Overcoming Adversarial Perturbations in Data-driven Prognostics Through Semantic Structural Context-driven Deep Learning

Xingyu Zhou[1], Robert Canady[2], Yi Li[3],and Aniruddha Gokhale[4]

[1,2,3,4] *Dept of EECS, Vanderbilt University, Nashville, TN 37235, US*
{*xingyu.zhou,robert.e.canady,yi.li,a.gokhale*}*@vanderbilt.edu*

ABSTRACT

Deep learning has shown impressive performance across a variety of domains, including data-driven prognostics. However, research has shown that deep neural networks are susceptible to adversarial perturbations, which are small but specially designed modifications to normal data inputs that can adversely affect the quality of the machine learning predictor. We study the impact of such adversarial perturbations in data-driven prognostics where sensor readings are utilized for system health status prediction including status classification and remaining useful life regression. We find that we can introduce obvious errors in prognostics by adding imperceptible noise to a normal input and that the hybrid model with randomization and structural contexts is more robust to adversarial perturbations than the conventional deep neural network. Our work shows limitations of current deep learning techniques in pure data-driven prognostics, and indicates a potential technical path forward. To the best of our knowledge, this work is the first to investigate the implications of using randomization and semantic structural contexts against current adversarial attacks for deep learning-based prognostics.

1. INTRODUCTION

Machine learning has gained much attention to solve a variety of challenges in cyber-physical systems (CPS). Data-driven methods, particularly deep learning, have made deep strides in health management and prognostics, such as anomaly detection and remaining useful life estimation. For a relatively complex system with a number of sensors, the service provider can utilize data flows from multiple smart data sources to perform data-driven prognostics using a deep learning model. However, recent research in prognostics has shown that statistical learning methods, e.g., deep neural networks, are susceptible to small adversarial perturbations, which are small but specially designed modifications to normal data inputs that can adversely affect the quality of the machine-learned predictor (Echauz et al., 2019). For instance, owing to the deployment of external sensors in many application scenarios, an attacker could intercept and maliciously modify sensor readings to conduct these kinds of adversarial attacks, which could subsequently lead to critical damage caused due to inaccurate system health status evaluation.

We study the impact of such adversarial perturbations in data-driven prognostics as shown in Figure 1. Data-driven prognostics methods are suitable for scenarios where complete analytical models of the physics are difficult or impossible to formulate. In this context, recent research has shown the success of combining physics-based semantic information (Chao, Kulkarni, Goebel, & Fink, 2020) with pure data-driven prognostics. For this paper, two typical prognostics problem settings comprising classification and regression are considered (Saxena, Celaya, et al., 2008). In the classification setting, the model predicts the remaining useful life of the system in a certain range and outputs a categorical result to indicate to which health status category the system belongs to. In the more general regression setting, the model outputs a numerical prediction value for the remaining useful life.

We demonstrate our ideas using the widely-used C-MAPSS turbine engine dataset (Saxena & Goebel, 2008).We explore the vulnerability of prediction models and potential ways to defend against adversarial attacks.To the best of our knowledge, this work is the first to investigate the implications of using semantic structural contexts against current adversarial attacks for deep learning-based prognostics (Y. Li et al., 2020). We find that we can introduce obvious errors in prognostics by adding imperceptible noise to a normal input and that the model involving randomization and structural contexts is more robust to adversarial perturbations. Our work highlights the limitations of current deep learning techniques in pure data-driven prognostics, and presents a potential technical path forward.

We make the following contributions in this paper:

- We present a framework that can formalize the security and resilience testing in data-driven prognostics settings.

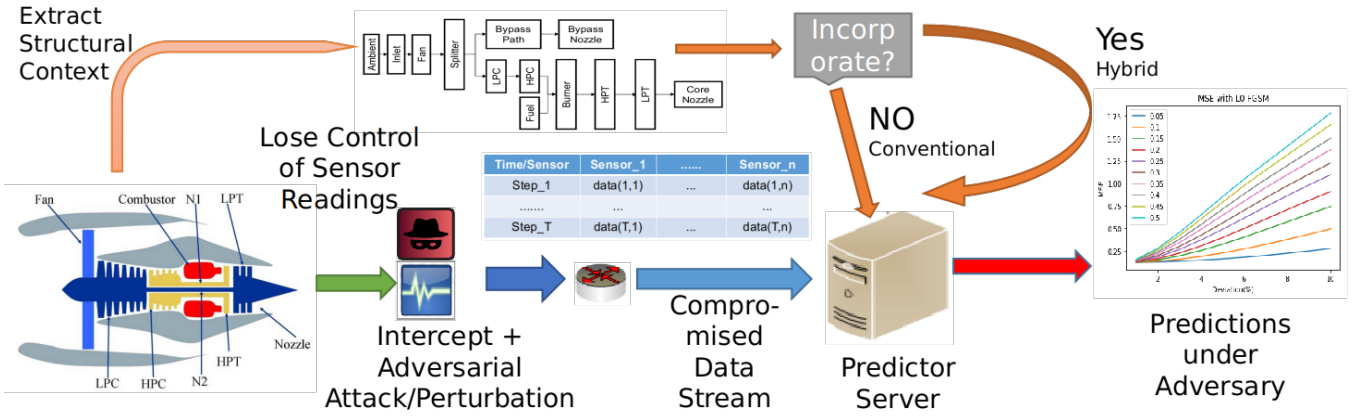

Figure 1. Overall Workflow for Robustness Evaluation of Deep Learning-based Prognostics on Conventional & Hybrid Models (with Randomization and Structural Contexts)

- We show the vulnerability of deep learning prognostics models under various settings.
- We investigate the possibility of inducing randomization elements and semantic structural context to mitigate adversarial impacts.
- We conduct a robustness case study using the engine degradation dataset on typical applications of health status classification and remaining useful life prediction.

The rest of the paper is organized as follows. Section 2 provides the motivation for evaluating and mitigating security risks in data-driven prognostics. Section 3 illustrates the theoretical background for our adversarial attack and defense settings. Section 4 presents a case study to demonstrate the capabilities of our framework on the C-MAPSS turbine engine dataset (Ramasso & Saxena, 2014). Finally, Section 5 concludes the paper and alludes to future research directions.

2. MOTIVATION

It is essential for service providers to maintain a capability to monitor and predict the health status of their system. Two primary system modeling technical paths have been widely used including model-based and data-driven. Classical model-based methods assume that the model must be accurate enough to depict system behaviors, e.g., Bond Graphs (Broenink, 1990) or analytical battery physical systems (Zhang & Lee, 2011). These methods need to know formally how different components of a system interact and how the system output can be computed analytically. For many real-world cases, however, due either to partial knowledge or the system's large-scale, it becomes impossible to reveal the pattern in a detailed analytical way. Consequently, as the complexity of the system increases and the amount of generated data volumes increases, data-driven health management and prognostics are preferred (Baraldi, Cadini, Mangili, & Zio, 2013).

Recent advances in machine learning, particularly deep neural networks, enable a lower threshold for building a prediction model for health management and prognostics. However, deep neural networks are susceptible to adversarial attacks, which are a small amount of additional data designed as perturbations to misguide the original neural network prediction systems. Recently, the impact of adversarial examples in deep learning (Szegedy et al., 2013) has given rise to many concerns (Goodfellow, Shlens, & Szegedy, 2014).Prior research (Vorobeychik & Kantarcioglu, 2018) has shown how these adversarial examples can pose threats to current machine learning systems. Most prior work in the field of adversarial machine learning during the past decade has paid attention to classification tasks (Biggio & Roli, 2018). As regression tasks start playing an increasingly important role in CPS scenarios, the topic of adversarial regression is attracting more research attention.

For the health management and prognostics field, deep learning techniques have shown to be successful in a number of tasks like power disturbance classification (Valtierra-Rodriguez, de Jesus Romero-Troncoso, Osornio-Rios, & Garcia-Perez, 2013) and remaining useful life prediction (X. Li, Ding, & Sun, 2018). Consequently, even though a health prediction model could be built easily using state-of-the-art deep neural networks and tool-flows, a more cautious view still needs to be taken due to potential risks of adversarial attacks with these kind of learning-based components (Echauz et al., 2019).

In a nutshell, although accurate status classification or prognostics is critical for efficient data-driven health management, the vulnerability in this broad practical scenario has not been carefully investigated to date. To address these issues in the prognostics and health management domain, we propose an approach to show model vulnerabilities from domain-specific settings and explore some potential ways to mitigate underlying adversarial impacts.

3. METHODOLOGY

In this section we delve into the details of the underlying methods used in our framework. These methods will be introduced in a step-by-step manner following the evaluation execution path of attack and defense on neural network predictors.

3.1. Adversarial Attack

As discussed above, attacks on neural network-based prognostics are essentially small adversarial data modifications. There are two important features for a successful adversarial attack. First, the attack perturbation should not be 'obvious' enough to be detected by the system that it is attacking. Secondly, the attack should lead to 'obvious' performance deviation in the compromised prediction system.

Overall, given a model $f(\cdot)$, one well-recognized way to define an adversarial attack (Carlini et al., 2019) is the worst-case target loss L for a given perturbation budget defined by an ϵ bounded distance magnitude $D(x, x')$ between the original data point x and the perturbated data point x'.

$$\max_{x':\mathcal{D}(x,x')<\epsilon} L\big(f(x'), y\big) \tag{1}$$

It is worth pointing out that the attack methods often regard the requirement of being stealthy as self-evident under the fixed maximum magnitude constraint of ϵ. This can be generalized to data with or can be pre-processed into a fixed range using techniques like MinMax Normalization (Patro & Sahu, 2015).

3.1.1. Gradient-based Attack

Based on the two features above, we can regard an adversarial attack against a prediction model as an optimization problem. The goal of this optimization problem is to solve for an adversarial perturbation on the original data input that on one hand maximizes the target loss function and on the other hand be under some realistic constraints. Here we consider a white-box setting where an attacker can obtain full knowledge of the prediction model. In this way, the attack procedure would be a gradient-based optimization computation using the loss function. We select two typical adversarial attack methods of Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD).

FGSM (Fast Gradient Sign Method) (Goodfellow et al., 2014) is one of the most well-known and popular adversarial attack methods. It formulates the optimization problem incorporating these two constraints using only one single equation:

$$x' = clip_{[0,1]}\{x + \epsilon \cdot sign(\nabla_x L\big(f(x'), y\big)\} \tag{2}$$

Here, x represents inputs to the model and is assumed to be in the range of $[0, 1]$, y refers to the targets associated with x (for tasks with targets) and $L\big(f(x'), y\big)$ is the goal loss function for deviating the neural network predictor $f(\cdot)$. The magnitude constraint added to the original sample is represented by ϵ. This method is quite simple and intuitive. The attacker adds fixed magnitude perturbations to maximize the loss function.

PGD (Projected Gradient Descent Method) (Madry, Makelov, Schmidt, Tsipras, & Vladu, 2017) is recognized as a universal attack procedure and by far one of the strongest attack methods. Instead of starting from the exact data point, it starts from a random perturbation in the $\|l\|_p$ norm ball around the input sample. It then utilizes the FGSM approach but with a much smaller gradient step towards the greatest loss and projects the perturbation back into the allowed $\|l\|_p$ ball range. In contrast to one-step FGSM, PGD iteratively adds small perturbations using updated gradients. With the same maximum level magnitude of the perturbation, PGD proves to be much stronger than FGSM.

3.1.2. Adversarial Goal Setting

In a real-world CPS scenario, the settings become more complex with a data input space that is potentially larger than a fixed range. As a result, we reformulate an adversarial attack as an optimization problem which attempts to find the best synthetic perturbations that maximize the prediction loss while keeping the modification magnitude at a small enough level so as to go undetected.

In CPS settings not only do input data formats show more complex patterns, the adversarial attack goals can also be more flexible than, say, computer vision classification tasks. For an adversarial regression scenario like remaining useful life prediction, we propose three adversarial attack settings using corresponding loss functions: **Mean Squared Error** (to maximize absolute prediction deviation), **Maximization** (to maximize prediction value) and **Minimization** (to maximize prediction value). The latter two are not commonly discussed but would be extremely meaningful for health management and prognostics application scenarios where the cost of underestimating or overestimating could be high and unbalanced (Ramasso & Saxena, 2014).

3.2. Randomization-based Defense

Although there has not been an overall defense technique for existing adversarial attacks (Akhtar & Mian, 2018), previous robustness improvement works that induced randomization elements have shown a high success rate in terms of detecting adversarial examples (Athalye, Carlini, & Wagner, 2018). As redundancy design is a significant ideology toward higher system reliability, it is intuitive to combine this internal ideology with a probabilistic implementation. Therefore, we choose two typical randomization-based methods and further propose a potential way utilizing semantic structural contexts to help mitigate adversarial impacts in data-driven prognostics.

3.2.1. Gaussian Augmented Adversarial Training

One mainstream defense technique that has attracted much attention is improving model robustness during the training phase. Research has shown that data augmentation during training using Gaussian data augmentation (Zantedeschi, Nicolae, & Rawat, 2017) helps in improving robustness of neural networks to adversarial attacks. The intuition behind this is straightforward: in the high dimensional space, data points within a $\|l\|_p$ ball range should own the same label. Based on this assumption, the original training dataset can be extended by adding a norm-bounded Gaussian noise but with the same label. The training based on this extended dataset generates more resilient models against small perturbations.

3.2.2. Input Dropout Inference

In the inference phase, there has been research (Xie, Wang, Zhang, Ren, & Yuille, 2017) showing that random cropping or random padding with resizing of the adversarial examples reduces their effectiveness. Furthermore, randomly deactivating input neurons in the inference phase (S. Wang et al., 2018) has also been discussed. For the model deployment strategy, a recent work (Liu, Cheng, Zhang, & Hsieh, 2018) adds small random noises to one input for several predictions and suggests the ensemble of prediction results with the highest probable class. However, these methods have only been under trial on computer vision and classification tasks and cannot be applied to other scenarios directly.

3.2.3. Generalized Random Ensemble

Based on the above discussions, we design a generalized model application strategy against adversarial examples using randomization elements applicable for both classification and regression problems.

The system holds a predictor and a self-representative auto-encoder detector for reconstruction. Given a data input, the system randomly sets input feature values to zero at a dropout rate level P_R. We discuss two kinds of dropout here: (1) Normal Dropout with a constant rate on all features and (2) $SemanticDropout$ using system structural contexts to assign features with different dropout rates as shown in Algorithm 1. The intuition behind this $SemanticDropout$ is to decouple most correlated features from the same subcomponent that are most likely to be adversarially modified in the same direction. Then the data with deactivated values is sent to a self-representative auto-encoder to recover the matrix. This recovered data is then used for prediction. This $\{Drop => Reconstruct => Predict\}$ procedure is conducted repeatedly for $nIter$ iterations and generates $nIter$ sets of predictions.

To deal with the classification problem, the output matrix contains likelihood values for classes. We sort these likelihood values for each class. The likelihood for one class is computed as the sum of the results of the highest $nThres$ rounds. The overall classification result refers to the class with the highest sum value.

To deal with regression problem, the system first uses some training data with the trained model and generate some adversarial examples using deviation maximization loss function (like mean squared error). These adversarial data from training data would be used to decide whether the model is more likely to be adversarially-maximized or adversarially-minimized or equal likely. Given that the output matrix contains numerical prediction values from the step above, we sort these prediction values for $nIter$ rounds. If the model is adversarially-maximized, the final prediction result would be the mean value of the smallest $nThres$ prediction results. In contrast, if the model is adversarially-minimized, the final prediction result would be the mean value of the largest $nThres$ prediction results. Otherwise, the final prediction result would be the mean value of the median $nThres$ prediction results.

Algorithm 1 Semantic Dropout

Require: $\boldsymbol{x}$: original observation with $nFeature$ feature rows; $adjMat$: adjacency matrix of features (same subcomponent features regarded as adjacent); $CorrCoef$: correlation matrix computation function; Dot: dot product; $Dropout$: normal dropout function; $MinMaxScale$: function to scale inputs into a range between a min and max value; P_R: base dropout rate.

$\boldsymbol{xcorr} \leftarrow Abs(CorrCoef(\boldsymbol{x}))$
$Diag(\boldsymbol{xcorr}) \leftarrow 0$
$Diag(\boldsymbol{xcorr}) \leftarrow MaxByRow(\boldsymbol{xcorr})$
$\boldsymbol{adjCorr} \leftarrow Dot(adjMat, \boldsymbol{xcorr}) + Dot(\boldsymbol{xcorr}, adjMat)$
$\boldsymbol{adjCorrMean} \leftarrow MeanByRow(\boldsymbol{adjCorr})$
$\boldsymbol{adjCorPr} \leftarrow MinMaxScale(\boldsymbol{adjCorrMean}, min = 0.5P_R, max = 1.5P_R)$
$\boldsymbol{xSemDrop} \leftarrow \boldsymbol{x}.copy()$
while $i < nFeature$ **do**
  $\boldsymbol{xSemDrop}[i] \leftarrow DropOut(\boldsymbol{x}[i], \boldsymbol{adjCorPr}[i])$
  $i \leftarrow i + 1$
end while
return $\boldsymbol{xSemDrop}$

4. EVALUATION

For demonstration purposes, we conduct a case study using the widely-used C-MAPSS jet engine degradation dataset (Saxena & Goebel, 2008). It is worth pointing out that the proposed attack and defense settings as well as implementation methods can also be generalized to other data-driven health management and prognostics settings without any constraints.

4.1. Dataset Description

The C-MAPSS dataset (Saxena & Goebel, 2008) was initially generated for the PHM08 Data Challenge. It is a dataset for data-driven remaining useful life (RUL) prediction for jet tur-

bofan engines. The standard version has four sub-datasets consisting of running data of the engine under a certain mode. The engine starts degrading from a time point and breaks down (RUL=0) at the end of the running cycle. Apart from time label, there are 24 features for each data point as shown in Table 2. The first three are the three operational settings (does not state exactly what they represent) that have a substantial effect on engine performance.The remaining ones represent the 21 sensor values.Some researchers also use trends to make a initial selection on features (T. Wang, Yu, Siegel, & Lee, 2008) to guarantee that the selected features have relative clear trends throughout the degradation process (Ellefsen, Bjørlykhaug, Æsøy, Ushakov, & Zhang, 2019). As we are trying to demonstrate the potential risk of data-driven predictors, we only choose the first set $FD001$ for our experiments here. There are some data preparation steps that could be applied on this dataset including duplicate removal and normalization like MinMax (Patro & Sahu, 2015) or Batch normalization (Ioffe & Szegedy, 2015). We implement MinMax normalization to transform feature values into a fixed range of 0 and 1.

4.2. System Models

Since we focus on sequence analysis, we choose the method of long short-term memory (LSTM) network (Hochreiter & Schmidhuber, 1997). A recurrent LSTM network enables us to input sequence data into a network, and make predictions per individual time steps of the sequence data. Given its good support for time series, LSTMs have been widely used for data-driven prognostics (Zheng, Ristovski, Farahat, & Gupta, 2017) for more than a decade (Heimes, 2008). Our work makes use of recurrent neural networks in two ways using sensor readings of different components. One is to to predict the health status or remaining useful life. The other is to use an LSTM auto-encoder to help reconstruct data matrices with deactivated zero values after randomization executions. The input dimension is $50 * 24$ indicating 24 features across 50 time steps. For the health predictor, we include two LSTM layers with 100 units. The status classifier has a Sigmoid (binary classification) or Softmax (multi-class classification) activation layer to generate likelihood for classification outputs. The regression predictor is added with a single neuron to compute the numerical output after the LSTM layers. For the auto-encoder, we include two LSTM layers with 50 units followed by a fully-connected layer with 24 units to generate an output with the same shape with the input.

4.3. Prediction and Attack Setting

For health management and prognostics applications, the input and output formats vary for different scenarios. For this engine dataset, research can be conducted on both remaining useful life value regression or health status classification (Umberto Griffo, 2019). To show the generalized existence of adversarial impacts across different data-driven settings, we consider three tasks with five adversarial settings as shown in Table 1. Here, the labels for classification tasks are transformed from the original numerical remaining useful life values.

Adv Loss	Attack Goal	Outcome
None	$f(x) = y$	Normal Prediction
Binary Crossentropy	$f(x') \neq y$	Misclassify
Categorical Crossentropy		
Minimization	$\min f(x')$	Decreased Prediction
Maximization	$\max f(x')$	Increased Prediction
Mean Squared Error	$\max(f(x') - y)^2$	Increased Absolute Prediction Deviation

Table 1. Attack Goal for Perturbated Sample $x' : \mathcal{D}(x, x') < \epsilon$

For the binary classification task, a predictor judges whether the engine is already in the failure state. We consider engines with no more than 15 cycles of life lengths as being in failure state. This predictor outputs a binary result of being true or being false. And as a result, the adversarial attacks make use of the known model knowledge and use the loss function of **Binary Crossentropy** to lead the predictor to misclassify the modified sample into another class.

For the multiclass classification task, a predictor judges whether the engine is in the steady phase, the degrading phase or the critical phase. We consider engines with no more than 50 cycles of life lengths as being critical, with 50 to 105 cycles as being degrading and with more than 105 cycles as being steady. This predictor outputs a class indicating the phase. As a result, the adversarial attacks make use of the known model knowledge and use the loss function of **Categorical Crossentropy** to lead the predictor to misclassify the modified sample into wrong classes.

For the regression task, a predictor outputs a positive numerical value of the estimated remaining useful life for the given time step of data input. This is the most fundamental task of this dataset. Here the attacker has three options for the attack goal setting using different loss functions:**Mean Squared Error** to maximize absolute prediction deviation, **Maximization** to maximize prediction value and **Minimization** to minimize prediction value.

The attack is a manipulation of sensor data under reasonable constraints with full knowledge of the prediction and detection model. Among the 24 features provided for data points, the first three are control settings and the remaining 21 are sensor readings. We assume the attacker can modify values of sensor readings but the control settings are kept untouched. For the attack strength, we conduct experiments on clean data along with different attack perturbation levels of $advEps = 0.01 - 0.10$. That is equivalent to $1 - 10\%$ of data value range based on the MinMax Normalization. Under these constraints, we generate adversarial examples using the strongly iterative PGD attack(50 step) to maximize the prediction deviation.

No.	Parameter	Detail	Group
1	C1	Control input 1	0
2	C2	Control input 2	0
3	C3	Control input 3	0
4	T2	Total temperature at fan inlet	1
5	T24	Total temperature at LPC outlet	2
6	T30	Total temperature at HPC outlet	3
7	T50	Total temperature at LPT outlet	5
8	P2	Pressure at fan inlet	1
9	P15	Total pressure in bypass-duct	2
10	P30	Total pressure at HPC outlet	3
11	Nf	Physical fan speed	4
12	Nc	Physical core speed	6
13	epr	Engine pressure ratio (P50/P2)	0
14	Ps30	Static pressure at HPC outlet	3
15	phi	Ratio of fuel flow to Ps30	3
16	NRf	Corrected fan speed	4
17	NRc	Corrected core speed	6
18	BPR	Bypass Ratio	2
19	farB	Burner fuel-air ratio	5
20	htBleed	Bleed Enthalpy	0
21	Nf_dmd	Demanded fan speed	4
22	PCNfR_dmd	Demanded corrected fan speed	4
23	W31	HPT coolant bleed	5
24	W32	LPT coolant bleed	5

Table 2. Sensor Feature Grouping according to Their Semantic Structural Contexts

4.4. Evaluating Attack and Defense

For comparison purposes, we conduct experiments on models trained from only clean natural data and also from Gaussian Augmented data (with a norm bound of 0.10). Moreover, for each model, we include three settings of no defense, normal dropout and $SemanticDropout$. In this section we first introduce how we incorporate semantic structural contexts and then we show more detailed experimental results.

4.4.1. Structural Information Embedding

Without any background knowledge of aircraft engine dynamics, we can get this high-level information from the layout description given by the original CMAPSS dataset (Saxena, Goebel, Simon, & Eklund, 2008). Among 24 column variables, the first three are control inputs and the rest are sensor values. Five features including three control inputs are regarded as conducting impact globally. Further, we use this mapping to build the feature adjacency matrix for the proposed $SemanticDropout$ method discussed. Using the semantic context knowledge of the high-level component structure, we can briefly relate features to where they work in the engine and how different features are connected as shown in Figure 2. The definition of 24 column features as well as the proposed semantic grouping are also shown in Table 2. Features in the same group would be marked as being adjacent to each other. The global impact features would be assumed to be adjacent to all other features.

4.4.2. Experimental Results

We conduct experiments on various settings. To evaluate the worst case, Without loss of generality, the number of detection iterations is set to 10 and the detection threshold is set as half of that. For all prediction result figures, the detection dropout rate is set at 40%. And we show results under increasing adversarial attack strengths($advEps$ ranging from $0.01(1\%)$ to $0.10(10\%)$).

Figure 3 shows binary classification accuracy under adversarial attack. Figure 4 shows multiclass classification accuracy under adversarial attack. From both these classification cases, the $SemanticDropout$ method shows highest adversarial robustness. But there are some obvious differences shown from our experiments. In the multiclass case, the Gaussian Augmented model always shows better robustness whereas in the binary case the Gaussian Augmented model does not hold a stable performance. This shows the potential impact of noise strength for the adversarial training phase. The other point is that the robustness of the multiclass model is higher and in other words the binary classification model itself is more vulnerable. From our empirical experiences this should have something to do with the balance of dataset splitting and output encoding (Buckman, Roy, Raffel, & Goodfellow, 2018).

Figure 5 shows adversarial regression errors under prediction deviation maximization attack setting. Figure 6 shows adversarial regression errors under prediction minimization attack setting. Figure 7 shows adversarial regression errors under prediction maximization attack setting. For most of these model and attack settings, the $SemanticDropout$ method shows highest adversarial robustness.

Figure 9-11 show prediction results under a medium level($advEps = 0.05$) adversarial PGD attack on the Gaussian Augmented model. Figure 9 shows prediction results when the attacker uses the mean square error between clean and adversarial prediction as the optimization loss function and aims to make prediction results on adversarial data that deviates from clean data as much as possible. Figure 10 shows prediction results when the attacker uses the negative derivative of the predictor as the optimization loss function and aims to make prediction results on adversarial data as small as possible. Figure 11 shows prediction results when the attacker uses the derivative of the predictor as the optimization loss function and aims to make prediction results on adversarial data as large as possible.

Figure 8 We provide an example adversarial perturbation as shown in with a ground truth RUL value of around 60. We can see features get different magnitudes of modification along time steps. And from this example here, we can also see why it is difficult to isolate the origin of vulnerability. Even for a certain feature, it might show different sensitivity levels along time steps. By far, we cannot provide a general description of how a certain sensor reading would be sensitive to adversarial attacks.

To make visualization easier, we sort test data according to

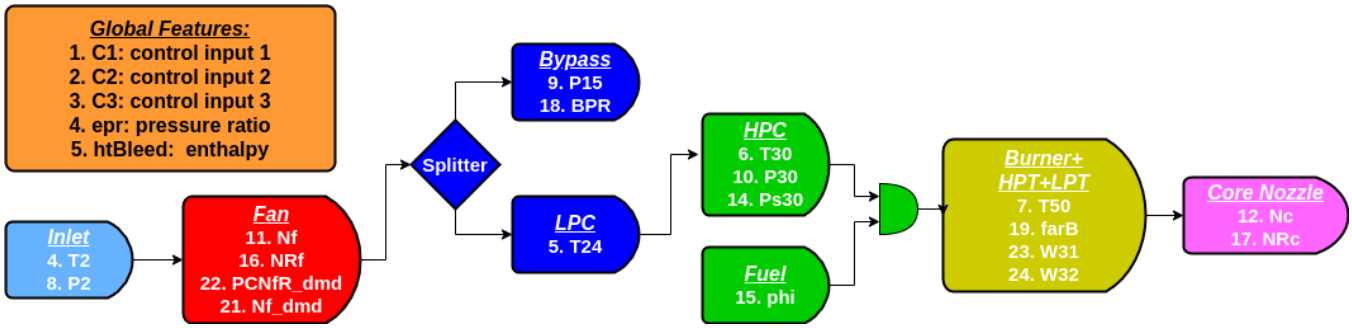

Figure 2. Groups of Features Along Layout Components (one color per group)

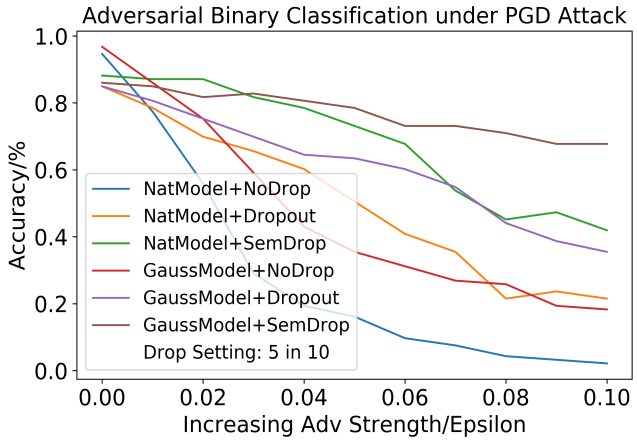

Figure 3. Failure Stutus Binary Accuracy under Increasing Adversarial Attack Strength

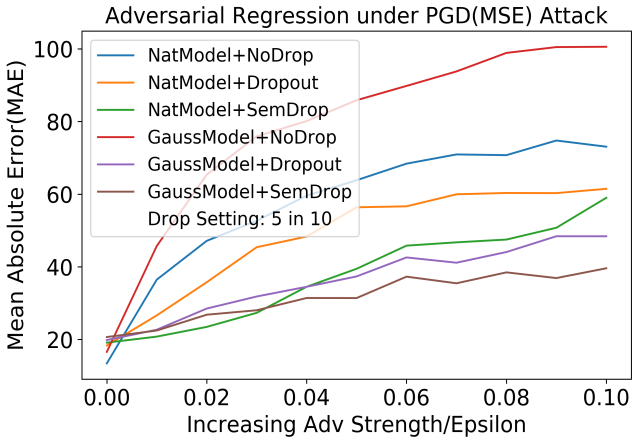

Figure 5. Remaining Useful Life Prediction Mean Absolute Error under Deviation Maximization Attack

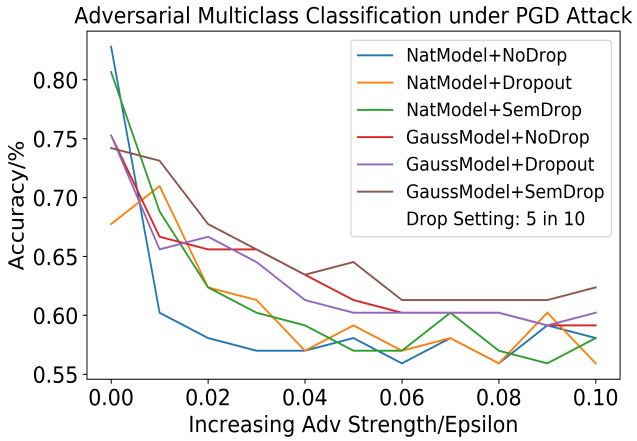

Figure 4. Health Status Multiclass Accuracy under Increasing Adversarial Attack Strength

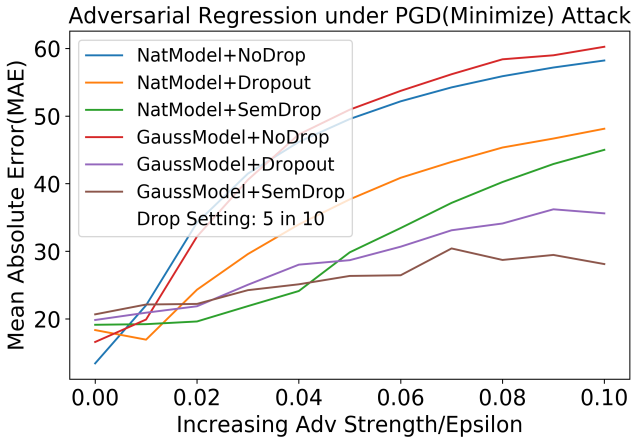

Figure 6. Remaining Useful Life Prediction Mean Absolute Error under Prediction Minimization Attack

their ground truth remaining useful life values. From figures above we can also see that adversarial impacts in this data-driven prognostics case are more prone to making prediction results larger. This is most obvious in Figure 9 where adversarial data misguides the predictor to in both directions in a greedy way but we can see the maximization shows up much than minimization. In addition, the proposed *SemanticDropout* method with autoencoder-based reconstruction successfully decreases prediction deviations at these vulnerable points without much impact on other locations. On the other hand, these

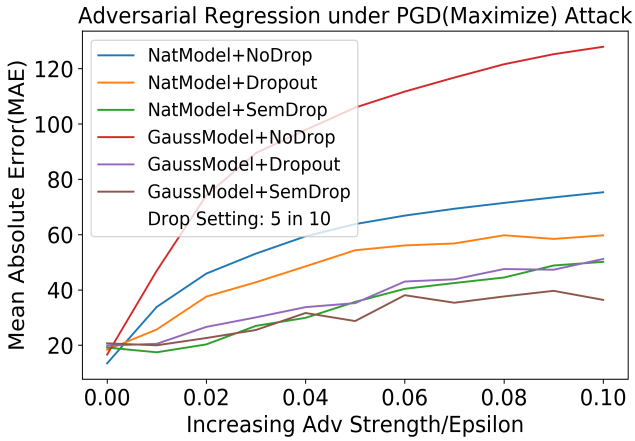

Figure 7. Remaining Useful Life Prediction Mean Absolute Error under Prediction Maximization Attack

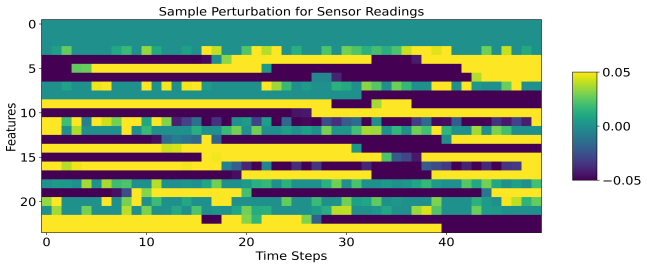

Figure 8. Example Adversarial Perturbation under Maximum Magnitude Constraint $advEps = 0.05(5\%)$

experiments also show the trade-off from potential defense mechanisms. With a large value of detection dropout, the chance of being totally stealthy for an adversarial input data is low. Meanwhile, erasing and reconstructing more elements of data obviously leads to higher reconstruction error and thus prediction error. This becomes an inevitable trade-off by inducing defensive steps since these extra processing would not generate extra information. As shown in our experiments, this potential risk is most obvious when the detection dropout rate is high. As a result, the combination of a medium level detection dropout rate along with reconstruction would render a better trade-off between natural error by the defense mechanism and robustness error by the adversarial attack.

We present experiment results under more flexible settings in Table 3. The table shows experimental results under three levels of detection dropout rate: $30\%, 40\%, 50\%$ (three columns from left to right in each setting) with two cycles of reconstruction using dropout and autoencoder. The error metric we choose to show here is the most commonly used mean absolute error (MAE) for regression and accuracy for classification. For different adversarial attack settings, the settings with best output results are marked in dark black. We can see that low detection dropout rates usually perform better on clean or less perturbed data but on the other hand are less robust against strong adversarial attacks.

In summary, we show the vulnerability and potential risks of deep learning based predictors in data-driven prognostics applications. We also show the efficiency as well as the potential of the proposed randomization-based defense technique involving semantic structural contexts. One significant advantage of our randomization-based framework is that it makes use of existing pre-trained models in a resilient way, which means it can work together with other defense techniques seamlessly.

5. CONCLUSION

Data-driven models in health management and prognostics scenarios. We explore the vulnerabilities of the state-of-the-art deep neural network data-driven health management and prognostics caused by the technique called adversarial attack. We show the significance of inducing randomization elements to improve model robustness. Furthermore, we investigate the possibility of inducing randomization elements from semantic structural contexts to mitigate adversarial impacts. Our discussions and experimental implementations on the engine degradation dataset cover the most typical settings in PHM applications. These general settings would help formalize the security testing in more scenarios.

Our future work is focused on the following three aspects. First, our current investigation is only on a single operation and error degradation mode. We are only incorporating semantic structural contexts using a high level of whether features belong to the same component part or not. We are planning to incorporate more hierarchical operational settings in the semantic feature analysis for more tasks (Pasareanu, Gopinath, & Yu, 2018). Secondly, we emphasize the significance of using supervised self-representative model for data reconstruction using an auto-encoder. But there are also other methods that are also worth exploring like matrix completion (Yang, Zhang, Katabi, & Xu, 2019) or generative adversarial networks (Samangouei, Kabkab, & Chellappa, 2018). Thirdly, even though our defense framework has shown promising performances on various settings, the optimal choice of the defense settings remains empirical. Therefore, we need to dive deeper into the origins of the vulnerabilities in the system and conduct the attack and defense evaluation in a more systematic way. Further research is necessary for measuring the efficiency of this framework and speed up computations towards more real-time applications.

ACKNOWLEDGMENTS

This work was supported in part by the AFOSR DDDAS FA9550-18-1-0126 program. Any opinions,findings, and conclusions or recommendations expressed are those of the author(s) and do not necessarily reflect the views of the sponsor.

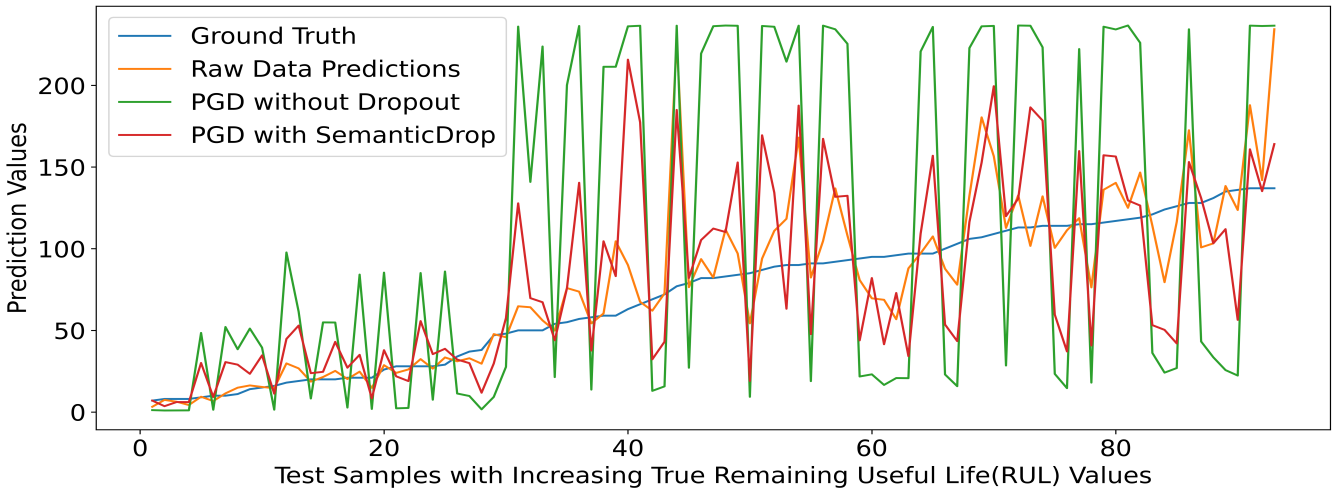

Figure 9. Regression Results (with GN) under Deviation Maximization PGD Adversarial Attack with $advEps = 0.05(5\%)$

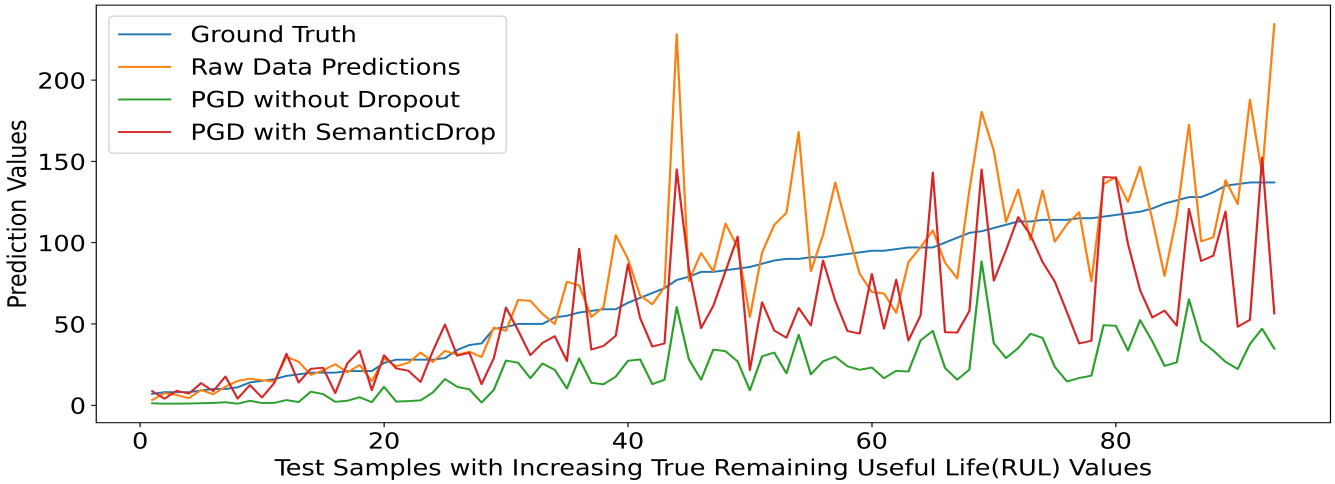

Figure 10. Regression Results (with GN) under Prediction Minimization PGD Adversarial Attack with $advEps = 0.05(5\%)$

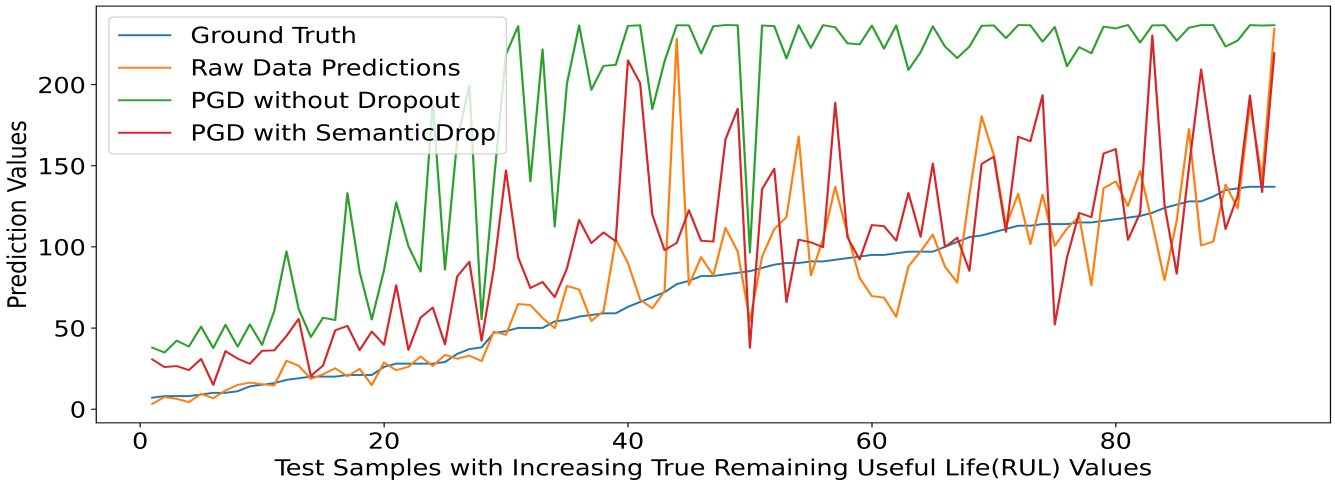

Figure 11. Regression Results (with GN) under Prediction Maximization PGD Adversarial Attack with $advEps = 0.05(5\%)$

Settings	AdvLoss	AdvEps	Nat+NoDrp	Nat+Drp	Nat+SemDrp	GN+NoDrp	GN+Drp	GN+SemDrp
Binary Classif-ication Accuracy /%	Binary Cross-Entropy	0	94.6	87.1 / 84.9 / 83.9	90.3 / 88.2 / 82.8	**96.8**	87.1 / 84.9 / 84.9	88.2 / 86 / 88.2
		0.02	55.9	71 / 69.9 / 71	**86 / 87.1 / 83.9**	75.3	78.5 / 75.3 / 73.1	86 / 81.7 / 82.8
		0.04	19.4	51.6 / 60.2 / 65.6	80.6 / 78.5 / 74.2	43	60.2 / 64.5 / 68.8	**82.8 / 80.6 / 78.5**
		0.06	9.7	30.1 / 40.9 / 51.6	63.4 / 67.7 / 60.2	31.2	52.7 / 60.2 / 65.6	**74.2 / 73.1 / 73.1**
		0.08	4.3	16.1 / 21.5 / 37.6	46.2 / 45.2 / 41.9	25.8	34.4 / 44.1 / 57	**71 / 71 / 69.9**
		0.1	2.2	11.8 / 21.5 / 26.9	38.7 / 41.9 / 37.6	18.3	31.2 / 35.5 / 45.2	**66.7 / 67.7 / 65.6**
Multiclass Classif-ication Accuracy /%	Categorical Cross-Entropy	0	**82.8**	69.9 / 67.7 / 62.4	75.3 / 80.6 / 77.4	75.3	72 / 75.3 / 65.6	76.3 / 74.2 / 72
		0.02	58.1	62.4 / 62.4 / 69.9	61.3 / 62.4 / 64.5	65.6	63.4 / 66.7 / 61.3	**66.7 / 67.7 / 71**
		0.04	57	55.9 / 57 / 61.3	58.1 / 59.1 / 58.1	63.4	61.3 / 61.3 / 62.4	**62.4 / 63.4 / 61.3**
		0.06	55.9	57 / 57 / 60.2	57 / 57 / 57	60.2	60.2 / 60.2 / 59.1	**61.3 / 61.3 / 61.3**
		0.08	55.9	55.9 / 55.9 / 55.9	57 / 57 / 57	60.2	**60.2 / 60.2 / 63.4**	60.2 / 61.3 / 60.2
		0.1	58.1	55.9 / 55.9 / 55.9	58.1 / 58.1 / 58.1	59.1	60.2 / 60.2 / 58.1	**60.2 / 62.4 / 61.3**
Regression Mean Absolute Error /cycle	Mean Squared Error	0	**13.4**	16.2 / 18.4 / 20.9	17.4 / 19.1 / 16.9	16.6	16.2 / 19.8 / 22.4	21.1 / 20.7 / 19.7
		0.02	47.1	36.8 / 35.8 / 33.5	**25.6 / 23.5 / 24.8**	65.4	28.9 / 28.5 / 29.1	27.9 / 26.8 / 26
		0.04	59.6	51.4 / 48.3 / 47.7	37.5 / 34.5 / 35.4	80.1	37.9 / 34.5 / 30.8	**34.9 / 31.4 / 32.6**
		0.06	68.4	59.5 / 56.6 / 52.6	50.6 / 45.8 / 47.5	89.8	44.6 / 42.6 / 37.1	**38.1 / 37.3 / 38**
		0.08	70.7	65.5 / 60.3 / 59.5	54.3 / 47.5 / 49.3	98.9	51.3 / 44.1 / 39.2	**40.9 / 38.5 / 36.6**
		0.1	73.1	65.9 / 61.5 / 59.4	66 / 59 / 59.5	100.6	57.7 / 48.4 / 44.9	**43.6 / 39.6 / 37.9**
	Minimize	0	**13.4**	16.2 / 18.4 / 20.9	17.4 / 19.1 / 16.9	16.6	16.2 / 19.8 / 22.4	21.1 / 20.7 / 19.7
		0.02	34.2	26.8 / 24.3 / 21.2	**19.6 / 19.6 / 18.8**	32.2	21.8 / 21.9 / 21.8	22.8 / 22.2 / 23.4
		0.04	46.2	38.3 / 34 / 30.7	**26.2 / 24.1 / 25.6**	47.3	29 / 28 / 26	25.5 / 25.1 / 26.2
		0.06	52.2	45.1 / 40.9 / 36.2	34.7 / 33.4 / 34.3	53.7	34.4 / 30.7 / 29.1	**26.4 / 26.5 / 28.5**
		0.08	55.9	49.8 / 45.3 / 40.9	41 / 40.2 / 41.2	58.4	39 / 34.1 / 31.7	**28.4 / 28.7 / 31.1**
		0.1	58.2	52.3 / 48.1 / 43.6	45.7 / 45 / 46.1	60.2	41.3 / 35.6 / 34	**30.2 / 28.1 / 31.6**
	Maximize	0	**13.4**	16.2 / 18.4 / 20.9	17.4 / 19.1 / 16.9	16.6	16.2 / 19.8 / 22.4	21.1 / 20.7 / 19.7
		0.02	45.9	35.6 / 37.6 / 35.2	24.4 / 20.3 / 21.7	74.1	26.6 / 26.6 / 28.3	**21.7 / 22.6 / 19.5**
		0.04	59.4	49.3 / 48.5 / 49.6	**37.7 / 29.9 / 32.5**	97.8	37.7 / 33.8 / 31.7	34.1 / 31.6 / 30.3
		0.06	66.9	57.5 / 56.1 / 55.3	48.7 / 40.4 / 42.4	111.7	45.5 / 43 / 36.5	**38.3 / 38.1 / 33**
		0.08	71.4	59.8 / 59.7 / 58	55 / 44.5 / 47.3	121.6	51.4 / 47.5 / 41	**46.5 / 37.6 / 38.5**
		0.1	75.3	63.2 / 59.7 / 59.5	61.9 / 50.1 / 51.6	127.9	59.7 / 51.2 / 46.2	**45 / 36.4 / 37.7**

Table 3. Adversarial Regression Error Rates under Projected Gradient Descent(PGD) Attacks

REFERENCES

Akhtar, N., & Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, *6*, 14410–14430.

Athalye, A., Carlini, N., & Wagner, D. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*.

Baraldi, P., Cadini, F., Mangili, F., & Zio, E. (2013). Model-based and data-driven prognostics under different available information. *Probabilistic Engineering Mechanics*, *32*, 66–79.

Biggio, B., & Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, *84*, 317–331.

Broenink, J. F. (1990). Introduction to physical systems modelling with bond graphs. *SiE Whitebook on Simulation Methodologies*.

Buckman, J., Roy, A., Raffel, C., & Goodfellow, I. (2018). Thermometer encoding: One hot way to resist adversarial examples.

Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., ... Kurakin, A. (2019). On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*.

Chao, M. A., Kulkarni, C., Goebel, K., & Fink, O. (2020). Fusing physics-based and deep learning models for prognostics. *arXiv preprint arXiv:2003.00732*.

Echauz, J., Kenemer, K., Hussein, S., Dhaliwal, J., Shintre, S., Grzonkowski, S., & Gardner, A. (2019). Adversarial campaign mitigation via roc-centric prognostics. In *Proceedings of the annual conference of the phm society* (Vol. 11).

Ellefsen, A. L., Bjørlykhaug, E., Æsøy, V., Ushakov, S., & Zhang, H. (2019). Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliability Engineering & System Safety*, *183*, 240–251.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples (2014). *arXiv preprint arXiv:1412.6572*.

Heimes, F. O. (2008). Recurrent neural networks for remaining useful life estimation. In *2008 international conference on prognostics and health management* (pp. 1–6).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Li, X., Ding, Q., & Sun, J.-Q. (2018). Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, *172*, 1–11.

Li, Y., Zhang, H., Bermudez, C., Chen, Y., Landman, B. A., & Vorobeychik, Y. (2020). Anatomical context protects deep learning from adversarial perturbations in medical imaging. *Neurocomputing*, *379*, 370–378.

Liu, X., Cheng, M., Zhang, H., & Hsieh, C.-J. (2018). Towards robust neural networks via random self-ensemble. In

Proceedings of the european conference on computer vision (eccv) (pp. 369–385).

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Pasareanu, C. S., Gopinath, D., & Yu, H. (2018). Compositional verification for autonomous systems with deep learning components. *arXiv preprint arXiv:1810.08303*.

Patro, S., & Sahu, K. K. (2015). Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*.

Ramasso, E., & Saxena, A. (2014). Performance benchmarking and analysis of prognostic methods for cmapss datasets. *International Journal of Prognostics and Health Management*, *5*(2), 1–15.

Samangouei, P., Kabkab, M., & Chellappa, R. (2018). Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*.

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. (2008). Metrics for evaluating performance of prognostic techniques. In *2008 international conference on prognostics and health management* (pp. 1–17).

Saxena, A., & Goebel, K. (2008). C-mapss data set. *NASA Ames Prognostics Data Repository*.

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine prognostics. *NASA Technical Reports*.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Umberto Griffo. (2019). *Recurrent Neural Networks for Predictive Maintenance.* {https://github.com/umbertogriffo/Predictive-Maintenance-using-LSTM}. ([Online; accessed 19-May-2020])

Valtierra-Rodriguez, M., de Jesus Romero-Troncoso, R., Osornio-Rios, R. A., & Garcia-Perez, A. (2013). Detection and classification of single and combined power quality disturbances using neural networks. *IEEE Transactions on Industrial Electronics*, *61*(5), 2473–2482.

Vorobeychik, Y., & Kantarcioglu, M. (2018). Adversarial machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *12*(3), 1–169.

Wang, S., Wang, X., Zhao, P., Wen, W., Kaeli, D., Chin, P., & Lin, X. (2018). Defensive dropout for hardening deep neural networks under adversarial attacks. In *Proceedings of the international conference on computer-aided design* (p. 71).

Wang, T., Yu, J., Siegel, D., & Lee, J. (2008). A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–6).

Xie, C., Wang, J., Zhang, Z., Ren, Z., & Yuille, A. (2017). Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*.

Yang, Y., Zhang, G., Katabi, D., & Xu, Z. (2019). Me-net: Towards effective adversarial robustness with matrix estimation. *arXiv preprint arXiv:1905.11971*.

Zantedeschi, V., Nicolae, M.-I., & Rawat, A. (2017). Efficient defenses against adversarial attacks. In *Proceedings of the 10th acm workshop on artificial intelligence and security* (pp. 39–49).

Zhang, J., & Lee, J. (2011). A review on prognostics and health monitoring of li-ion battery. *Journal of power sources*, *196*(15), 6007–6014.

Zheng, S., Ristovski, K., Farahat, A., & Gupta, C. (2017). Long short-term memory network for remaining useful life estimation. In *2017 ieee international conference on prognostics and health management (icphm)* (pp. 88–95).