial \mathcal{L}_{max}}{\partial \theta_{CNN}}$
        Update the parameters of GP and weights $\theta^{m+1}_{CNN}$of the CNN feature extractor for the $m^{th}$ episode: $\theta^{m+1}_{CNN} \leftarrow \theta^m_{CNN} - \gamma \frac{\partial \mathcal{L}_{max}}{\partial \theta^m_{CNN}} . \mathcal{L}_{max}$, $\theta^{m+1}_{GP} \leftarrow \theta^m_{GP} - \gamma \frac{\partial \mathcal{L}_{max}}{\partial \theta^m_{GP}} . \mathcal{L}_{max}$
    end

    BEGIN  backpropagation of regularized loss $\mathcal{L}^{MLE} + \mathcal{L}^{SIM}$
        Make synthetic images via selecting top $k$ neighbours to get $\hat{x}_i$
        do → forward pass of the CNN base feature extractor $f^{CNN}: \hat{x}_i \rightarrow \hat{z}_i$
        do → forward pass of the GP $f^{GP} : \hat{z}_i \rightarrow p(\hat{y}_i | f^{GP}(\hat{z}_i) ; \hat{\mu}_i, \hat{\sigma}_i^2) = \mathcal{N}(\hat{\mu}_i, K_{\hat{x}_i})$
        Calculate $\mathcal{L}^{MLE} + \mathcal{L}^{SIM}$ between the labels $y_i$ and the GP classifier posterior mean $\hat{\mu}_i$ from the choice of KLD, Wasserstein and maximum correntropy
        Update the new parameters of $\hat{\theta}^m_{GP}$ GP: $\hat{\theta}^m_{GP} \leftarrow \theta^m_{GP} - \gamma \frac{\partial \mathcal{L}^{MLE} + \partial \mathcal{L}^{SIM}}{\partial \theta^m_{GP}} . \mathcal{L}^{MLE} + \mathcal{L}^{SIM}$
        Update the weights of the CNN feature extractor: $\hat{\theta}^m_{CNN} \leftarrow \theta^m_{CNN} - \gamma \frac{\partial \mathcal{L}^{MLE} + \partial \mathcal{L}^{SIM}}{\partial \theta^m_{CNN}} . \mathcal{L}^{MLE} + \mathcal{L}^{SIM}$
    end → End training loop
```

B. Loss Functions

Consider two sets of probability mass functions $p(x)$ and $q(x)$ that take a data point x. Finding the shift of mass from one set to the other requires calculating the discrepancy between the two. The Kullback-Leibler divergence [24] D_{KL}, shown in (1), represents this discrepancy as a measure of entropy. It quantifies the shift of probability mass by taking the difference of entropy across the distributions.

The Wasserstein distance [25] solves the problem from the point of view of optimal transport. These problems are divided into two parts: assignment and cost. The assignment strategy determines how much mass is moved across the supports of the distributions. The cost measures the effort required for the assignment strategy. Two versions of the Wasserstein metric are used. One is an approximation (2), where matrices P and C represent assignment and cost respectively. The total cost can be obtained by taking the Frobenius inner product of the two (i.e., $\langle C, P \rangle$). The transport plan is to obtain the minimum of the product. This is subtracted from the regularized entropy in (2). Here, η is denoted as the scalar multiplier. For these experiments, we choose the default value for $\eta = 0.1$ and a quadratic distance-based cost function as an approximation to the exact Wasserstein-1 distance formulation in (3). The exact form takes the infimum of the absolute difference between the masses where γ denotes the transport plan. This work uses the differences between the pair of successive values across two masses $p(x)$ and $q(x)$ as the transport plan.

Finally, the maximum correntropy loss function [26] has also been implemented in the second backpropagation step. The maximum correntropy loss function uses a kernel to compute the difference across two variables instead of using entropy-based methods such as in KLD and Wasserstein functions. The formulation can be seen in (3). The Gaussian kernel is a popular one: $k_\sigma(p(x) - q(x))^2 = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(p(x)-q(x))^2}{2\sigma^2}\right)$, where σ^2 represents the variance of the distribution. The considered cost functions are given below.

$$\mathcal{L}^{KLD} = D_{KL}\,(p \,||\, q) = -\sum_x q(x) \log q(x) + \sum_x p(x) \log p(x) \quad (1)$$

$$\mathcal{L}^{WASS} = \, min\langle C, P \rangle - \eta \sum_x p(x) \log p(x) \quad (2)$$

$$\mathcal{L}^{WASS-FIRST} = \inf \int | \, p(x) - p(y) \, | \, \gamma(p(x) - p(y)) \quad (3)$$

$$\begin{aligned} \mathcal{L}^{MC} = \, V_\sigma\big(p(x), q(x)\big) &= \mathbb{E}\big[k_\sigma\big(p(x) - q(x)\big)\big] \\ &= \frac{1}{N} \sum_{x=1}^{N} k_\sigma(p(x) - q(x)) \end{aligned} \quad (4)$$

The term V_σ refers to the MC and $\mathbb{E}$ refers to the expected value. This measure has been proven to be less sensitive to outliers [27]. This is found in many second-order statistics measures such as cross-entropy. It is heavily studied in outlier suppression [27]. The next Section V presents results with different data sets and analyses them.

V. PERFORMANCE VALIDATION

A. Accuracy, Precision-Recall and ROC Curves

Before the experiments, the CNN-GP classifier is trained with the three different similarity losses. The purpose is to observe the accuracy as a means of performance evaluation. The average results are calculated by dividing the averaged correct samples by the total number of samples. Experiments are run ten times and accuracy values are averaged. The standard deviation is ± 2%. Then, the system is disrupted using: a) an additive white Gaussian noise (AWGN) and b) motion blur (MB). The results are compared with the system version where no similarity losses are used (i.e. without regularization). These results are presented in Table II. Next, the precision-recall and the ROC results characterize the performance of the proposed CNN-GP framework. These results are plotted for each dataset side to side in Fig. 2. The average precision (AP) and ROC area are obtained by averaging the individual curve entities. They help in grouping entities that give similar results and make it easy to read the curves individually.

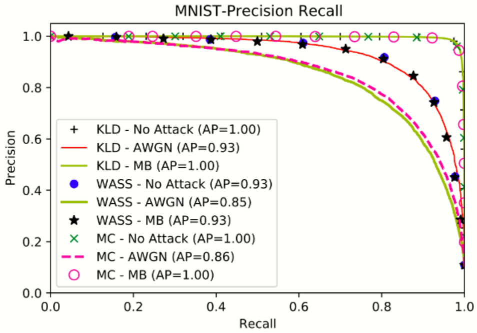
TABLE II: PERFORMANCE VALIDATION BASED ON TEST ACCURACY FOR EACH ATTACK TYPE ON THREE DATASET TYPE

MNIST	No Attack (%)	AWGN (%)	Motion Blur (%)
No regularization	88	51	65
KLD	97	89	72
WASS	86	77	70
MC	97	78	75
WASS-FIRST	97	85	88

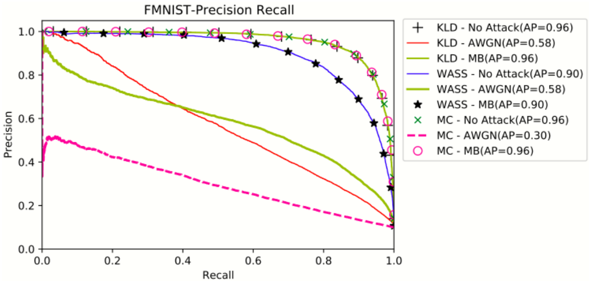
Fashion-MNIST	No Attack (%)	AWGN (%)	Motion Blur (%)
No regularization	85	32	12
KLD	88	53	76
WASS	81	56	72
MC	89	35	80
WASS-FIRST	89	54	77

CIFAR10	No Attack (%)	AWGN (%)	Motion Blur (%)
No regularization	67	10	11
KLD	73	26	38
WASS	40	28	28
MC	65	25	38
WASS-FIRST	68	26	40

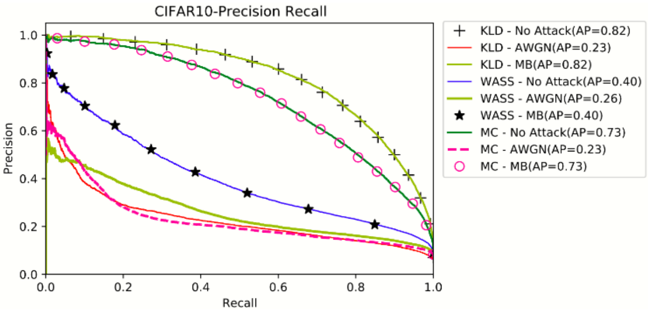
CIFAR100	No Attack (%)	AWGN (%)	Motion Blur (%)
No regularization	24	10	8
KLD	30	10	12
WASS	27	10	10
MC	28	8	8
WASS-FIRST	27	10	10



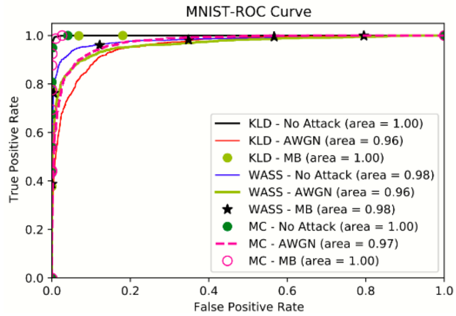
(A)



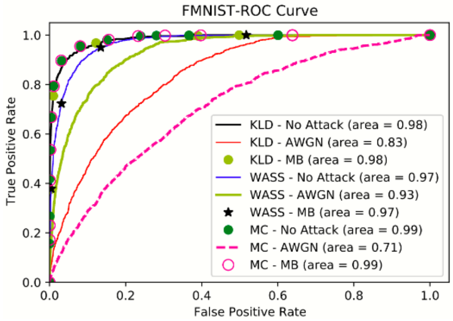
(B)



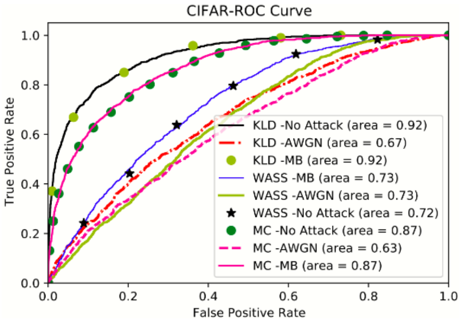
(C)



(D)



(E)

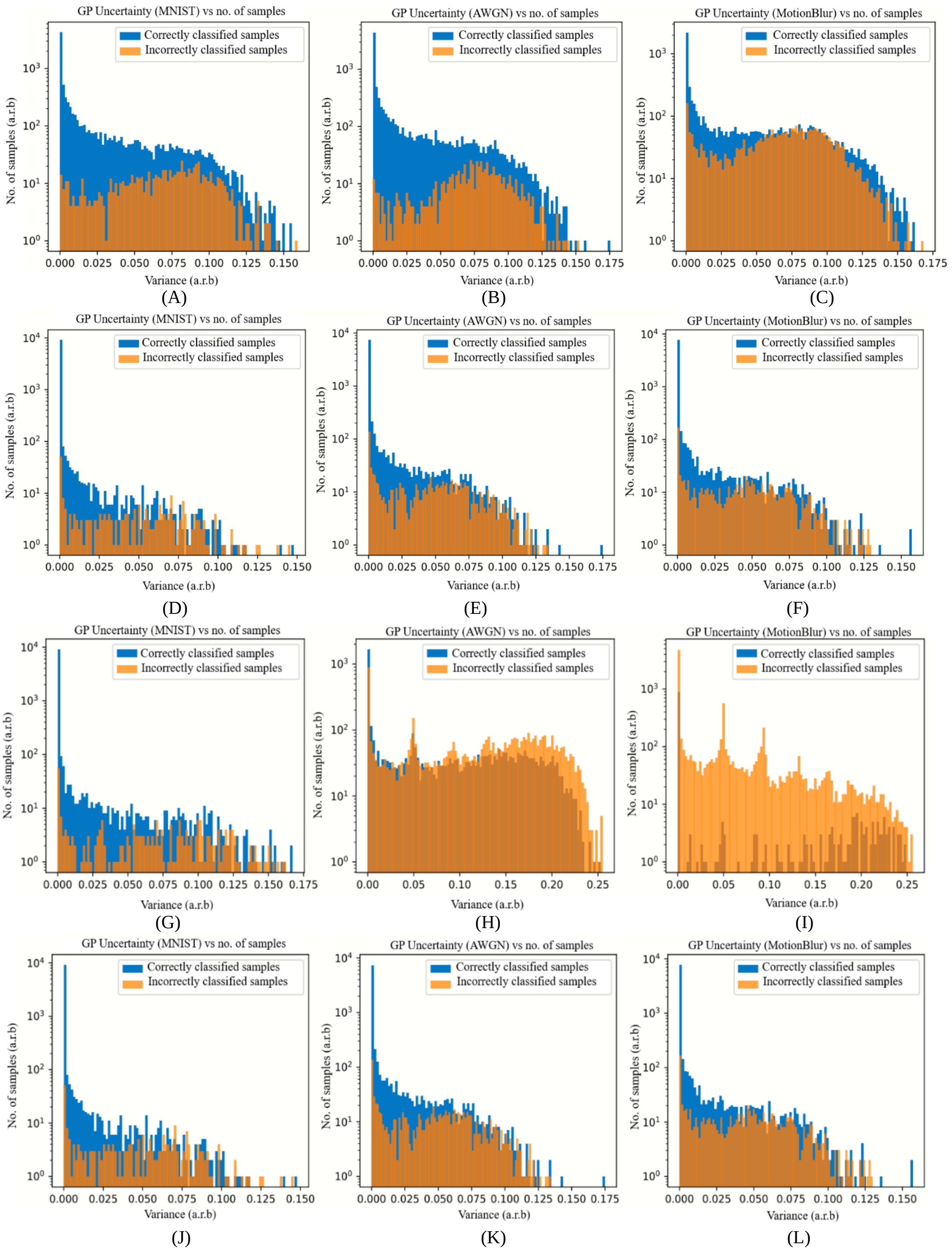


(F)

Fig. 2. The respective precision-recall and ROC curves for CNNGP framework trained on MNIST, FMNIST and CIFAR and on three loss functions; KLD, WASS and MC. Each plot considers three attack configurations; no attack, a gaussian noise and motion blurring. A) and B) show precision-recall and ROC curves for MNIST dataset, C) and D) for FMNIST and E) and F) for CIFAR-10.

B. *Uncertainty Analysis*

To further test the hypothesis, considering MNIST only, the output mean predictions from the GP classifier and the post-softmax sample variance are plotted as bar graphs. As shown in Fig. 3. The purpose of this experiment is to demonstrate the improved performance with the reduced amount of uncertainty on AWGN and motion blurring. Every time the label is correct, the appropriate variance is computed from the likelihood. Blue bars represent the variance of correct samples and orange for the incorrect. This is carried for each of the samples in the test set (10000 MNIST images).



(A) (B) (C)

(D) (E) (F)

(G) (H) (I)

(J) (K) (L)

Fig. 3. Output variance plots from GP classifier for MNIST dataset for four configurations. The first row considers the case of the model trained without any regularization from similarity losses, the second row is for GP classifier trained on KLD similarity loss, the third row for Wasserstein distance and the fourth for maximum correntropy. Each of the columns represents the black-box attack types, the first column is for clean MNIST images, the second column for white Gaussian noise and the third for motion blurring.

The proposed framework is tested with different data sets. One case study is on clean MNIST images (column 1), the other two on MNIST corrupted by AWGN (column 2) and MB (column 3). On Fig. 3 the dense blue bars indicate accurate model results, whereas the orange bars indicate incorrectly classified samples. If the predictions have high orange bars than the blue ones, the algorithm is more susceptible to attacks.

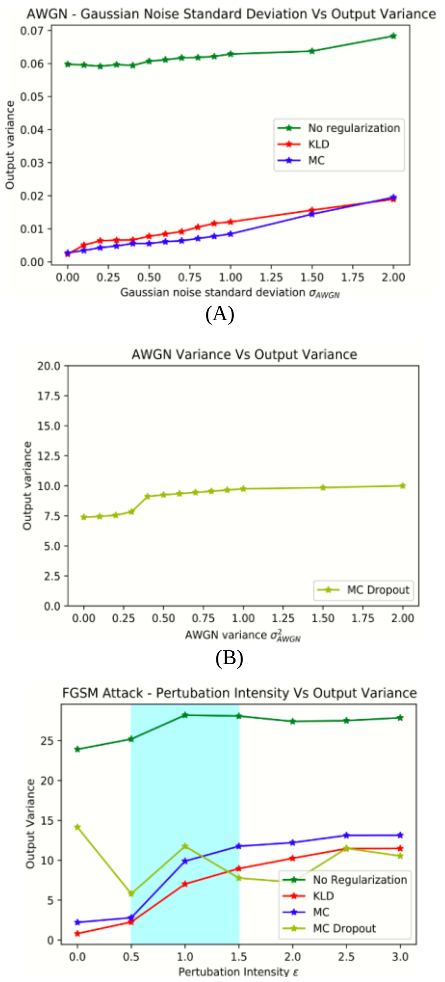
C. Variance Sensitivity to Attack Strength

To test the sensitivity of the GP classifier to AWGN and white-box attacks, two cases are shown in Fig.4A and Fig. 4B. Fig. 4A presents results with input MNIST images after being perturbed with AWGN. The input images are fed to the GP classifier and the post-softmax sample variance from the classifier is obtained. Fig. 4A presents the softmax sample output variance with respect to the σ^2_{AWGN} of the AWGN, varying in range 0.0 to 2.0.

We then test the system with the white-box attack fast gradient sign method (FGSM) [27]. This particular method works by computing the gradients of the output from the CNN feature extractor with respect to the image through a sign function to generate a new image that is imperceptible to the human eye. However, it can easily mislead the system.

The strength of the attack is denoted by ϵ that increases the level of perturbation. The highlighted region in Fig. 4C denotes the vital change of state in the system that can alert the system of the attack. This serves as a region where a high variance can lead to early detection of the attack before its intensity builds over time. Beyond this region, any change in variance would not be beneficial for a safety-critical system.

The proposed CNN-GP approach is also compared with the standard MC dropout method [13] and results are presented in Fig. 4B. The MC dropout results are obtained by isolating the pretrained CNN feature extractor and running forward passes 100 times. From this, the variance is computed and later averaged across the samples.



(A)

(B)

(C)

Fig. 4. Output variance computed from the GP classifier compared with the strength of both the additive white Gaussian noise in A), similarly for MC dropout in B) and fast-gradient sign method in C.

TABLE III: RUN TIME ANALYSIS FOR PROPOSED MODEL CNN-GP AND MC DROPOUT

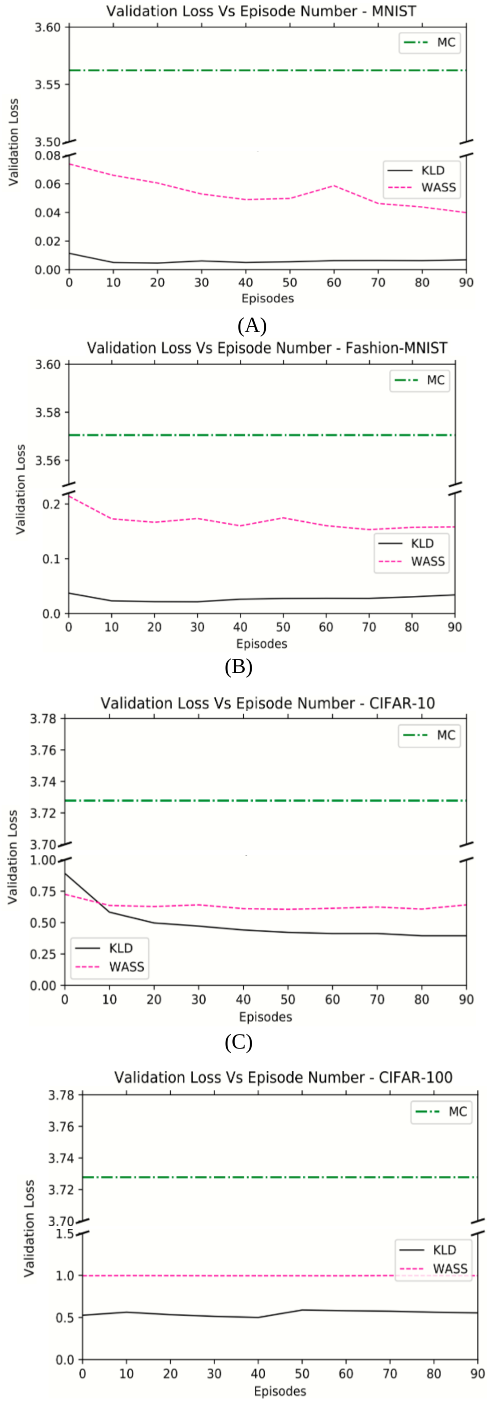
Model Type	Run Time (minutes)
CNN-GP	1.18
MC dropout	7.27

D. Computational Time

The computational time of the proposed CNN-GP framework is compared with the MC dropout method [13]. Both models provide output variance information on simple MNIST input images. The sampling rate for MC dropout method is set to 100. The respective run-time for each is then computed on the University of Sheffield provided GPU cluster (NVIDIA K80). The testing time is measured in minutes and the results are tabulated in Table III.

E. Convergence of Similarity Loss Functions in CNN Only Training

In this experiment, the convergence of the similarity loss functions is studied. This involves training the CNN component of the CNNGP framework to classify the four datasets: MNIST, Fashion-MNIST, CIFAR10 and 100. The training. The training is carried out for a total of 100 episodes. Every 10th episode, the validation loss is observed and plotted as shown in Figures 5(A)-(D).



(A)

(B)

(C)

(D)

Fig. 5. Validation losses observed during training CNN on the three similarity losses for 100 episodes. (A) shows validation losses during training CNN on MNIST, (B) for Fashion-MNIST, (C) and (D) for CIFAR-10 and CIFAR-100 respectively.

A further test to obtain the test accuracies on each of the four datasets for the CNN training is shown in Table IV. The results were run 100 times and averaged. This was carried out for all three losses. During training, it is observed that placing softmax activation after the final layer of CNN and taking the log of the probabilities allows the KLD loss to converge, but not for WASS, which converges when the final activation is switched to hyperbolic tangent. However, the MC loss does not converge with neither softmax nor hyperbolic tangent.

TABLE IV: Test Accuracy of the CNN Component Trained on KLD, WASS and MC On MNIST, Fashion-MNIST, CIFAR-10 and CIFAR-100

Dataset Type	KLD	WASS	MC
MNIST	98.75	96.48	8.92
Fashion-MNIST	90.69	84.92	10.01
CIFAR-10	73.83	27.00	5.84
CIFAR-100	28.57	1.00	1.08

VI. Discussion

Considering the results from Table II, we see that when there is no attack, the CNN-GP configurations that backpropagate regularized losses, excluding the case with the Wasserstein metric (WASS), perform better than without backpropagation (no regularization). This confirms that the CNN-GP with the regularized loss functions demonstrates reliable performance. This is supported further by the uncertainty charts in Fig. 3 where the uncertainty measures of KLD (row 2) and MC (row 4) have lower bar heights for incorrect sample variance (orange bars) than those for the cases that do not use any form of regularization (row 1).

We further see that the prediction results with the Wasserstein metric are comparable with the other data, regardless of the attack when tested on the complex CIFAR10 dataset (40%), it performs rather poorly than expected. This agrees with the hypothesis of [28] which claims that the Wasserstein metric yields biased gradients that have a higher chance of leading to a false local minimum than the KLD during optimization.

The exact Wasserstein metric [25] (WASS-FIRST) outperforms KLD in MNIST and Fashion-MNIST examples. The approximated version of Wasserstein distance, on the other hand, faces loss in performance. This is further supported by the precision-recall diagram for the approximated Wasserstein metric for all attacks. These show that the precision for these methods slowly drop when the dataset complexity is increased (from MNIST to CIFAR-10). This is further supported in the drop of the true positive rate of the ROC curve for MC in Fig. 2D-F for both the case of AWGN and MB.

In order to characterize the robustness of the approaches, the recall function is calculated. Precision is heavily affected by uncertainties and impacts the results of all methods. However, the approaches with the MC dropout and KLD maintain a good level of precision despite having poor recalls (e.g., in AWGN attacks for MC and KLD). Hence, it is possible to diagnose the recall aspect as a measure of sensitivity to the attack.

Then, considering the MC and KLD results, it is evident that using these losses results in high accuracies in motion blurring when compared with the Wasserstein metric results. The performances of the MC and KLD are similar. This is further evident in Fig. 3 where uncertainty charts for both KLD and MC have a greater number of correct sample variance (blue) as compared to those for the Wasserstein metric (row 3). For MC, this is expected since this type of loss is ideal for robust algorithm design. This is further supported in Fig. 2 where the precision-recall for both KLD and MC for motion blurring (MB) remains the highest as the dataset complexity increases (MNIST to CIFAR10).

Regarding the variance sensitivity to attack strength, it can be seen from Fig. 4A and Fig. 4C that CNN-GP trained on the MC similarity loss is more responsive than both KLD as well as the no regularization configuration. This also demonstrates that the MC is suitable for robust algorithm design. The graphs show that both the MC and the KLD functions, start with higher confidence in predictions (i.e., low variance) before the attack strength is increased when compared to the case without regularization. This confirms both our hypothesis and our results in Fig. 3 that backpropagation of regularized maximum likelihood loss in the CNN-GP framework reduces the impact of uncertainties and attacks on the classification results and characterizes the model's confidence. For the MC dropout method, it is seen from both Fig. 4B and Fig. 4C that this model is not representing the uncertainty estimates well when compared with the CNN model. Hence, it is not reliable for uncertainty quantification. The computational complexity of the compared approaches is characterized by Table III which shows that the MC dropout method is much slower than the CNN-GP framework.

The convergence properties of the similarity losses on a CNN only component of the CNNGP shows that KLD converges only when the log of softmax probabilities is taken, this confirms with the equation (1). WASS converges if the hyperbolic tangent is used. This was initially spotted in the experiments when the input images were normalized in both negative and positive ranges, using hyperbolic tangent allowed gradients to flow in both negative and positive ranges.

VII. Conclusions and Future Works

This paper proposes a CNN-GP framework that can characterize the impact of uncertainties on the classification results. Three loss functions – the Kulback-Leibler divergence, the Wasserstein distance, and the maximum correntropy are used for regularization CNN-GP and their performance is compared. The GP layer serves for quantifying the uncertainty. A small variance corresponds to a small uncertainty, a high variance means high uncertainty and hence means that the classification result cannot be trusted. The proposed CNN-GP framework is compared with a Monte Carlo dropout and it is shown that the CNN-GP is more efficient than the MC dropout method, especially with respect to computational time. The main limitation of the framework is that it is not able to get high accuracies on large and complex datasets e.g. CIFAR10 and CIFAR100. That is pointing to architecture issues more than the algorithm since the state-of-the-art architecture for CIFAR10 uses up to more than 15 convolutional layers [29].

In the future, we will focus on training large complex networks. Also, consider the possibility of feeding the CNN feature extractor as a covariance kernel to the GP. This may be computationally more feasible and may also improve the uncertainty representation in the GP since it will give the GP a holistic view of the impact of the dataset on the performance of the CNN. This work also investigates the relationship between reliable AI and robust AI via backpropagation of maximum likelihood loss regularized with the three similarity losses and leverages information to improve AI reliability.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Author Mahed Javed contributed to the main idea behind the proposed framework, conducted the experiments, responsible for the write-up of the paper as well as the amendment. Authors Lyudmila Mihaylova and Nidhal Bouaynaya contributed with ideas to the paper and improved its presentation. Lyudmila Mihaylova is also a PhD advisor.

Acknowledgment

We are grateful to UK EPSRC for funding this work through EP/T013265/1 project NSF-EPSRC:ShiRAS. Towards Safe and Reliable Autonomy in Sensor Driven Systems and UK Research and Innovation (UKRI) Trustworthy Autonomous Systems (TAS) programme [EPSRC Ref: EP/V026747/1]. This work was also supported by the National Science Foundation under Grant USA NSF ECCS 1903466.

References

[1] R. Tomsett, A. Widdicombe, T. Xing, S. Chakraborty, S. Julier, P. Gurram, R. Rao, and M. Srivastava, "Why the failure? How adversarial examples can provide insights for interpretable machine learning," in *Proc. the 21st International Conference on Information Fusion*, Cambridge, UK, 2018, pp. 838-845.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. the 26th Advances in Neural Information Processing Systems*, Harrahs and Harveys, Lake Tahoe, 2012, pp. 1097-1105.

[3] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with Bernoulli approximate variational inference," arXiv preprint arXiv:1506.02158, 2015.

[4] G. Marcus, "The next decade in AI: Four Steps towards robust artificial intelligence," eprint arXiv:2002.06177, February 2020.

[5] T. G. Dietterich, "Steps towards robust artificial intelligence," *AI Magazine*, vol. 3, pp. 3-24, 2020.

[6] N. Narodytska and S. Kasiviswanathan, "Simple black-box adversarial attacks on deep neural networks," in *Proc. the 30th IEEE Conference on Computer Vision and Pattern Recognition Workshops*, IEEE Press, Honolulu USA, 2017, pp. 1310-1318.

[7] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Proc. the 31st Advances in Neural Information Processing Systems*, Long Beach CA, 2017, pp. 5574-5584.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proc. the IEEE*, 1998, vol. 11, 86, pp. 2278-324.

[9] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," CoRR. abs/1708.07747, 2017.

[10] A. Krizhevsky and G. E. Hinton, "Learning multiple layers of features from tiny images," Technical Report TR-2009, University of Toronto, Toronto, 2009.

[11] C. M. Bishop, *Neural Networks for Pattern Recognition*, 1st ed. Oxford University Press, 1995, ch. 1, pp. 5-20.

[12] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, 2006, ch. 1, pp. 10-15.

[13] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: representing model uncertainty in deep learning," in *Proc. the 33rd International Conference on Machine Learning*, New York, USA, 2016, pp. 1050-1059.

[14] J. Schmidhuber, "Evolutionary principles in self-referential learning, or on learning how to learn: The meta-meta-meta-... hook," Ph.D dissertation, Faculty of Computer Science, Technical University of Munich, Munich, Germany, 1987.

[15] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *Proc. the 5th International Conference on Learning Representations*, Palais des Congrès Neptune, Toulon, France, 2016, pp. 1-17.

[16] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay. Adversarial attacks and defences: A survey. [Online]. Available: https://arxiv.org/pdf/1810.00069.pdf

[17] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," in *Proc. the 6th International Conf. on Learning Representations*, Vancouver, Canada, 2018, arXiv:1805.06605.

[18] A. Harakeh. (September 2017). *Adversarial Robustness of Uncertainty Aware Deep Neural Networks*. [Online]. Available: http://www.cs.toronto.edu/~chechik/courses19/csc2125/project/ali-final.pdf

[19] M. Javed and L. S. Mihaylova, "Leveraging uncertainty in adversarial learning to improve deep learning based segmentation," in *Proc. the 13th Symposium Sensor Data Fusion Trends, Solutions and Applications*, IEEE, Bonn, Germany, 2019, pp. 1-8.

[20] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. the 31st Advances in Neural Information Processing Systems*, Long Beach CA, 2017, pp. 1195-1204.

[21] J. Li, Y. K. Wong, Q. Zhao, and M. S. Kankanhalli, "Learning to learn from noisy labelled data," in *Proc. the 33rd IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'19)*, IEEE Press, Long Beach CA, 2019, pp. 5051-5059.

[22] L. Smith and Y. Gal, "Understanding measures of uncertainty for adversarial example detection," in *Proc. the 34th Conference on Uncertainty in Artificial Intelligence*, Monterey, California, USA, 2018, pp. 1-10.

[23] A. G. Wilson, C. Dann, and H. Nickisch, "Thoughts on massively scalable gaussian processes," arXiv preprint arXiv: 1511.01870, 2015.

[24] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79-86, 1951.

[25] I. Olkin and F. Pukelsheim, "The distance between two random vectors with given dispersion matrices," *Linear Algebra and its Applications*, vol. 48, pp. 257-263, 1982.

[26] Y. Qi, Y. Wang, J. Zhang, J. Zhu, and X. Zheng, "Robust deep network with maximum correntropy criterion for seizure detection," *BioMed Research International*, Article ID 703816, 2014.

[27] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv: 1312.6572, 2014.

[28] M. G. Bellemare, I. Danihelka, W. Dabney, S. Mohamed, B. Lakshminarayanan, S. Hoyer, and R. Munos. (May 2017). The Cramer distance as a solution to biased Wasserstein gradients. [Online]. Available: https://arxiv.org/pdf/1705.10743.pdf

[29] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (BiT): General visual representation learning," in *Proc. the 16th European Conference on Computer Vision*, 2020, pp. 491-507.

Mahed Javed is currently with the Department of Automatic Control and Systems, Sheffield, UK. Currently he is working as a research student. He obtained his undergraduate degree in engineering from University of Liverpool, UK in 2016 and postgraduate degree from the University of Sheffield, UK in 2017.

He has worked as a software engineer during his university study period more than once. Firstly, as the software designer for IMechE Unmanned Aerial Vehicle challenge from 2015 to 2016. Secondly, as the software developer for the Sheffield University Nova Balloon Telescope (SUNBYTE). His research interests include machine learning and deep learning in the applications including computer vision and uncertainty quantification.

Lyudmila Miahylova is currently a professor of signal processing and control with the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK. Her research interests include machine learning and autonomous systems with various applications such as navigation, surveillance, and sensor network systems. She is an associate editor for the IEEE Transactions on Aerospace and Electronic Systems and for the Elsevier Signal Processing Journal. She was the president of the International Society of Information Fusion (ISIF) from 2016 to 2018. She is on the Board of Directors of ISIF. She was the program chair of the International Conference on Information Fusion 2020 in South Africa, the general vice-chair in 2018 at Cambridge, UK of the IET Data Fusion and Target Tracking 2014 and 2012 Conferences, program co-chair for the 19th International Conference on Information Fusion 2020, 2016, and others. She is a senior IEEE member.

Nidhal Buoaynaya holds a Ph.D. in electrical and computer engineering (ECE) and an M.S. in pure mathematics from the University of Illinois at Chicago. She is a professor of ECE and the director of Rowan’s Artificial Intelligence Lab (RAIL). She is currently serving as the associate dean for Research and Graduate Studies of the Henry M. Rowan College of Engineering. Her research interests are in big data analytics, machine learning and mathematical optimization. She co-authored more than 100 refereed journal articles, book chapters and conference proceedings. Dr. Bouaynaya won numerous Best Paper Awards, the most recent was at the 2019 IEEE International Workshop on Machine Learning for Signal Processing.

Her research is primarily funded by the National Science Foundation, the National Institutes of Health (NIH), and industry. She is also interested in entrepreneurial endeavors. She is the Co-founder and Chief Executive Officer (CEO) of MRIMATH, LLC, a start-up company that uses artificial intelligence to improve patient oncology outcome and treatment response.