Sensor-based Threats to Internet-of-Things and Machine Learning enabled Predictive Analytics

Gautam Raj Mode
Department of Electrical Engineering & Computer Science
University of Missouri, Columbia, MO, USA
gmwyc@mail.missouri.edu

Khaza Anuarul Hoque*
Department of Electrical Engineering & Computer Science
University of Missouri, Columbia, MO, USA
hoquek@missouri.edu

ABSTRACT

Predictive maintenance (PdM) is a modern solution which facilitates in predicting faults in a component powered by state-of-the-art machine learning algorithms. Particularly, PdM plays a prominent role in Aircraft Engine Health Monitoring (EHM) systems since early detection of defects in an aircraft engine can save valuable time, cost, and human lives. PdM uses IoT sensor data for predicting Remaining Useful Life (RUL) of an engine. However, IoT sensors are well-known for their susceptibility to cyber-attacks which possess a vital threat for aircraft PdM. In this work, we utilize the Long Short-Term Memory (LSTM), a deep learning approach for RUL prediction of an aircraft engine. Afterward, we apply the false data injection attack (FDIA), replay, and denial of service (DoS) attacks on aircraft engine sensor data and evaluate their impact on the LSTM-based RUL prediction. Our experimental results demonstrate that the cyber-attacks on IoT sensors strongly defects the RUL prediction. Furthermore, the results show that increasing the number of layers in LSTM also improves the resiliency of LSTM-based PdM against the FDIA attacks.

KEYWORDS

Predictive Maintenance, Aircraft engine maintenance, Long short-term memory, remaining useful life, denial-of-service attack, false-data injection attack, replay attack

ACM Reference format:
Gautam Raj Mode and Khaza Anuarul Hoque. 2019. Sensor-based Threats to Internet-of-Things and Machine Learning enabled Predictive Analytics. In *Proceedings of First International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things (AIChallengeIoT 2019), New York, USA, November 10–13 2019 (AIChallengeIoT'19),* 12 pages.
https://doi.org/00.0001

1 INTRODUCTION

Nowadays, due to the rapid development in the field of avionics, the aircraft engines have become increasingly sophisticated and complex, which means unexpected faults could result in consequences that range from flight delays to an accident that may cost millions to the airline, but most importantly loss of lives. Therefore, traditional maintenance strategies such as preventive maintenance [5] in which the equipment is regularly checked and cleaned, which may no longer fulfill the high demands of the modern aviation industry. Recently, predictive maintenance (PdM) [9, 28, 36] has been recognized as a better-suited strategy for aircraft engines as it considers wear and tear of the equipment, and most importantly it depends on the data from the equipment, removing the need for periodical checks [21].

As an ideal maintenance policy, PdM collects different parameters from an aircraft engine, detects minuscule changes in the functioning of an engine, and discovers faults, including when, where, and which type of failure that may occur by using machine learning algorithms [10]. With the fault information, PdM could arrange appropriate maintenance requests to the engine manufacturer even before the fault is encountered. This, in turn, helps in maximizing the service life of the engine without increasing the risk of failure. To design the best-suited PdM for an aircraft, a comprehensive understanding of Engine Health Monitoring (EHM) [15] systems is required. Figure 1 shows a generic EHM architecture. An EHM system has several IoT (Internet of Things) sensors connected to wireless network, which monitor different parameters of an aircraft engine, and sends out alerts to the engine manufacturer if the Remaining Useful Life (RUL) [29] of the engine is approaching its end of life. [1] An EHM system employs PdM systems to predict the RUL using the data collected from the IoT sensors. Unfortunately, IoT sensors are susceptible to cyber attacks [30], which possess a significant threat for the overall PdM system.

Sensor-based threats in IoT sensors have gained a lot of attention from researchers in academia in industry [1, 6, 26]. In [26], the author mentioned about different security and privacy challenges of Industrial Internet of Things (IoT) systems, whereas in [1] the authors classify different threat types, analyze and characterize intruders, and attacks on IoT devices and services. The work in [39] proposes a pairwise inconsistency based algorithm to enhance attack detection capability. In contrast, the work in [24] considered the problem of detection and identification of sensor attacks in the presence of transient faults when multiple sensors measure the same physical variable. Apart from IoT sensors, the attacks on machine learning algorithms, known as *adversarial machine learning* is also a very active research area [14, 17, 19, 23, 34]. These works discuss attacks on different stages of a machine learning pipeline which includes adversarial attacks on training and test data. Note, in these works noise is considered as the medium of generating adversarial examples. In contrast, the IoT sensors in a

*corresponding author

AIChallengeIoT'19, November 10–13 2019, New York, USA

ACM ISBN 12343…$15.00
https://doi.org/00.0001

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

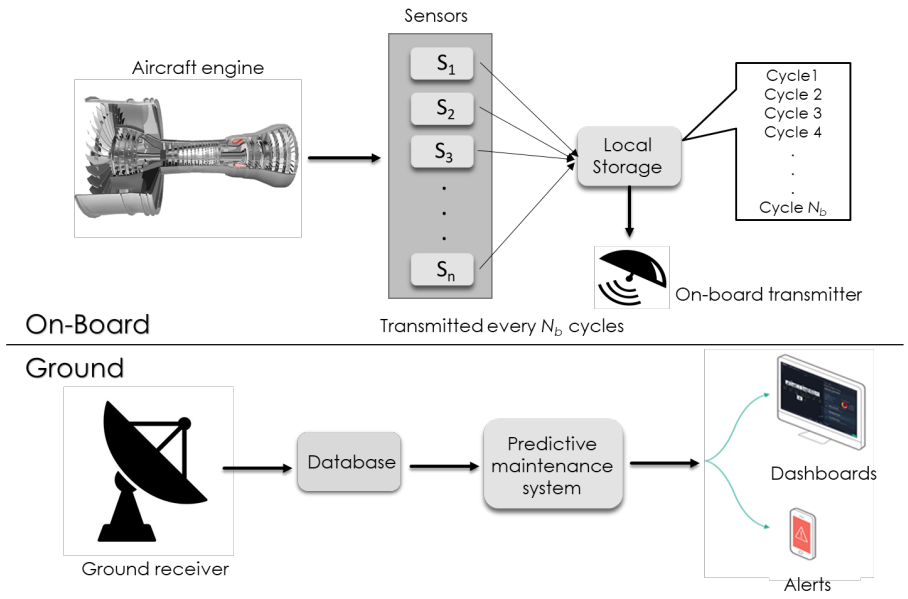


Figure 1: Engine health monitoring (EHM) system architecture

machine learning based PdM system are susceptible to different types of attacks such as Denial of Service (DoS), replay attack and False Data Injection Attack (FDIA), which formulates a novel *adversarial predictive maintenance (APdM)* problem. However, the effect of such attacks on a PdM system that utilizes machine learning algorithms is still underexplored which motivates this research. In the case of an aircraft engine, such attacks may lead to an incorrect prediction of RUL, resulting in the delay of timely maintenance leading to catastrophic consequences.

In this paper, we investigate the effects of cyber attacks on IoT sensors in a PdM system that relies on a machine learning algorithm. To be more specific, we apply the Long Short-Term Memory (LSTM), a deep learning approach for predicting the RUL in an aircraft PdM system. We perform the False Data Injection Attack (FDIA), replay, and Denial of Service (DoS) attacks on different engine sensors in the C-MAPSS [27] (Commercial Modular Aero-Propulsion System Simulation) data set [2] that is utilized by LSTM for RUL prediction. We evaluate these attacks on three different attack scenarios, analyze and measure their impacts on the RUL prediction. To the best of our knowledge, this is the first work that demonstrates the effects of cyber attacks on IoT sensors in a PdM system.

The rest of the paper is organized as follows. Section II briefly discusses the Engine Health Monitoring (EHM) systems and Predictive Maintenance (PdM). Section III introduces the network architectures of LSTM for the RUL prediction, and the NASA C-MAPSS data set used for our experiment. In Section IV, we present three different attack scenarios for the RUL prediction. Experiment results of those attack scenarios are described in Section V. Section VI presents the observations from those obtained results, and Section VII concludes the paper.

[2]a popular turbofan engine degradation data set published by NASA's Prognostics Center of Excellence (PCoE)

2 PRELIMINARIES

2.1 Engine health monitoring (EHM) system

An aircraft engine is a complex system, so it requires adequate monitoring to ensure safe operation and in-time maintenance [15]. 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 several parameters in an engine, Engine health monitoring (EHM) systems [33] have been in service for three decades. In an EHM system, IoT sensors are mounted inside and outside of an engine to monitor different parameters. All the IoT sensors are connected to a wireless network [3], which uses radio frequency for transmitting sensor output to central engine control [3]. Currently, modern engines are equipped with more than 5000 sensors and can significantly record and analyze large data. Considering Rolls-Royce engines, which operate 10 million flight hours every month, more than 5000 engines in use and each of them generates 1 Gigabyte of data per flight [22], then the whole system can generate hundreds of Gigabytes of data every day. Processing this amount of sensor data is a cumbersome task. There exist some work using dimensionality reduction [47], [44], [43], [42] and cloud computing [46], [37], [41] for processing large data. Deep learning can garner the benefits of big data as it uses GPU for parallel computing.

Figure 1 shows a general EHM architecture. The sensors onboard the engine send time series data (cycles) every hour to the local

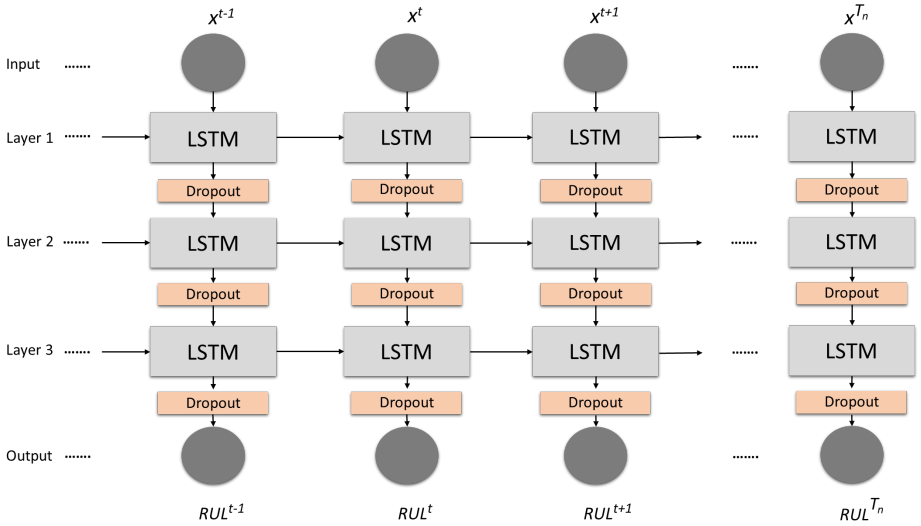


Figure 2: Deep LSTM model for RUL prediction

storage onboard 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.

2.2 Predictive maintenance (PdM)

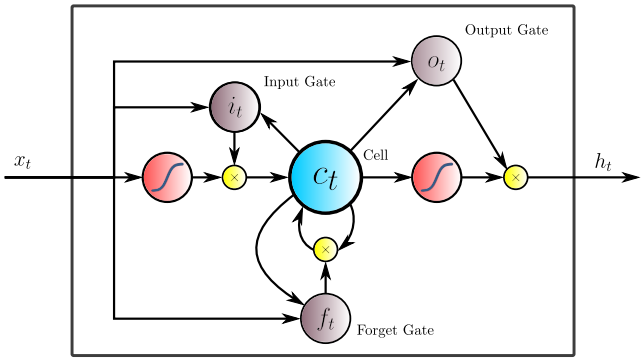
In manufacturing supply chains, unexpected failures are considered as primary operational risk as it can hinder productivity and can incur huge losses. For example, in modern automotive industry, an assembly line has several robots working on a car, and even if one of the robots fails, it will result in the total halt of the assembly line, causing loss of valuable production time and money. To overcome this problem, PdM strategies are employed. PdM is an industry 4.0 solution, which assists in predicting the future state of physical assets. It helps in better-informed maintenance decisions, to prevent unexpected delays.

PdM systems are employed in major industries like Nuclear power plants, aviation industry, automotive industry, and health care services. PdM allows for convenient scheduling of corrective maintenance parts for the equipment can be ordered beforehand to avoid last minute hassle, which saves a lot of valuable production time. PdM is well suited for making an informed decision when dealing with time-series data. A data-driven based model of PdM employ some the remarkable strategies like Random Forest algorithm [35], Artificial Neural Networks (ANN) [4], fuzzy models [31], Big data frameworks [7]. In this paper, an LSTM based deep learning approach is employed in predicting RUL of an aircraft engine.

3 LSTM NETWORK FOR RUL PREDICTION

In this section, the LSTM network for RUL prediction is presented, including the architecture of the network, the NASA C-MAPSS data set, and feature extraction from the data set. Evaluation of LSTM network has also been discussed.

Figure 3: A single memory block in LSTM



3.1 Long short-term memory model

Long Short-Term Memory Network [12] is a special kind of Recursive Neural Network (RNN), capable of learning long-term dependencies. LSTM is explicitly designed to avoid long term dependency problem, which is prevalent is RNN. It has achieved great praise in the field of machine learning and speech recognition. Some 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.

The LSTM contains special units called memory blocks in the recurrent hidden layer [13]. The memory blocks include memory cells with self-connections storing the temporal state of the network

in addition to special multiplicative units called gates to control the flow of information. Each memory block in the original architecture contains input gates, output gates, and forget gates. A single memory block is shown for clarity in Figure 3. The LSTM architecture solves the vanishing gradient problem, which is recurrent in other neural networks at small computational extra-cost.

Figure 2 shows deep LSTM architecture used in the system. The proposed LSTM architecture has three LSTM layers. A dropout [32] is used after the first, second and third LSTM layers, and L2 regularization to control model overfitting. The input is fed to the first LSTM layer, and the output of one layer is input for the next layer. RMSprop optimizer is utilized to train models. The training process is stopped when the error reaches the minimum.

Suppose that training data has N equipment (aircraft engines) 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. 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. As shown in Figure 2, at time t, LSTM network takes r-dimensional sensor data x_t and gives predicted RUL_t.

LSTM cell structure at time t is shown in Figure 3. Let the 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, and the LSTM cells in the network are implemented as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \quad (1)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \quad (2)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f), \quad (3)$$

$$a_t = tanh(W_c x_t + U_c h_{t-1} + b_c), \quad (4)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ a_t, \quad (5)$$

$$h_t = o_t \circ tanh(c_t), \quad (6)$$

σ is element wise logistic sigmoid function. $tanh$ is element-wise hyperbolic tangent function. The variable weights and bias to be computed during training of the network are $W_i, W_o, W_f, W_c \in \mathbb{R}^{q \times r}$, $U_i, U_o, U_f, U_c \in \mathbb{R}^{q \times r}$, $b_i, b_o, b_f, b_c \in \mathbb{R}^{q \times r}$. $\circ$ is element-wise multiplication of two vectors. The similar work can be found at [45]. As the architecture uses multiple layers and multiple nodes in each layer, it effectively discovers complex relationships within sensor data.

3.2 C-MAPSS data set

To evaluate the performance of the proposed LSTM network, 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 a different number of operating conditions and fault conditions. In this dataset, there are four sub-datasets with various operating conditions and fault conditions. Every subset has training data and test data, as shown in Table 1. 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.

Table 1: C-MAPSS Data Set

Data Set	FD001	FD002	FD003	FD004
Train trajectories	100	260	100	249
Test trajectories	100	259	100	248
Operating conditions	1	6	1	6
Fault conditions	1	1	2	2

Table 2 shows a list of all the sensors [3]. The time series data stops 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 failure based on a given incomplete data.

Feature generation. The C-MAPSS data set contains raw data of sensors and operating conditions, which needs to be processed before giving them as inputs to the LSTM network. In real-world applications, an engine generates raw data including several sensor parameters, loads, failure times, and operating conditions. Before the raw data is sent for PdM, the data of each sensor needs to be normalized. The data is normalized in the scale of [0, 1] or [−1, 1]. Selection of the target range depends on the nature of the data. For this data set, the features have been scaled in the scale of [0, 1].

3.3 Evaluation metric

To evaluate the performance of the LSTM model on the test data, we utilize the mean absolute error (MAE) metric. MAE is widely used as an evaluation metric in model evaluation studies. MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. Hence, MAE can be considered as a more natural measure of average error in many cases when compared to the Root Mean Square Error (RMSE) metric [8, 38]. MAE gives equal penalty weights to the model when the estimated RUL is either larger or smaller than the true RUL. Equation (7) represents the formula for MAE where y_i is true RUL whereas $\hat{y_i}$ is the predicted RUL.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y_i}| \quad (7)$$

4 CYBER-ATTACKS ON A PdM SYSTEM

An aircraft engine has different types of IoT sensors mounted on, or inside the engine, to sense various physical parameters (such as operation temperature, oil temperature, vibration, pressure, etc.)

[3] More details about these 21 sensors can be found in [25]

Table 2: Decryption 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

associated with the engine operation. These physical parameters can be measured using a network of sensors, and then transmitted to a central engine control through a wireless network [3]. Since these sensors are connected wirelessly, if an attacker can gain access to the communication network, they can directly manipulate the sensors measurements. As mentioned earlier, in this paper, we consider performing False Data Injection Attack (FDIA), replay, and Denial of Service (DoS) attacks on the PdM system.

4.1 False data injection attack (FDIA)

False Data Injection Attack (FDIA) [18] is capable of disrupting the RUL prediction process in a PdM system. A successful FDIA can cause the engine sensors to output erroneous values to the central engine control, and thus make 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 Equation (8).

$$Z_i = X_i + F_i \tag{8}$$

For an effective FDIA vector formulation, the adversary needs to have knowledge of the communication topology used by the sensors to communicate with central engine control, have physical access to the sensors, know whereabouts of the airplane and its maintenance history. In this paper, both *constrained* and *unconstrained* FDIAs are considered, which are further defined as follows:

1) *Constrained*: The attacker has access to a limited number of sensors, and some part of the communication network.
2) *Unconstrained*: The attacker has access to all of the sensors and also has total control of the communication network.

4.2 Replay attack

A replay attack is also known as a playback attack [20]. It is a category of network attack in which the attacker fraudulently repeats the data or has it delayed. The delay or repeat of the data transmission is carried out by the attacker, who intercepts the data and re-transmits it.

In this paper, we consider a replay attack that repeats the previously obtained data over a certain period of time T. For instance, if x_i is the original sensor output of i^{th} sensor and x_r is the captured sensor data of i^{th} sensor for the attack period from T_{start} to T_{end}, then the replay attack is formulated as follows:

$$r_i = \begin{cases} x_r & T_{start} < T < T_{end} \\ x_i & else \end{cases} \tag{9}$$

In a replay attack, the attacker can capture healthy sensor data after the engine has completed its maintenance and then replay the captured data when the degradation of the engine initiates. In order for a replay attack to be effective, the attacker needs to know the communication topology used by the sensors in the network, have physical access to the sensors, know the locations of the airplane and its maintenance history. In this paper, both *constrained* and *unconstrained* replay attack are considered. The properties of constrained and unconstrained attacks are similar to those mentioned in the FDIA attack.

4.3 Denial of service attack (DoS)

Denial of service attack (DoS), also known as interruption attacks, is mainly targeted against the communication network/infrastructure in a system. It can partially or entirely disrupt the data exchange between sensors and central engine control. The data exchange can be interrupted by either flooding the system with data packets or either dropping all the data packets [40].

In this paper, we consider a DoS attack that entirely disrupts the communication between sensors and central engine control for a period of T. For instance, if x_i is the original sensor output of i^{th} sensor, and d_i is the attacked senor output of i^{th} sensor for the attack period from T_{start} to T_{end}, then the DoS attack is formulated as follows:

$$d_i = \begin{cases} 0 & T_{start} < T < T_{end} \\ x_i & else \end{cases} \tag{10}$$

Note that, for a DoS attack to be effective, the intruder needs to know the communication topology used by the IoT sensors, have physical access to the sensors, have access to maintenance records of the airplane, and know the whereabouts of the airplane. In this paper, both *constrained* and *unconstrained* DoS attacks are considered. The properties of constrained and unconstrained attacks are similar to those mentioned in the FDIA attack.

4.4 Attack scenarios

As shown in Figure 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 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 data set 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 three different cyber-attacks on PdM systems, we consider three different attack scenarios.

Scenario 1

In this scenario, when the engine is called for maintenance, the adversary replaces the sensors in the engine with malicious ones. These malicious sensors are programmed to initiate the attacks after N^d_{avg}. The attacks performed are *unconstrained* as access to the sensors is not restricted. To perform an effective cyber-attack, the adversary shall meet all of the following requirements:

- Have access to maintenance records of the airplane.
- Have knowledge of whereabouts of the airplane, that includes flight schedules, and a number of flight hours after maintenance.
- Have access to the physical sensors of the engine.

Scenario 2

This scenario is very similar to scenario 1, but the only difference is that the attacks are initiated from the beginning time cycles of the engine. In this scenario, the adversary can perform *unconstrained* attacks as he has access to all of the sensors, making the attack more catastrophic. To perform an effective cyber-attack, the adversary shall meet all of the requirements mentioned in scenario 1.

Scenario 3

In this scenario, the attackers are on board the airplane, they have intruded the communication network, which is used by IoT sensors to send data to central engine control. The attackers have limited access to the communication network and are only able to manipulate a certain group of sensors for the flight duration. In this scenario, the attacks are *constrained* as the adversary has limited control on the attack. To perform an effective cyber-attack, the adversary shall meet all of the following requirements:

- Have knowledge of the aircraft's flight hours after maintenance.
- Have knowledge of the communication topology used in the aircraft.
- Should possess devices capable of intruding the communication network.

5 EXPERIMENTAL RESULTS

Section 5.1 compares different LSTM network architectures. In Section 5.2, the impact of cyber-attacks on the PdM system has been presented, which includes the study of different attack signatures, impact of cyber-attacks on the RUL prediction, and piece-wise RUL prediction.

5.1 Comparison of LSTM network architectures

In order to select the best network architecture of LSTM for PdM, we compare three different architectures of LSTM for the FD001 data set. Figure 4 represents the comparison between LSTM with architectures L(100,50), lh(70), L(100, 100), lh(60) and L(100, 100, 100), lh(80). The notation L(100,100,100) lh(80) refers to network that has 100 nodes in the hidden layers of the first LSTM layer, 100 nodes in the hidden layers of the second LSTM layer, 100 nodes in the hidden layers of the third LSTM layer, and a sequence length of 80. At the end, there is a 1-dimensional output layer.

Table 3: MAE comparison for different network architectures

Network architectures	FD001
	MAE
L(100,50), lh(70)	12.21
L(100,100), lh(60)	11.56
L(100,100,100), lh(80)	8.89

LSTMs are stochastic, which means that their outputs can be different corresponding to the same given inputs. Hence, we run each LSTM network architecture in table 3 for five times, and the best structure is chosen based on the least MAE value for each architecture.

In Figure 4, we show the engine IDs on x-axis and their respective true RUL and predicted RUL on y-axis. From Figure 4 and Table 3 it is evident that the architecture L(100, 100, 100) with a sequence length equal to 80 has the least MAE of 8.89. That MAE value is for test data set including data of 100 engines. Hence, we choose this network for modeling the attacks on the PdM system. It can be observed that both the sequence length and the number of LSTM layers are important parameters in achieving accurate RUL.

5.2 Cyber-attacks on the PdM system

In this section, the impact of different cyber-attacks on RUL prediction of the LSTM network has been presented.

5.2.1 Attack model setup

The average degradation point of the engine N^d_{avg} is considered as 130 for the FD001 data set [11] [2] [45], and aircraft sends 20 time cycles (N_b) of data to the ground at a time. In scenario 3, the attacker has initiated the cyber-attacks after N^d_{avg}, which is 130 time cycles, and the attack duration is 20 hours, which is 20 time cycles. So each engine in the test data set should have at least 150 time cycles of data. Hence, we remove the data of the engines that

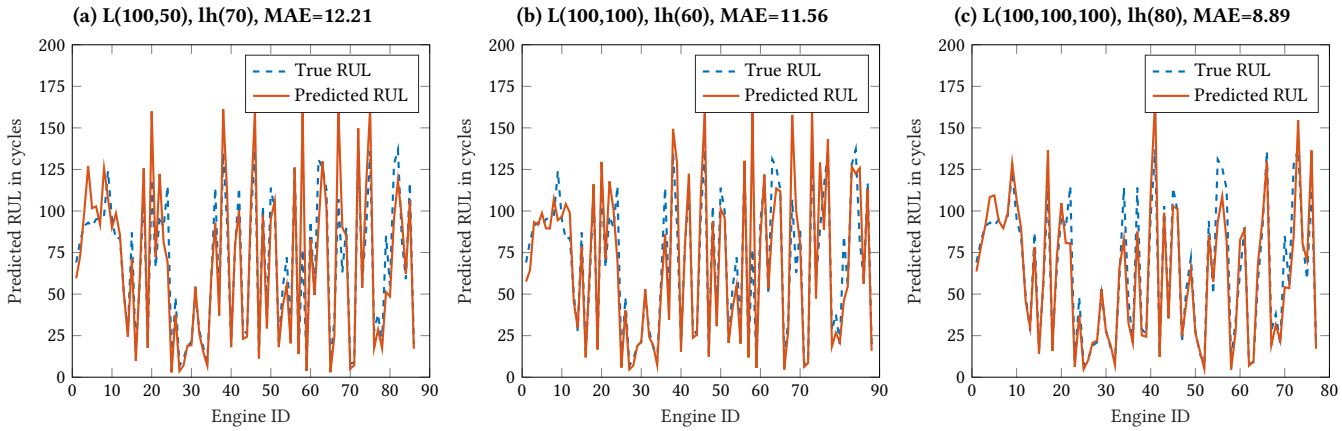


Figure 4: Comparison of LSTM architectures

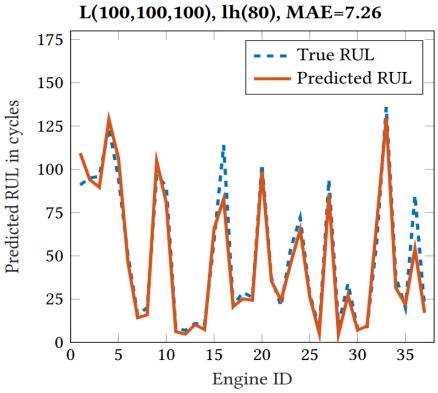


Figure 5: RUL prediction on attack model test data set

have less than 150 time cycles from the FD001 test data set, which gives us test data for 37 engines.

Figure 5 shows RUL prediction on the new data set, which has data of 37 engines, and the MAE of the new test data set is found to be 7.26. The new test data set will be used in further experiments. In train and test data set, 7 out of 21 sensors can be ignored as their values after normalization remain constant. Adversarial attacks can be performed on the remaining 14 sensors.

5.2.2 Attack signatures

In this section, we present the attack signatures of FDIA, replay and DoS attacks as described in the section 4.

FDIA: To model the FDIA attack on sensors, we add a vicious vector to the original vector modifies the sensor output by a very small margin (0.2% to 0.3%) in both constrained and unconstrained attacks. Figure 6(a) and 6(d) shows the comparison between the original and FDIA attacked output signal of sensor 2 for engine ID 3 for both constrained and unconstrained attacks, and they are described as follows.

a. *Constrained*: In this attack, the sensor output from time cycles 130 to 150 are attacked. As shown in Figure 6(a), the attack signature is very similar to the original signal. Note, in the constrained attack the adversary has limited access to sensors. Hence, it has lesser impact on the PdM system in comparison to the the unconstrained attack.

b. *Unconstrained*: In this kind of attack, the sensor output from time cycles 130 to 303 are attacked. As shown in Figure 6(d), the attack signature is also very similar to the original signal, making it harder to detect. As the adversary has access to all of the sensors, it makes the attack even more effective.

Replay attack: This attack can be catastrophic as healthy data is replayed in place of degrading data, making it harder to decipher. Figure 6(b) and 6(e) represents the comparison between the original and the replay attacked output signal of sensor 2 for engine ID 3 for both constrained and unconstrained attacks, and they are described as follows.

a. *Constrained*: In this attack, the sensor output from time cycles 130 to 150 are attacked. The attacker captures healthy sensor data from initial flights and replays the same data after 130 time cycles of data.

b. *Unconstrained*: In this kind of attack, the sensor output from time cycles 130 to 303 are attacked. The malicious sensors on board the engine, capture healthy data from time cycles 1 to 130 and replay the same data after 130 time cycles. The unconstrained attack is more effective as the adversary has access to all of the sensors.

DoS attack: As the communication network is attacked using a DoS attack, the data packets are dropped, making the sensor output to zero. Figure 6(c) and 6(e) represents the comparison between the original and DoS attacked output of sensor 2 for engine ID 3 for both constrained and unconstrained attacks, and they are described as follows.

a. *Constrained*: In this attack, the sensor output from time cycles 130 to 150 are attacked. As the sensor output falls to zero, it has a significant impact on the PdM system.

b. *Unconstrained*: In this type of attack, the sensor output from time cycles 130 to 303 are attacked. The intruders can access to all sensors in the communication channel so that they can drop all the packets sent from the sensors to the central engine control, and this could result in erroneous results from the PdM system.

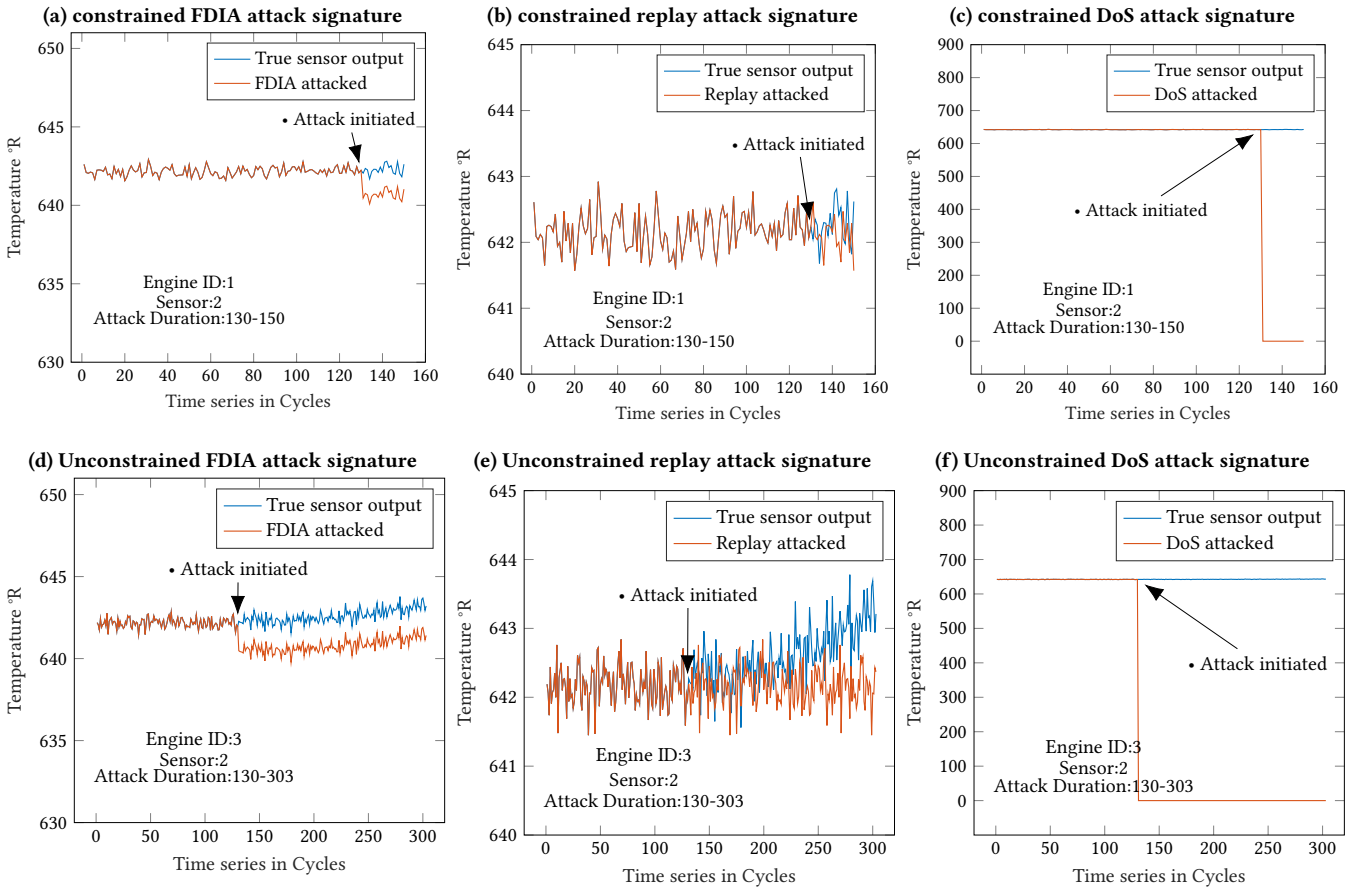


Figure 6: Attack signatures

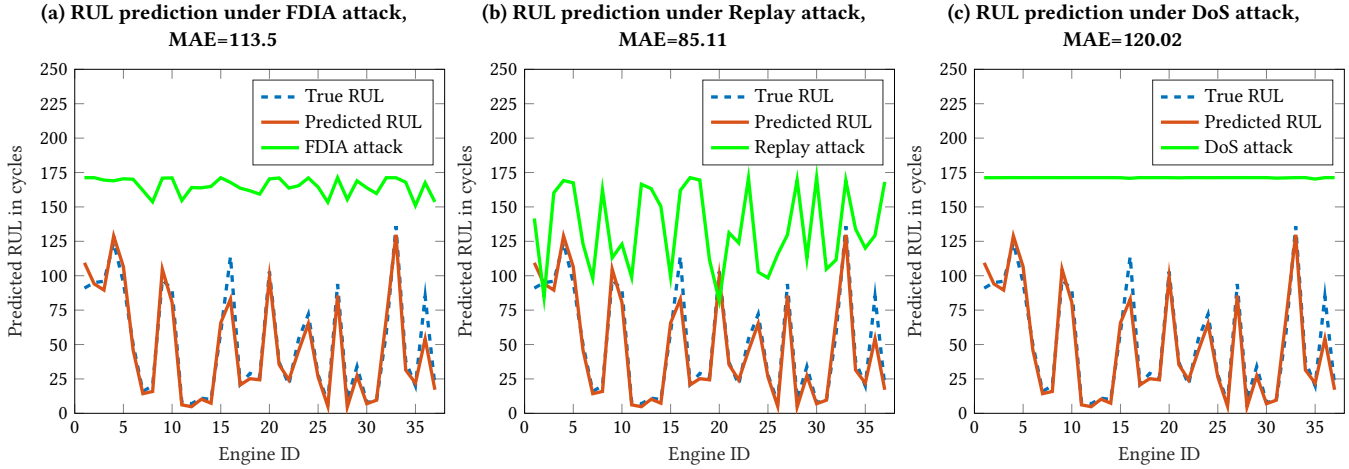


Figure 7: Attack scenario 1

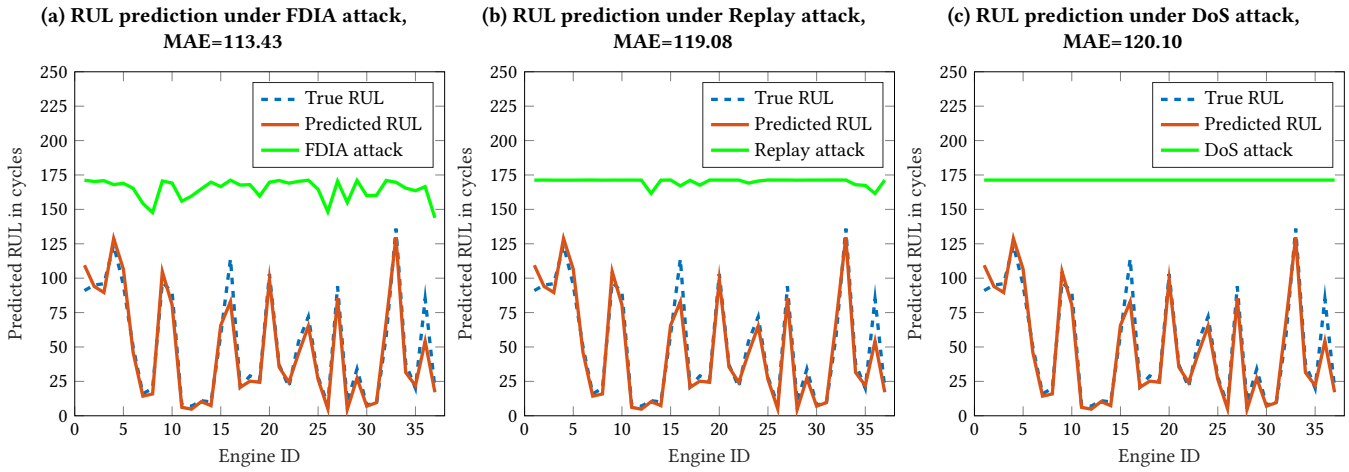


Figure 8: Attack scenario 2

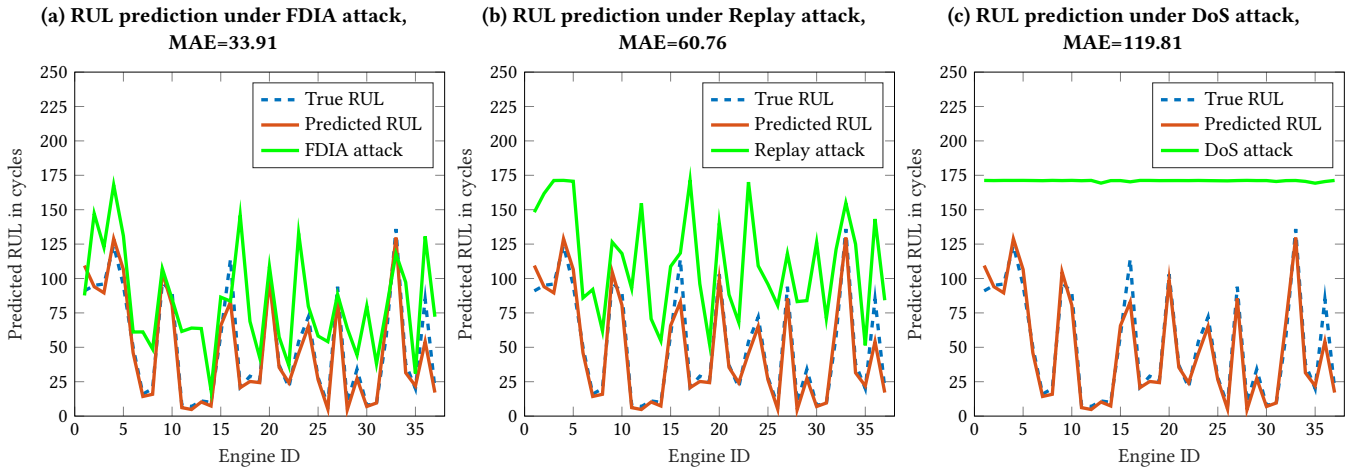


Figure 9: Attack scenario 3

5.2.3 Impact analysis of different attacks on the aircraft PdM system

To show the impact of FDIA, replay, and DoS attacks on the aircraft PdM system, we implement the three cyber-attacks for different scenarios mentioned previously in Section 4.4. Table 4 shows a number of sensors attacked for each scenario.

Table 4: Number of sensors attacked per scenario

Scenario	No. of sensors attacked out of 14		
	FDIA	Replay	DoS
1	7	14	7
2	7	14	7
3	3	4	3

Scenario 1: In this scenario, the adversary performs attacks after 130 (N^d_{avg}) cycles to end of data. The attacker has access to all of the sensors in the engine. For replay attack, the malicious sensors capture initial 130 (N^d_{avg}) time cycles of data (healthy sensor data), and replays the captured data for the attack duration. Figure 7 represents the effected RUL prediction for three different attacks. From Figures 7(a), 7(b), and 7(c), we observe that the attacks in scenario 1 has more impact on RUL prediction.

Scenario 2: In this scenario, the adversary performs attacks from beginning time cycles of data. The attacker has access to all of the sensors in the engine. For replay attack, the malicious sensors capture initial 20 time cycles of data (healthy sensor data), and replays the captured data after 20 time cycles to the rest of the attack duration. Figures 8(a), 8(b), and 8(c) represents the effected RUL prediction for three different attacks. In Figures 7 and 8, it can

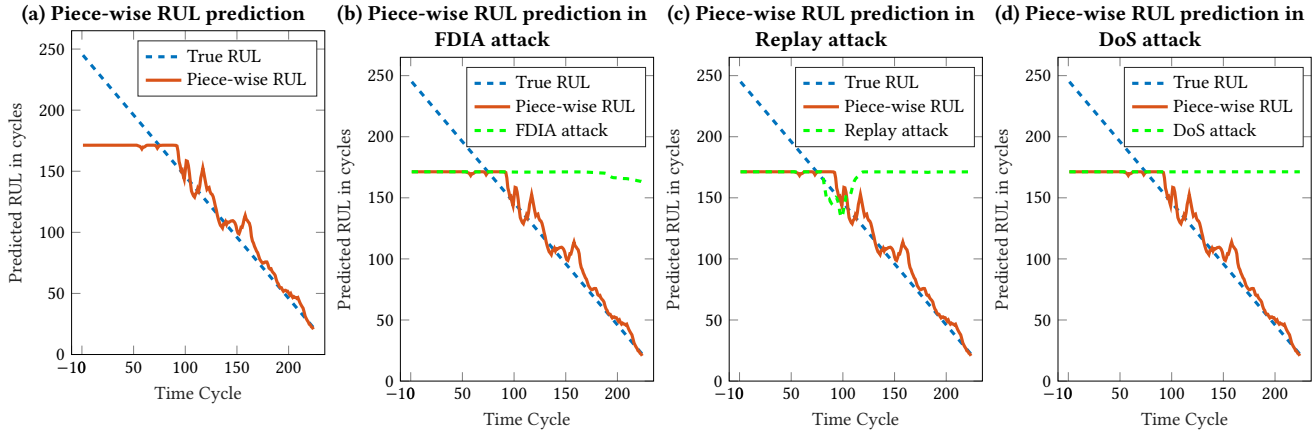


Figure 10: Piece-wise RUL prediction

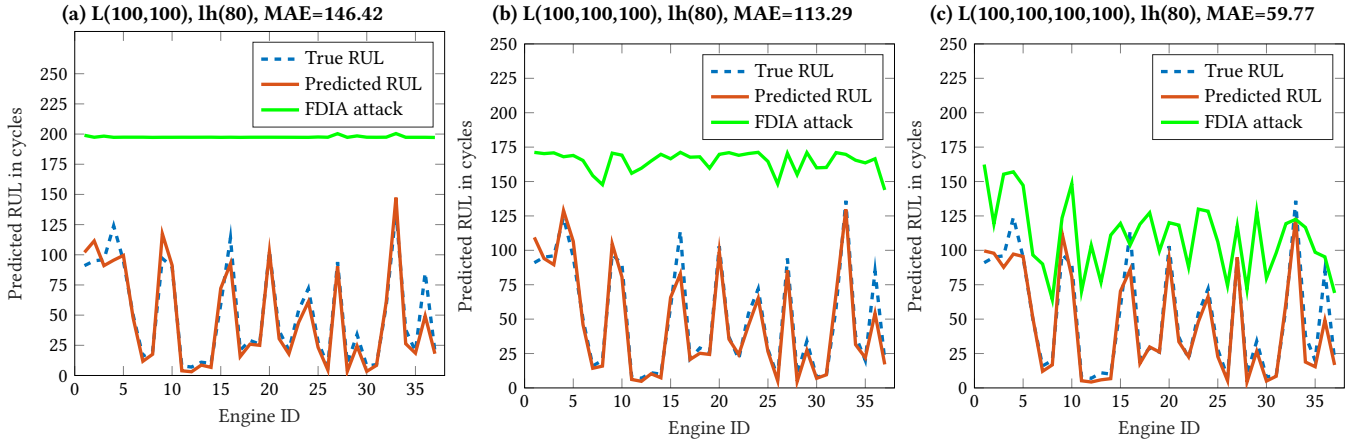


Figure 11: Comparison of multi-layer LSTM networks under FDIA attack of scenario 2

be observed that the scenario 1 and 2 have greater impact on the RUL prediction.

Scenario 3: In this scenario, the adversary performs attacks between 130 and 150 time cycles. The attacker has limited access to the sensors in the engine. For replay attack, the malicious sensors capture initial 20 time cycles of data (healthy sensor data), and replays the captured data for the attack duration. Figures 9(a), 9(b),and 9(c) represents the effected RUL prediction for three different attacks. When compared to scenarios 1 and 2, we observe that the scenario 3 has less effect on RUL prediction. In order to make this scenario more effective, the adversary has to take multiple flights to attacks more time cycles of data.

5.2.4 Piece-wise RUL prediction

The cyber-attacks evaluated in scenarios 1, 2, and 3 take into account the attacks performed on all of the engine data in the C-MAPSS test data set. To show the impact of cyber-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.

Figure 10(a) shows an example of an engine data from the data set of 100 engines, and depicts the predicted RUL using LSTM at each time step of that engine data. For example, if X is the time series data of a particular engine, then $X_i = [x_1, x_2, x_3 \ldots x_{t-k}]$ 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 $RUL_i^p = [RUL_1^p, RUL_2^p, RUL_3^p \ldots RUL_{t-k}^p]$. From Figure 10(a), it is evident that as the time series approaches the end of life, the predicted RUL is close to the true RUL, because the LSTM model has more time series data to accurately predict the RUL. Figure 10(b), 10(c), and 10(d) shows three different attacks of scenario 1 on the aircraft PdM system. The attacks are performed on an engine with 242 time cycles of test data.

For FDIA and DoS attacks shown in Figure 10(b) and Figure 10(d), we observe that the predicted RUL during the attack remains constant at 170 for the attack duration. In contrast, for the replay

Table 5: MAE comparison for attacks scenarios

Scenario	MAE			
	True	FDIA	Replay	DoS
1	7.26	113.5	85.11	120.2
2		113.43	119.08	120.10
3		33.91	60.76	119.81

attack from as shown in Figure 10(c), we observe that the predicted RUL follows the trend of piece-wise RUL until 130 time cycles and goes back to the initial RUL as healthy data is replayed after 130 time cycles.

Table 6: MAE comparison of multi-layer LSTM networks for attack scenario 2

Network	MAE			
	True	FDIA	Replay	DoS
L(100,100), lh(80)	7.65	146.42	129.55	120.66
L(100,100,100), lh(80)	7.26	113.29	119.08	120.10
L(100,100,100,100), lh(80)	6.21	59.77	117.11	119.77

6 DISCUSSION

In this work, we first evaluate the deep learning approach in predicting aircraft engine RUL, and obtained results show a great prospect for deep learning in PdM. It is observed that sequence length and network architecture are crucial in predicting accurate RUL. An LSTM network with smaller sequence length or inadequate LSTM layers results in the network having larger MAE.

The impact analysis of different cyber attacks on aircraft sensors in the C-MAPSS dataset provides some interesting insights. In the case of FDIA and replay attacks, we observe that the constrained attacks have less impact on PdM system in comparison with unconstrained attacks. In the event of DoS attack, both constrained and unconstrained had a similar influence on the PdM system. Note that the attack signature of FDIA is very close to the original sensor output. However, it has a significant impact on the aircraft RUL prediction. An FDIA attack on the PdM system may result in an increase or decrease in the predicted RUL of the aircraft engine. The decrease of the predicted RUL does not have a significant impact on the airline, whereas the increase in RUL will have a significant impact, which may result in substantial losses to the airline and most importantly the loss of life. A replay attack, when executed efficiently will result in an incorrect prediction of RUL, and most importantly it conveys that the engine has not reached its end of life, which may result in unexpected delays and untimely maintenance.

Table 5 shows the outcome of different attacks on the LSTM network. In the table, the *True* column represents MAE of the LSTM architecture without any attack, and the rest of the columns are MAE of the network after different attacks. It can be observed that DoS attack in all of the scenarios has a significant impact on the PdM system, with MAE ranging from 119 to 121. Whereas FDIA attack in scenario 3 has a minimal impact (MAE of 33.91). Note, even though the impact is minimum, the MAE is 4 times greater than the true MAE (7.26), making it disastrous for the PdM system.

Figure 11 compares multi-layer LSTM networks under FDIA attack in scenario 2. All of the networks in this figure have similar nodes in the hidden layers of each LSTM layer, and similar sequence length. It can be observed that as the number of LSTM layers increases the network becomes more resilient to the FDIA attack. As shown in Table 6, we also observe that for the FDIA attack, the 4-layer LSTM has MAE of 59.77, which is half the MAE of 3-layer LSTM. Even though, increasing the LSTM layers results in the network becoming more resilient to the FDIA attack, overdoing it, might result in overfitting [16], which is not desirable. In the case of replay and DoS attacks from Table 6, the MAE of the networks did not show a considerable change with the increase of LSTM layers.

All of these obtained results show that machine learning-based PdM systems have a great prospect for aircraft maintenance. However, our results indicate that they are very susceptible to cyber attacks. Hence, while designing PdM systems, the designer must consider incorporating the ability to make the systems less vulnerable and more resilient to different kinds of cyber attacks, such as those investigated in this paper.

7 CONCLUSIONS AND FUTURE WORKS

This paper presents an LSTM approach for predicting Remaining Useful Life (RUL) of an aircraft engine and explores the impacts of cyber attacks on such systems. We model three different attacks (DoS, FDIA, and replay attack) on the IoT sensors that provide data to PdM system and evaluate their impacts on RUL prediction. The obtained results show that DoS attack is the most lethal attack for the aircraft PdM system. Besides, the FDIA and replay also have a substantial impact on the RUL prediction. We also show that the number of layers in LSTM has a direct relationship with the FDIA attack, e.g., increasing the number of layers in LSTM also enhances the resiliency against the FDIA attack. In the future, we plan to develop an end-to-end methodology for detecting and mitigating cyber attacks in a PdM system.

REFERENCES

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

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

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

[4] Pramod Bangalore and Lina Bertling Tjernberg. 2013. An approach for self evolving neural network based algorithm for fault prognosis in wind turbine. In *2013 IEEE Grenoble Conference*. IEEE, 1–6.

[5] Richard Barlow and Larry Hunter. 1960. Optimum preventive maintenance policies. *Operations research* 8, 1 (1960), 90–100.

[6] Chakib Bekara. 2014. Security issues and challenges for the IoT-based smart grid. *Procedia Computer Science* 34 (2014), 532–537.

[7] Mikel Canizo, Enrique Onieva, Angel Conde, Santiago Charramendieta, and Salvador Trujillo. 2017. Real-time predictive maintenance for wind turbines using Big Data frameworks. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, 70–77.

[8] Tianfeng Chai and Roland R Draxler. 2014. Root mean square error (RMSE) or mean absolute error (MAE)? *Geoscientific Model Development Discussions* 7 (2014), 1525–1534.

[9] Federico Civerchia, Stefano Bocchino, Claudio Salvadori, Enrico Rossi, Luca Maggiani, and Matteo Petracca. 2017. Industrial Internet of Things monitoring solution for advanced predictive maintenance applications. *Journal of Industrial Information Integration* 7 (2017), 4–12.

[10] Matthias Auf der Mauer, Tristan Behrens, Mahdi Derakhshanmanesh, Christopher Hansen, and Stefan Muderack. 2019. Applying Sound-Based Analysis at Porsche Production: Towards Predictive Maintenance of Production Machines Using Deep Learning and Internet-of-Things Technology. In *Digitalization Cases*. Springer, 79–97.

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

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

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

[14] Ling Huang, Anthony D Joseph, Blaine Nelson, Benjamin IP Rubinstein, and JD Tygar. 2011. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*. ACM, 43–58.

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

[16] Saurabh Karsoliya. 2012. Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture. *International Journal of Engineering Trends and Technology* 3, 6 (2012), 714–717.

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

[18] Kebina Manandhar, Xiaojun Cao, Fei Hu, and Yao Liu. 2014. 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 (2014), 370–379.

[19] Patrick McDaniel, Nicolas Papernot, and Z Berkay Celik. 2016. Machine learning in adversarial settings. *IEEE Security & Privacy* 14, 3 (2016), 68–72.

[20] Yilin Mo and Bruno Sinopoli. 2009. Secure control against replay attacks. In *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 911–918.

[21] Toshio Nakagawa. 1984. Periodic inspection policy with preventive maintenance. *Naval Research Logistics Quarterly* 31, 1 (1984), 33–40.

[22] M Ong, Xiaoxu Ren, Jeff Allan, H.A. Thompson, and Peter Fleming. 2003. Future trends in aircraft engine monitoring. 8/1– 8/7. https://doi.org/10.1049/ic:20030008

[23] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ACM, 506–519.

[24] Junkil Park, Radoslav Ivanov, James Weimer, Miroslav Pajic, and Insup Lee. 2015. Sensor attack detection in the presence of transient faults. In *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*. ACM, 1–10.

[25] Emmanuel Ramasso and Abhinav Saxena. 2014. Performance Benchmarking and Analysis of Prognostic Methods for CMAPSS Datasets. *International Journal of Prognostics and Health Management* 5, 2 (2014), 1–15.

[26] Ahmad-Reza Sadeghi, Christian Wachsmann, and Michael Waidner. 2015. Security and privacy challenges in industrial internet of things. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE, 1–6.

[27] Abhinav Saxena and Kai Goebel. 2008. C-MAPSS data set. *NASA Ames Prognostics Data Repository* (2008).

[28] Sule Selcuk. 2017. Predictive maintenance, its implementation and latest trends. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 231, 9 (2017), 1670–1679.

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

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

[31] Silvio Simani, Saverio Farsoni, and Paolo Castaldi. 2014. Fault tolerant control of an offshore wind turbine model via identified fuzzy prototypes. In *2014 UKACC International Conference on Control (CONTROL)*. IEEE, 486–491.

[32] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

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

[34] JD Tygar. 2011. Adversarial machine learning. *IEEE Internet Computing* 15, 5 (2011), 4–6.

[35] Anoop Prakash Verma. 2012. Performance monitoring of wind turbines: a data-mining approach. (2012).

[36] Yupeng Wei, Dazhong Wu, and Janis Terpenny. 2018. Predictive maintenance for aircraft engines using data fusion. In *2018 Institute of Industrial and Systems Engineers Annual Conference and Expo, IISE 2018*.

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

[38] Cort J Willmott and Kenji Matsuura. 2005. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate research* 30, 1 (2005), 79–82.

[39] Kang Yang, Rui Wang, Yu Jiang, Houbing Song, Chenxia Luo, Yong Guan, Xiaojuan Li, and Zhiping Shi. 2018. Sensor attack detection using history based pairwise inconsistency. *Future Generation Computer Systems* 86 (2018), 392–402.

[40] Ping Yi, Ting Zhu, Qingquan Zhang, Yue Wu, and Li Pan. 2016. Puppet attack: A denial of service attack in advanced metering infrastructure network. *Journal of Network and Computer Applications* 59 (2016), 325–332.

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

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

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

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

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

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

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