Robustness Analysis of Gaussian Process Convolutional Neural Network with Uncertainty Quantification

Mahed Javed, Lyudmila Mihaylova, Nidhal Bouaynaya

***Abstract*—This paper presents a novel framework for training convolutional neural networks (CNNs) to quantify the impact of gradual and abrupt uncertainties in the form of adversarial attacks. Uncertainty quantification is achieved by combining the CNN with a Gaussian process (GP) classifier algorithm. The variance of the GP quantifies the impact on the uncertainties and especially their effect on the object classification tasks. Learning from uncertainty provides the proposed CNN-GP framework with flexibility, reliability and robustness to adversarial attacks. The proposed approach includes training the network under noisy conditions. This is accomplished by comparing predictions with classification labels via the Kullback-Leibler divergence, Wasserstein distance and maximum correntropy. The network performance is tested on the classical MNIST, Fashion-MNIST, CIFAR10 and CIFAR 100 datasets. Further tests on robustness to both black-box and white-box attacks are also carried out for MNIST. The results show that the testing accuracy improves for networks that backpropogate uncertainty as compared to methods that do not quantify the impact of uncertainties. A comparison with a state-of-art Monte Carlo dropout method is also presented and 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 by 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]. The first half of this decade saw a surge of machine learning algorithms which encouraged the development of DNNs 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, misspecified 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. Misspecified 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 explicate 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 both white and black-box attacks. The tests are carried on relatively simple datasets MNIST [8] and FMNIST [9], as well as on complex datasets CIFAR10 [10] and on large dataset CIFAR100 [10]. The main idea is to use similarity cost as a tool to backpropagate the uncertainty information. This has a regularization effect on the loss functions. The entire problem of learning from uncertainty is casted as an example of backpropagation. The proposed framework trains a simple convolutional neural network (CNN) [11] feature extractor with a Gaussian process (GP) classifier [12] at a higher level. The GP is introduced for two purposes, one to characterize uncertainty and second to use the features from CNN for classifying the input images. The uncertainty is characterized by the variance of the GP. The CNN model transforms large complex input spaces to simple, low dimensional features for the GP to interpret.

The CNN-GP training is carried out in two stages: backpropagation of epistemic uncertainty and then of aleatoric uncertainty. The validation results demonstrate that these two stages influence each other and cannot achieve good results as isolated training materials. The main contributions of this work are highlighted below.

• A CNN-GP framework is proposed for classification and uncertainty quantification. The framework performance is validated both with gradual and abrupt uncertainties (random attacks in the data) and is compared with a state-of-the-art approach with dropout

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 (email: bouaynaya@rowan.edu)

• The framework has been extensively tested on four types of datasets with increasing complexity; MNIST, FMNIST, CIFAR10 and CIFAR100. The framework demonstrates that backpropagation of uncertainty is vital for developing CNNs and DNNs with strong robustness against black-box and white-box adversarial attacks

• The uncertainty quantification is based on the GP variance. The analysis charts that show reduced uncertainty in predictions. Precision-recall and receiver operating characteristics (ROC) curves characterize the accuracy of the results

• The proposed framework provides reliable uncertainty estimates and has an increased computational efficiency compared with a state-of-art Monte Carlo dropout approach [23]. The validation is performed with increasing strength of the black-box and white-box attacks

• The paper shows that explainable AI is linked to robust AI, such that robustness in AI can be achieved by accounting for uncertainty measures in both the model and datasets

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 in deep learning. Section III presents the proposed framework. Followed by Section IV on robustness analysis and tests on the accuracy of the framework. These tests include black-box and white-box attacks on four different datasets varying in complexity and size. The propagated uncertainty is analyzed with the GP variance, precision-recall and ROC curves. The variance information is further tested with the increase of attack strength. Section V presents discussion of the results and finally ends with the section on future works in Section VI.

II. Related Works

Learning from uncertainty is an actively developing field in Bayesian deep learning. It is practised in many forms and under several learning monikers, of the most popular ones being meta-learning [13] and adversarial learning [14]. Meta-learning treatment of uncertainty-based learning consists of recognizing the fact that learning from uncertainty is a "meta" step operating on top of the main learning step (i.e. backpropagation of gradients). 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 [15] as well as defence strategies. However, there are a few that leverage uncertainty. Amongst these are the works of [1], which focus on the detection of attacks, while [16] and [17] focus more on their mitigation. There have even been some methods that merged the two fields. For example, in [18], a generative adversarial network (GAN) based discrimination is used to backpropagate epistemic uncertainty. In this section, we study the literature and compare the latest techniques to our approach.

A. Comparison of Current Approaches

Uncertainty related research in meta-learning is usually adopted in semi-supervised tasks. These tasks entail learning from a dataset with limited labels. This is also carried out in noisy, uncertain conditions. Examples of this in literature can be seen practised in [19] and [20]. The main difference in the individual approaches is that [19] adopts a global averaging scheme on DNN weights as a means of modelling noise in the labels, while [20] generates an external noise model and a student-teacher learning scheme to teach their network to be consistent in predictions. Methods that involve external noise generation do not require alteration of their training architecture. They are also easy to scale.

Research in the field of adversarial learning, [16] and [17], 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 [21] and [22] uses Bayesian methods. Specifically, [21] uses softmax variance to account for uncertainty while [22] uses Monte Carlo (MC) dropout. MC dropout quantifies uncertainty by sampling via multiple forward passes and then computing the variance from these samples [23]. GAN methods, on the other hand, don't 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, [21] 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 [21] that mutual information is more informative at detecting attacks. Here, information criteria characterize how well the uncertainty is represented and its sensitivity to adversarial attacks.

The drawbacks of these approaches [16], [17] are that GAN based methods are difficult to train since they involve optimizing of two separate 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 poorly calibrated representation of uncertainty as opposed to the better calibrated softmax variance in [21].

B. Comparison with Proposed Methods

The aforementioned techniques [16], [17] and [21] provide sound solutions in uncertainty-based robustness. However, they only consider the forward propagation of uncertainty. In this work, we confirm the theory posed by [1] and improve the methods by both [16] and [18] which are indirectly accomplishing backpropagation of uncertainty. The framework, proposed in this paper, is faster than [16] and [18] and less computationally expensive. This is because GANs require training two separate networks, and the MC dropout methods require long sampling time. The proposed framework uses a Gaussian process classifier that allows fast quantification of uncertainties. By backpropagating the uncertainty information, it is possible to reduce the uncertainty in the predictions as well as improve the sensitivity of the framework to adversarial attack strength.

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 our proposed framework. Dealing separately their formal definitions and descriptions.

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 w.r.t the loss function and updating the weights of each kernel by adding on the scaled value (via learning rate γ) of this gradient.

The proposed framework combines a CNN feature extractor and a GP after it, in one architecture (Figure 1). The CNN has two convolution layers of 32 and 64 filters of 3x3 kernel size. The padding size of convolutional layers varies. This is because MNIST and FMNIST datasets share the same input size of 28x28x1 as opposed to CIFAR10 and CIFAR100 i.e. 32x32x3. 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 layer and the final fully-connected layer. Pooling layers downsample the features and dropout layers are used as a regularizer. The fully connected layer, on the other hand, flattens the features to a 128x10 (for MNIST and FMNIST, 128x16 for CIFAR10, 128 x 100 for CIFAR100) feature vector. These features are then fed to the GP half of the framework discussed in the next sub-section.

C. Gaussian Process

A Gaussian Process is a Bayesian nonparametric approach [12] which 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 respectively the likelihood function is problem-dependent. In our 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).

MSGPs are the preferred methods for many applications, thanks to their scalability. MSGPs were introduced in [24] and have 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 1xN vector (N is the batch size) is then estimated via maximum likelihood.

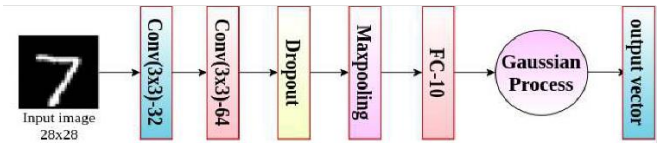


Fig. 1. The GP-CNN framework at test time. It consists of a CNN base feature extractor with a GP after it.

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 ith 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 ith batch
$\hat{\sigma}_i^2$	Aleatoric variance / uncertainty for the ith batch
δx_i	The difference between the ith 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 zth 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

IV. A CONVOLUTIONAL NEURAL NETWORKS COMBINED WITH A GAUSSIAN PROCESS AND LOSS FUNCTIONS FOR UNCERTAINTY QUANTIFICATION

A. Training Algorithm for the Proposed Framework

The training algorithm for the proposed framework consists of two halves a) backpropagation of epistemic uncertainty and b) backpropagation of aleatoric uncertainty. Both are carried out independently. In step a), the prediction from the GP classifier is compared with the labels using the maximum likelihood $\mathcal{L}_{max}$. The error obtained is then backpropagated by the parameters of the GP (the lengthscale λ and amplitude σ_E) and the CNN (convolutional layers). For inferring, we use the approximated variational inference since categorical likelihood is used for the classification.

In step b), synthetic training samples are created. This step is inspired by the work of [20] where randomly sampled mini-batches are ranked. This is proceeded by the random selection stage where the top k nearest neighbors of the mini-batch samples are selected to replace the original samples. These synthetic samples are then fed to both the CNN and GP. Similarly, the loss function $\mathcal{L}^{GP}$ is used to backpropagate aleatoric uncertainty by encouraging GP classifier to remain consistent in its predictions. These losses encourage the development of noise-tolerant weights and also have a regularization effect. Three functions characterize similarity losses: 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 for characterizing the uncertainties. Different loss functions are used and these are described in Section IV.

All experiments in our paper use the following default arguments; batch size=16, episodes=100, learning rate of GP=0.1, neighbors sampling no.=10, KLD scaling factor = 1

Require: M: episodes, γ: learning rate (GP), k: neighbors sampling no., N: batch size, β: KLD scaling factor

DO: initialization of weights: θ_{CNN}^{m}, θ_{GP}^{m}

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 $x_i \in \mathbb{R}^{h \times w \times c}$ (image height, width and channel) and $y_i \in \mathbb{R}^{1 \times C}$ (C is total number of classes).

BEGIN backpropagation of epistemic uncertainty σ_i^2

do → forward pass of CNN base represented as a function $f^{CNN} : x_i \rightarrow z_i$, where $z_i \in \mathbb{R}^{Z \times C}$ 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 μ_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 : $\mathcal{L}_{max} \approx \sum_{i=1}^{N} \mathbb{E}_q\,[\log(p(y_i | f^{GP}(x_i); \mu_i, \sigma_i^2) - \beta D_{KL}(q(u_i) \,||\, p(u_i)))]$

Compute gradients of loss w.r.t 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 CNN feature extractor for m^{th} episode: $\theta_{CNN}^{m+1} \leftarrow \theta_{CNN}^{m} - \gamma \frac{\partial \mathcal{L}_{max}}{\partial \theta_{CNN}^{m}} . \mathcal{L}_{max}$, $\theta_{GP}^{m+1} \leftarrow \theta_{GP}^{m} - \gamma \frac{\partial \mathcal{L}_{max}}{\partial \theta_{GP}^{m}} . \mathcal{L}_{max}$

end

BEGIN backpropagation of aleatoric uncertainty $\hat{\sigma}_i^2$

Make synthetic images via 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 $f^{GP} : \hat{z}_i \rightarrow p(\hat{y}_i \mid f^{GP}(\hat{z}_i) ; \hat{\mu}_i, \hat{\sigma}_i^2) = \mathcal{N}(\hat{\mu}_i, K_{\hat{x}_i})$

Calculate similarity loss $\mathcal{L}^{GP}$ 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 $\hat{\theta}_{GP}^{m}$ GP: $\hat{\theta}_{GP}^{m} \leftarrow \theta_{GP}^{m} - \gamma \frac{\partial \mathcal{L}^{GP}}{\partial \theta_{GP}^{m}} . \mathcal{L}^{GP}$

Update parameters of CNN feature extractor: $\hat{\theta}_{CNN}^{m} \leftarrow \theta_{CNN}^{m} - \gamma \frac{\partial \mathcal{L}^{GP}}{\partial \theta_{CNN}^{m}} . \mathcal{L}^{GP}$

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 [25] 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 [26] 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. Both are represented as matrices P and C, respectively. The total cost can be obtained by taking the Frobenius inner product of the two (i.e. $\langle C, P \rangle$). The objective then is to obtain the minimum of the product and subtract from the regularized entropy in (2). Here, η is denoted as the regularization term. For our experiments, we choose the default value for $\eta = 0.1$ and a quadratic distance-based cost function as an approximation to the primal Wasserstein distance formulation [26].

Finally, the maximum correntropy loss function [27] has also been implemented in the 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 equation (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.

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

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

$$\mathcal{L}^{MC} = V_\sigma\big(p(x), q(x)\big) = \mathbb{E}\big[k_\sigma\big(p(x) - q(x)\big)\big] = \frac{1}{N} \sum_{x=1}^{N} k_\sigma(p(x) - q(x)) \quad (3)$$

The term V_σ refers to the MC across two masses $p(x)$ and $q(x)$ where $\mathbb{E}$ refers to the expected value. This measure has been proven to be less sensitive to outliers. This is found in many second-order statistics measures such as cross-entropy. It is heavily studied in outlier suppression [27] and is ideally suitable for robust algorithm design. The next Section V presents results and analyses them.

V. PERFORMANCE VALIDATION

A. Validation Accuracy, Precision-Recall and ROC Curves

Before the experiments, the CNN-GP classifier is trained with the three different similarity losses. The purpose was to observe the accuracy as a means of performance evaluation. The average results were calculated by dividing the averaged correct samples by the total number of samples. Experiments were run ten times and accuracy values were averaged. The standard deviation was ± 2%. Then, the system was disrupted using black-box attacks of two types: a) an additive white Gaussian noise (AWGN) and b) motion blur (MB). The results were compared with the system version where no similarity losses were used (i.e. without regularization). These results are presented in Table II. Next, the precision-recall and the ROC results characterize the accuracy of the proposed CNN-GP framework. These results are plotted for each dataset side to side in Figure 2. The average precision (AP) and ROC area are two quantities that 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

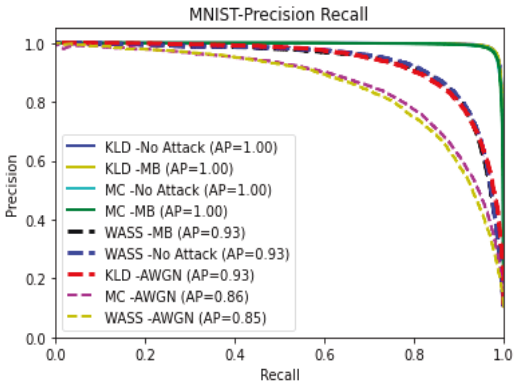
MNIST	No Attack (%)	AWGN (%)	Motion Blur (%)
No regularization	88	51	65
KLD	97 (86)	89 (75)	72 (60)
WASS	86 (35)	77 (11)	70 (13)
MC	97 (40)	78 (21)	75 (33)

Fashion-MNIST			
No regularization	85	32	12
KLD	88	53	76
WASS	81	56	72
MC	89	35	80

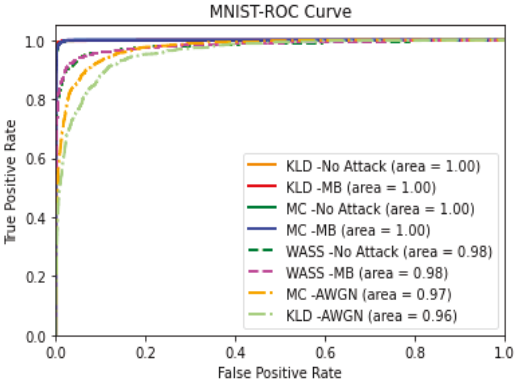
CIFAR10			
No regularization	67	10	11
KLD	73	26	38
WASS	40	28	28
MC	65	25	38

CIFAR100			
No regularization	55	10	8
KLD	60	10	12
WASS	35	10	10
MC	61	8	8

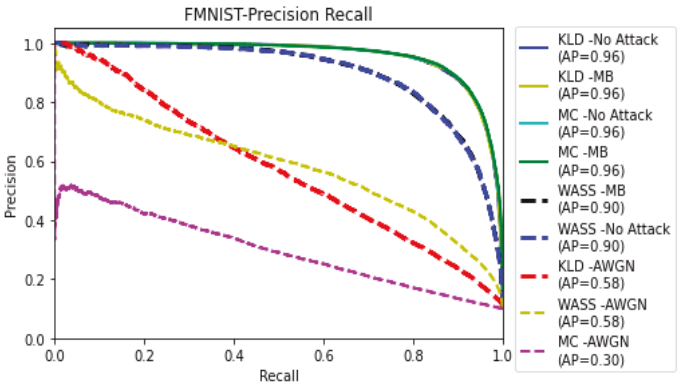
(A)



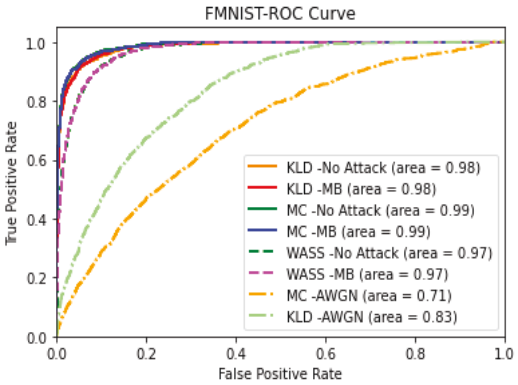
(B)



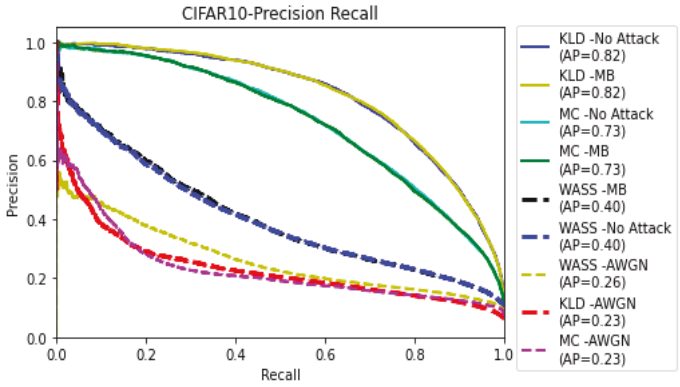
(C)



(D)



(E)



(F)

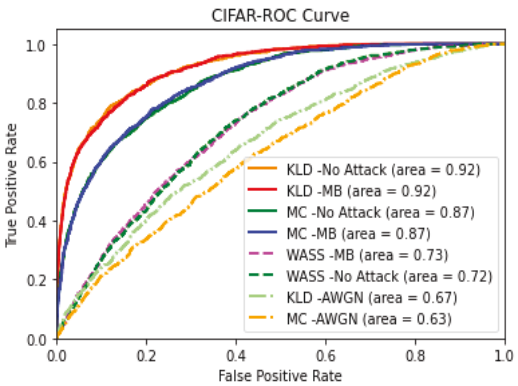


Fig. 2. The respective precision-recall and ROC curves for GPCNN framework trained on MNIST, FMNIST and CIFAR and on three loss functions; KLD, Wasserstein distance and maximum correntropy. Each plot considers three black-box attack configurations; no attack, a gaussian noise and motion blurring. A) 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 that reducing uncertainty allows the framework to be more reliable, we considered the four cases and analyzed the evolution of uncertainty separately. First, only the MNIST data was considered. The output mean predictions from the GP classifier and the standard deviation are plotted as bar graphs. This is shown in Figure 4. Every time the label is correct, the appropriate variance is computed from the likelihood. Blue bars represent the variance of correct samples and yellow for the incorrect. This is carried for each of the samples in the test set (10000 MNIST images).

The proposed framework is tested with three examples. One on clean MNIST images (column 1), the other two on MNIST corrupted by AWGN (column 2) and MB (column 3). The results show that the more the number of blue bars, the more accurate the model is, the lower the height of the blue or yellow bars, the more reliable the model is. If the predictions have higher yellow bars than blue, it is more susceptible to black-box attacks.

C. *Variance Sensitivity to Attack Strength*

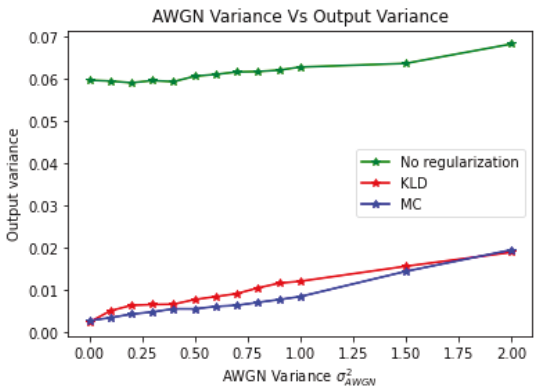
To test the sensitivity of the GP classifier to both black-box and white-box, we consider two cases as shown in Figure 3A and 3B. In one, we consider the case of white Gaussian noise to perturb input images from MNIST dataset. The input images are fed to GP classifier and the output variance from the classifier is obtained. We plot these in Figure 3A for standard deviations σ_{AWGN} in range 0.0 to 2.0.

We then test the system with the white-box attack fast gradient sign method (FGSM) [28]. 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, can easily mislead the system's representation.

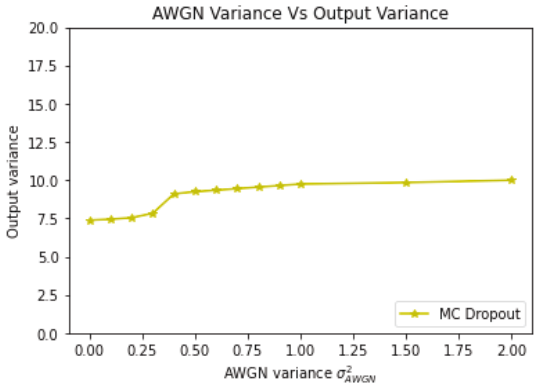
The strength of the attack is denoted by ϵ that increases the level of perturbation. The highlighted region in Figure 4B denotes the vital change of state in the system that can alert the system of the attack. This serves as a region where a high variance can lead to early detection of the attack before its intensity builds overtime. 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 [23] and results are presented in Figure 3B. 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)

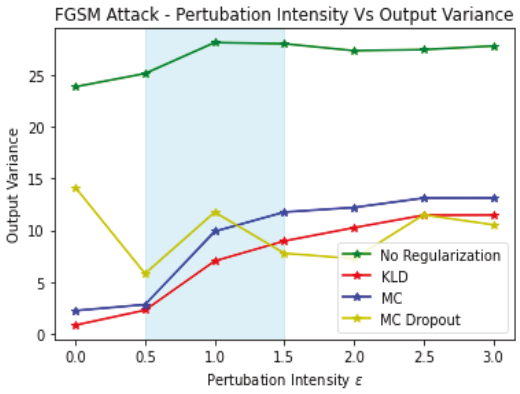


Fig. 3. Output variance computed from the GP classifier compared with the strength of both the additive white Gaussian noise in A), similarly for MC dropout in B) and fast-gradient sign method in C

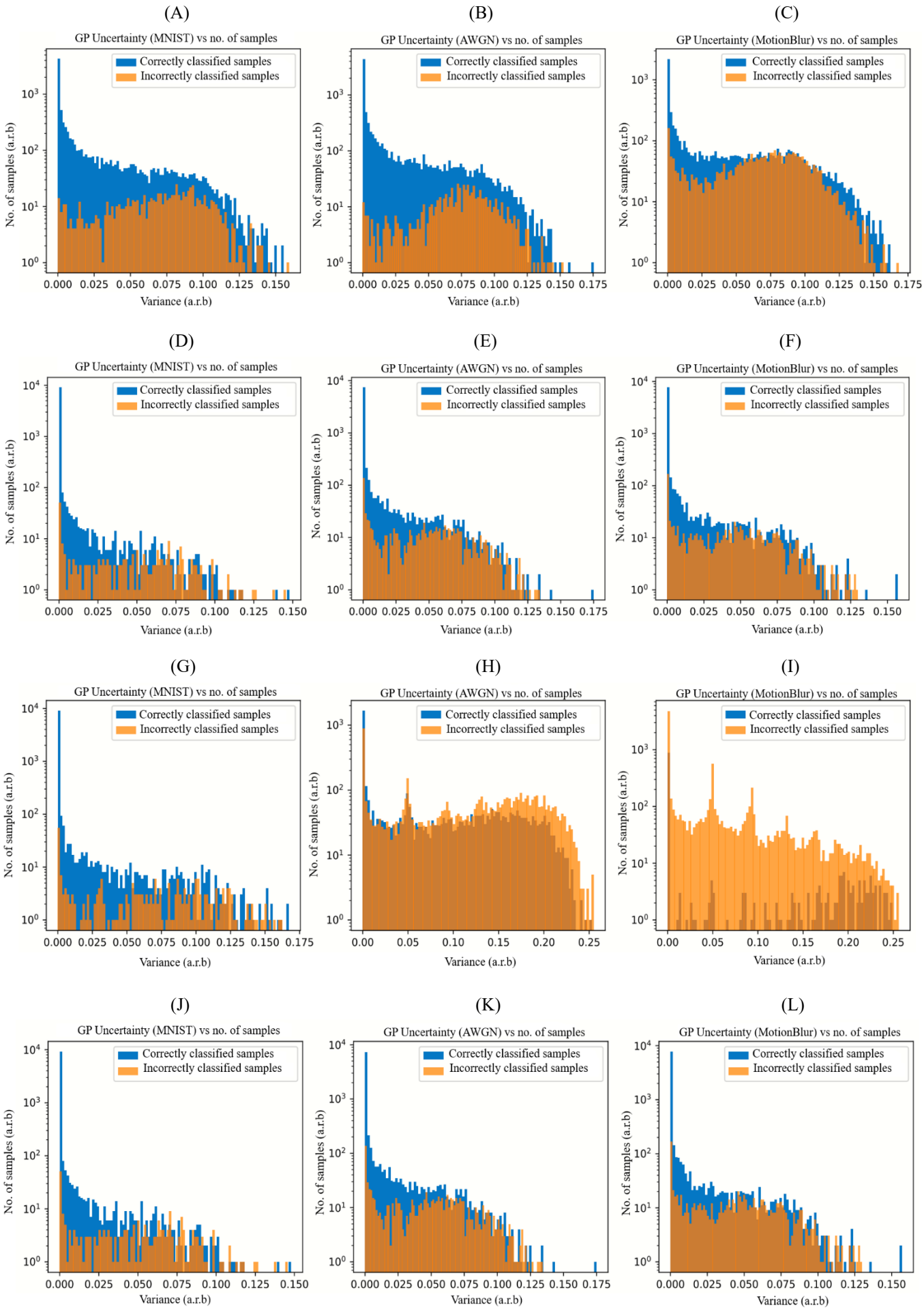


Fig. 4. Output variance plots from GP classifier for MNIST dataset for four configurations. The first row considers the case of model trained without any regularization from similariy losses, the second row is for GP classifier trained on KLD similarity loss, third row for Wasserstein distance and fourth for maximum correntropy. Each of the column represent the black-box attack types, the first column is for clean MNIST images, second column for white gaussian noise and the third for motion blurring

D. Computational Time

The computational time of the proposed framework CNN-GP is compared with the MC dropout method [23]. Both of the models are made to 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.

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

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

VI. DISCUSSION

Considering the results from Table II, we see that when there is no attack, the CNN-GP configurations that backpropagate both epistemic and aleatoric uncertainties, excluding the case with the Wasserstein metric, perform better than without backpropagation (no regularization). Furthermore, backpropagation of epistemic uncertainty influences the backpropagation of aleatoric uncertainty since the networks perform much worse when each of the processes is done separately (bracketed accuracies represent aleatoric only). This confirms that both stages of the training are necessary for reliable results. This is further demonstrated by uncertainty charts in Figure 4 where the uncertainty measures of KLD (row 2) and MC (row 4) have lower bar heights for incorrect sample variance (yellow bars) than those for the cases that backpropagates epistemic uncertainty only (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 [29] which claims that the Wasserstein metric yields biased gradients that have a higher chance in leading to a false local minimum than the KLD during optimization. This may also explain why KLD results on the backpropagation of aleatoric uncertainty are higher (75% and 60%) than the Wasserstein metric (11% and 13%).

This result may be due to the fact that an approximated version of the Wasserstein metric is implemented. An approximate implementation is performed to avoid the complexity and intractability of computing the infimum of double integrals in the primal version [26]. This is further supported by the precision-recall diagram for the Wasserstein metric for all attacks which shows that the precision for these methods slowly drop when the dataset complexity is increased (from MNIST to CIFAR10). The downward shift of blue, black and yellow dashed lines in Figure 2 visualizes these drops.

In order to characterize the robustness of the approaches, the recall function is calculated. Precision is heavily affected by the 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 Figure 4 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 was expected since this type of loss is ideal for robust algorithm design. This is further supported in Figure 2 where the precision-recall for both KLD and MC for motion blurring (MB) remain the highest (solid blue, green and yellow lines) as the dataset complexity increases (MNIST to CIFAR10).

Regarding the variance sensitivity to attack strength, we can see from Figures 3A and 3C 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 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 Figure 4 that backpropagation in the CNN-GP framework reduces the impact of uncertainties and attacks on the classification results and characterize the model's confidence. For the MC dropout method, it is seen from both Figure 3B and 3C 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.

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 were embedded in the backpropagation step of the CNN-GP and their performance was compared. The GP layer serves for quantifying the uncertainty, based on the GP variance. 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 results show that the models become robust and reliable and can cope with attacks, after learning from uncertainty. 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 [30]. In 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 uncertainty 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 paper, conducted the experiments, responsible for the write-up of the paper as well as the amendment.

Author Lyudmila Mihaylova contribute with ideas to the paper, its presentation and served as the main advisor for the improvements.

Author Nidhal Bouaynaya contribute with ideas to the paper, amendment of the paper and served as the secondary advisor for the improvements.

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. 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, M. Srivastava, "Why the Failure? How Adversarial Examples Can Provide Insights For Interpretable Machine Learning," *In Proceedings of the 21st International Conference on Information Fusion*, Cambridge, UK, 2018, pp. 838-845.

[2] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *In Proceedings of 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," *In Proceedings of the 4th International Conference on Learning Representations*, San Juan, Puerto Rico, 2015.

[4] G. Marcus, "The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence," 2020.

[5] T.G. Dietterich, "Steps Towards Robust Artificial Intelligence," AI Magazine, Vol.3, 2020, pp. 3-24.

[6] N. Narodytska and S. Kasiviswanathan, "Simple Black-box Adversarial Attacks on Deep Neural Networks," *In Proceedings of 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 Proceedings of the 31st Advances in Neural Information Processing Systems*. Long Beach CA, 2017, pp. 5574-5584.

[8] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition," *In Proceedings of the IEEE*, 1998 Vol. 11, 86, pp. 2278-324.

[9] H. Xiao, K. Rasul, R. Vollgraf, "Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms". CoRR. abs/1708.07747, 2017, http://arxiv.org/abs/1708.07747.

[10] A. Krizhevsky, I. Sutskever, G.E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *In Proceedings of the 25th Advances in Neural Information Processing Systems*, Harrahs and Harveys, Lake Tahoe, 2012, pp. 1097-1105.

[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] 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.

[14] A. Kurakin, I. Goodfellow, S. Bengio, "Adversarial machine learning at scale" *In Proceedings of the 5th International Conference on Learning Representations*, Palais des Congrès Neptune, Toulon, France, 2016.

[15] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, D. Mukhopadhyay. (Feburary 15). Adversarial Attacks and Defences: A Survey [Online]. Available: https://arxiv.org/pdf/1810.00069.pdf

[16] P. Samangouei, M. Kabkab, R. Chellappa, "Defense-GAN: Protecting Classifiers against Adversarial Attacks using Generative Models," *In Proc. of the 6th International Conf. on Learning Representations*, Vancouver, Canada, 2018.

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

[18] M. Javed and L. S. Mihaylova, "Leveraging Uncertainty in Adversarial Learning to Improve Deep Learning Based Segmentation," *In Proceedings of the 13th Symposium Sensor Data Fusion Trends, Solutions and Applications*. IEEE, Bonn, Germany, 2019, pp. 1-8.

[19] A. Tarvainen and H. Valpola, "Mean Teachers are Better Role Models: Weight-averaged Consistency Targets Improve Semi-supervised Deep Learning Results," *In Proceedings of the 31st Advances in Neural Information Processing Systems*, Long Beach CA, 2017, pp. 1195-1204.

[20] J. Li, Yongkang Wong, Qi Zhao, Mohan S. Kankanhalli. 2019. Learning to Learn from Noisy Labelled Data. In Proceedings of the 33rd IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'19). IEEE Press, Long Beach CA, pp. 5051-5059.

[21] L. Smith, Y. Gal, "Understanding Measures of Uncertainty for Adversarial Example Detection," *In Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence*, Monterey, California, USA, 2018, pp. 1-10.

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

[23] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning" *In Proceedings of the 33rd International Conference on Machine Learning*, New York, USA, 2016, pp. 1050-1059.

[24] A.G. Wilson, C. Dann, H. Nickisch, "Thoughts on Massively Scalable Gaussian processes", 2015. arXiv preprint arXiv:1511.01870.

[25] S. Kullback and R.A. Leibler. *On Information and Sufficiency*. The Annals of Mathematical Statistics, 1951, vol. 22, no. 1, pp. 79-86.

[26] I. Olkin and F. Pukelsheim, "The Distance between two Random Vectors with Given Dispersion Matrices," *Linear Algebra and its Applications. Linear Algebra and its Applications,* 1982, vol. 48, pp. 257-263.

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

[28] I. Goodfellow, J. Shlens and C. Szegedy, "Explaining and Harnessing Adversarial Examples", 2014, arXiv preprint arXiv:1312.6572.

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

[30] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, N. Houlsby, "Big Transfer (BiT): General Visual Representation Learning," *In Proceedings of the 16th European Conference on Computer Vision*, 2020.

(**All authors should include biographies with photo at the end of regular papers**.)

Mahed Javed is currently with the Department of Automatic Control and Systems, Sheffield, UK. Currently 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 a software designer for IMechE Unmanned Aerial Vehicle challenge from 2015-2016. Secondly as a software developer for the Sheffield University Nova Balloon Telescope (SUNBYTE). His research interests include machine learning and deep learning in the specific area of uncertainty quantification and adversarial robustness.

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) in the period 2016–2018. She is on the Board of Directors of ISIF. She was the Program Chair of the International Conference on Information Fusion 2020 (South Africa), the General Vice-Chair for the 2018 (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.