ADVERSARIAL ROBUSTNESS OF DEEP LEARNING ENABLED INDUSTRY 4.0 PROGNOSTICS

A Thesis presented to

the Faculty of the Graduate School

at the University of Missouri

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

GAUTAM RAJ MODE

Thesis Supervisor: Prof. Khaza Anuarul Hoque

December 2020

The undersigned, appointed by the Dean of the Graduate School, have examined the thesis entitled:

ADVERSARIAL ROBUSTNESS OF DEEP LEARNING ENABLED INDUSTRY 4.0 PROGNOSTICS

presented by Gautam Raj Mode, a candidate for the degree of Master of Science and hereby certify that, in their opinion, it is worthy of acceptance.

Prof. Khaza Anuarul Hoque

Prof. Prasad Calyam

Prof. Yaw Adu-Gyamfi

ACKNOWLEDGMENTS

I would like to thank Dr. Khaza Anuarul Hoque for offering me this opportunity to pursue my Master's degree, guiding my research, and providing moral and emotional support during my Master's journey. I was given the wonderful opportunity by Khaza Anuarul Hoque, to work in the Dependable Cyber Physical Systems (DCPS) Lab and was provided the higher end lab facilities in the pursuit of my challenging research which I am very much grateful. I would like to express my gratitude to Dr. Prasad Calyam and Dr. Yaw Adu-Gyamfi for their interest and consent to be part of my thesis committee.

Finally, but most importantly, special thanks must go to my loving and supportive parents, Sudhakar and Rajani, and my wonderful sister, Raasi, who provided endless support, understanding, and unending inspiration and always put a smile on my face. Thank you for all your love.

Gautam Raj Mode

Contents

List of Tables

List of Figures

Figure Page

ABSTRACT

The advent of Industry 4.0 in automation and data exchange leads us toward a constant evolution in smart manufacturing environments, including extensive utilization of Internet-of-Things (IoT) and Deep Learning (DL). Specifically, the state-of-the-art Prognostics and Health Management (PHM) has shown great success in achieving a competitive edge in Industry 4.0 by reducing maintenance cost, downtime, and increasing productivity by making data-driven informed decisions. These state-of-the-art PHM systems employ IoT device data and DL algorithms to make informed decisions/predictions of Remaining Useful Life (RUL). Unfortunately, IoT sensors and DL algorithms, both are prone to cyber-attacks. For instance, deep learning algorithms are known for their susceptibility to adversarial examples. Such adversarial attacks have been extensively studied in the computer vision domain. However, it is surprising that their impact on the PHM domain is yet not explored. Thus, modern data-driven intelligent PHM systems pose a significant threat to safety- and cost-critical applications. Towards this, in this thesis, we propose a methodology to design adversarially robust PHM systems by analyzing the effect of different types of adversarial attacks on several DL enabled PHM models. More specifically, we craft adversarial attacks using Fast Gradient Sign Method (FGSM) and Basic Iterative Method (BIM) and evaluate their impact on Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Convolutional Neural Network (CNN), Bi-directional LSTM, and Multi-layer perceptron (MLP) based PHM models using the proposed methodology. The obtained results using NASA's turbofan engine, and a well-known battery PHM dataset show that these systems are vulnerable to adversarial attacks and can cause a serious defect in the RUL prediction. We also analyze the impact of adversarial training using the proposed methodology to enhance the adversarial robustness of the PHM systems. The obtained results show that adversarial training is successful in significantly improvising the robustness of these PHM models.

Keywords: Adversarial attack, Deep learning, Adversarial training, Prognostics and health management (PHM), Industry 4.0

Chapter 1

Introduction

The goal of this chapter is to discuss certain key concepts that provide an understanding of the background to prognostics and health management, the threats to PHM systems, and the motivation of this thesis. This thesis addresses the PHM approach applied to condition-based maintenance technology and also addresses different threats to the PHM systems.

1.1 Motivation

The advent of Industry 4.0 in automation and data exchange leads us toward a constant evolution in smart manufacturing environments, including an intensive utilization of Internet-of-Things (IoT) and Deep Learning (DL). Specifically, the state-of-the-art Prognostics and Health Management (PHM) [47] has shown great success in achieving a competitive edge in Industry 4.0 by reducing the maintenance cost, downtime, and increasing productivity by making data-driven informed decisions. For instance, modern PHM techniques can help reduce downtime by 35%-45%, maintenance cost by 20%-25%, and can increase production by 20%-25% [7]. The ability to sense changes in the physical world (such as temperature, vibration, pressure, etc.) using IoT sensors, and to analyze the sensed data using the state-of-the-art DL algorithms for different prognostic tasks such

as the Remaining Useful Life (RUL) prediction has enabled a highly reliable and cost-efficient industrial automation framework. Unfortunately, IoT sensors are also known for their vulnerability to cyber attacks [106, 144], and DL algorithms can also be easily fooled by adversarial examples [113].

According to a recent report from the *Malwarebytes*, cyber-threats against businesses/factories have increased by more than 200% over the past year [2]. Specifically, it is very hard to detect stealthy attacks, such as False Data Injection Attack (FDIA) [79] on the PdM system due to the nature of the attack. In false data injection attack (FDIA) [79], an attacker stealthily compromises measurements from IoT sensors, such that the manipulated sensor measurements bypass the sensor's basic 'faulty data' detection mechanism and propagates to the sensor output undetected. An FDI attack can be implemented by compromising physical sensors, sensor communication network, and data processing programs. Such attacks on a PdM system can act as a "time bomb" since FDIAs on a PdM system do not show their effect immediately, which also helps in bypassing basic anomaly detection mechanisms. Instead, the attack propagates from the sensor to the ML part of the PdM system and fools the system by predicting a delayed asset failure or maintenance interval. This might incur a significant cost by inducing an unplanned failure or loss of human lives in safety-critical applications [6, 8, 10].

From the perspective of computer vision, an adversarial example can be an image formed by making small perturbations (insignificant to the human eye) so that a classifier misclassifies it with high confidence. Such adversarial attacks have been extensively studied in the computer vision domain [14]. Even though advanced data-driven PHM depends on DL, it is very surprising that the impact of adversarial attacks on the PHM domain has not been studied yet. In manufacturing, adversarial attacks can lead to a wrong prognostic decision, e.g., a wrong estimation of RUL can delay the maintenance of a machine leading to unexpected failures. Such unexpected failures are considered a primary operational risk, as they can hinder productivity and can incur a huge loss. For example, in the mod-

ern automotive industry, an assembly line has several robots working on a car, and if even one robot fails, it will result in the halt of the entire assembly line, causing loss of valuable production time and increased production cost. In another situation, a wrong prognostic prediction in an operating autonomous vehicle, or aircraft may lead to loss of human lives. Thus, even though the utilization of IoT and ML is revolutionizing the smart industry, the vulnerabilities related to IoT and ML possesses a great challenge for Industry 4.0.

In this thesis, we provide a methodology to design and build robust PHM systems. We employ our methodology to build a PHM system, craft adversarial attacks for the PHM system and also implement adversarial training to make a robust PHM system. A researcher/engineer can use this methodology in the pre-deployment stage to build robust PHM models. In this thesis, we apply cyber-attacks (FDI) and craft adversarial attacks from the image domain to deep learning regression models for prognostics. We present experimental studies using NASA's C-MAPSS [99] and Li-ion battery datasets [101]. For adversarial attacks, we consider Fast Gradient Sign Method (FGSM) [52] and Basic Iterative Method (BIM) [67] adversarial attacks. The obtained results show that state-of-the-art DL regression models are prone to cyber-threats and adversarial attacks. We also transfer the adversarial examples crafted for one DL model to another, also known as a black-box attack. This shows that adversarial examples crafted for one network architecture can be transferred to other architectures. We also enhance the adversarial robustness of the PHM system by performing adversarial training to mitigate the impact of adversarial threats. The thesis highlights the importance of learning and defending against cyber-threats and adversarial attacks in deep learning regression models for prognostics.

1.2 Problem statement

The goal of this thesis is to answer the research question formulated below, i.e., to improve the prognostic and health management of engineering systems to facilitate better operation and maintenance decision making. The following research questions (RQs) have been formulated based on the defined prognostic needs:

1. How does one incorporate time-varying contextual, operational, and condition data as well as historical data to predict the remaining useful life of an asset in a PHM model?

2. Does cyber-attack impact predictions of deep learning-enabled PHM systems?

3. What is the impact of carefully crafted adversarial examples on state-of-the-art PHM model?

4. Can adversarial examples crafted for one DL model be transferred to other DL models? Does it have the same impact as of white-box attack?

5. How does one measure adversarial robustness? Does adversarial training mitigate the impact of adversarial attacks on DL enabled PHM systems?

1.3 Contributions

To summarize, the main contributions of this paper are:

1. An empirical study of False Data Injection (FDI) attack on state-of-the-art PHM domain and analyze the impact of real-life scenarios using PHM datasets specifically NASA's C-MAPSS and Li-ion battery datasets.

2. Formalize and Craft adversarial examples for DL based regression models for prognostics using adversarial attacks that are primarily designed for the

image domain and apply them to the PHM domain. To be specific, we use untargeted (FGSM and BIM) attacks to craft adversarial examples for Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Bi-directional LSTM, Multilayer Perceptron (MLP), and Convolutional Neural Network (CNN) regression models.

3. An empirical study of adversarial attacks on state-of-the-art PHM domain and we highlight the impact of adversarial attacks in real-life scenarios using PHM datasets.

4. A comprehensive study of the transferability property of adversarial examples crafted by untargeted adversarial attacks in DL based regression models for prognostics.

5. A discussion on the potential defense techniques to improve the adversarial robustness of a PHM system. Also, employing adversarial training (one of the adversarial defense strategies) to build a robust PHM model.

1.4 Thesis organization

The thesis is structured into six chapters. Chapter 1 introduces the motivation and contribution of this thesis. Chapter 2 presents a review of literature relevant to PHM, application of deep learning, and IoT in PHM, cyber-attacks on PHM systems, adversarial examples in PHM. Chapter 3 provides helpful insight into the impact of cyber-attack (False Data Injection attacks) on DL enabled PHM system. Chapter 4 describes the methodology employed in the general structure of the proposed framework to build a PHM model, crafting adversarial examples for the proposed PHM model and checking the adversarial robustness of the adversarially trained PHM model. Chapter 5 does an empirical study of crafting adversarial examples for a PHM system and also implements adversarial training to increase the adversarial robustness of a PHM system. Chapter 6 concludes the thesis and

proposes further research.

Chapter 2

Literature review

The goal of this section is to introduce Prognostic and Health Management (PHM) system and architecture of the state-of-the-art PHM system. This chapter also presents the vulnerabilities of a PHM system.

2.1 Prognostic and Health Management (PHM)

One of the major problems in industry is the extension of the useful life of high-performance systems. Proper maintenance plays an important role by extending the useful life, reducing the life cycle costs and improving the reliability and availability. The reliability of a component or system is a measurement of its performance regarding its intended function above a minimum standard for a specified period of time in defined circumstances. Prog- nostics and health management (PHM) is an engineering discipline that aims to maintain the system behavior and function while ensuring mission success, safety and effectiveness. Health management using a proper condition-based maintenance (CBM) deployment is a worldwide accepted technique and has grown to be very popular in many industries over the past few decades. These techniques are relevant in environments where the prediction of a failure and the prevention and mitigation of its consequences increase profits and enhance the safety of the facilities concerned.

Prognosis is the most critical part of this process and is currently recognised as a key feature in maintenance strategies since the estimation of the remaining useful life (RUL) is essential. PHM can provide a state assessment of the future health of a system or of components of interest, e.g., when a degraded state has been found. Using this technology, one can estimate how long it will take before the equipment will reach a failure threshold under future operating and environmental conditions. PHM technology is relatively immature compared to diagnosis technology, and a challenging task facing the research community is to overcome some of the major barriers obstructing the application of PHM technology to real-world industrial systems. One major challenge is in actually predicting the RUL since it often depends on multiple parameters that are time and operational dependent. These relationships and models have to be derived from a physical understanding about the system or by measuring its degradation behaviour.

PHM addresses the prediction of future conditions and how to manage the health of an asset. In complex industrial applications, fault propagation is often difficult to predict, especially where multiple dependencies and complex relations exist between different process parameters and the asset health. In most cases, the degradation data for the entire life of the components are not available, and no explicit relationship has been established between damage and the measured health condition. Therefore, there is a need for methods that can be used in these cases to classify the health states and predict the remaining useful life.

2.2 Deep learning in time-series forecasting

Time series forecasting is a challenging and important problem in the data science and data mining community. Therefore, hundreds of methods have been proposed for their analysis [37]. With the success of machine learning (ML) algorithms in different domains, ML techniques for time series forecasting is also popular [85, 115]. However, among these methods, only a few (when compared to

the non-DL methods) have considered DL methods for time series forecasting[24, 46, 120].

In this work, we focus on the time/cost-sensitive and safety-critical applications of deep learning time series forecasting, which motivates us for investigating the impact of adversarial attacks on them. Specifically, we explore the impact of adversarial attacks on LSTM, CNN, GRU, MLP and Bi-LSTM. All of these models are known for their effectiveness in time series forecasting. LSTM is capable of learning long-term dependencies using several gates and thus suits well the time series forecasting problems. In [116], authors employ an LSTM model for predicting the traffic flow with missing data. The other successful applications of LSTM in time series forecasting includes petroleum production forecasting [97], financial time series forecasting [26], solar radiation forecasting [109], and remaining useful life prediction of aircraft engines [129]. GRU is an improvised version of Recurrent Neural Network(RNN) [63], and also effective in time series forecasting [126]. For instance, in [114], authors employ 1D convnets and bidirectional GRUs for air pollution forecasting in Beijing, China. The other applications of GRU models in time series forecasting include personalized healthcare and climate forecasting [36], mine gas concentration forecasting [62], smart grid bus load forecasting [103]. In [41], authors present a CNN-based bagging model for forecasting hourly loads in a smart grid. Apart from the energy domain, CNNs are also useful for financial time series forecasting [15, 102].

2.3 PHM cloud-edge architecture and threat model

Prognostics and Health Management (PHM) has gained a lot of attention in recent years for intelligent manufacturing in the context of Industry 4.0 [131]. The state-of-the-art PHM incorporates both IoT devices for sensing [47], and machine learning algorithms for analyzing the sensed data, and thus making a way for smart analytics to predict the future state of equipment. The equipment moni-

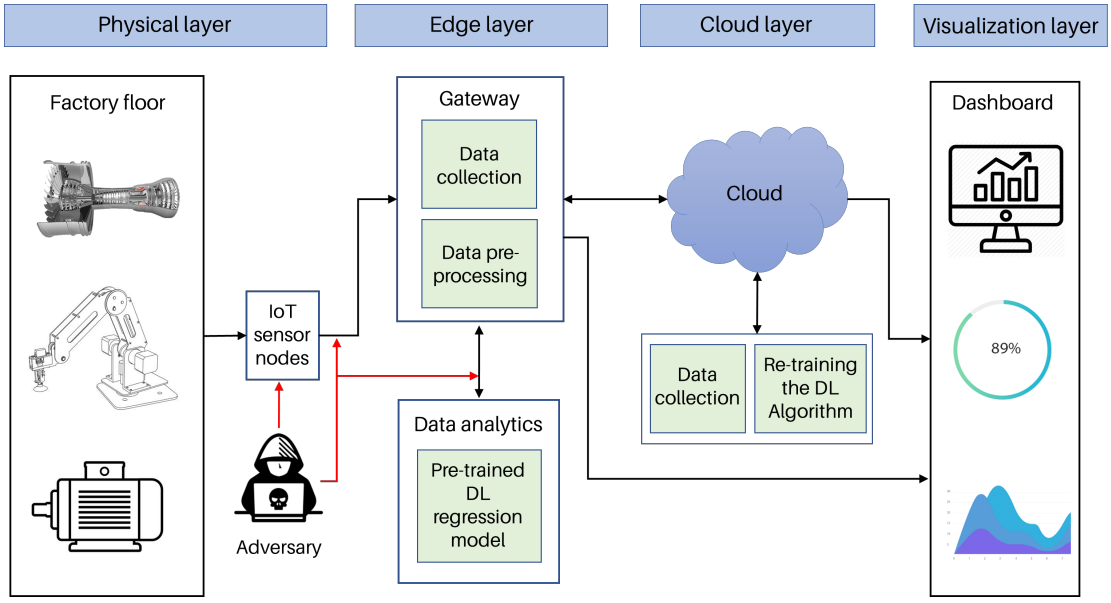

Figure 2.1: PHM cloud-edge architecture and threat model

toring system includes input from IoT sensors that measure different parameters, such as pressure, temperature, speed, vibration, etc. Employing these IoT sensors eliminates the requirement for manual inspection/analysis, and hence the operating condition of equipment can be monitored using automated PHM systems. In PHM, Remaining Useful Lifetime (RUL) indicates the amount of time left before a piece of equipment or machine fails or degrades to a point at which it cannot perform its intended function anymore. Indeed, it is important to have an accurate RUL estimation since an early prediction may result in over-maintenance and a late prediction could lead to catastrophic failures. From a machine learning perspective, the prognostic is a regression problem, as the target value (RUL) is in the real domain. Thus, an RUL estimation involves learning a function that maps the condition of machines to its RUL estimates.

In the era of big data, an efficient way to analyze the huge amount of data is crucial. Deep learning (DL) provides a complementary approach for analyzing data obtained from IoT devices to provide accurate insights by identifying failure signatures, profiles, and providing an actionable prediction of failure through RUL estimation [44, 131]. DL algorithms, especially LSTM, GRU, and CNN have shown great success in prognostics tasks [31, 71, 90, 94, 127, 133]. Figure2.1 shows an

overview of an IoT and DL enabled cloud-edge architecture, divided into physical, edge, cloud, and visualization layer [13], and also a potential attack scenario. The *physical layer* involves several IoT sensors nodes, which collect different sensor measurements from equipment. The data from the sensor nodes are sent to the edge for processing. The *edge layer* pre-process the data for passing to the next *cloud* layer for training/re-training purposes. The *edge layer* also has a pre-trained deep learning model to predict the future health state of the equipment. *The cloud* layer performs in-depth analysis and also performs training/re-training of the DL models on the newly arrived data from the edge layer. After the re-training, the re-trained DL model is sent back to the edge layer for improved accuracy of the machine's health state prediction. The *visualization layer* uses the data collected from the field, along with the results from the PHM model, and provides a visual representation of actionable insights to an engineer.

PHM attack scenarios: Let us consider an attack scenario as shown in the Figure 2.1. An adversary can either compromise the physical sensors or the communication network used between IoT sensor nodes and the edge layer. If successful, an adversary can capture the sensor data, craft adversarial examples for the captured data, and inject the crafted adversarial examples into the PHM system through a False Data Injection (FDI) [92] attack. Indeed, the cloud layer also has its vulnerabilities [65], however, typically third-party cloud services such as AWS, Azure, etc. have their security measures [17] which makes them less vulnerable when compared to the physical and edge layer. More detail about the IoT and DL related threats to the PHM systems can be found in the extended version of this paper [83].

2.4 Vulnerabilities of the PHM system

2.4.1 Attacks on IoT sensors

IoT devices and deep learning have paved the way for smart analytics, however, their vulnerabilities also create new means of attacks. As mentioned earlier, in the PHM context, IoT sensors collect a variety of parameters including temperature, pressure, speed, vibration, etc., which is then transmitted via a network (in many cases using a wireless network) to a centralized processing unit to make informed decisions. The data collected from the sensors greatly influence the prognostics and maintenance related decisions. Even if the network is secured, any attacks on the sensors or the infrastructure could result in incorrect decisions and would result in a considerable impact on the performance of the PHM system [12, 96]. For instance, attackers can use the sensors to transfer malicious code, trigger messages to activate a malware planted in an IoT device [111], capture sensitive information shared between devices [78, 144], or even capture encryption and depreciation keys for extracting encrypted information [38]. Understanding these sensor-based threats is necessary for researchers to design reliable solutions to detect and prevent these threats efficiently. As a result, IoT sensor-based threats have gained a lot of attention from researchers in academia and in industry [12, 21, 96]. Unfortunately, most of the existing IoT security frameworks are not suitable for detecting sensor-based threats at the system level [106]. IoT sensors are typically small and hence they limited by power and resource constraints. This is indeed an obstacle for implementing the complicated security mechanisms on IoT sensors and devices. Note, sensor attacks detection and mitigation is also an active research problem in the cyber-physical system domain [40, 104]. However, in the context of industrial automation, how attacks on IoT sensors influence ML algorithms for making incorrect decisions is yet to be explored in detail.

2.4.2 Adversarial attacks on deep learning

In addition to IoT attacks, the performance of a PHM system can also be considerably affected by orchestrated security attacks on DL algorithms. The study of the effect of adversarial attacks on machine learning techniques is known as *adversarial machine learning*, and is one of the most active research topics in the deep learning community [68]. In [67, 76, 113], the authors have shown how adversarial examples can be used for deep learning algorithms to make a wrong classification. Since attacks on DL algorithms may have catastrophic consequences, their detection and mitigation have been explored in many recent literature [51, 95]. However, most of them are venerable to future attacks [28]. For instance, in [48, 66], the authors explored the adversarial training technique to mitigate the adversarial attacks. Adversarial training injects adversarial examples into training data to increase robustness. Unfortunately, later on, researchers found the adversarial training technique to be inefficient for mitigating those attacks [130]. A detailed survey of proposed defenses in the adversarial ML domain can be found in [30]. Even though adversarial machine learning is an active research area, most of the works in this area are limited to the computer vision domain or its variants. The effect of adversarial attacks in other domains, such as regression tasks for PHM is not yet explored. In our work, we study the impact of adversarial attacks on DL based regression models for PHM.

Interestingly, the adversarial attack approaches for multivariate time series DL regression models have been ignored by the community. There are only two previous works that consider adversarial attacks on time series. In [86], the authors adopt a soft K-Nearest-Neighbours (KNN) coupled with Dynamic Time Warping (DTW) and show that the adversarial examples can fool the proposed classifier on a simulated dataset. Unfortunately, the KNN classifier is no longer considered the state-of-art classifier for time series data [18]. The authors in [43], utilize the FGSM and BIM attacks to fool Residual network (ResNet) classifiers for *univariate* time series *classification* tasks. In our work, we also employ the FGSM and BIM

attacks, however, we apply and evaluate their impacts on DL regression models for *mutivariate* time series *forecasting*.

In summary, this work sheds light on the resiliency of DL regression models for multivariate time series forecasting in real-world safety-critical and cost-critical applications. This will guide the data mining, data science, and machine learning researchers to develop techniques for detecting and mitigating adversarial attacks in time series data.

2.5 Summary

This chapter has introduced the development of Prognostics and health management, along with various techniques and models that could be chosen to implement it. This chapter also introduced the state-of-the-art architecture of a PHM system and also its vulnerabilities. In the next chapter, we do a motivational study of cyber-threats to the PHM system. This study analyzes the impact of cyber-threats to the PHM system by performing False Data injection (FDI) attacks on the IoT sensors of a PHM system.

Chapter 3

False data injection attacks on PHM systems

The goal of this chapter is to discuss the cyber-threats associated with a well-connected PHM system. In this Chapter, we analyze the impact of False Data Injection (FDI) attack on Deep Learning enabled PHM systems. The main goal of this chapter is to show that PHM systems are vulnerable to cyber-threats.

3.1 Cyber-attacks on PHM systems

Current advances in machine learning (ML) techniques and Internet-of-Things (IoT) sensors has enabled the emergence of predictive maintenance (PdM), which is a method of preventing asset failure by analyzing production data and identifying patterns to predict issues before they happen. State-of-the-art PdM techniques can help reduce downtime by 35%-45%, maintenance cost by 20%-25%, and can increase production by 20%-25% [7]. Due to these benefits, IoT and ML-enabled PdM solutions are reshaping automotive, aerospace, oil and gas, transportation, manufacturing industries and also reshaping the national defense. Specifically, deep learning (DL) algorithms have recently shown tremendous success in such PdM applications [39]. Unfortunately, IoT sensors and DL algorithms are both

susceptible to attacks [107], which poses a significant threat to the overall PdM system. According to a recent report from the *Malwarebytes*, cyber-threats against businesses/factories have increased by more than 200% over the past year [2].

Specifically, it is very hard to detect stealthy attacks, such as False Data Injection Attack (FDIA) [79] on the PdM system due to the nature of the attack. In false data injection attack (FDIA) [79], an attacker stealthily compromises measurements from IoT sensors, such that the manipulated sensor measurements bypass the sensor's basic 'faulty data' detection mechanism and propagates to the sensor output undetected. An FDI attack can be implemented by compromising physical sensors, sensor communication network, and data processing programs. Such attacks on a PdM system can act as a "time bomb" since FDIAs on a PdM system do not show their effect immediately, which also helps in bypassing basic anomaly detection mechanisms. Instead, the attack propagates from the sensor to the ML part of the PdM system and fools the system by predicting a delayed asset failure or maintenance interval. This might incur a significant cost by inducing an unplanned failure or loss of human lives in safety-critical applications [6, 8, 10].

FDI attacks have already caused many known disastrous incidents, such as the Northeast blackout of 2003 in the USA and the Ukrainian power grid attack affecting over 230,000 people, leaving them without electricity for several hours. Extensive research has been performed on the detection and mitigation of FDI attacks in cyber-physical systems (CPS) domain [54, 70, 73]. Unfortunately, the effect of FDIA on a PdM system is yet not explored which motivates our research. In the case of aircraft engine PdM systems, FDIAs may result in the delay of timely maintenance and lead to mid-air engine failures which are catastrophic. Current users of PdM systems for aircraft engine maintenance include Pratt and Whitney, Rolls-Royce, Honeywell, General electronics and the US Air force [1, 8, 9, 11]. For example, Bombadier's new jetliner uses a Pratt and Whitney turbofan engine that boasted more than 5,000 sensors [3, 5]. Powered with the modern DL algorithms, this engine can predict the future demands of the engine, perform adjustments,

and thus save 15% of fuel usage. However, the vulnerability of sensor-attacks against for such IoT and ML-based engines is considered a challenge [3, 25, 35]. The existing sensor attack detection solutions in the IoT and cyber-physical system domain is not sufficient to address this problem due to the fact that, when deployed individually to the thousands of sensors, most of the existing techniques suffer from scalability problems and resource overheads as many IoT sensors are power and resource-constrained.

In this chapter, we model continuous and interim FDIAs on IoT sensors and show their impact on a PdM model by performing a case study on the aircraft Predictive Maintenance (PdM) system. We use the C-MAPSS [99] (Commercial Modular Aero-Propulsion System Simulation) dataset[1]. At first, to build an accurate predictive model, we train the Long Short-Term Memory (LSTM), Gated recurrent unit (GRU), and Convolutional neural network (CNN) algorithms using the C-MAPSS dataset. We evaluate these three predictive models, and the obtained results show that the GRU-based model predicts the RUL[2] most accurately. The obtained results from the GRU-based model outperforms the recent works that use DL for RUL prediction using the C-MAPSS dataset in [42, 136, 140] (by predicting RUL 1.3-1.9 times more accurately).

Afterward, we model two types of false data injection attacks (FDIA) on the C-MAPSS dataset and evaluate their impact on CNN, LSTM, and GRU-based PdM models. To be more realistic, we model attack only on 3 sensors among the 21 sensors in the turbofan engine. The obtained results show that all the PdM models are greatly defected by the FDIA even if only 3 out of the 21 sensors are attacked. However, the GRU-based PdM model is comparatively more accurate and resilient to FDIA when compared to the other evaluated PdM models. In terms of sensitivity, we also explore that CNN is way more sensitive to FDIAs when compared to the LSTM and GRU. This is indeed an important observation

[1]a popular turbofan engine degradation dataset published by NASA's Prognostics Center of Excellence (PCoE)

[2]Remaining useful life (RUL) is the length of time a machine is likely to operate before it requires repair or replacement.

since CNN-based techniques are quite popular in asset maintenance [27, 60, 108] and our results indicate that special measures should be taken for designing a CNN-based PdM. Afterward, we analyze the GRU-based PdM model using four different sequence lengths. The obtained results show an interesting relationship between the accuracy, the resiliency and the sequence length of the models. To the best of our knowledge, this is the *first work* that demonstrates the effects of IoT sensor attacks on a deep learning-enabled PdM system.

3.2 DL algorithms for RUL prediction

As mentioned earlier, RUL can be predicted using different ML algorithms. For this chapter, we utilize LSTM, GRU, and CNN algorithms and compare their performance.

3.2.1 Long short-term memory model (LSTM)

An LSTM [57] is a special kind of Recursive Neural Network (RNN), capable of learning long-term dependencies. LSTM is explicitly designed to avoid long term dependency problems, which is prevalent in RNN. It has achieved great praise in the field of machine learning and speech recognition. Some of the neural networks have a dependency problem, but an LSTM can overcome the problem of dependency by controlling the flow of information using input, output and forget gate. The input gate controls the flow of input activation into the memory cell. The output gate controls the output flow of cell activation into the rest of the network.

Suppose that training data has N equipment of the same make and type that provide failure data, and each equipment provides set multivariate time-series data from the sensors of the equipment. Also, assume that there are r sensors of the same type on each equipment. Then data collected from each equipment can be represented in a matrix form $X_n = [x_1, x_2, ..., x_t, ..., x_{T_n}] \in \mathbb{R}^{r \times T_n}$ $(n =$

$1, ..., N$) where T_n is time of the failure and at time t the r-dimensional vector of sensor measurements is $x_t = [s_t^1, ..., s_t^r] \in \mathbb{R}^{r\times 1}, t = 1, 2, ..., T_n$. The data of each equipment in X_n is fed to LSTM network and the network learns how to model the whole sequence with respect to target RUL. At time t, LSTM network takes r-dimensional sensor data x_t and gives predicted RUL_t.

Let the LSTM cell has q nodes, then $c_t \in \mathbb{R}^{q\times 1}$ is output of cell state, $h_t \in \mathbb{R}^{q\times 1}$ is output of LSTM cell, $o_t \in \mathbb{R}^{q\times 1}$ is output gate, $i_t \in \mathbb{R}^{q\times 1}$ is input gate, and $f_t \in \mathbb{R}^{q\times 1}$ is forget gate at time t. At time $t-1$, the output h_{t-1}, and hidden state c_{t-1} will serve as input to LSTM cell at time t. The input x_t is fed as input to the cell. In LSTM, the normalized data are calculated using the following equations:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{3.1}$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{3.2}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{3.3}$$

$$\widetilde{c}_t = act(W_c \cdot [h_{t-1}, x_t] + b_c), \tag{3.4}$$

$$c_t = f_t * c_{t-1} + i_t * \widetilde{c}_t, \tag{3.5}$$

$$h_t = o_t * act(c_t), \tag{3.6}$$

Where σ is the sigmoid layer. c_t and $\widetilde{c}_t$ are each internal memory cell and temporary value to make a new internal memory cell at time t. $*$ is element-wise multiplication of two vectors.

3.2.2 Gated recurrent unit (GRU)

The GRU was proposed by *Cho et al.* [32]. It operates using a reset gate and update gate. GRUs are improved version of standard recurrent neural network. Similar to the LSTM unit, the GRU has gating units that modulate the flow of information, however, without having a separate memory cell. GRU's performance on certain tasks of polyphonic music modeling and speech signal modeling was

found to be similar to that of LSTM. GRUs have been shown to exhibit even better performance on certain smaller datasets [135]. The memory block of GRU is simpler than that of LSTM. The forget, input and output gates are replaced with an update and a reset gate. Also, GRU combines the hidden state and the internal memory cell. In GRU, the normalized data are calculated using the following equations:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z), \tag{3.7}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r), \tag{3.8}$$

$$\widetilde{h}_t = act(W \cdot [r_t * h_{t-1}, x_t] + b_h), \tag{3.9}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h}_t, \tag{3.10}$$

where z_t and r_t are the update gate and reset gate at time t, respectively. $\widetilde{h}_t$ is a temporary value to make new hidden state at time t.

3.2.3 Convolutional neural network (CNN)

CNN a deep learning algorithm has achieved exceptional success in various research fields [22] because it has many advantages over traditional machine learning approaches such as MLP [112]. CNNs are fundamentally inspired from feed-forward ANNs. Like any other advanced DL algorithm, CNN also find their applications in different areas including CNN-based PdM system [27, 60, 108]. A CNN consists of one or more convolutional layers and then followed by one or more fully connected layers as in a standard multi-layer neural network. A 1D CNN model is utilized in this paper to predict the RUL of the engine. Details about CNN construction and network design are presented in detail in [61].

3.2.4 C-MAPSS dataset

To evaluate the performance of the CNN, LSTM, and GRU DL algorithms, we use a well-known dataset, NASA's turbofan engine degradation simulation dataset C-MAPSS (Commercial Modular Aero-Propulsion System Simulation). This dataset includes 21 sensor data with different number of operating conditions and fault conditions. Table 3.1 gives details about the sensors in the engine[3]. In this dataset, there are four sub-datasets (FD001-04). Every subset has training data and test data. The test data has run to failure data from several engines of the same type. Each row in test data is a time cycle which can be defined as an hour of operation. A time cycle has 26 columns where the 1st column represents engine ID, and the 2nd column represents the current operational cycle number. The columns from 3 to 5 represent the three operational settings and columns from 6-26 represent the 21 sensor values. The time series data terminates only when a fault is encountered. For example, an engine with ID 1 has 192 time cycles of data, which means the engine has developed a fault at the 192nd time cycle. The test data contains data only for some time cycles as our goal is to estimate the remaining operational time cycles before a fault.

3.3 Modeling of FDIA

In this section, we describe in detail about an FDIA, ways of modeling it and also provide an attack scenario.

3.3.1 False data injection attack (FDIA)

As mentioned earlier, false data injection attack (FDIA) [79] can be injected into the system by compromising physical sensors, sensor data communication

[3]More details about these 21 sensors can be found in [93]

Table 3.1: Description of sensor signals for aircraft gas turbine engine

Index	Symbol	Description	Units
1	T2	Total temperature at fan inlet	R
2	T24	Total temperature at LPC outlet	R
3	T30	Total temperature at HPC outlet	R
4	T50	Total temperature at LPC outlet	R
5	P2	Pressure at fan inlet	psai
6	P15	Total pressure at LPC bypass-duct	psai
7	P30	Total pressure at HPC outlet	psai
8	Nf	Physical fan speed	rpm
9	Nc	Physical core speed	rpm
10	Epr	Engine pressure ratio	-
11	Ps30	Static pressure at HPC outlet	psia
12	Phi	Ratio of fuel flow to Ps30	pps/psi
13	NRf	Corrected fan speed	rpm
14	NRc	Corrected core speed	rpm
15	BPR	Bypass ratio	-
16	farB	Burner fuel-air ratio	-
17	htBleed	Bleed enthalpy	-
18	Nf_dmd	Demanded fan speed	rpm
19	PCNfR_dmd	Demanded corrected fan speed	rpm
20	W31	HPT coolant bleed	lbm/s
21	W32	LPT coolant bleed	lbm/s

links, and data processing programs. Compromising physical sensors requires physical access to the sensors and hence is a tedious task. In contrast, hacking the sensor data communication links and data processing programs is an easier option for an attacker (explained in detail in the *attack surface* section). A successful FDIA can cause the engine sensors to output erroneous values to the central engine control, and thus make an either physical or economic impact on the predictive maintenance model. For example, X_i represents the information transmitted by the i^{th} sensor. In an FDIA, the adversary contaminates the original vector with a vicious vector. Let $X_i = [x_1, x_2, ..., x_k]$ be the original vector data containing k sensor reading for the i^{th} sensor. The original vector could be contaminated by adding an FDIA vector with the same dimension as the original vector. Let the contaminated vector for the i^{th} sensor be $F_i = [\lambda_1, \lambda_2, ..., \lambda_k]$, then the compromised vector is given by Eq. 3.11.

$$Z_i = X_i + F_i \tag{3.11}$$

An FDIA can be *constrained*, where the attacker has access to a limited num-

ber of sensors, and some part of the communication network, and an FDIA can also be *unconstrained*, where the attacker has access to all of the sensors and also has total control of the communication network. In this work, we consider the constrained attack since it is more practical that an attacker has access to only a limited number of sensors (for the case study, the attack scenario considers only 3 sensors from a total of 21 sensors). We model two variations of FDIAs to explore and compare their impact, specifically, *continuous FDIA* and *interim FDIA*. In the case of continuous FDIA, the attack is continuous, which means, once the attack starts, from that point on-wards all the sensor reading are compromised. For instance, if the attack starts at the time instant $atck\_start = 3$ and ends at $atck\_end$ then F_i can be expressed as $F_i = [\lambda_1, \lambda_2, \lambda_{atck\_start}, ..., \lambda_{atck\_end}]$, where $atck\_start \geq 1$ and $atck\_end = k$. In the case of interim FDIA, the duration of attack is a short time interval, where $atck\_start > 1$ and $atck\_end < k$.

3.3.2 Attacker's stealthiness

An FDIA can be stealthy if it is not detected by the defense mechanism. In order to achieve that objective, the attack vector should remain in the boundary conditions of the sensor measurements. There exist constant vectors Z_{min} and Z_{max}, such that for any FDIA vector Z_i, the compromised vector passes undetected through the defense if

$$Z_i = X_i + F_i \; and \; Z_{min} \leq Z_i \leq Z_{max} \tag{3.12}$$

We assume the attacker knows Z_{min} and Z_{max} to construct attack vectors satisfying Eq.3.12. Such information is easily available from the sensor data sheets provided by the vendor.

3.3.3 Attacker's objective

The attacker's objective is to cause a delay in aircraft engine maintenance. This objective can be achieved by altering the IoT sensors readings that are fed to the PdM systems. Injecting false data to the sensor readings result in incorrect predictions from PdM systems which in turn results in a delay of timely maintenance. As timely maintenance is a crucial factor of engine performance, a lapse of maintenance may result in mid-air engine failures which are catastrophic.

One can argue that the attacker having access to the physical sensors or the communication network of the sensors would directly attack the main systems (flight navigation and instrument landing systems) rather than just altering the sensor values for the PdM. However, there is a higher chance that a direct attack on the main system will easily get detected by the defense mechanisms. In contrast, introducing FDIA to sensors is an easier and safer option for an attacker since such attacks are more stealthy, hard to get detected as they are in sensor's acceptable range and also the impact on the aircraft does not show up immediately. Instead, it causes the erroneous calculation of RUL and delays the maintenance cycle leading to a catastrophic incident.

3.3.4 Attack surface

In this thesis, only the *constrained attacks* are considered. Note, one of the ways to launch an FDIA is using spoofing techniques. For instance, Tippenhauer *et al.* [117] showed a spoof attack scenario on GPS-enabled devices. In this attack scenario, a forged GPS signal is transmitted to the device to alter the location. In this way, the true location of the device is disguised and the attacker can perform a physical attack on the device. In another work, Giannetos *et al.* [50] introduced an app named *Spy-sense*, which monitors behaviors of several sensors in a device. The app can manipulate sensor data by deleting or modifying it. *Spy-sense* exploits the active memory region in a device and relays sensitive data covertly. These

works show that FDI attacks can be performed even without gaining direct access to a system.

One of the recent articles [69] considers cyber-attacks as one of the reasons behind the two recent Boeing 737 Max 8 crashes. According to that article, a passenger, vehicle or drone carrying a sonic device capable of impacting the MCAS sensor controlling the plane could have been responsible for such an attack. Recently, ICS-CERT published an alert on certain controlled area network (CAN) bus systems aboard aircraft that might be vulnerable to hacking. It cited a report that an attacker with access to the aircraft could attach a device to avionics CAN bus to inject false data, resulting in incorrect readings in an avionic equipment [119]. Using such a device attached to the bus could lead to incorrect engine telemetry readings, incorrect compass and attitude data, and incorrect altitude, airspeed, and angle of attack (AoA) data. Pilots might not be able to distinguish between false and legitimate readings. This alert explores the possibility of injecting false data into IoT sensor readings of aircraft engine which are transmitted on a CAN. In this work we consider FDIA using malicious device attached to a avionics CAN.

3.3.5 Attack scenario

An aircraft engine is a complex system, so it requires adequate monitoring to ensure safe operation and in-time maintenance [59]. Several displays and dials in the cockpit give different measurements like exhaust gas temperatures, engine pressure ratio, the pressure at fan inlet, rotational speeds, etc. All these parameters are crucial in indicating the health of the engine; they serve as early indicators of failure and prevent costly component damage. In order to accomplish the task of monitoring these parameters in an engine, Engine health monitoring (EHM) systems [118] have been in service for three decades. Fig. 3.1 shows a generic EHM architecture. An EHM system has several IoT (Internet of Things) sensors mounted inside and outside of an engine to monitor different parameters.

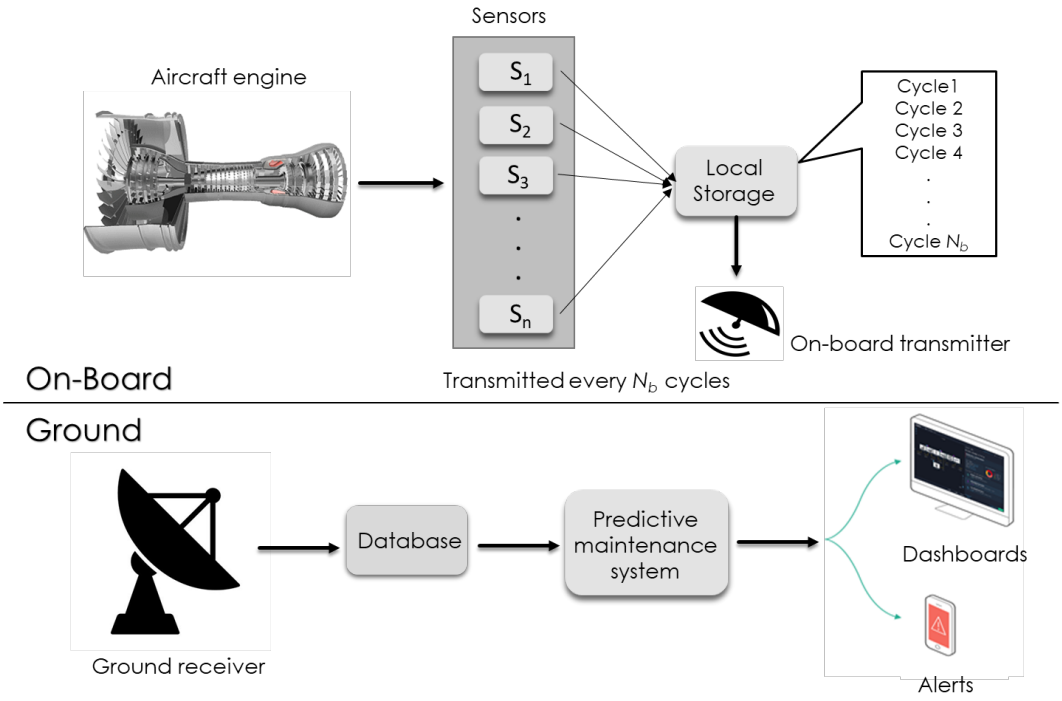

Figure 3.1: Engine health monitoring (EHM) system architecture

All these IoT sensors are connected to a wireless network [19], which uses radio frequency for transmitting sensor output to central engine control [19]. These IoT sensors monitor different parameters of an aircraft engine and sends out alerts to the engine manufacturer if the Remaining Useful Life (RUL) [105] of the engine is approaching its end of life. [4] An EHM system employs PdM systems to predict the RUL using the data collected from the IoT sensors.

The sensors on-board the engine send time series data (cycles) every hour to the local storage on-board the airplane. After every N_b cycles of data are captured, the data is transmitted to the ground station. At the ground station, the incoming live data is stored in the database and sent to PdM system to predict RUL of the engine. The PdM system sends out alerts if the predicted RUL is less than the permissible safe operation RUL of the engine.

As shown in Figure 3.1 of the EHM architecture, the aircraft sends N_b cycles of data at a time to the ground station/engine manufacturer. At the ground station, the PdM system performs data analytics on the received data and send

[4]Remaining useful life (RUL) is the length of time a machine is likely to operate before it requires repair or replacement.

out alerts if the RUL is close to the threshold N_{th}. The value of N_{th} can vary from engine to engine, and it is manufacturer-dependant. An adversary having this knowledge can perform the attacks more effectively. In a more practical sense, the degradation of the engine is very negligible at the beginning, but as time proceeds, the degradation follows a linear trend, and it increases as the engine approaches the end of life. Assuming in an engine, the linear degradation initially starts at N^d cycle. The value of N^d is different for different engines, as the wear of the engines may be different. If the average of N^d for all the engines in the dataset is taken, it is found to be N^d_{avg}. An adversary having the knowledge of N^d_{avg} can perform the attacks after the degradation initiates, making the attack more destructive.

To study the impact of FDIA on PdM systems, we consider an attack scenario where the attacker has access to the aircraft and could attach a device to avionics CAN bus [119] as mentioned previously in section 4 (attack surface). The device attached to CAN bus can inject false data into engine sensor readings, resulting in incorrect predictions of RUL of the aircraft engine. Note, as mentioned in section (attack surface), it is also possible to launch an FDI attack without direct access to the aircraft by using the sensor spoofing technique [64], or using a drone carrying a special device capable of interfering and impacting the on-board aircraft sensor measurements [69]. In this work, we consider two variations of FDIA which are continuous and interim FDIA. In continuous FDIA, the attack is initiated after N^d and continues to the end of life of the engine. In Interim FDIA, the attack is initiated after N^d and continues to the next 20 time cycles. In both the variations of FDIA, random and biased FDIAs are used to evaluate the PdM model's performance. Here, random FDIA means the noise added to the sensor output has a range (0.01% to 0.05%). Whereas, biased FDIA has a constant amount of noise added to the sensor output.

3.4 EXPERIMENTAL RESULTS

In this section, we first compare three different DL algorithms for RUL prediction. Next, we present both continuous and interim FDIA signatures, and the impact of FDIAs on the RUL prediction. Lastly, we present piece-wise RUL prediction and detail the impact of sequence length on resiliency.

3.4.1 Comparison of deep learning algorithms

In order to select the best machine learning algorithm for the PdM, we compare the performance of LSTM, GRU, and CNN algorithms for the C-MAPSS dataset. To evaluate the performance of these DL models, we utilize the root mean square error (RMSE) metric which is widely used as an evaluation metric in model evaluation studies. Table 5.1 represents the comparison of these DL algorithms with architectures LSTM(100,100,100,100) lh(80), GRU(100,100,100) lh(80), and CNN(64,64,64,64) lh(100). The notation GRU(100,100,100) lh(80) refers to a network that has 100 nodes in the hidden layers of the first GRU layer, 100 nodes in the hidden layers of the second GRU layer, 100 nodes in the hidden layers of the third GRU layer, and a sequence length of 80. In the end, there is a 1-dimensional output layer.

Table 3.2: RMSE comparison for different DL algorithms

Predictor architecture	**RMSE**
	Test
CNN(64,64,64.64) lh(100)	9.94
LSTM(100,100,100,100) lh(80)	8.76
GRU(100,100,100) lh(80)	7.26

From Table 5.1 it is evident that the DL algorithm GRU(100, 100, 100) with a sequence length 80 has the least RMSE of 7.26. It means that GRU is very accurate in predicting accurate RUL for this dataset. Note, the obtained results in Table 5.1 show that our GRU-based predictive model performs 1.9, 1.7 and 1.3 times better (in terms of accuracy) when compared to the recent works in [42, 122, 136],

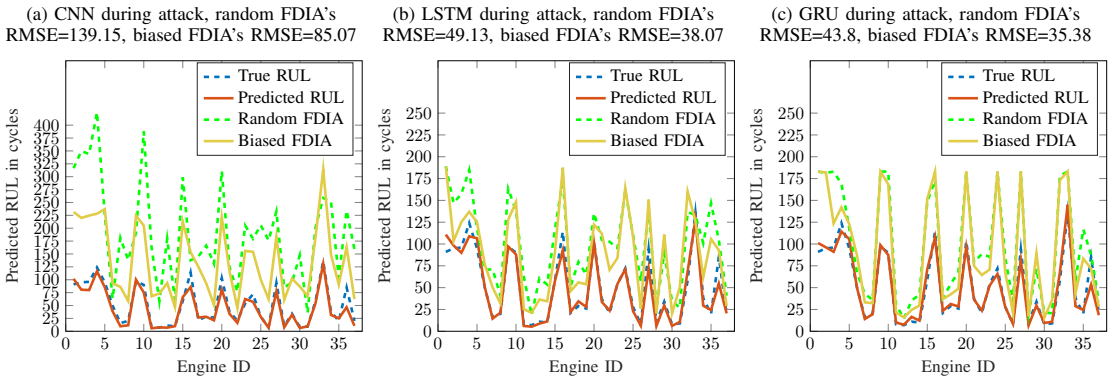

Figure 3.2: FDI attack scenario for continuous period

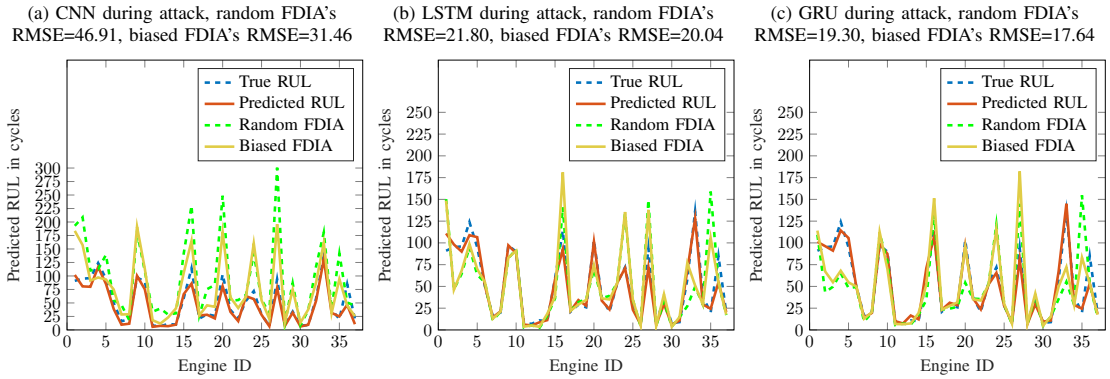

Figure 3.3: FDI attack scenario for interim period

respectively, on RUL estimation using DL algorithms and the C-MAPSS dataset.

3.4.2 Impact of attacks on a PdM system

The average degradation point of the engine N_{avg}^{d} is considered as 130 for the FD001 dataset [56] [16] [141], and we assume that the EHM system of the aircraft sends 20-time cycles (N_b) of data to the ground at a time. The details of EHM, the parameter N_b and how the data is sent to the ground are discussed in [81]. The FDIA can be performed on 21 sensors, but to make the attack more realistic, we perform FDIA on only 3 sensors (specifically, T24, T50, and P30). More information about these sensors can be found in [45]. In FDIA continuous scenario, the attacker has initiated the attacks after N_{avg}^{d}, which is 130-time cycles (a one-time cycle is equivalent of one flight hour), and the attack duration is until end of life of the engine. In FDIA interim scenario, the attacker has initiated the attacks after N_{avg}^{d}, which is 130-time cycles, and the attack duration is 20

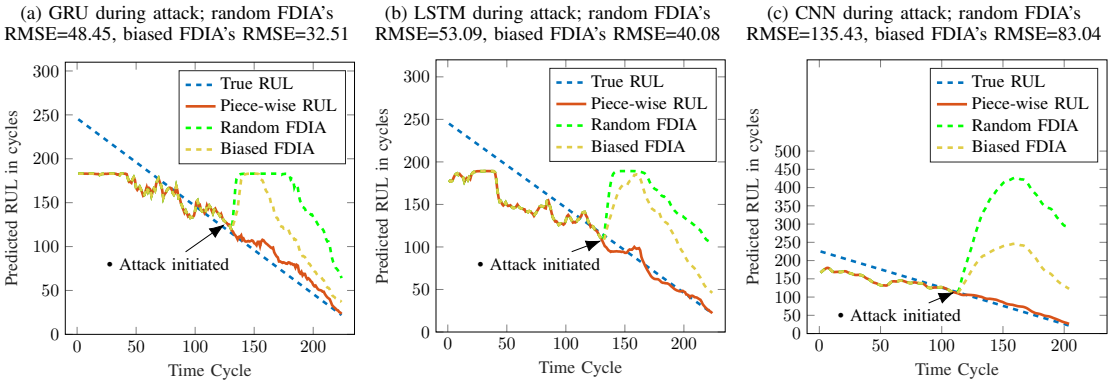

Figure 3.4: Piece-wise RUL prediction for continuous FDIA

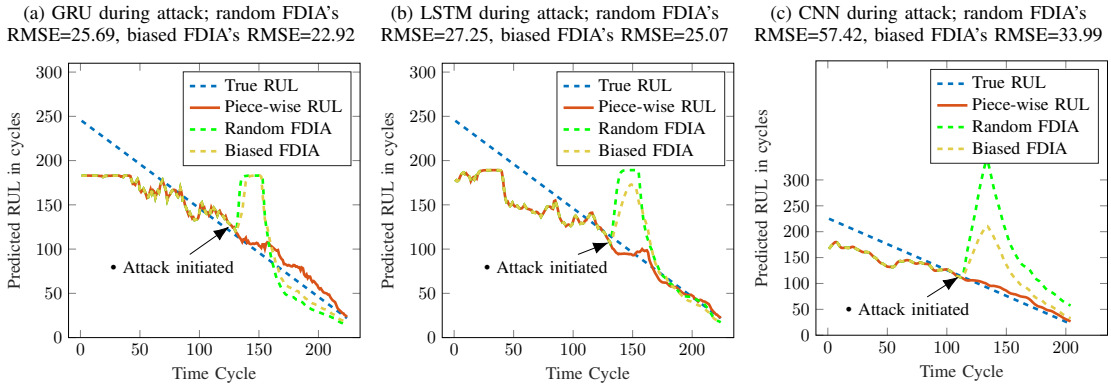

Figure 3.5: Piece-wise RUL prediction for Interim FDIA

hours (20-time cycles). Since the attack is initiated after 130-time cycles, we only consider the engines which have data for more than 130 cycles which gives us 37 engines in the FD001 dataset. The resultant dataset is re-evaluated using the LSTM, CNN and GRU-based PdM models and the obtained RMSEs are 6.09, 7.50, and 5.36, respectively.

FDIA signature: To model the FDIA on sensors, we add a vicious vector to the original vector, which modifies the sensor output by a very small margin (0.01% to 0.05%) for random FDIA and 0.02% for biased FDIA. Here, random FDIA means the noise added to the sensor output has a range (0.01% to 0.05%). Whereas, biased FDIA has a constant amount of noise added to the sensor output. The attack signatures can be found in our extended paper [81].

Impact of FDIA on CNN, LSTM and GRU: To show the impact of an FDIA on the aircraft PdM system, we implement an attack for the scenario mentioned previously in Section II (attack scenario). The FDIA is performed on three sensors

(T24, T50, and P30) instead of attacking all the 21 sensors in the dataset. In FDIA continuous scenario, the adversary performs attacks from 130-time cycles to end of life of the engine. It is evident from Fig. 3.2 that LSTM, GRU, and CNN are greatly affected by the continuous FDI attack. In the case of random and biased FDIA, random FDIA showed a considerable impact on all PdM models. The CNN based PdM model is the most affected by the continuous FDIA as random FDIA's RMSE is 139.15 and biased FDIA's RMSE is 85.07 (true RMSE is 7.50) which is almost 18 times and 11 times higher when compared to the true RMSE, respectively. In contrast, the GRU based PdM model is the least affected by the continuous FDIA as random FDIA's RMSE is 43.8 and biased FDIA's RMSE is 35.38 (true RMSE is 5.36). Even though the GRU is least affected by both random and biased FDIA, their RMSE is 8 and 6 times higher than the true RMSE, respectively, making it also deadly for a PdM system.

In the FDIA interim scenario, the adversary performs attacks between 130 and 150-time cycles (20-time cycles). It is evident from Fig. 3.3 that LSTM, GRU, and CNN are greatly affected by the interim FDI attack. Once again, the CNN based PdM model is greatly affected by the continuous FDIA as random FDIA's RMSE is 46.91 and biased FDIA's RMSE is 31.46 (true RMSE is 7.50) which is almost 6 times and 4 times higher than the true RMSE, respectively. In contrast, the GRU based PdM model is the least affected by the interim FDIA as random FDIA's RMSE is 19.30 and biased FDIA's RMSE is 17.64 (true RMSE is 5.36). This indicates that GRU-based PdM models are comparatively resilient to both continuous and interim FDIA. Even though the GRU is least affected by both random and biased FDIA, their RMSE is still 4 times and 3 times higher than the true RMSE, respectively, making it deadly for a PdM system. When comparing both continuous and interim FDIA, it observed that continuous FDIA's RMSE is almost twice the interim FDIA's RMSE. Hence, continuous FDIAs are more potent than interim FDIA.

3.4.3 Piece-wise RUL prediction

In order to show the impact of FDIA attacks on a specific engine data, we apply the piece-wise RUL prediction. The piece-wise RUL prediction gives a better visual representation of degradation in an aircraft engine. Fig. 3.4(a) shows an example of an engine data from the dataset of 100 engines, and depicts the predicted RUL using GRU at each time step of that engine data. For example, if X is the time series data of a particular engine, then $X_i = [x_1, x_2, x_3...x_{t-k}]$ represents time series data until time $t-k$. RUL^p is predicted RUL at each time step in X, which is can be defined as $RUL_i^p = [RUL_1^p, RUL_2^p, RUL_3^p...RUL_{t-k}^p]$. From Fig. 3.4(a), it is evident that as the time series approaches the end of life, the predicted RUL (red line) is close to the true RUL (blue dashes), because the DL model has more time series data to accurately predict the RUL.

In the case of piece-wise RUL prediction during continuous FDIA, it is observed from Fig. 3.4 that both random and biased FDIAs are initiated from 130-time cycles to 242-time cycles for engine ID 17. Here, the green and yellow dashes in the figures are predicted RUL after random and biased FDIA, respectively. In the GRU, LSTM, and CNN based piece-wise RUL prediction (for both random and biased FDIA), the attacker initiates the FDIA after 130-time cycles. The impact of the attack is quite interesting as the RUL jumps upwards (around 200 for GRU and LSTM) with a possible indication to the engine maintenance operator that the engine is quite healthy. This may influence a 'no maintenance required' decision from the maintenance engineers' point of view, however, in reality, the RUL is decreasing continuously and going below the 100-time cycles which might require to schedule urgent maintenance leading to a catastrophic event. For CNN, the continuous FDIA causes a longer jump (even beyond the initial RUL value) when compared to the FDIA in LSTM and GRU. Of course, there is a higher chance that this will be flagged as a potential fault either in the engine or in the PdM system, and will cause unnecessary engine maintenance and will increase the aircraft downtown causing a financial loss to the flight operator.

In the case of piece-wise RUL prediction for engine ID 17 under interim FDIA, it is observed in Fig. 3.5 that the attack causes a similar jump as shown in the case of continuous FDIA in Fig. 3.4. However, the effect of the attack flushes away way sooner when compared to the continuous FDIA case. However, note that the attack duration was only 20 cycles, but it took more than 45 cycles to flush out the effect by the PdM system. Hence, if maintenance is due around that period, it may lead to catastrophic consequences. Once again, the piece-wise RUL prediction results indicate that employing CNN in PdM systems may result in systems that are very sensitive to the FDIA and hence special measures should be taken for designing a CNN-based PdM.

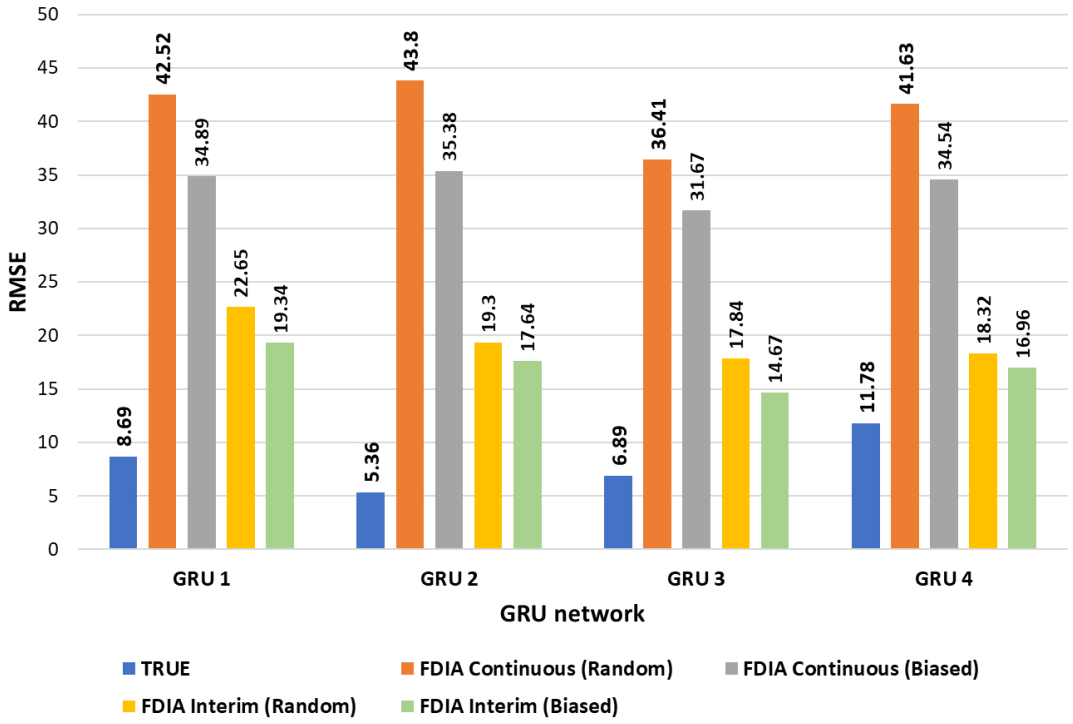

Figure 3.6: RMSE comparison of different GRU networks

3.4.4 Impact of sequence length on resiliency of GRU

Since GRU has performed best among the DL algorithms as shown in the experimental results in the previous subsections, in Figure 3.6 we compare four different GRU networks under FDI attack. The GRU networks have structures

GRU1(100,100,100) lh(90), GRU2(100,100,100) lh(80), GRU3(100,100,100) lh(70), and GRU4(100,100,100) lh(60). We observe that the GRU network with architecture GRU2(100,100,100) lh 80 has the least value of true RMSE (5.36), which means that it predicts RUL quite accurately, however, it is less resilient to both continuous and interim FDIA. In contrast, GRU with network architecture GRU3(100,100,100) lh(70) shows the second-best performance in predicting the RUL (RMSE of 6.89), however, in terms of resiliency, this network is the least affected by continuous and interim FDIA. This indeed shows an interesting insight that the sequence length affects not only the accuracy but also the resiliency of the model. It also indicates that accuracy should not be the only factor while designing a PdM system. For instance, in terms of accuracy GRU2 is the typical choice. However, if both accuracy and resiliency are considered, GRU3 is can be an ideal choice (at the cost of losing some accuracy).

3.5 Discussion

In this work, we first evaluate three different DL algorithms on the C-MAPSS dataset and obtained results show a great prospect for deep learning in PdM. Results show that the GRU performed 1.3-1.9 times better than the recent works that use deep learning on the C-MAPSS dataset [42, 122, 136]. The impact analysis of FDIA on aircraft sensors in the C-MAPSS dataset provides some interesting insights. We observe that CNN based PdM model is greatly affected by both random and biased FDIA. In the case of interim FDIA, CNN's random and biased RMSE are 18 and 11 times higher than the true RMSE, respectively, and in the case of continuous, the random and biased RMSE are 6 and 4 times higher than the true RMSE, respectively. We also observe that the GRU-based PdM model is more resilient to both random and biased in comparison with CNN and LSTM-based PdM models. Even though the GRU is least affected by both random and biased FDIA, their RMSE is 8 and 6 times higher than the true RMSE in the

case of continuous FDIA, respectively. In the case of interim FDIA, the random and biased RMSE are 4 and 3 times higher than the true RMSE, respectively, making it disastrous for the PdM system. This may result in the delay of timely maintenance for the aircraft engine and eventually result in engine failure at some point. Note, the attack signature of FDIA is very close to the original sensor output making it harder to be detected by common defense mechanisms in an EHM system.

A piece-wise RUL predicting approach is used in visualizing the impact of attacks on the sensors, which clearly shows that the PdM system is susceptible to sensor attacks. CNN based piece-wise RUL prediction results show that special measures should be taken when designing and adopting CNN-based PdM systems (such as the cases in [27, 55, 60, 108]) as they are very sensitive to the FDIA. Figure3.6, gives an interesting insight into the relationship between accuracy and resiliency of the GRU network. It shows the need for considering the relationship between the accuracy, resiliency and sequence length of a DL mode (such as GRU in our case) in the design phase. Indeed, such an analysis can serve as empirical guidance to the development of subsequent data-driven PdM systems.

All of these obtained results show that DL-based PdM systems have a great prospect for aircraft maintenance, however, they are very susceptible to sensor attacks. Hence it is required to investigate proper detection techniques to detect such stealthy attacks and special care should be taken when manufacturing IoT sensors for DL/AI applications. For the same reason, while designing a PdM system, the designer also must consider the resiliency of the DL algorithm instead of just emphasizing on the algorithm's accuracy.

In this chapter, we have seen that a randomly generated noise can have a great impact on the PHM system. From this, we were inspired to study the impact of carefully crafted adversarial noise on the PHM system. Especially, the deep learning algorithms in PHM systems are prone to these adversarial threats. Adversarial attacks in deep learning have been extensively explored for image

recognition and classification applications. However, their application to the non-image domain is vastly under-explored and hence, pose a significant threat to the PHM system. Hence, it is very important to build PHM systems that are reliable and also robust to adversarial threats. In the next chapter, we formulate a methodology to build robust PHM systems that are resilient to cyber/adversarial threats. The methodology consists of building a PHM system, crafting adversarial examples to the PHM system and improving the adversarial robustness of the PHM system using adversarial training.

Chapter 4

Methodology for building robust PHM systems

In the previous chapter, we have seen that a randomly generated noise can have a great impact on the PHM system. From this, we were inspired to study the impact of carefully crafted adversarial noise on the PHM system. Adversarial attacks in deep learning have been extensively explored for image recognition and classification applications. However, their application to the non-image domain is vastly under-explored and hence, pose a significant threat to the PHM system. Hence, it is very important to build PHM systems that are reliable and also robust to adversarial threats. so this thesis provides a methodology to design and build adversarially robust PHM systems by employing adversarial training as a defense strategy.

In this chapter, a systematic approach to designing and implementing PHM for industrial applications is provided. Also, a comprehensive approach on generating adversarial examples to a PHM system along with approaches on making a PHM model robust to adversarial threats is provided, as described in Figure 4.1. A researcher/engineer can use this methodology in pre-deployment stage to build a robust PHM model. The methodology is separated into three steps, i.e. (1) Building a PHM system, (2) Crafting an adversarial attack, and lastly, (3) Adversarial

robustness. In step 1, we build a PHM system that employs DL for predictions. In step 2, we attack the PHM system by crafting adversarial examples and feeding it to the DL model in the PHM system, which may result in incorrect predictions. In step 3, we choose one of the adversarial defense strategies, to make a PHM model robust to adversarial threats. In this work, we consider adversarial training to make a robust PHM model.

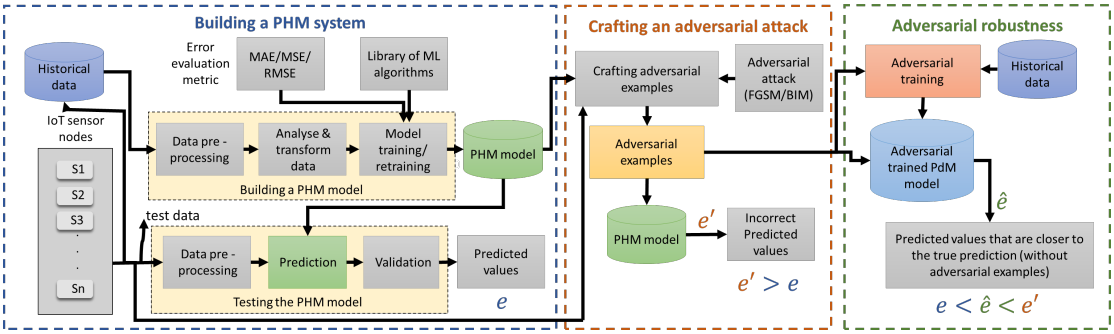

Figure 4.1: The proposed methodology

4.1 Building a PHM system

A PHM system is a complex system of interconnected components, which involves IoT devices and state-of-the-art DL algorithms to make informed decisions/predictions. The designing of the PHM model involves several steps as shown in figure 4.1, and each step plays a prominent role in designing an efficient PHM model.

Historical data, it is a repository of time-series IoT data, which is used for building a PHM model. Real-time data from the IoT sensors are sequentially stored in historical data and then sent for *building PHM model* block. IoT devices in general generate large amount of data and processing it can be a cumbersome task. There exist some work using dimensionality reduction [143], [139], [138], [137] and cloud computing [142], [123], [132] for processing large data.

Bulding PHM model, this block comprises of three main blocks which are key in building a PHM model. Firstly, *Data pre-processing*, in order to build

an efficient PHM model, the quality of data provided is crucial. The raw sensor readings can be noisy, incomplete, and inconsistent. To make it ready for the model building it has to go through the following steps

- **Data cleaning**: Detect and fill missing values in the data and also to detect and remove noisy data points and outliers.
- **Data transformation**: Normalize IoT data to reduce data dimensions to make it easier to process and also to reduce the effect of noise.
- **Data reduction**: Sample data records for easier data handling and also to reduce the computational power needed to handle data.
- **Data discretization**: Convert continuous attributes to categorical attributes for ease of use. It is also known as binning.

Secondly, *Analyze and Transform data*, this block deals with selecting important features in the prepared data. Feature selection is the process of identifying the most crucial or influential features in the data. Some features may be less important and some may be pivotal for predicting equipment failure. It is key to understand the role played by every feature, especially when working with huge amount of data. Reducing the features will result in reducing the computational power needed to train and run the model, which in turn saves a lot of precious time.

Lastly, *Model training and retraining*, in order to build a quality model, the importance of choosing the appropriate ML algorithm is crucial. The algorithm is selected based on data and desired model outcomes. After selecting the appropriate algorithm, the next phase would be to train the model. The dataset from *Analyze and Transform data* block is split into two samples. The training sample comprises the major portion of the dataset and the rest is the test sample. The model is trained using the training dataset and in order to evaluate the performance of the trained model, few methods such as precision, recall, and F1 are employed, and also error evaluation metrics such as Mean Absolute error (MAE),

Root Mean Square Error (RMSE) and Mean Square Error (MSE) can also be used to evaluate the performance. The model retraining is scheduled at regular intervals to accommodate the changing conditions and also to maintain a consistent performance of the model.

In **Testing the PHM model** block, the test data is fed to *data pre-processing* block to handle noisy, incomplete and inconsistent data. In the next phase, the prepared data is sent to *prediction* block, where the trained PHM model makes predictions on the data provided. The predictions have an RMSE (e), which is helpful in evaluating the performance of the PHM system. In *validation* block the accuracy of the prediction made by the model is evaluated. In this block, the efficacy of the PHM model is evaluated and changes to the training data set are made based on the predicted outcomes. This step is necessary to maintain the quality of the model.

4.2 Crafting an adversarial attack

Adversarial examples are inputs to machine learning models that an attacker has intentionally designed to cause the model to make incorrect predictions. Adversarial attacks have a great impact on computer vision tasks. Adversarial attacks are less explored in the PHM domain. In this work, we craft adversarial examples using FGSM and BIM attacks. Adversarial attack block in Figure 4.1 shows the steps involved in the implementation of an adversarial attack on a PHM system.

Crafting adversarial examples block needs three inputs to craft adversarial examples. First, a trained DL model, second, sensor data from IoT sensors, and lastly, adversarial attack. The sensor data and trained DL model are inputs from the previous step (Building a PHM system). A researcher/engineer has the ability to choose between stat-of-the-art adversarial attacks like C & W's Attack [29], DeepFool [84, 128], One-pixel attack [110], Fast-Gradient Sign Method (FGSM) [52], Projected Gradient Descent (PGD) [77], and Basic Iterative Method (BIM)

[67]. In this work, we craft adversarial attacks using FGSM and BIM attacks. The adversarial examples that are crafted by these attacks are closer to the input signal, which makes it a tedious task to detect and mitigate the attack. The adversarial examples that are crafted from this step are given as input to the PHM model, which results in incorrect predictions and also results in predictions with higher RMSE (e'). The RMSE from these predictions (e') is greater than the RMSE of the prediction from the previous step (Building a PHM system) i.e. $e' > e$. These incorrect predictions in safety-critical PHM systems may result in failure of equipment and may incur a huge loss. For example, in the case of aircraft engine maintenance, incorrect RUL predictions may result in a lapse of timely maintenance and result in mid-air engine failures.

4.3 Adversarial robustness

As we seek to deploy deep learning algorithms in PHM systems, it becomes critical that the system is truly robust and reliable. Although many notions of robustness and reliability exist, the adversarial robustness raised a great deal of interest in recent years. Adversarial robustness is a model's ability to mitigate the impact of adversarial threats from an adversary, crafted with the intention of fooling the model. The PHM system is vulnerable to adversarial threats and hence it is important to make a PHM system robust to adversarial threats. There are many strategies proposed [91] to make a model robust to adversarial threats and they can be divided into three categories: *modifying data*, *modifying models*, and *using auxiliary tools*. *Modifying data* refers to the reconstruction of the training data. Adversarial training is one of the techniques of modifying data, which adds adversarial examples to the training data to make a robust DL model. Other data modifying techniques include data compression [34], gradient hiding [88], and data randomization [124]. *Modifying models* refers to the modification of ML/DL models to make it robust against adversarial threats, it includes techniques like

feature squeezing [125], regularization [23], distillation [89], and mask defense [49]. Lastly, *auxiliary tools* include MagNet [80], defense-GAN [98], and high-level representation guided denoiser [74]. All these strategies are mainly proposed for the computer vision domain. In contrast, there are very few strategies proposed to detect adversarial examples in DL regression models, and inductive conformal anomaly detection method [20, 121] is one of the potential strategies. The IoT and DL based prognostics is currently revolutionizing the Industry 4.0 domain, however, their security vulnerabilities are often ignored. In this work, we evaluate the adversarial robustness of a model using the adversarial training method.

The *adversarial robustness* block in Figure 4.1 gives an overview of the steps involved in making a DL model adversarially robust. The process of adversarial training, adds crafted adversarial examples from the previous step (Adversarial attack) to the training dataset. A DL model is trained on this new dataset to learn on the adversarial examples and its data distribution, which helps in making a model robust to adversarial threats. When crafted adversarial examples are fed to the adversarially trained DL model, it results in predictions that are closer to the true prediction (without adversarial attack) and also have RMSE ($\hat{e}$) closer to the model without adversarial examples (e) and less than PHM model with adversarial examples (e') i.e. $e < \hat{e} < e'$. The process of adversarial training makes a PHM model robust and is helpful in mitigating the effect of adversarial attacks on a PHM system.

4.4 Summary

This chapter has introduced a systematic approach to designing and implementing PHM for industrial applications. The chapter also provided a comprehensive approach on crafting adversarial examples to a PHM system along with approaches on making a PHM model robust to adversarial threats. In the next chapter, we use the proposed methodology to build a PHM system that employs

DL for predictions, attack the PHM system by crafting adversarial examples and feeding it to the DL model in the PHM system, and finally, employ adversarial training to make a PHM model robust to adversarial threats.

Chapter 5

Crafting Adversarial Examples for Deep Learning Based Prognostics

The goal of this chapter is to discuss the adversarial-threats associated with a well-connected PHM system. In this Chapter, we use the methodology proposed in the previous chapter to design a PHM system, craft adversarial examples to a PHM system and employ adversarial training to build robust PHM systems. In this chapter, we analyze the impact of Fast Gradient Sign Method (FGSM) and Basic Iterative Method (BIM) attacks on Deep Learning enabled PHM systems. We perform a comprehensive study of the transferability property of adversarial examples in DL-based PHM models. We also enhance the adversarial robustness of the PHM system by performing adversarial training to mitigate the impact of adversarial threats.

5.1 Adversarial attacks on PHM systems

The advent of Industry 4.0 in automation and data exchange leads us toward a constant evolution in smart manufacturing environments, including an intensive utilization of Internet-of-Things (IoT) and Deep Learning (DL). Specifically, the state-of-the-art Prognostics and Health Management (PHM) [47] has shown great

success in achieving a competitive edge in Industry 4.0 by reducing the maintenance cost, downtime and increasing the productivity by making data-driven informed decisions. For instance, modern PHM techniques can help reduce downtime by 35%-45%, maintenance cost by 20%-25%, and can increase production by 20%-25% [7]. The ability to sense changes in the physical world (such as temperature, vibration, pressure, etc.) using IoT sensors, and to analyze the sensed data using the state-of-the-art DL algorithms for different prognostic tasks such as the Remaining Useful Life (RUL) prediction has enabled a highly reliable and cost-efficient industrial automation framework. Unfortunately, IoT sensors are also known for their vulnerability to cyber attacks [106, 144], and DL algorithms can also be easily fooled by adversarial examples [113]. From the perspective of computer vision, an adversarial example can be an image formed by making small perturbations (insignificant to the human eye) so that a classifier misclassifies it with high confidence. The highly-connected IoT sensors and DL utilized in PHM systems tend to inherit their respective vulnerabilities, thus making them a lucrative target for cyber-attackers [4]. According to a recent report from the *Malwarebytes*, cyber-threats against businesses/factories have increased by more than 200% over the past year [2]. Another interesting fact is that the adversarial examples can often transfer from one model to another model, which means that it is possible to attack models to which the attacker does not have access [113]. Such adversarial attacks have been extensively studied in the computer vision domain [14]. Even though advanced data-driven PHM depends on DL, it is very surprising that the impact of adversarial attacks on the PHM domain has not been studied yet.

In manufacturing, adversarial attacks can lead to a wrong prognostic decision, e.g., a wrong estimation of RUL can delay the maintenance of a machine leading to unexpected failures. Such unexpected failures are considered a primary operational risk, as they can hinder productivity and can incur a huge loss. For example, in the modern automotive industry, an assembly line has several robots working

on a car, and if even one robot fails, it will result in the halt of the entire assembly line, causing loss of valuable production time and increased production cost. In another situation, a wrong prognostic prediction in an operating autonomous vehicle, or aircraft may lead to loss of human lives. Thus, even though the utilization of IoT and ML is revolutionizing the smart industry, the vulnerabilities related to IoT and ML possesses a great challenge for Industry 4.0.

In chapter III, we showed that even a very small amount of randomly generated noise injected to the IoT sensors can greatly defect the RUL estimation. In this chapter, we go beyond the concept of randomly generated noise for PHM. We adopt adversarial example generation algorithms from the computer vision domain (for classification), formalize them (for regression), craft adversarial examples for the PHM domain. Our work provides a methodology to design and build adversarially robust PHM systems by employing one of the adversarial defense strategies i.e. adversarial training. We employ this methodology to first build two PHM systems i.e. turbofan engine and battery PHM systems using Long Short-Term Memory (LSTM) [58], Gated Recurrent Unit (GRU) [33], Convolutional Neural Network (CNN) [53], Multilayer Perceptron (MLP) [87], and Bi-directional Long Short-Term Memory (Bi-LSTM) [100] DL algorithms. Then, we craft adversarial attacks i.e. Fast Gradient Sign Method (FGSM) [52] and Basic Iterative Method (BIM) [67] using the proposed methodology for those PHM systems. We also analyze the impact of adversarial training using the proposed methodology to enhance the adversarial robustness of the PHM systems.

5.2 Deep learning for prognostics

In this section, we present the deep learning architectures that we employed in this work, to be specific LSTM, GRU, CNN, MLP and Bi-LSTM. We choose these models as they are quite popular in the PHM domains. We use MLP as a baseline for the other DL models.

5.2.1 Long Short-Term Memory (LSTM)

An LSTM [57] is a type of Recursive Neural Network (RNN), which is known for its applicability in time-series classification and regression [134]. The details about LSTM equations and architecture can be found in Chapter 3.

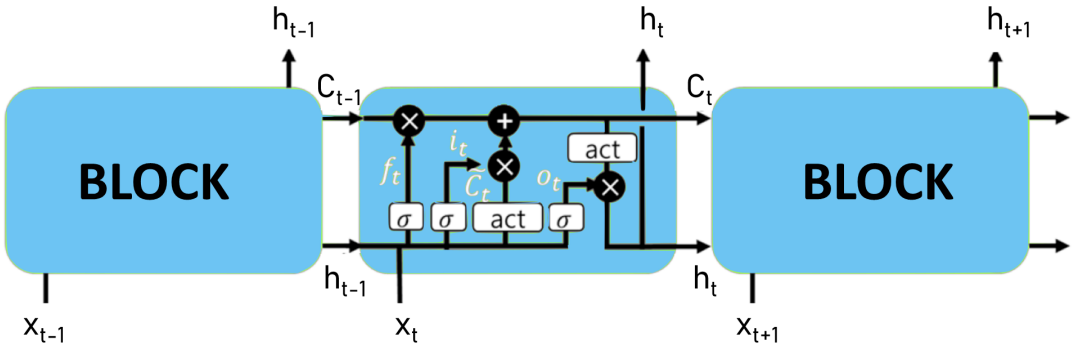

Figure 5.1: LSTM cell structure at time t

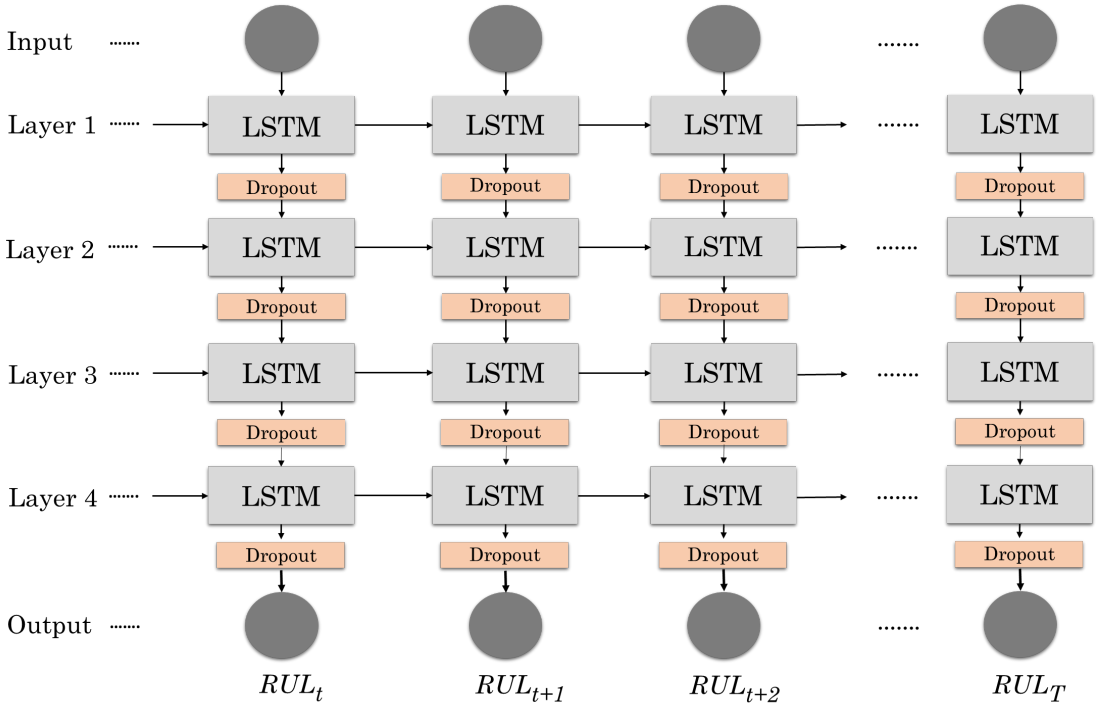

Figure 5.2: LSTM Architecture

For C-MAPSS dataset, we employ LSTM of the architecture LSTM(100,100,100,100) lh(80). This notation refers to a network that has 100 nodes in the hidden layers of the first, second, third, and fourth LSTM layers, and a sequence length of 80. In

the end, there is a 1-dimensional output layer. Figure 5.2 shows the architecture of LSTM employed in our work.

For battery dataset, we employ LSTM of the architecture LSTM(180,180,120,120) lh(80). This notation refers to a network that has 180 nodes in the hidden layers of the first and second, 120 nodes in third and fourth LSTM layers, and a sequence length of 60. In the end, there is a 1-dimensional output layer. Figure 5.2 shows the architecture of LSTM employed in our work.

5.2.2 Gated Recurrent Unit (GRU)

The GRU was proposed by *Cho et al.* [32]. It operates using a reset gate and update gates. GRUs are improved versions of standard recurrent neural networks. Similar to the LSTM unit, the GRU has gating units that modulate the flow of information, however, GRU has two gates (reset and update gates). The GRU does not have a memory unit. It just exposes the full hidden content without any control. GRUs have been shown to exhibit even better performance on certain smaller datasets [135]. The details about GRU equations and architecture can be found in Chapter 3.

For C-MAPSS dataset, we employ GRU of the architecture GRU(100,100,100) lh(80). This notation refers to a network that has 100 nodes in the hidden layers of the first, second, and third GRU layers, and a sequence length of 80. In the end, there is a 1-dimensional output layer. Figure 5.3 shows the architecture of GRU employed in our work.

For battery dataset, we employ GRU of the architecture GRU(180,180,120,120) lh(80). This notation refers to a network that has 180 nodes in the hidden layers of the first and second, 120 nodes in third and fourth GRU layers, and a sequence length of 60. In the end, there is a 1-dimensional output layer. Figure 5.4 shows the architecture of GRU employed in our work.

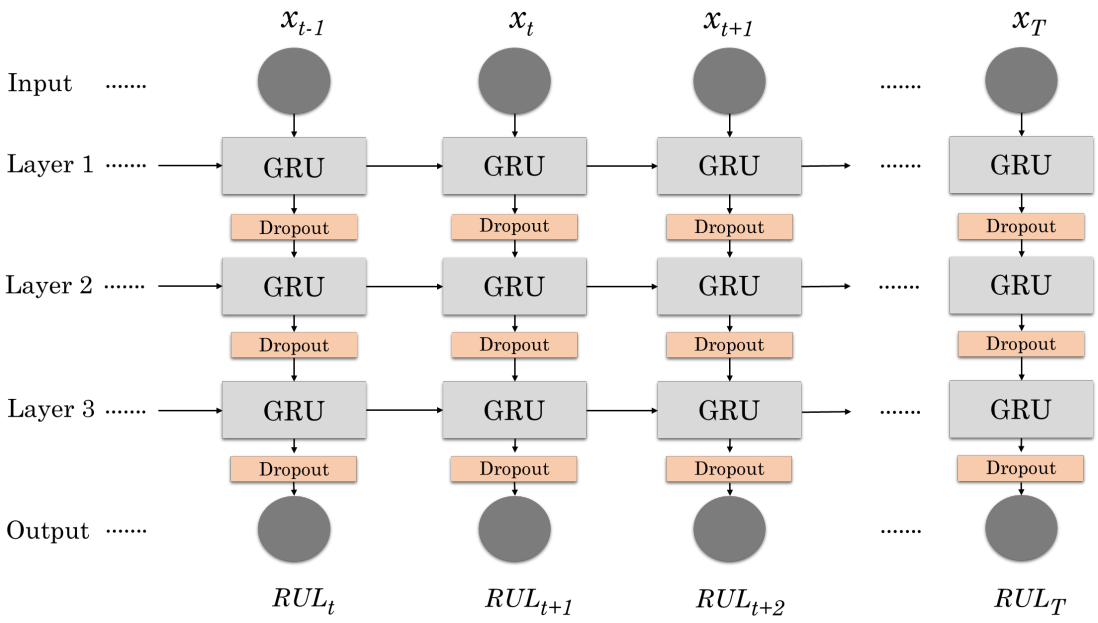

Figure 5.3: GRU Architecture

5.2.3 Bi-directional LSTM network (Bi-LSTM)

Bidirectional LSTMs are an extension of traditional LSTMs that can improve model performance on sequence regression problems. Bidirectional recurrent neural networks (BRNN) were first introduced by Schuster and Paliwal in 1997 [100]. The Bi-LSTM allows the network to have both backward and forward information about the sequence at every time step by running the inputs in two ways, one from past to future and one from future to past. During the training process, the Bi-LSTM is trained in both forward and backward directions. Figure 5.5 shows the general structure of Bi-LSTM. Bi-LSTMs are employed in several applications of PHM to predict the RUL of an equipment [72, 75].

For C-MAPSS dataset, we employ Bi-LSTM of the architecture Bi-LSTM(180,180,120) lh(80). This notation refers to a network that has 180 nodes in the hidden layers of the first, second, 120 hidden nodes in the third Bi-LSTM layer, and a sequence length of 60. In the end, there is a 1-dimensional output layer. Figure 5.6 shows the architecture of Bi-LSTM employed in our work.

For battery dataset, we employ Bi-LSTM of the architecture Bi-LSTM(180,120,120) lh(80). This notation refers to a network that has 180 nodes in the hidden layers

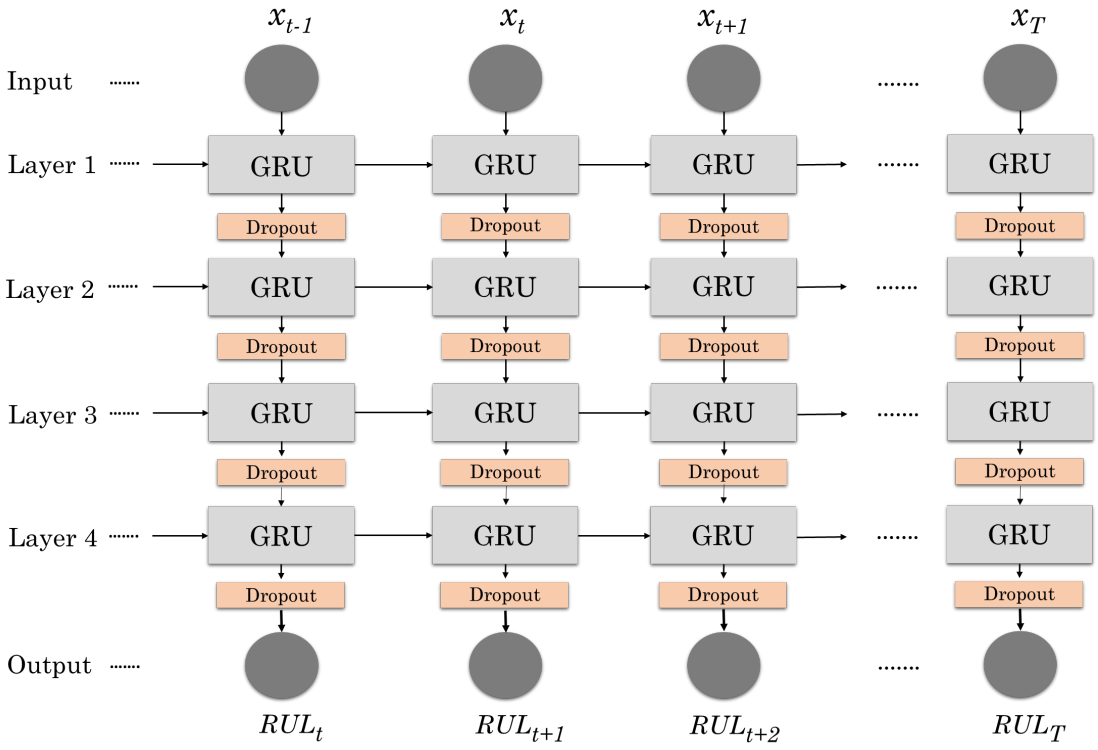

Figure 5.4: GRU Architecture

of the first Bi-LSTM layer and 120 hidden nodes in third and fourth Bi-LSTM layer, and a sequence length of 60. In the end, there is a 1-dimensional output layer. Figure 5.6 shows the architecture of Bi-LSTM employed in our work.

5.2.4 Multilayer Perceptron (MLP)

A multilayer perceptron (MLP) [87] is a class of feedforward artificial neural network (ANN). An MLP network consists of at least three layers: an input layer, a hidden layer and an output layer. Each node in a layer is a neuron that uses a nonlinear activation function, except for the nodes of the input layer. A MLP is one of the most common neural network models used in the field of deep learning. In this work, we use MLP (even though it is deemed insufficient for modern day advanced regression tasks) as a baseline for other DL models.

For C-MAPSS dataset, we employ MLP of the architecture MLP(128,128,64) lh(60). This notation refers to a network that has 128 nodes in the hidden layers of the first, second layers and 64 hidden nodes of the third MLP layer, and a sequence

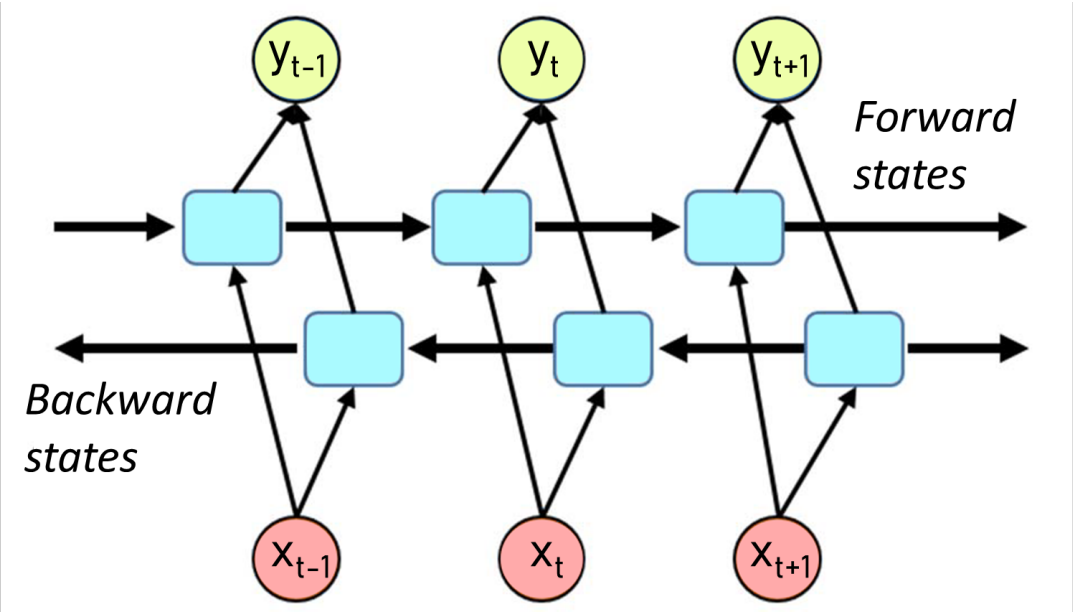

Figure 5.5: Bi-LSTM cell structure at time t

length of 60. In the end, there is a 1-dimensional output layer. Figure 5.7 shows the architecture of MLP employed in our work.

For battery dataset,, we employ MLP of the architecture MLP(256,256,256,128) lh(60). This notation refers to a network that has 256 nodes in the hidden layers of the first, second, third layers and 128 hidden nodes of the fourth MLP layer, and a sequence length of 60. In the end, there is a 1-dimensional output layer. Figure 5.8 shows the architecture of MLP employed in our work.

5.2.5 Convolutional Neural Network (CNN)

CNN is a type of deep learning algorithm, which is fundamentally inspired by feed-forward ANNs. CNN find their applications in different areas including prognostics [27, 108]. A CNN consists of one or more convolutional layers and then followed by one or more fully connected layers as in a standard multi-layer neural network. A 1D CNN model is utilized in this work to predict the RUL of the engine. Details about CNN construction and network design are presented in detail in [61].

CNN has achieved exceptional success in prognostics. Figure 5.9 shows the

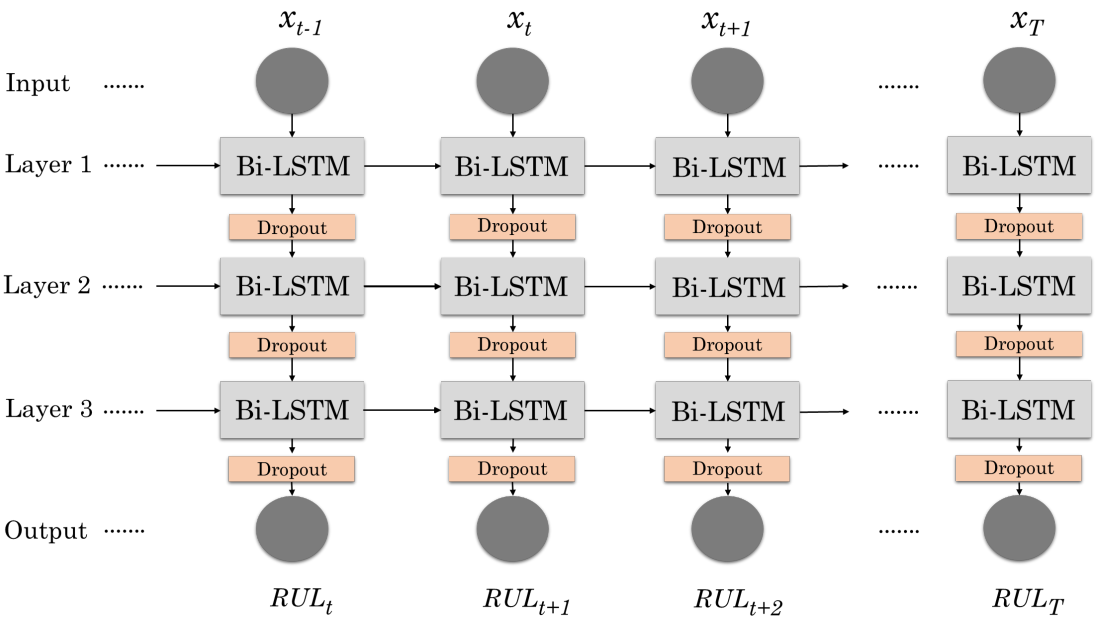

Figure 5.6: Bi-LSTM Architecture

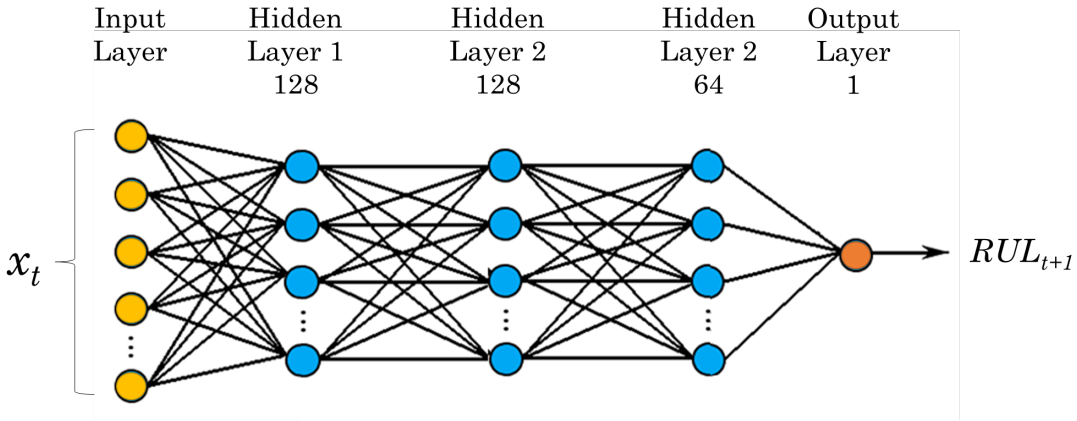

Figure 5.7: MLP Architecture

architecture of CNN employed for C-MAPSS dataset. The architecture employs an input layer, three 1-D convolutional layers with each having 64 filters and three dense layers of size 40, 40, and 1.

Figure 5.9 shows the architecture of CNN employed for battery dataset. The architecture employs an input layer of length 100, three 1-D convolutional layers with 256 filters in first, second layers and 128 filters in third and fourth layers. The architecture also has three dense layers of size 40, 40, and 1.

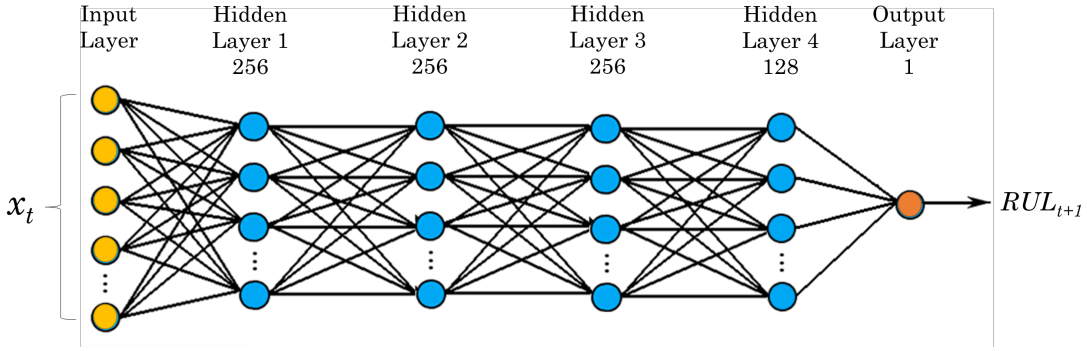

Figure 5.8: MLP Architecture

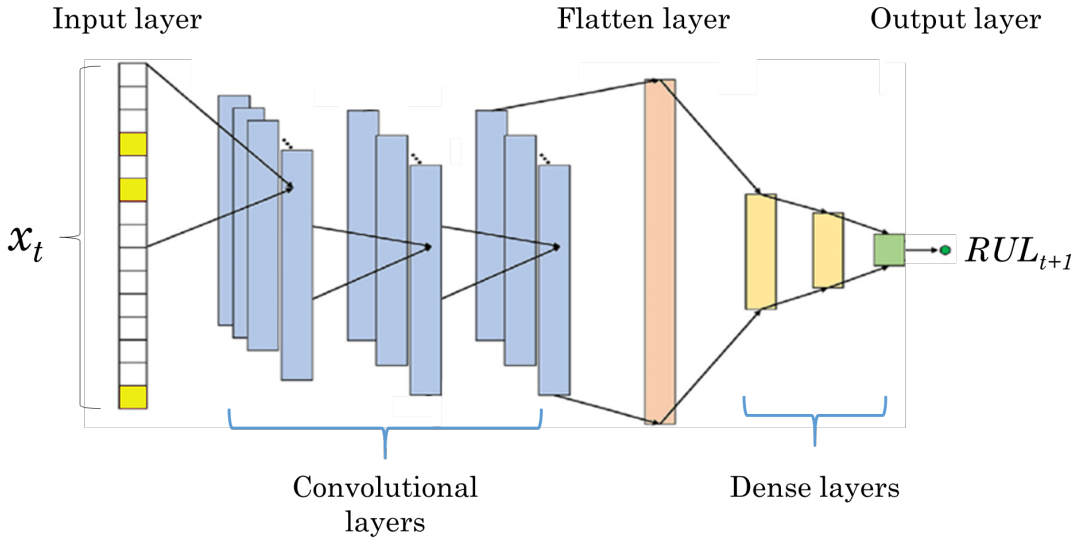

Figure 5.9: CNN Architecture

5.2.6 Prediction model

All the DL models employed predict the future state (RUL) of an equipment. If an equipment has N time cycles data at time instant t. We train our DL models to predict the RUL of $N + 1$ time cycles using the data of N time cycles. For example, if an equipment has 250 time cycles of data. Our DL model predicts RUL for 251 time cycles.

5.3 Adversarial Attacks on Prognostics

In this section, we formalize the adversarial attacks in the PHM domain and present the adversarial example generation algorithms.

5.3.1 Formalization of the problem

A machine or a piece of equipment in a PHM system has several sensors that record different parameters. These sensor measurements are recorded at every time step and hence constitute for multivariate time-series data [82].

Definition 1: Let M be a multivariate time-series (MTS). Assuming there are N sensors in an equipment, the multi-variate time-series can be defined as a sequence such that $M = [m_1, m_2, ..., m_T]$, $T =\mid M \mid$ is the length of M, and $m_i \in \mathbb{R}^N$ is a N dimension data point at time $i \in [1, T]$ representing the sensor measurements.

Definition 2: $D = (m_1, RUL_1), (m_2, RUL_2), ..., (m_T, RUL_T)$ is the dataset of pairs (m_i, RUL_i) where RUL_i is a label (RUL value at that time instant) corresponding to m_i.

Definition 3: Time series regression task consists of training the model on D in order to predict $\hat{RUL}$ from the possible inputs. Let $f(\cdot) : \mathbb{R}^{T \times N} \rightarrow \hat{RUL}$ represent a DL for regression. $J_f(\cdot, \cdot)$ denotes the cost function of the model f.

Definition 4: M' denotes the adversarial example, a perturbed version of M such that $\hat{RUL} \neq \hat{RUL}'$ and $\|M - M'\| \leq \epsilon$. where $\epsilon \geq 0 \in \mathbb{R}$ is a maximum perturbation magnitude.

Given a trained DL model f and an input MTS M, crafting an adversarial example M' can be described as a box-constrained optimization problem.

$$\min_{M'} \|M' - M\| \; s.t.$$
$$f(M') = \hat{RUL}', \; f(M) = \hat{RUL} \; and \; \hat{RUL} \neq \hat{RUL}'$$

Adversarial robustness: Given a model f', which is generated employing one of the adversarial defense strategies and M' is adversarial example from definition 4, then adversarial robustness can be described as box-constrained optimization problem.

$$\min_{\grave{RUL}} \left\| \grave{RUL} - \hat{RUL} \right\| \; s.t.$$

$$f'(M^{'}) = \grave{RUL}, \; f(M) = \hat{RUL} \; and \; \hat{RUL} \neq \grave{RUL}$$

5.3.2 Adversarial example generation for PHM

We craft adversarial examples (M') that defect the RUL predictions by increasing the cost of the model. In this work, we adopt and apply two adversarial example generation algorithms, Fast Gradient Sign Method (FGSM) [52] and Basic Iterative Method (BIM) [67].

Algorithm 1: FGSM algorithm for PHM

Input	**:** Original multivariate time series M from an equipment and its corresponding label $\hat{RUL}$
Output	**:** Perturbed multivariate time series M'
Parameter	**:** ϵ

$\eta = \epsilon \cdot sign(\nabla_m J_f(M, \hat{RUL}))$;
$M' = M + \eta$;

Fast Gradient Sign Method (FGSM): The FGSM was first proposed in [52] to generate adversarial images to fool the GoogLeNet model. The FGSM works by using the gradients of the neural network to create an adversarial example. This attack is also known as the one-shot method as the adversarial perturbation is generated by a single step computation. The attack is based on a one-step gradient update along the direction of the gradient's sign at each time step. This can be summarised using the equation $\eta = \epsilon \cdot sign(\nabla_m J_f(M, \hat{RUL}))$, where J_f is the cost function of model f, ∇_m indicates the gradient of the model with respect to the original MTS M with the correct label $\hat{RUL}$, ϵ denotes the hyper-parameter which controls the amplitude of the perturbation and M' is adversarial MTS. Algorithm 1 shows different steps of the FGSM attack.

Basic Iterative Method (BIM): The BIM [67] is an extension of FGSM. In BIM,

Algorithm 2: BIM algorithm for PHM

Input : Original multivariate time series M from an equipment and its corresponding label $\hat{RUL}$
Output : Perturbed multivariate time series M'
Parameter : I, ϵ, α

$M' \leftarrow M$;
while $i = 1 \leq I$ **do**
  $\eta = \alpha \cdot sign(\nabla_m J_f(M', \hat{RUL}))$;
  $M' = M' + \eta$;
  $M' = min\{M + \epsilon, max\{M - \epsilon, M'\}\}$;
  $i++$;
end

FGSM is applied multiple times with small step size, and clipping is performed after each step to ensure that they are in the range $[M - \epsilon, M + \epsilon]$ i.e. $\epsilon -$ *neighbourhood* of the original time series M. BIM is also known as Iterative-FGSM since FGSM is iterated with smaller step sizes. The adversarial examples generated through BIM are closer to the original input as perturbations are added iteratively and hence have a greater chance of fooling the network. However, compared to FGSM, BIM is computationally more expensive and slower. Algorithm 2 shows different steps of the BIM attack. The algorithm requires three hyperparameters: 1. the per step small perturbation (α); 2. the amount of maximum perturbation (ϵ) and 3. the number of iterations (I). Here, the value of α is calculated using $\alpha = \epsilon/I$.

5.4 Crafting adversarial examples for turbofan engine PHM case study

In this section, we evaluate adversarial attacks by performing a case study. At first, we explore the performance of five DL models without adversarial attacks and then apply the adversarial attacks to evaluate the impact on those models. We evaluate the transferability property of adversarial attacks. We also evaluate a adversarial robustness strategy to make a robust PHM model.

Table 5.1: RMSE comparison for different DL algorithms

Predictor architecture	**RMSE**
	Test
CNN(64,64,64.64) lh(100)	9.93
LSTM(100,100,100,100) lh(80)	8.80
GRU(100,100,100) lh(80)	7.62
MLP(128,128,64) lh(60)	11.60
Bi-LSTM(180,180,120) lh(60)	8.43

5.4.1 Deep learning models for the turbofan engine case study

To show the impact of adversarial examples on DL PHM models, we need to develop a PHM model first. For that, in this work, we perform a PHM case study using an aircraft Predictive Maintenance (PdM) [136] system. We use NASA's turbofan C-MAPSS [93] (Commercial Modular Aero-Propulsion System Simulation) dataset. This dataset includes 21 sensors data with a different number of operating conditions and fault conditions. We use the FD001 sub-dataset from the dataset for our experiments. We use five DL models, specifically, MLP, CNN, LSTM, Bi-LSTM, and GRU for predicting the RUL of the aircraft engines as they are known for their applicability in the PdM domain. The architecture of these DL algorithms are mentioned in section 5.2. As shown in Table 5.1, when there is no attack present, GRU(100, 100, 100) with a sequence length 80 provides the most accurate RUL prediction among these five models with the least Root Mean Square Error (RMSE) of 7.62. Next, we craft adversarial examples for these models and evaluate their impacts by comparing the performance of these DL models.

5.4.2 Threat model for the turbofan engine PdM

Before proceeding to the results, we describe the threat model for the turbofan engine PdM case study as follows.

Attack objective: The objective of the attacker is to trigger either an early or

delayed maintenance. An early, or in other words unnecessary maintenance can result in flight downtime and unnecessary maintenance, both of which lead to a loss of flight time, loss of human effort, loss of resources, and also incurs an extra maintenance cost. On the other hand, delayed maintenance may lead to an engine failure which might cause the loss of human lives in the worst case.

Attack surface: One of the ways to launch an adversarial attack is using spoofing techniques. For instance, Tippenhauer *et al.* [117] showed a spoof attack scenario on GPS-enabled devices. In this attack scenario, a forged GPS signal is transmitted to the device to alter the location. In this way, the true location of the device is disguised and the attacker can perform a physical attack on the device. In another work, Giannetos *et al.* [50] introduced an app named *Spy-sense*, which monitors behaviors of several sensors in a device. The app can manipulate sensor data by deleting or modifying it. *Spy-sense* exploits the active memory region in a device and relays sensitive data covertly. These works show that adversarial attacks can be performed even without gaining direct access to a system.

In this work, we consider both, the white-box and black-box attacks. The results of the black-box attack are demonstrated through the transferability property of the adversarial examples [88]. In the white-box attack, the adversary has access to the data and internal parameters of the DL model. In the case of aircraft predictive maintenance, the attacker can have access to the sensor data by exploiting controlled area network (CAN) bus systems aboard aircraft. The ICS-CERT published an alert on certain CAN bus systems on-board certain aircraft that might be vulnerable to cyber-threats. In this alert, an attacker with access to the aircraft could attach a malicious device to the avionics CAN bus to record and inject false data. This may result in incorrect readings in an avionic equipment [119]. This alert explores the possibility of capturing sensor measurements, crafting adversarial examples using the captured data, and injecting the crafted adversarial examples back into the system.

Attack signatures: As mentioned earlier, the FD001 sub-dataset has 100 en-

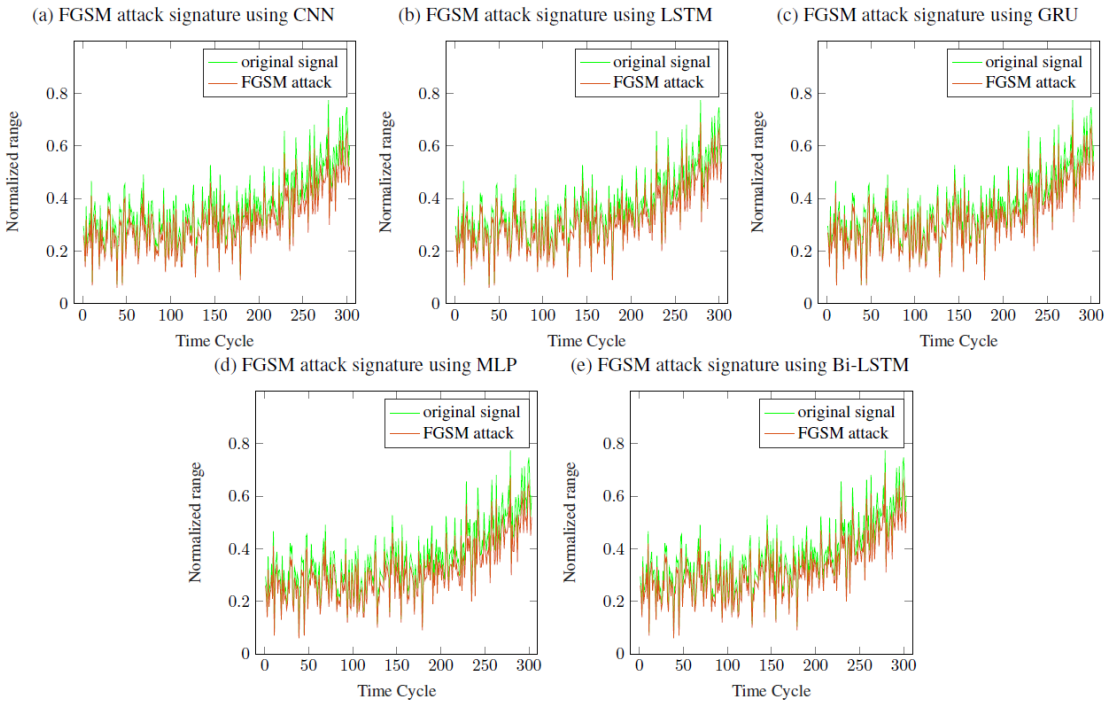

Figure 5.10: FGSM ($\epsilon = 0.3$) attack signature for sensor 2 of engine ID 49

gines, each of which has 21 sensors. Note, 7 out of these 21 engines can be ignored since their measurements remain constant. For the rest of the 14 sensors, we used the normalization technique to convert the raw sensory data into a normalized scale. We use the resultant normalized dataset to generate adversarial examples using FGSM and BIM. For illustration, Figure 5.10 and Figure 5.11 shows examples of a perturbed data from sensor 2 of engine ID 49 crafted using FGSM (with $\epsilon = 0.3$) and BIM (with $\alpha = 0.003$, $\epsilon = 0.3$ and $I = 100$), respectively. From Figure 5.11 it can be observed that the BIM attack generates adversarial examples that are closer to the input. We choose $\epsilon = 0.3$ for both FGSM and BIM attacks to make sure that the crafted adversarial examples are stealthy. Such stealthy attacks often fall within the boundary conditions of the sensor measurements, and hence they are indeed hard to detect using the common attack detection mechanisms.

5.4.3 Impact of adversarial attacks on turbofan engine PdM

To analyze the impact of attacks, we create a subset of test data from FD001 in which each engine has at least 150 time cycles of data. This gives us 37 engines

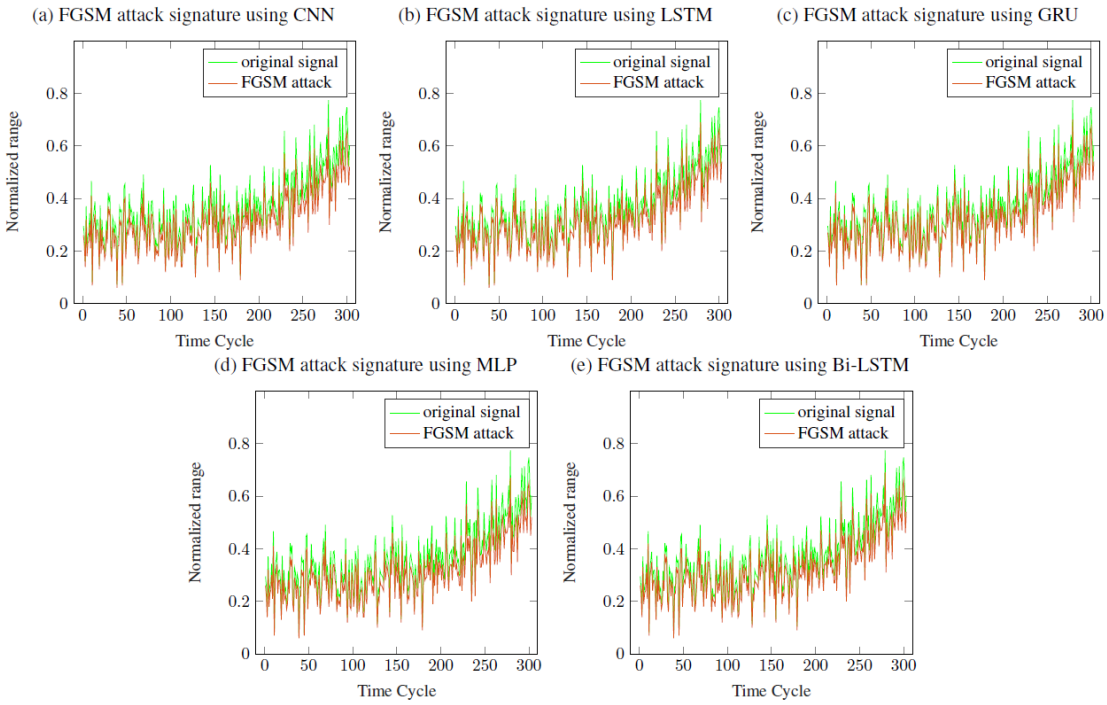

Figure 5.11: BIM ($\alpha = 0.003$, $\epsilon = 0.3$, and $I = 100$) attack signature for sensor 2 of engine ID 49

in the FD001 dataset. This is done since the engine's more time cycles data helps the DL models to make more accurate RUL predictions. This gives us 37 engines in the FD001 dataset. The resultant dataset is re-evaluated using the MLP, Bi-LSTM LSTM, CNN, and GRU-based PHM models and the obtained RMSEs are 10.98, 5.81, 5.83, 7.92, and 5.77, respectively.

To analyze the impact of FGSM and BIM attacks on the C-MAPSS, we craft adversarial examples using the proposed methodology and apply them to the DL models. From Figure 5.12, we observe that the FGSM attack (with $\epsilon = 0.3$) increases the RMSE of CNN, LSTM, GRU, MLP, and Bi-LSTM models by 131%, 134%, 94%, 56%, and 119% respectively, when compared to the DL models without attack. For the BIM attack (with $\alpha = 0.003$, $\epsilon = 0.3$ and $I = 100$), we also observe a similar trend, that is the RMSE for the CNN, LSTM, GRU, MLP, and Bi-LSTM model is increased by 294%, 351%, 346%, 175%, and 349% respectively, when compared to the DL models without attack. In all cases, as shown in Figure 5.12, the BIM attack results in a larger RMSE when compared to the FGSM attack.

The FGSM and BIM attacks can cause an under-prediction or over-prediction

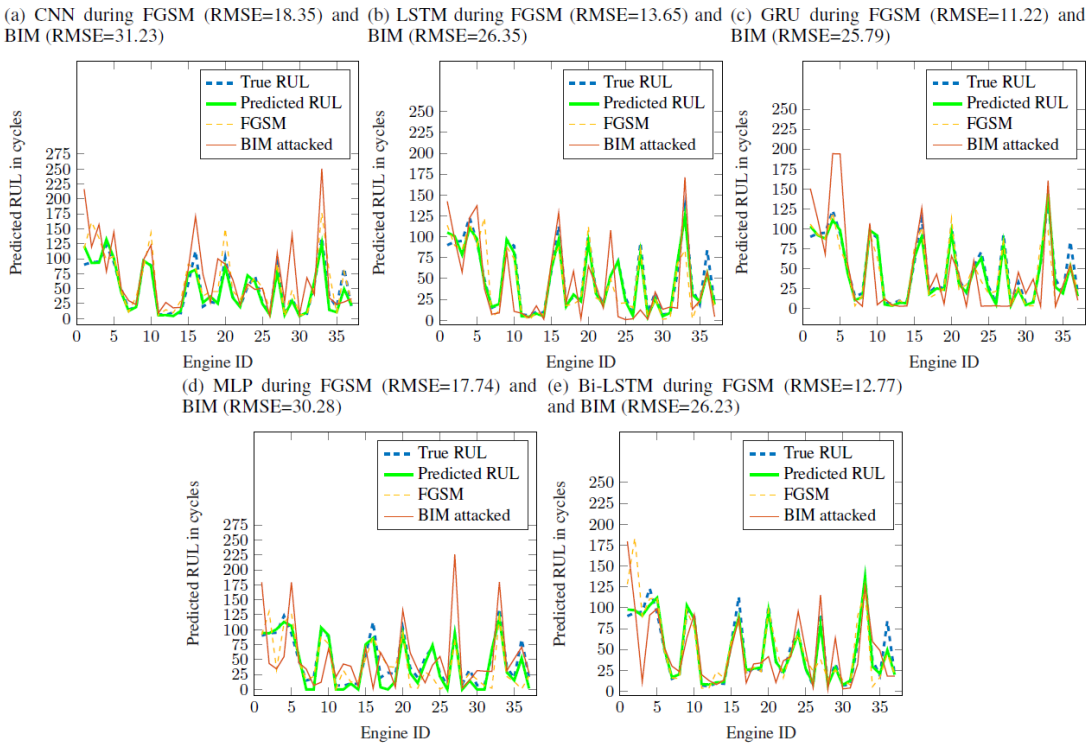

Figure 5.12: RUL estimation under FGSM ($\epsilon = 0.3$) and BIM ($\alpha = 0.003$, $\epsilon = 0.3$, and $I = 100$) attack

as mentioned in the attacker's objective. For instance, as shown in Figure 5.12, the CNN model predicts the RUL (without attack) of 125 (in hours) for engine ID 33 and 132 (in hours) for engine ID 4. After performing the FGSM and BIM attacks for engine ID 9, the same CNN model predicts the RUL (in hours) as 176 and 250, respectively. This represents a 14% and 20% increase in RUL after FGSM and BIM attacks. For engine ID 4, the FGSM and BIM attacks result in RUL of 97 and 78, respectively. This represents 26.51% and 40.9% decrease in the predicted RUL after FGSM and BIM attacks. An over-prediction, as shown in the first case, may cause delayed maintenance, whereas an under-prediction, as shown in the latter case may cause early maintenance, both of which have catastrophic consequences.

To elucidate the impact of FGSM and BIM attacks on specific engine data, we first apply the piece-wise RUL prediction (using the same DL models) for a single-engine (in this case engine ID 17) and then apply the crafted adversarial examples. The piece-wise RUL prediction gives a better visual representation of

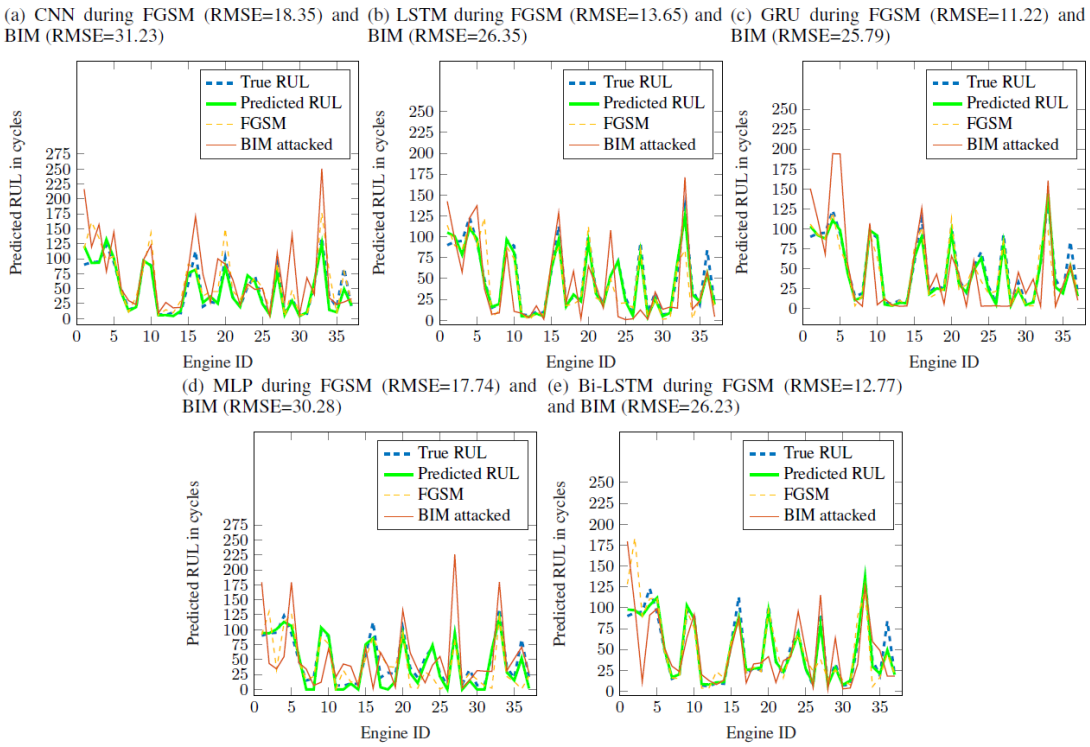

Figure 5.13: Piece-wise RUL prediction under FGSM ($\epsilon = 0.3$) and BIM ($\alpha = 0.003$, $\epsilon = 0.3$, and $I = 100$) attack

degradation (health status) in an aircraft engine. Figure 5.13(a), Figure 5.13(b), Figure 5.13(c), Figure 5.13(d), and Figure 5.13(e) shows the piece-wise RUL prediction using CNN, LSTM, GRU, MLP, and Bi-LSTM models, respectively, at each time step. From Figure 5.13, it is evident that as the time approaches towards the end of life, the predicted RUL (green solid line) is closer to the true RUL (blue dashes). This is because once the RUL predictions get more accurate with the increasing amount of data.

Next, we craft adversarial examples using both FGSM and BIM for that engine (engine ID 17), apply them for piece-wise RUL prediction, and compare their impact, as shown in Figure 5.13. We observe that the crafted adversarial examples have a strong impact from the beginning of the RUL prediction on the CNN model when compared to the MLP, Bi-LSTM, LSTM and GRU models. The piece-wise RUL prediction after the attack on the CNN model follows the same trend of the piece-wise RUL without attack, however, the attacked RUL values remain quite far from the actual prediction. On the other hand, the impact of adversarial attacks

on the GRU model is interesting since we observe that the RUL remains almost constant up to 104 time cycles and 129 time cycles for FGSM and BIM attacks, respectively, then starts decreasing. Such a phenomenon is deceiving in nature as it indicates that the engine is quite healthy and may influence a 'no maintenance required' decision by the maintenance engineer. Once again, it is evident that the BIM attack has a stronger impact on piece-wise RUL prediction when compared to the FGSM attack.

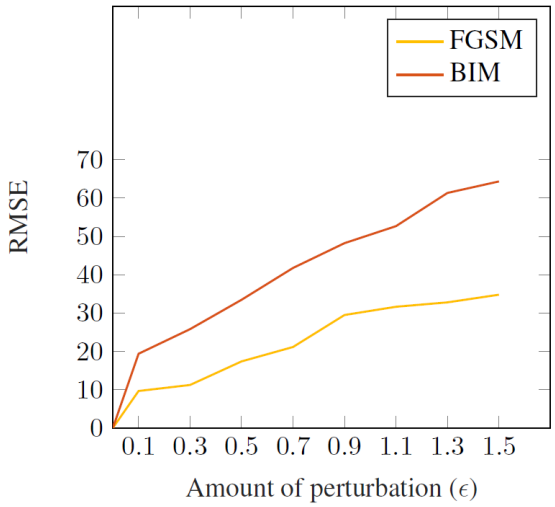

Figure 5.14: RMSE variation with respect to the amount of perturbation (ϵ) for FGSM and BIM attacks

Performance variation vs. the amount of perturbation: In this part of the experiments, we explore the impact of the amount of perturbation ϵ on the GRU model performance in terms of RMSE. We picked the GRU model as it showed the best performance in predicting the RUL. The obtained result is shown in Figure 5.14. We observe that for the larger values of ϵ, the BIM attack results in higher RMSE when compared to the FGSM. For instance, for $\epsilon = 1.3$, the FGSM attack results in an RMSE of 32.78, whereas the BIM attack results in an RMSE of 61.34. This shows that for the same value of ϵ, BIM can generate adversarial examples impacting the RMSE approximately twice when compared to the FGSM. This is due to the fact [67] that BIM adds a small amount of perturbation α on each iteration whereas FGSM adds the total amount of perturbation ϵ on each data point.

Transferability of attacks: To evaluate the transferability of adversarial attacks, we apply the adversarial examples crafted for a PHM model on the other PHM models using the data of 37 engines as mentioned in the experimental setup section. As mentioned earlier, such an attack is known as the *black box* attack [88], where the attacker has no knowledge of the target model's internal parameters, but still cause a considerable impact on the target model. The obtained results are shown in Table 5.2. The first column (DL models) of the Table 5.2 represents the RMSE of the models without attack. We observe that the FGSM and BIM adversarial examples crafted for the CNN model gives a higher RMSE when transferred to other DL models. Also, another interesting fact that we observe is when transferred, adversarial examples crafted using BIM results in a higher RMSE. For instance, the CNN-based PHM model has an RMSE of 7.92. When we craft adversarial examples for the CNN model using FGSM and BIM, and transfer to the GRU model, we observe that BIM adversarial attack increases the RMSE almost two times (27.12) when compared to the FGSM (14.23). A similar trend is also observed for all other PHM models when adversarial attacks are transferred.

Table 5.2: Transferability of FGSM and BIM attacks. The notation X/Y represents RMSE using FGSM/BIM

DL models	**RMSE**				
	MLP	**CNN**	**LSTM**	**Bi-LSTM**	**GRU**
MLP (RMSE = 10.98)	-	18.78 / 30.61	17.32 / 28.14	14.32 / 24.46	12.46 / 24.96
CNN (RMSE = 7.92)	18.47 / 30.56	-	18.76 / 29.45	14.44 / 24.62	14.23 / 27.12
LSTM (RMSE = 5.83)	19.96 / 32.45	18.23 / 31.13	-	14.74 / 23.26	11.33 / 18.65
Bi-LSTM (RMSE = 5.81)	18.42 / 31.88	17.44 / 30.67	12.32 / 21.36	-	10.72 / 19.66
GRU (RMSE = 5.77)	19.83 / 32.64	16.22 / 30.45	10.89 / 19.52	9.45 / 17.66	-

5.5 Adversarial training in turbofan engine PHM case study

In this work, we improve the adversarial robustness of a model using the adversarial training method from the proposed methodology. Adversarial training is one of the adversarial robustness strategies employed to make a model robust to

adversarial threats. This process consists of training a DL/ML model on adversarial examples crafted using adversarial attacks (FGSM and BIM), this process makes the model robust to adversarial threats.

Table 5.3: Adversarial training using FGSM and BIM attacks. The notation X/Y represents RMSE of test data after adversarial training using FGSM/BIM

DL models	**RMSE after testing on adversarial examples crafted using** $\epsilon = 0.3$				
	$\epsilon = 0.1$	$\epsilon = 0.3$	$\epsilon = 0.5$	$\epsilon = 0.7$	$\epsilon = 0.9$
MLP (RMSE = 11.23)	11.78/12.56	11.43 / 11.61	11.68 / 12.04	11.72 / 12.16	12.13 / 12.35
CNN (RMSE = 8.12)	8.82 / 9.23	8.32 / 8.46	8.48 / 8.65	8.75 / 9.11	9.12 / 9.42
LSTM (RMSE = 6.45)	6.93 / 7.42	6.55 / 6.77	6.72 / 6.89	6.96 / 7.19	7.25 / 7.75
Bi-LSTM (RMSE = 6.41)	6.82 / 7.37	6.50 / 6.72	6.66 / 6.81	6.93 / 7.11	7.18 / 7.68
GRU (RMSE = 6.36)	6.75 / 7.18	6.46 / 6.64	6.52 / 6.78	6.85 / 7.05	7.13 / 7.59

In adversarial training, we train a DL model on adversarial examples crafted for epsilon value in the range 0.1 to 0.9 and test the adversarially trained model on adversarial examples crafted using an epsilon value of 0.3 for all the experiments. From Table 5.3, the first column on the right consists of DL models with their respective RMSEs after adversarial training. It is observed that there is a slight increase in the RMSE of the model after adversarial training. From Table 5.3, it is observed that the effect of an adversarial attack is greatly reduced after adversarial training for both FGSM and BIM attacks. For example, in BIM attack, the increase of RMSE for the CNN, LSTM, GRU, MLP, and Bi-LSTM model for adversarial training epsilon of 0.3 is reduced from 294%, 351%, 346%, 175%, and 349% respectively, to 6.8%, 16%, 15%, 5.7%, and 15.6%, respectively. In FGSM attack, the increase of RMSE for the CNN, LSTM, GRU, MLP, and Bi-LSTM model for adversarial training epsilon of 0.3 is reduced from 131%, 134%, 94% 56%, and 119% respectively, to 5%, 12.43%, 11.95%, 4%, and 11.8%, respectively. This indeed shows that adversarial training is successful in mitigating adversarial threats.

From Table 5.3, it is observed that for adversarial training epsilon of 0.1 results in greater RMSE than the remaining adversarial training epsilons for all the DL models. This is because the DL models are trained on a smaller value of epsilon and tested on adversarial examples that are crafted using an epsilon, which is

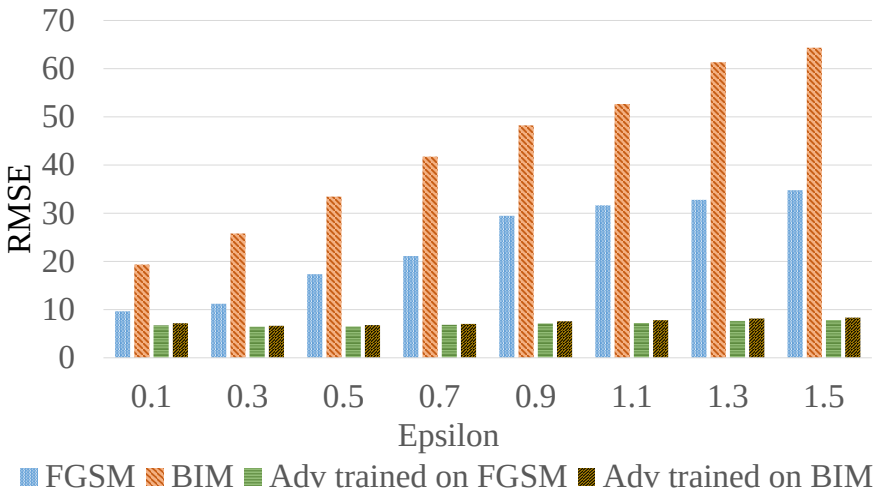

Figure 5.15: Comparision of adversarial trained models with non-adversarial trained models to FGSM and BIM attacks with respect to the increasing amount of perturbation (ϵ)

greater than adversarial training epsilon. It is also observed that as the value of training epsilon increases, we see an increasing trend in the RMSE for both the attack. For example, the CNN model of training epsilon 0.3 gives RMSE of 8.32 for FGSM attack, in comparison the same CNN model of training epsilon of 0.9 gives RMSE of 9.12 for FGSM attack.

The impact of adversarial training can be clearly seen in Figure 5.15. From Figure 5.15, it can be observed that as the value of epsilon increases the value of RMSE after FGSM and BIM attacks increases, but after adversarial training, the value of RMSE after FGSM and BIM attack is almost constant. For instance, for $\epsilon = 1.3$, the FGSM and BIM attacks on non-adversarial trained model results in an RMSE of 32.78 and 61.34, respectively, but after adversarial training, it results in RMSE of 7.67 and 8.17 for FGSM and BIM, respectively. From these experiments, we discern that adversarial training is successful in mitigating adversarial threats.

Table 5.4: RMSE comparison for different DL algorithms

Predictor architecture	**RMSE**
	Test
CNN(256,256,128,128) lh(60)	185.65
LSTM(180,180,120,120) lh(60)	144.97
GRU(180,180,120,120) lh(60)	139.13
MLP(256,256,256,128) lh(60)	190.25
Bi-LSTM(180,120,120) lh(60)	152.37

5.6 Crafting adversarial examples for battery PHM case study

In this section, we evaluate adversarial attacks by performing a case study. At first, we explore the performance of five PHM DL models without adversarial attacks and then apply the adversarial attacks to evaluate the impact on those models. We evaluate the transferability property of adversarial attacks. We also evaluate a adversarial robustness strategy to make a robust PHM model.

5.6.1 Deep learning models for the battery case study

To show the impact of adversarial examples on DL PHM models, we need to develop a PHM model first. For that, in this work, we perform a PHM case study using an battery PHM [101] system. The battery dataset [101] consists of 124 commercial lithium iron phosphate/graphite cells cycled under fast-charging conditions, with widely varying cycle lives ranging from 150 to 2,300 cycles. This dataset includes 7 sensors data recorded at regular intervals. We perform data prepossessing and remove three noisy battery data, so we are left with 121 battery data. We divide the battery dataset into train and test pairs, with train dataset having 100 battery data and test with 21 battery data. In the test data, we truncate the data at 300 cycles for all batteries, we make predictions of RUL based on the 300 cycles of data. We use five DL models, specifically, MLP, CNN, LSTM, Bi-LSTM, and GRU for predicting the RUL of the batteries as they are known for their applicability in the PHM domain. The architecture of these DL

algorithms are mentioned in section 5.2. As shown in Table 5.4, when there is no attack present, GRU(180,180,120,120) with a sequence length 60 provides the most accurate RUL prediction among these five models with the least Root Mean Square Error (RMSE) of 139.13. Next, we craft adversarial examples for these models and evaluate their impacts by comparing the performance of these DL models.

5.6.2 Threat model for the battery PdM

Before proceeding to the results, we describe the threat model for the battery PHM case study as follows.

Attack objective: The objective of the attacker is to trigger either an early or delayed maintenance. An early, or in other words unnecessary maintenance can result in downtime and unnecessary maintenance, both of which lead to a loss of critical time, loss of human effort, loss of resources, and also incurs an extra maintenance cost. On the other hand, delayed maintenance may lead to an equipment failure which may incur huge loss to the industry

Attack surface: In this work, we consider both, the white-box and black-box attacks. The results of the black-box attack are demonstrated through the transferability property of the adversarial examples [88]. In the white-box attack, the adversary has access to the data and internal parameters of the DL model. One of the ways to launch an adversarial attack is using spoofing techniques. For instance, Tippenhauer *et al.* [117] showed a spoof attack scenario on GPS-enabled devices. In this attack scenario, a forged GPS signal is transmitted to the device to alter the location. In this way, the true location of the device is disguised and the attacker can perform a physical attack on the device. In another work, Giannetos *et al.* [50] introduced an app named *Spy-sense*, which monitors behaviors of several sensors in a device. The app can manipulate sensor data by deleting or modifying

it. *Spy-sense* exploits the active memory region in a device and relays sensitive data covertly. These works show that adversarial attacks can be performed even without gaining direct access to a system. These attack surfaces explores the possibility of capturing sensor measurements, crafting adversarial examples using the captured data, and injecting the crafted adversarial examples back into the system.

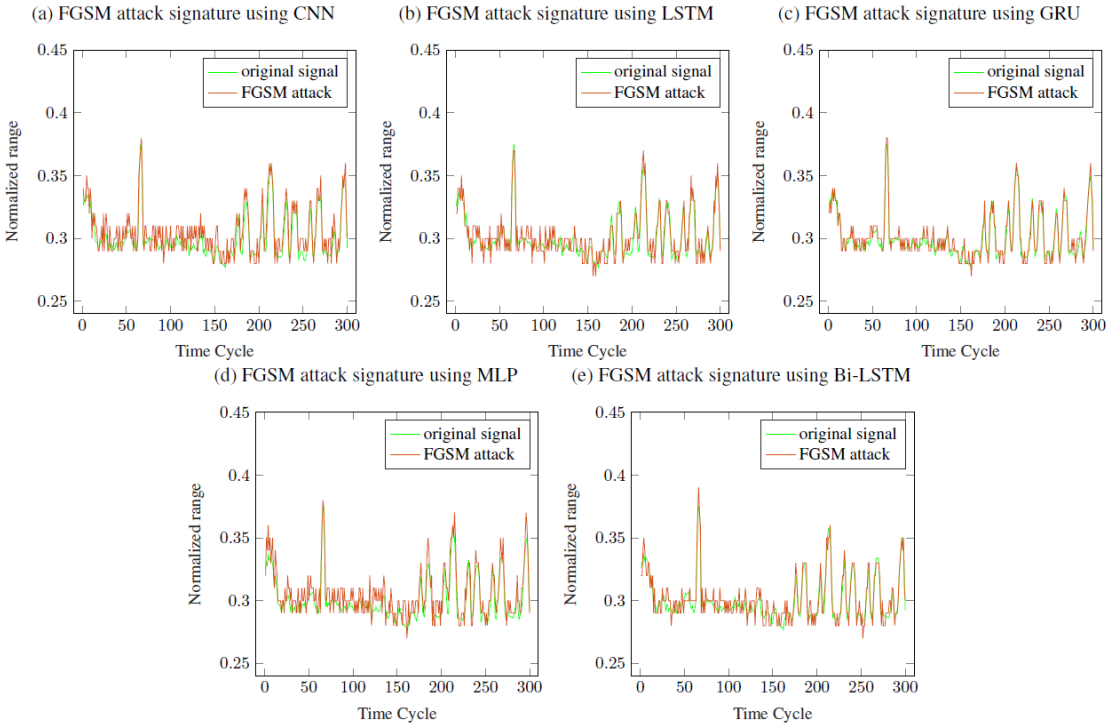

Figure 5.16: FGSM ($\epsilon = 0.6$) attack signature for sensor 4 of engine ID 1

Attack signatures: As mentioned earlier, the test dataset has 100 batteries, each of which has 7 sensors. For the 7 sensors, we used the normalization technique to convert the raw sensory data into a normalized scale. We use the resultant normalized dataset to generate adversarial examples using FGSM and BIM. For illustration, 5.16 and 5.17 shows examples of a perturbed data from sensor 4 (Tavg) of battery ID 1 in test dataset, crafted using FGSM (with $\epsilon = 0.6$) and BIM (with $\alpha = 0.006$, $\epsilon = 0.6$ and $I = 100$), respectively. From 5.17 it can be observed that the BIM attack generates adversarial examples that are closer to the input. We choose $\epsilon = 0.6$ for both FGSM and BIM attacks to make sure that the crafted adversarial examples are stealthy. Such stealthy attacks often fall within

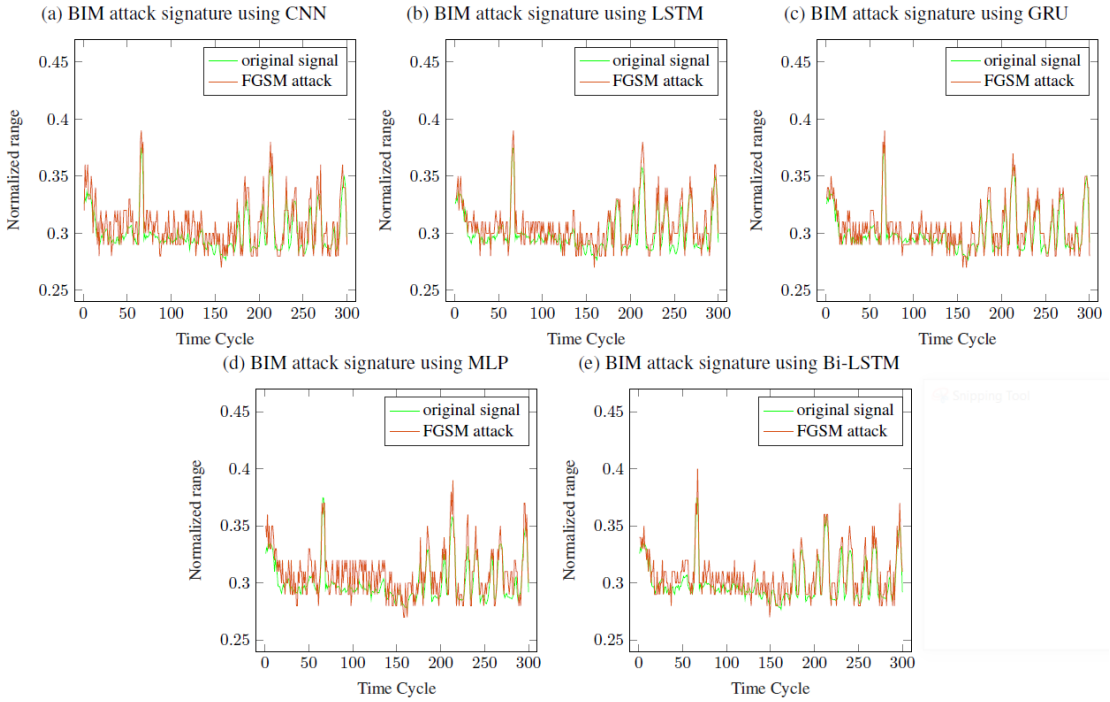

Figure 5.17: BIM ($\alpha = 0.006$, $\epsilon = 0.6$, and $I = 100$) attack signature for sensor 4 of battery ID 1

the boundary conditions of the sensor measurements, and hence they are indeed hard to detect using the common attack detection mechanisms.

5.6.3 Impact of adversarial attacks on battery PdM

To analyze the impact of FGSM and BIM attacks on the battery dataset, we craft adversarial examples using the proposed methodology and apply them to the DL models. From Figure 5.18, we observe that the FGSM attack (with $\epsilon = 0.6$) increases the RMSE of CNN, LSTM, GRU, MLP, and Bi-LSTM models by 33%, 65%, 46% 21%, and 47% respectively, when compared to the DL models without attack. For the BIM attack (with $\alpha = 0.006$, $\epsilon = 0.6$ and $I = 100$), we also observe a similar trend, that is the RMSE for the CNN, LSTM, GRU, MLP, and Bi-LSTM model is increased by 71%, 83%, 76%, 62%, and 83% respectively, when compared to the DL models without attack. In all cases, as shown in Figure 5.18, the BIM attack results in a larger RMSE when compared to the FGSM attack.

The FGSM and BIM attacks can cause an under-prediction or over-prediction

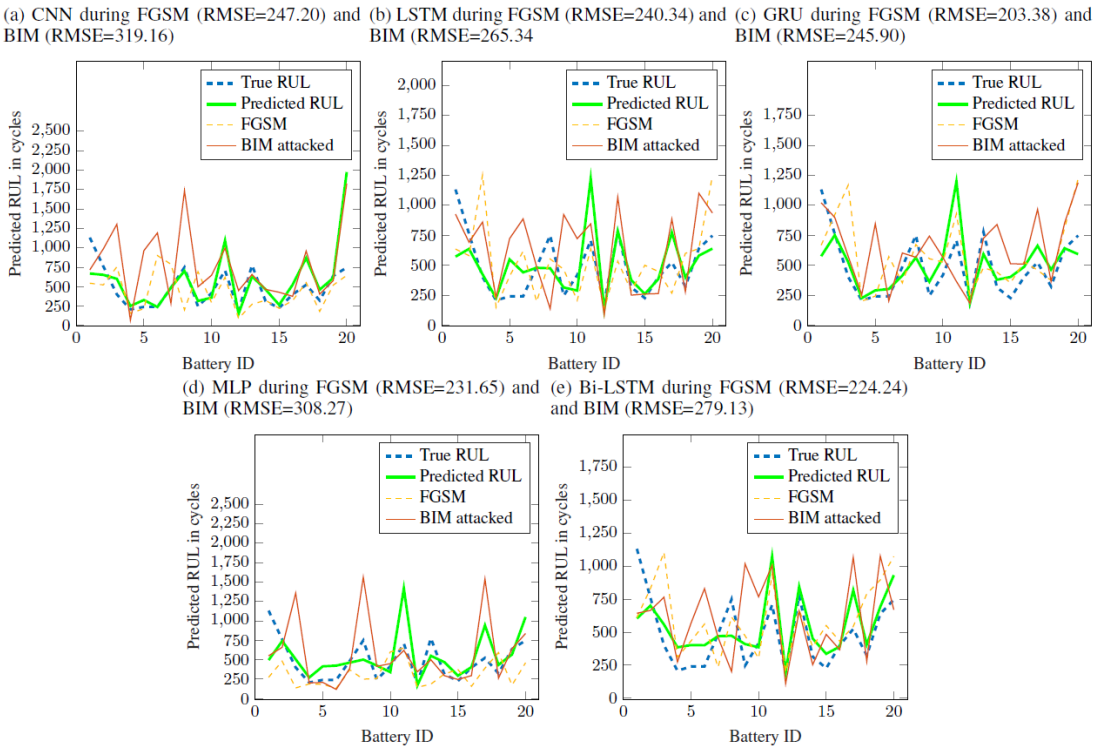

Figure 5.18: RUL estimation under FGSM ($\epsilon = 0.6$) and BIM ($\alpha = 0.006$, $\epsilon = 0.6$, and $I = 100$) attack

as mentioned in the attacker's objective. For instance, as shown in Figure 5.18, the CNN model predicts the RUL (without attack) of 238 (in cycles) for battery ID 6 and 251 (in cycles) for battery ID 4. After performing the FGSM and BIM attacks for battery ID 6, the same CNN model predicts the RUL (in hours) as 901 and 1192, respectively. This represents a 238% and 400% increase in RUL after FGSM and BIM attacks. For battery ID 4, the FGSM and BIM attacks result in RUL of 157 and 68, respectively. This represents 37% and 72% decrease in the predicted RUL after FGSM and BIM attacks. An over-prediction, as shown in the first case, may cause delayed maintenance, whereas an under-prediction, as shown in the latter case may cause early maintenance, both of which have catastrophic consequences.

To elucidate the impact of FGSM and BIM attacks on specific battery data, we first apply the piece-wise RUL prediction (using the same DL models) for a single-battery (in this case battery ID 2) and then apply the crafted adversarial examples. The piece-wise RUL prediction gives a better visual representation of degradation

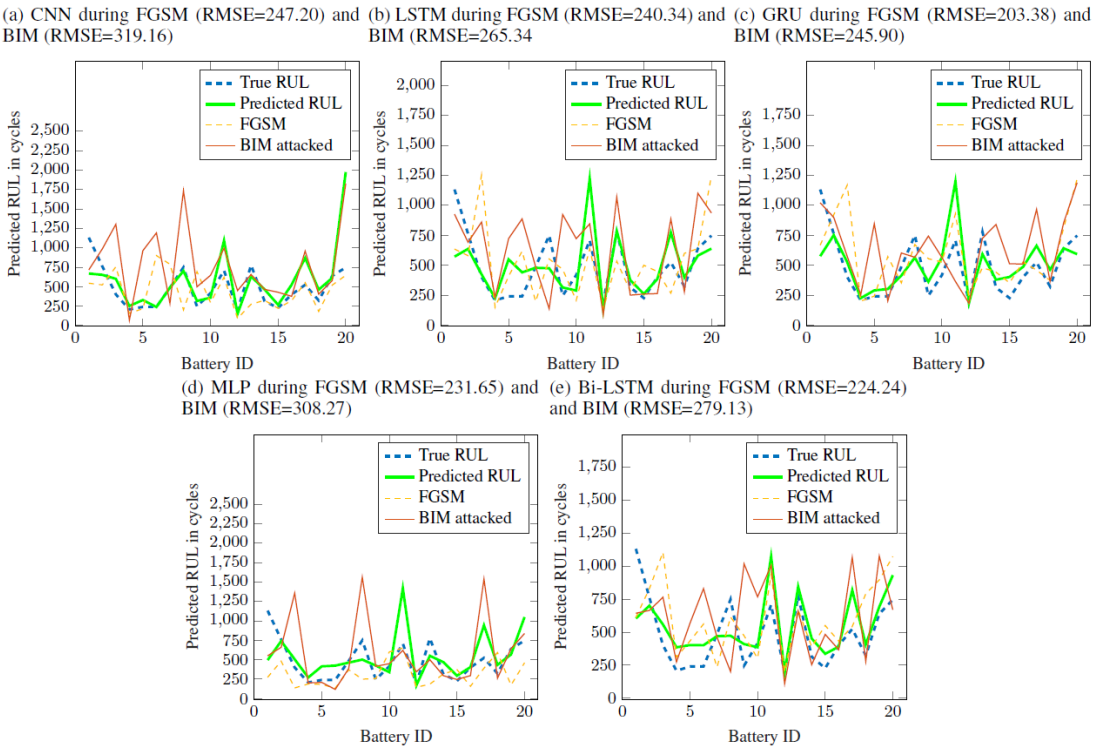

Figure 5.19: Piece-wise RUL prediction under FGSM ($\epsilon = 0.6$) and BIM ($\alpha = 0.006$, $\epsilon = 0.6$, and $I = 100$) attack

(health status) in an battery. Figure 5.19(a), Figure 5.19(b), Figure 5.19(c), Figure 5.19(d), and Figure 5.19(e) shows the piece-wise RUL prediction using CNN, LSTM, GRU, MLP, and Bi-LSTM models, respectively, at each time step. From Figure 5.19, it is evident that as the time approaches towards the end of life, the predicted RUL (green solid line) is closer to the true RUL (blue dashes). This is because once the RUL predictions get more accurate with the increasing amount of data. For LSTM model, it can observed that the model gives oscillating predictions but as the RUL approaches end of life the predictions become linear and also closer to the true prediction. The same can be observed for Bi-LSTM. This is because LSTM and Bi-LSTM models require more data to make accurate predictions.

Next, we craft adversarial examples using both FGSM and BIM for that battery (battery ID 2), apply them for piece-wise RUL prediction, and compare their impact, as shown in Figure 5.19. We observe that the crafted adversarial examples have a strong impact from the beginning of the RUL prediction on the MLP model

when compared to the CNN, Bi-LSTM, LSTM and GRU models. The piece-wise RUL prediction after the attack on the CNN model follows the same trend of the piece-wise RUL without attack, however, the attacked RUL values remain quite far from the actual prediction. The impact of BIM attack can be clearly seen in LSTM model, it is observed that the RUL after BIM attack is always above the true prediction. Such a phenomenon is deceiving in nature as it indicates that the battery is quite healthy and may influence a 'no maintenance required' decision by the maintenance engineer. Once again, it is evident that the BIM attack has a stronger impact on piece-wise RUL prediction when compared to the FGSM attack.

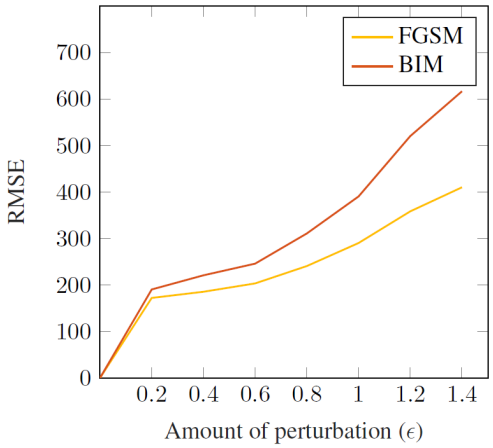

Figure 5.20: RMSE variation with respect to the amount of perturbation (ϵ) for FGSM and BIM attacks

Performance variation vs. the amount of perturbation: In this part of the experiments, we explore the impact of the amount of perturbation ϵ on the GRU model performance in terms of RMSE. We picked the GRU model as it showed the best performance in predicting the RUL. The obtained result is shown in Figure 5.20. We observe that for the larger values of ϵ, the BIM attack results in higher RMSE when compared to the FGSM. For instance, for $\epsilon = 1.2$, the FGSM attack results in an RMSE of 358.43, whereas the BIM attack results in an RMSE of 520.35. This shows that for the same value of ϵ, BIM can generate adversarial examples that have a higher impact when compared to the FGSM. This is due to

the fact [67] that BIM adds a small amount of perturbation α on each iteration whereas FGSM adds the total amount of perturbation ϵ on each data point.

Transferability of attacks: To evaluate the transferability of adversarial attacks, we apply the adversarial examples crafted for a PHM model on the other PHM models using the data of 37 engines as mentioned in the experimental setup section. As mentioned earlier, such an attack is known as the *black box* attack [88], where the attacker has no knowledge of the target model's internal parameters, but still cause a considerable impact on the target model. The obtained results are shown in Table 5.5. The first column (DL models) of the Table 5.5 represents the RMSE of the models without attack. We observe that the FGSM and BIM adversarial examples crafted for the CNN model gives a higher RMSE when transferred to other DL models.

Table 5.5: Transferability of FGSM and BIM attacks. The notation X/Y represents RMSE using FGSM/BIM

DL models	**RMSE**				
	MLP	**CNN**	**LSTM**	**Bi-LSTM**	**GRU**
MLP (RMSE = 190.25)	-	244.57 / 309.23	242.54 / 301.29	234.87 / 296.55	228.28 / 293.18
CNN (RMSE = 185.65)	247.92 / 311.56	-	246.77 / 310.67	238.35 / 303.79	231.11 / 297.61
LSTM (RMSE = 144.97)	206.78 / 251.38	199.89 / 244.33	-	184.94 / 227.98	180.21 / 223.45
Bi-LSTM (RMSE = 152.37)	209.55 / 255.23	203.45 / 248.42	189.21 / 232.94	-	185.43 / 229.72
GRU (RMSE = 139.13)	204.57 / 249.67	197.32 / 243.37	182.92 / 224.37	177.31 / 220.47	-

5.7 Adversarial training in battery PHM case study

Table 5.6: Adversarial training using FGSM and BIM attacks. The notation X/Y represents RMSE of test data after adversarial training using FGSM/BIM

DL models	**RMSE after testing on adversarial examples crafted using $\epsilon = 0.6$**				
	$\epsilon = 0.2$	$\epsilon = 0.4$	$\epsilon = 0.6$	$\epsilon = 0.8$	$\epsilon = 1.0$
MLP (RMSE = 198.56)	213.43/218.56	211.38 / 214.72	199.23 / 202.31	201.55 / 205.67	204.79 / 209.08
CNN (RMSE = 191.92)	208.32 / 211.55	206.43 / 209.57	193.64 /198.77	195.82 / 200.13	197.41 / 201.75
LSTM (RMSE = 148.32)	155.76 / 158.44	153.63 / 156.82	152.72 / 155.39	154.77 / 157.52	155.27 / 159.80
Bi-LSTM (RMSE = 157.55)	166.57 / 169.82	163.30 / 166.38	159.63 / 162.41	160.97 / 163.38	162.46 / 165.38
GRU (RMSE = 145.82)	153.54 / 157.21	151.35 / 154.88	149.54 / 153.72	151.27 / 154.21	153.56 / 156.48

In adversarial training, we train a DL model on adversarial examples crafted for epsilon value in the range 0.2 to 1 and test the adversarially trained model on

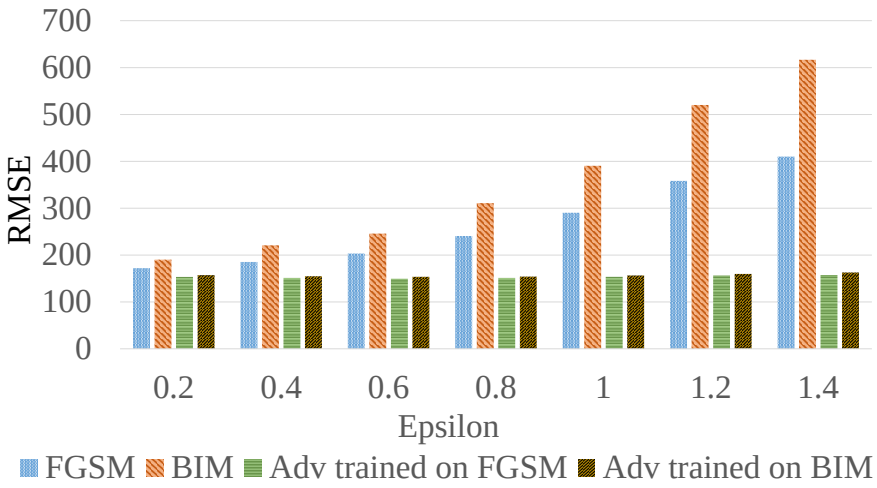

Figure 5.21: Comparision of adversarial trained models with non-adversarial trained models to FGSM and BIM attacks with respect to the increasing amount of perturbation (ϵ)

adversarial examples crafted using an epsilon value of 0.6 for all the experiments. From Table 5.6, the first column on the right consists of DL models with their respective RMSEs after adversarial training. It is observed that there is a slight increase in the RMSE of the model after adversarial training. From Table 5.6, it is observed that the effect of an adversarial attack is greatly reduced after adversarial training for both FGSM and BIM attacks. For example, in BIM attack, the increase of RMSE for the CNN, LSTM, GRU, MLP, and Bi-LSTM model for adversarial training epsilon of 0.6 is reduced from 71%, 83%, 76%, 62%, and 83% respectively, to 7%, 7.1%, 10%, 6.3%, and 6.5%, respectively. In FGSM attack, the increase of RMSE for the CNN, LSTM, GRU, MLP, and Bi-LSTM model for adversarial training epsilon of 0.6 is reduced from 33%, 65%, 46% 21%, and 47% respectively, to 4.3%, 5.3%, 7.4%, 4.7%, and 4.7%, respectively. This indeed shows that adversarial training is successful in mitigating adversarial threats.

From Table 5.6, it is observed that for adversarial training epsilon of 0.2 and 0.4 resultsin greater RMSE than the remaining adversarial training epsilons for all the DL models. This is because the DL models are trained on a smaller value of epsilon and tested on adversarial examples that are crafted using an epsilon, which is greater than adversarial training epsilon. It is also observed that as the value

of training epsilon increases, we see an increasing trend in the RMSE for both the attack. For example, the CNN model of training epsilon 0.6 gives RMSE of 193.64 for FGSM attack, in comparison the same CNN model of training epsilon of 1.0 gives RMSE of 197.41 for FGSM attack.

The impact of adversarial training can be clearly seen in Figure 5.21. From Figure 5.21, it can be observed that as the value of epsilon increases the value of RMSE after FGSM and BIM attacks increases, but after adversarial training, the value of RMSE after FGSM and BIM attack is almost constant. For instance, for $\epsilon = 1.2$, the FGSM and BIM attacks on non-adversarial trained model results in an RMSE of 358.43 and 520.35, respectively, but after adversarial training, it results in RMSE of 156.55 and 159.73 for FGSM and BIM, respectively. From these experiments, we discern that adversarial training is successful in mitigating adversarial threats.

5.8 Discussion

In the previous section, we built two robust PHM systems. In the process, we observed that GRU model performs better than the remaining DL models in predicting RUL for both the cases studies. For LSTM and Bi-LSTM model, it is observed that the model gives oscillating predictions in the case of battery PHM. This is because LSTM and Bi-LSTM models require more data to make accurate predictions. For adversarial attacks, the BIM generates adversarial examples that result in higher RMSE than FGSM. The impact of adversarial examples on turbofan engine case study is more prominent when compared to the battery PHM. It is because crafting of adversarial attacks depend not only on the DL algorithms but also on the dataset. It is also observed that the CNN generates adversarial examples that result in higher RMSE and also the adversarial examples are highly transferable. The adversarial training in both the case studies was able to reduce the impact of adversarial attacks and also the RMSE after the attacks remained

constant for increasing values of epsilon for both the case studies.

5.9 Summary

In this chapter, We used the proposed methodology to craft adversarial attacks to the PHM domain to reveal their vulnerabilities and show how they can be used to defect the deep learning-enabled prognostic models. We crafted adversarial examples using the FGSM and BIM algorithms for LSTM, GRU, CNN, MLP, and Bi-LSTM based PHM models using NASA's turbofan engine case study. The obtained results showed that the crafted adversarial examples that include very small perturbation to the original sensor measurements can cause serious defects to the remaining useful life estimation. The BIM performed better in defecting the RUL with a smaller perturbation when compared to the FGSM. We also observed that adversarial examples crafted for CNN models are more transferable to the other DL models when compared to the LSTM, GRU, MLP, and Bi-LSTM. We employed adversarial training as one of the defense strategies to mitigate the impact of adversarial threats to the PHM domain and the results showed that it is successful in mitigating the threats. Our work is the first one to shed light on the importance of defending such adversarial attacks in the PHM domain and also the first one to employ one of the adversarial defense strategies to mitigate the impact of adversarial threats in the PHM domain. In the next chapter, we conclude the thesis and also propose the future scope of the research.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we provided a methodology to design and build robust PHM systems. We employed our methodology to build a PHM system, craft adversarial attacks for the PHM system and also implement adversarial training to make a robust PHM system. A researcher/engineer can use this methodology in the pre-deployment stage to build a robust PHM model. We implemented our methodology in real-life scenarios using NASA's C-MAPSS dataset and Li-ion battery dataset. At first, we built PHM systems using state-of-the-art DL algorithms like LSTM, GRU, CNN, and Bi-LSTM. Second, we introduced adversarial attacks from the image domain to the PHM domain to reveal their vulnerabilities and show how they can be used to defect the deep learning-enabled prognostic models. We crafted adversarial examples using the FGSM and BIM algorithms for LSTM, GRU, CNN, MLP, and Bi-LSTM based PHM models. The obtained results showed that the crafted adversarial examples that include very small perturbation to the original sensor measurements can cause serious defects to the remaining useful life estimation. The BIM attack performed better in defecting the RUL with a smaller perturbation when compared to the FGSM for both the

case studies. We also observed that adversarial examples crafted for CNN models are more transferable to the other DL models.

Lastly, We employed adversarial training as one of the defense strategies to mitigate the impact of adversarial threats to the PHM domain and the results showed that it is successful in mitigating the threats. Our work is the first one to shed light on the importance of defending such adversarial attacks in the PHM domain and also the first one to employ one of the adversarial defense strategies to mitigate the impact of adversarial threats in the PHM domain.

6.2 Future Work

The future scope for this solution is implementing several state-of-the-art adversarial attacks and also defense strategies that are proposed to the image domain and implement them on regression-based PHM systems. The defense strategies are useful in mitigating the effect of adversarial attacks but are not useful in detecting adversarial attacks. In the future, we also plan to implement adversarial attack/cyber-attack detection algorithms to detect the attacks beforehand and alert the engineers about the threats.

Bibliography

[1] Artificial intelligence for predictive maintenance, Aerospace Manufacturing and Design Magazine. Available: https://www.aerospacemanufacturinganddesign.com/article/artificial-intelligence-for-predictive-maintenance/.

[2] Cyber-attacks On Smart Factories Are On The Rise. Available: https://smartmachinesandfactories.com/news/fullstory.php/aid/459/Cyber-attacks$_o n_s mart_f actories_a re_o n_t he_r ise.html$.

[3] IOT Use Cases and Innovation in IOT. Available: https://medium.com/@billsoftnet/iot-use-cases-and-innovation-in-iot-6b4e49fbc9dc.

[4] Is Industrial Machine Learning Predictive Maintenance a Cyber Security Risk? Available: https://www.presenso.com/blog/machine-learning-predictive-maintenance-cyber-security.

[5] Maintaining the data-rich Pratt Whitney GTF engine. Available: https://www.sae.org/news/2018/10/maintaining-the-data-rich-pratt–whitney-gtf-engine.

[6] Predictive digonostics, Bosch. Available: https://www.bosch-mobility-solutions.com/en/products-and-services/mobility-services/predictive-diagnostics/.

[7] Predictive maintenance benefits for the freight logistics industr. Available: https://www.ibm.com/downloads/cas/AVNOLWQW.

[8] The Rolls-Royce Intelligent Engine — Driven by data. Available: https://www.rolls-royce.com/media/press-releases/2018/06-02-2018-rr-intelligentengine-driven-by-data.aspx.

[9] The US Air Force Is Adding Algorithms to Predict When Planes Will Break, Defense One Magazine. Available: https://www.defenseone.com/business/2018/05/us-air-force-adding-algorithms-predict-when-planes-will-break/148234/.

[10] Transforming Railroad Asset Management: Going Smart with Predictive Maintenance. Available: https://www.tcs.com/content/dam/tcs/pdf/Industries/travel-and-hospitality/Transforming-Railroad-Asset-Management.pdf.

[11] USAF Launches Predictive Maintenance For Three Fleets. Available: https://aviationweek.com/defense/usaf-launches-predictive-maintenance-three-fleets.

[12] Mohamed Abomhara et al. Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks. *Journal of Cyber Security and Mobility*, 4(1):65–88, 2015.

[13] Jigsaw Academy. Layers Of The Internet Of Things. https://analyticstraining.com/4-layers-of-the-internet-of-things/, 2018. [Online; accessed 03-Dec-2018].

[14] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018.

[15] Argimiro Arratia and Eduardo Sepúlveda. Convolutional neural networks, image recognition and financial time series forecasting. In *Workshop on Mining Data for Financial Applications*, pages 60–69. Springer, 2019.

[16] Giduthuri Sateesh Babu, Peilin Zhao, and Xiao-Li Li. Deep convolutional neural network based regression approach for estimation of remaining useful life. In *International conference on database systems for advanced applications*, pages 214–228. Springer, 2016.

[17] John Backes, Pauline Bolignano, Byron Cook, Andrew Gacek, Kasper Soe Luckow, Neha Rungta, Martin Schaef, Cole Schlesinger, Rima Tanash, Carsten Varming, et al. One-click formal methods. *IEEE Software*, 36(6):61–65, 2019.

[18] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31(3):606–660, 2017.

[19] Haowei Bai, Mohammed Atiquzzaman, and David Lilja. Wireless sensor network for aircraft health monitoring. In *First International Conference on Broadband Networks*, pages 748–750. IEEE, 2004.

[20] Vineeth Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk. *Conformal prediction for reliable machine learning: theory, adaptations and applications*. Newnes, 2014.

[21] Chakib Bekara. Security issues and challenges for the iot-based smart grid. *Procedia Computer Science*, 34:532–537, 2014.

[22] Ashwin Bhandare, Maithili Bhide, Pranav Gokhale, and Rohan Chandavarkar. Applications of convolutional neural networks. *International Journal of Computer Science and Information Technologies*, 7(5):2206–2215, 2016.

[23] Battista Biggio, Blaine Nelson, and Pavel Laskov. Support vector machines under adversarial label noise. In *Asian conference on machine learning*, pages 97–112, 2011.

[24] Anastasia Borovykh, Sander Bohte, and Cornelis W Oosterlee. Conditional time series forecasting with convolutional neural networks. *arXiv preprint arXiv:1703.04691*, 2017.

[25] Bruce Jackson. Threat of a remote cyberattack on today's aircraft is real, 2019. [Online; accessed 01-07-2019].

[26] Jian Cao, Zhi Li, and Jian Li. Financial time series forecasting model based on ceemdan and lstm. *Physica A: Statistical Mechanics and its Applications*, 519:127–139, 2019.

[27] R Caponetto, F Rizzo, L Russotti, and MG Xibilia. Deep learning algorithm for predictive maintenance of rotating machines through the analysis of the orbits shape of the rotor shaft. In *International Conference on Smart Innovation, Ergonomics and Applied Human Factors*, pages 245–250. Springer, 2019.

[28] Nicholas Carlini, Guy Katz, Clark Barrett, and David L Dill. Provably minimally-distorted adversarial examples. *arXiv preprint arXiv:1709.10207*, 2017.

[29] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[30] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*, 2018.

[31] Jinglong Chen, Hongjie Jing, Yuanhong Chang, and Qian Liu. Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process. *Reliability Engineering & System Safety*, 185:372–382, 2019.

[32] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[33] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[34] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E Kounavis, and Duen Horng Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv preprint arXiv:1705.02900*, 2017.

[35] David Cenciotti. Cybersecurity in the sky: Internet of things capabilities making aircraft more exposed to cyber threats than ever before, 2017. [Online; accessed 20-June-2017].

[36] Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. Gru-ode-bayes: Continuous modeling of sporadically-observed time series. In *Advances in Neural Information Processing Systems*, pages 7377–7388, 2019.

[37] Jan G De Gooijer and Rob J Hyndman. 25 years of time series forecasting. *International journal of forecasting*, 22(3):443–473, 2006.

[38] Santos Merino Del Pozo, François-Xavier Standaert, Dina Kamel, and Amir Moradi. Side-channel attacks from static power: When should we care? In *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, pages 145–150. EDA Consortium, 2015.

[39] Matthias Auf der Mauer, Tristan Behrens, Mahdi Derakhshanmanesh, Christopher Hansen, and Stefan Muderack. Applying sound-based analysis at porsche production: Towards predictive maintenance of production

machines using deep learning and internet-of-things technology. In *Digitalization Cases*, pages 79–97. Springer, 2019.

[40] Derui Ding, Qing-Long Han, Yang Xiang, Xiaohua Ge, and Xian-Ming Zhang. A survey on security control and attack detection for industrial cyber-physical systems. *Neurocomputing*, 275:1674–1683, 2018.

[41] Xishuang Dong, Lijun Qian, and Lei Huang. A cnn based bagging learning approach to short-term load forecasting in smart grid. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 1–6. IEEE, 2017.

[42] André Listou Ellefsen, Emil Bjørlykhaug, Vilmar Æsøy, Sergey Ushakov, and Houxiang Zhang. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliability Engineering & System Safety*, 183:240–251, 2019.

[43] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Adversarial attacks on deep neural networks for time series classification. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[44] Olga Fink, Qin Wang, Markus Svensén, Pierre Dersin, Wan-Jui Lee, and Melanie Ducoffe. Potential, challenges and future directions for deep learning in prognostics and health management applications. *Engineering Applications of Artificial Intelligence*, 92:103678, 2020.

[45] Dean K Frederick, Jonathan A DeCastro, and Jonathan S Litt. User's guide for the commercial modular aero-propulsion system simulation (c-mapss). 2007.

[46] John Cristian Borges Gamboa. Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*, 2017.

[47] Chong Leong Gan. Prognostics and health management of electronics: Fundamentals, machine learning, and the internet of things, 2020.

[48] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[49] Ji Gao, Beilun Wang, Zeming Lin, Weilin Xu, and Yanjun Qi. Deepcloak: Masking deep neural network models for robustness against adversarial samples. *arXiv preprint arXiv:1702.06763*, 2017.

[50] Thanassis Giannetsos and Tassos Dimitriou. Spy-sense: spyware tool for executing stealthy exploits against sensor networks. In *Proceedings of the 2nd ACM workshop on Hot topics on wireless network security and privacy*, pages 7–12. ACM, 2013.

[51] Akhil Goel, Anirudh Singh, Akshay Agarwal, Mayank Vatsa, and Richa Singh. Smartbox: Benchmarking adversarial detection and mitigation algorithms for face recognition. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–7. IEEE, 2018.

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

[53] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.

[54] Yanpeng Guan and Xiaohua Ge. Distributed attack detection and secure estimation of networked cyber-physical systems against false data injection attacks and jamming attacks. *IEEE Transactions on Signal and Information Processing over Networks*, 4(1):48–59, 2017.

[55] Nikou Günnemann and Jürgen Pfeffer. Predicting defective engines using convolutional neural networks on temporal vibration signals. In *First International Workshop on Learning with Imbalanced Domains: Theory and Applications*, pages 92–102, 2017.

[56] Felix O Heimes. Recurrent neural networks for remaining useful life estimation. In *2008 international conference on prognostics and health management*, pages 1–6. IEEE, 2008.

[57] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[58] Sepp Hochreiter and Jürgen Schmidhuber. Lstm can solve hard long time lag problems. In *Advances in neural information processing systems*, pages 473–479, 1997.

[59] Klaus Hünecke. *Jet engines: Fundamentals of theory, design and operation.* Number BOOK. Airlife, 1997.

[60] Timo Huuhtanen and Alexander Jung. Predictive maintenance of photovoltaic panels via deep learning. In *2018 IEEE Data Science Workshop (DSW)*, pages 66–70. IEEE, 2018.

[61] Turker Ince, Serkan Kiranyaz, Levent Eren, Murat Askar, and Moncef Gabbouj. Real-time motor fault detection by 1-d convolutional neural networks. *IEEE Transactions on Industrial Electronics*, 63(11):7067–7075, 2016.

[62] Pengtao Jia, Hangduo Liu, Sujian Wang, and Peng Wang. Research on a mine gas concentration forecasting model based on a gru network. *IEEE Access*, 8:38023–38031, 2020.

[63] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *International conference on machine learning*, pages 2342–2350, 2015.

[64] Andrew J Kerns, Daniel P Shepard, Jahshan A Bhatti, and Todd E Humphreys. Unmanned aircraft capture and control via gps spoofing. *Journal of Field Robotics*, 31(4):617–636, 2014.

[65] Rakesh Kumar and Rinkaj Goyal. On cloud security requirements, threats, vulnerabilities and countermeasures: A survey. *Computer Science Review*, 33:1–48, 2019.

[66] Alex Kurakin, Dan Boneh, Florian Tramèr, Ian Goodfellow, Nicolas Papernot, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. 2018.

[67] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[68] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[69] Lee Neubecker. Could a sonic weapon have caused the two recent boeing 737 max 8 crashes?, 2019. [Online; accessed 22-March-2019].

[70] Beibei Li, Rongxing Lu, Wei Wang, and Kim-Kwang Raymond Choo. Distributed host-based collaborative detection for false data injection attacks in smart grid cyber-physical system. *Journal of Parallel and Distributed Computing*, 103:32–41, 2017.

[71] Jialin Li, Xueyi Li, and David He. A directed acyclic graph network combined with cnn and lstm for remaining useful life prediction. *IEEE Access*, 7:75464–75475, 2019.

[72] Xueyi Li, Jialin Li, Chengying Zhao, Yongzhi Qu, and David He. Early gear pitting fault diagnosis based on bi-directional lstm. In *10th Prognostics and System Health Management Conference, PHAI-Qingdao 2019*, page 8942949. Institute of Electrical and Electronics Engineers Inc., 2019.

[73] Gaoqi Liang, Junhua Zhao, Fengji Luo, Steven R Weller, and Zhao Yang Dong. A review of false data injection attacks against modern power systems. *IEEE Transactions on Smart Grid*, 8(4):1630–1638, 2016.

[74] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1787, 2018.

[75] Chenyu Liu and Konstantinos Gryllias. Unsupervised domain adaptation based remaining useful life prediction of rolling element bearings. In *PHM Society European Conference*, volume 5, pages 10–10, 2020.

[76] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[77] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[78] Anindya Maiti, Murtuza Jadliwala, Jibo He, and Igor Bilogrevic. (smart) watch your taps: side-channel keystroke inference attacks using smartwatches. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pages 27–30. ACM, 2015.

[79] Kebina Manandhar, Xiaojun Cao, Fei Hu, and Yao Liu. Detection of faults and attacks including false data injection attack in smart grid using kalman filter. *IEEE transactions on control of network systems*, 1(4):370–379, 2014.

[80] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM, 2017.

[81] Gautam Raj Mode, Prasad Calyam, and Khaza Anuarul Hoque. False data injection attacks in internet of things and deep learning enabled predictive analytics. *arXiv preprint arXiv:1910.01716*, 2019.

[82] Gautam Raj Mode and Khaza Anuarul Hoque. Adversarial examples in deep learning for multivariate time series regression. *arXiv preprint arXiv:2009.11911*, 2020.

[83] Gautam Raj Mode and Khaza Anuarul Hoque. Crafting adversarial examples for deep learning based prognostics (extended version). *arXiv preprint arXiv:2009.10149*, 2020.

[84] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[85] Nyein Naing, Wai Yan, and Zaw Zaw Htike. State of the art machine learning techniques for time series forecasting: A survey. *Advanced Science Letters*, 21(11):3574–3576, 2015.

[86] Izaskun Oregi, Javier Del Ser, Aritz Perez, and Jose A Lozano. Adversarial sample crafting for time series classification with elastic similarity measures. In *International Symposium on Intelligent and Distributed Computing*, pages 26–39. Springer, 2018.

[87] Sankar K Pal and Sushmita Mitra. Multilayer perceptron, fuzzy sets, classifiaction. 1992.

[88] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine

learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519. ACM, 2017.

[89] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

[90] Kyungnam Park, Yohwan Choi, Won Jae Choi, Hee-Yeon Ryu, and Hongseok Kim. Lstm-based battery remaining useful life prediction with multi-channel charging profiles. *IEEE Access*, 8:20786–20798, 2020.

[91] Shilin Qiu, Qihe Liu, Shijie Zhou, and Chunjiang Wu. Review of artificial intelligence adversarial attack and defense technologies. *Applied Sciences*, 9(5):909, 2019.

[92] Md Ashfaqur Rahman and Hamed Mohsenian-Rad. False data injection attacks with incomplete information against smart power grids. In *2012 IEEE Global Communications Conference (GLOBECOM)*, pages 3153–3158. IEEE, 2012.

[93] Emmanuel Ramasso and Abhinav Saxena. Performance benchmarking and analysis of prognostic methods for cmapss datasets. *International Journal of Prognostics and Health Management*, 5(2):1–15, 2014.

[94] Lei Ren, Xuejun Cheng, Xiaokang Wang, Jin Cui, and Lin Zhang. Multi-scale dense gate recurrent unit networks for bearing remaining useful life prediction. *Future Generation Computer Systems*, 94:601–609, 2019.

[95] Ishai Rosenberg, Asaf Shabtai, Yuval Elovici, and Lior Rokach. Defense methods against adversarial examples for recurrent neural networks. *arXiv preprint arXiv:1901.09963*, 2019.

[96] Ahmad-Reza Sadeghi, Christian Wachsmann, and Michael Waidner. Security and privacy challenges in industrial internet of things. In *2015*

52nd ACM/EDAC/IEEE Design Automation Conference (DAC), pages 1–6. IEEE, 2015.

[97] Alaa Sagheer and Mostafa Kotb. Time series forecasting of petroleum production using deep lstm recurrent networks. *Neurocomputing*, 323:203–213, 2019.

[98] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

[99] Abhinav Saxena and Kai Goebel. C-mapss data set. *NASA Ames Prognostics Data Repository*, 2008.

[100] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[101] Kristen A Severson, Peter M Attia, Norman Jin, Nicholas Perkins, Benben Jiang, Zi Yang, Michael H Chen, Muratahan Aykol, Patrick K Herring, Dimitrios Fraggedakis, et al. Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy*, 4(5):383–391, 2019.

[102] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, 90:106181, 2020.

[103] Maoya Shen, Qifeng Xu, Kaijie Wang, Mengfu Tu, and Bingxiang Wu. Short-term bus load forecasting method based on cnn-gru neural network. In *Proceedings of PURPLE MOUNTAIN FORUM 2019-International Forum on Smart Grid Protection and Control*, pages 711–722. Springer, 2020.

[104] Yasser Shoukry, Pierluigi Nuzzo, Alberto Puggelli, Alberto L Sangiovanni-Vincentelli, Sanjit A Seshia, and Paulo Tabuada. Secure state estimation for cyber-physical systems under sensor attacks: A satisfiability modulo

theory approach. *IEEE Transactions on Automatic Control*, 62(10):4917–4932, 2017.

[105] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, and Dong-Hua Zhou. Remaining useful life estimation–a review on the statistical data driven approaches. *European journal of operational research*, 213(1):1–14, 2011.

[106] Amit Kumar Sikder, Giuseppe Petracca, Hidayet Aksu, Trent Jaeger, and A Selcuk Uluagac. A survey on sensor-based threats to internet-of-things (IoT) devices and applications. *arXiv preprint arXiv:1802.02041*, 2018.

[107] Amit Kumar Sikder, Giuseppe Petracca, Hidayet Aksu, Trent Jaeger, and A Selcuk Uluagac. A survey on sensor-based threats to internet-of-things (iot) devices and applications. *arXiv preprint arXiv:1802.02041*, 2018.

[108] Willamos Silva. Cnn-pdm: A convolutional neural network framework for assets predictive maintenance. 2019.

[109] Murat Cihan Sorkun, ÖZLEM DURMAZ İNCEL, and Christophe Paoli. Time series forecasting on multivariate solar radiation data using deep learning (lstm). *Turkish Journal of Electrical Engineering & Computer Sciences*, 28(1):211–223, 2020.

[110] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 2019.

[111] Venkatachalam Subramanian, Selcuk Uluagac, Hasan Cam, and Raheem Beyah. Examining the characteristics and implications of sensor side channels. In *2013 IEEE International Conference on Communications (ICC)*, pages 2205–2210. IEEE, 2013.

[112] Rakesh Ranjan Swain and Pabitra Mohan Khilar. A fuzzy mlp approach for fault diagnosis in wireless sensor networks. In *2016 IEEE region 10 conference (TENCON)*, pages 3183–3188. IEEE, 2016.

[113] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[114] Qing Tao, Fang Liu, Yong Li, and Denis Sidorov. Air pollution forecasting using a deep learning model based on 1d convnets and bidirectional gru. *IEEE Access*, 7:76690–76698, 2019.

[115] Ahmed Tealab. Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal*, 3(2):334–340, 2018.

[116] Yan Tian, Kaili Zhang, Jianyuan Li, Xianxuan Lin, and Bailin Yang. Lstm-based traffic flow prediction with missing data. *Neurocomputing*, 318:297–305, 2018.

[117] Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. On the requirements for successful gps spoofing attacks. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 75–86. ACM, 2011.

[118] Irem Tumer and Anupa Bajwa. A survey of aircraft engine health monitoring systems. In *35th Joint Propulsion Conference and Exhibit*, page 2528, 1999.

[119] US Department of Homeland Security CISA Cyber + Infrastructure. Can bus network implementation in avionics, 2019. [Online; accessed 13-September-2019].

[120] Dmitry Vengertsev. Deep learning architecture for univariate time series forecasting. *Cs229*, pages 3–7, 2014.

[121] Denis Volkhonskiy, Ilia Nouretdinov, Alexander Gammerman, Vladimir Vovk, and Evgeny Burnaev. Inductive conformal martingales for change-point detection. *arXiv preprint arXiv:1706.03415*, 2017.

[122] Biao Wang, Yaguo Lei, Naipeng Li, and Tao Yan. Deep separable convolutional network for remaining useful life prediction of machinery. *Mechanical Systems and Signal Processing*, 134:106330, 2019.

[123] Dan Williams, Shuai Zheng, Xiangliang Zhang, and Hani Jamjoom. Tidewatch: Fingerprinting the cyclicality of big data workloads. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 2031–2039. IEEE, 2014.

[124] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1369–1378, 2017.

[125] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

[126] Peter T Yamak, Li Yujian, and Pius K Gadosey. A comparison between arima, lstm, and gru for time series forecasting. In *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, pages 49–55, 2019.

[127] Boyuan Yang, Ruonan Liu, and Enrico Zio. Remaining useful life prediction based on a double-convolutional neural network architecture. *IEEE Transactions on Industrial Electronics*, 66(12):9521–9530, 2019.

[128] Minghao Yin, Yongbing Zhang, Xiu Li, and Shiqi Wang. When deep fool meets deep prior: Adversarial attack on super-resolution network. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 1930–1938. ACM, 2018.

[129] Mei Yuan, Yuting Wu, and Li Lin. Fault diagnosis and remaining useful life estimation of aero engine using lstm neural network. In *2016 IEEE*

International Conference on Aircraft Utility Systems (AUS), pages 135–140. IEEE, 2016.

[130] Huan Zhang, Hongge Chen, Zhao Song, Duane Boning, Inderjit S Dhillon, and Cho-Jui Hsieh. The limitations of adversarial training and the blind-spot attack. *arXiv preprint arXiv:1901.04684*, 2019.

[131] Liangwei Zhang, Jing Lin, Bin Liu, Zhicong Zhang, Xiaohui Yan, and Muheng Wei. A review on deep learning applications in prognostics and health management. *IEEE Access*, 7:162415–162438, 2019.

[132] Xiangliang Zhang, Zon-Yin Shae, Shuai Zheng, and Hani Jamjoom. Virtual machine migration in an over-committed cloud. In *2012 IEEE Network Operations and Management Symposium*, pages 196–203. IEEE, 2012.

[133] Xinyun Zhang, Yan Dong, Long Wen, Fang Lu, and Wei Li. Remaining useful life estimation based on a new convolutional and recurrent neural network. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pages 317–322. IEEE, 2019.

[134] Yongzhi Zhang, Rui Xiong, Hongwen He, and Michael G Pecht. Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. *IEEE Transactions on Vehicular Technology*, 67(7):5695–5705, 2018.

[135] Rui Zhao, Dongzhe Wang, Ruqiang Yan, Kezhi Mao, Fei Shen, and Jinjiang Wang. Machine health monitoring using local feature-based gated recurrent unit networks. *IEEE Transactions on Industrial Electronics*, 65(2):1539–1548, 2017.

[136] Caifeng Zheng, Weirong Liu, Bin Chen, Dianzhu Gao, Yijun Cheng, Yingze Yang, Xiaoyong Zhang, Shuo Li, Zhiwu Huang, and Jun Peng. A data-driven approach for remaining useful life prediction of aircraft engines. In

2018 21st International Conference on Intelligent Transportation Systems (ITSC), pages 184–189. IEEE, 2018.

[137] Shuai Zheng, Xiao Cai, Chris Ding, Feiping Nie, and Heng Huang. A closed form solution to multi-view low-rank regression. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[138] Shuai Zheng and Chris Ding. Kernel alignment inspired linear discriminant analysis. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 401–416. Springer, 2014.

[139] Shuai Zheng, Feiping Nie, Chris Ding, and Heng Huang. A harmonic mean linear discriminant analysis for robust image classification. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 402–409. IEEE, 2016.

[140] Shuai Zheng, Kosta Ristovski, Ahmed Farahat, and Chetan Gupta. Long short-term memory network for remaining useful life estimation. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 88–95. IEEE, 2017.

[141] Shuai Zheng, Kosta Ristovski, Ahmed Farahat, and Chetan Gupta. Long short-term memory network for remaining useful life estimation. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 88–95. IEEE, 2017.

[142] Shuai Zheng, Zon-Yin Shae, Xiangliang Zhang, Hani Jamjoom, and Liana Fong. Analysis and modeling of social influence in high performance computing workloads. In *European Conference on Parallel Processing*, pages 193–204. Springer, 2011.

[143] Yunmin Zhu, Enbin Song, Jie Zhou, and Zhisheng You. Optimal dimensionality reduction of sensor data in multisensor estimation fusion. *IEEE Transactions on Signal Processing*, 53(5):1631–1639, 2005.

[144] Li Zhuang, Feng Zhou, and J Doug Tygar. Keyboard acoustic emanations revisited. *ACM Transactions on Information and System Security (TISSEC)*, 13(1):3, 2009.