PROGNOS: An Automatic Remaining Useful Life (RUL) Prediction Model for Military Systems Using Machine Learning

Onur Sürücü*[a], Connor Wilkinson[b], Uygar Yeprem[c], Waleed Hilal[b], S. Andrew Gadsden[b], John Yawney[a], Naseem Alsadi[b], Alessandro Giuliano[b]

[a]Adastra Corporation, 200 Bay St. Suite, Toronto, ON, Canada; [b]McMaster University, 1280 Main St. W, Hamilton, Hamilton, ON Canada; [c]University of Guelph, 50 Stone Rd E, Guelph, ON Canada

ABSTRACT

In modern industrial settings, the quality of maintenance efforts directly influence equipment's operational uptime and efficiency. Condition monitoring is a common process employed for predicting the health of a technical asset, whereby a predictive maintenance strategy can be adopted to minimize machine downtime and potential losses. Throughout the field, machine learning (ML) methods have become noteworthy for predicting failures before they occur, thereby preventing significant financial costs and providing a safer workplace environment. These benefits from predictive maintenance techniques, are particularly useful in the context of military equipment. Such equipment is often significantly expensive, and untimely machine failure could result in significant human endangerment. In this paper, a prognostic model (PROGNOS) is proposed to predict military equipment's remaining useful life (RUL) based on their monitoring signals. The main considerations of PROGNOS are expectation maximization tuned Kalman Filter (EM-KF) for signal filtering, a recently introduced feature extraction algorithm (PCA-mRMR-VIF), and predictive LSTM model with an adaptive sliding window. The viability and performance of the proposed model were tested on a highly complex competition dataset: the NASA aircraft gas turbine engine degradation dataset, wherein readings from multiple sensor channels were recorded for degrading machines. According to testing results, we can confidently say that the proposed PROGNOS model was viable and robust overall, proving its general usefulness on all military equipment that emit signals.

Keywords: predictive maintenance, condition monitoring, remaining useful life, NASA gas turbine engine, military equipment, feature engineering, artificial intelligence, machine learning

1. INTRODUCTION

Production plants are expected to run 24 hours a day to meet market demand. Unexpected equipment breakdowns may result in tremendous economic stresses through significant process downtimes. Most companies require these interruptions to be anticipated in advance to take the necessary precautions before stoppages occur unexpectedly. A non-intrusive procedure for tracking and detecting potential faults in systems is obligatory for all industrial assets. The record shows that machines may fail for diverse reasons depending on the frequency of maintenance. The manufacturing industry has reported a considerable increase in the frequency of accidents due to poor and dangerous maintenance practices [1]. Every year, industrial expenses in the U.S. reach up to $200 billion on maintaining plant equipment and facilities, while poor maintenance causes losses of up to $60 billion [2].

Predictive maintenance (PdM) or condition-based monitoring is an advanced diagnostic technique to reveal the operating machinery faults in their incipient phase before any breakdowns occur, and the proper maintenance can be identified by monitoring the equipment's diagnostic data. In order to quantify a machinery's health state, remaining useful life (RUL) value is estimated by exclusively monitoring the machinery's emitted signals.

Machine Learning (ML) is a branch of artificial intelligence specialized in building algorithms that learn from data and continuously improve its performance over time without requiring an human intervention [3]. ML models have provided many advantages for many fields, including stoppage reduction, maintenance cost reduction, spare-part life increases, operator safety, increased production, repair verification, an increase in overall fit, and many more [4]. To construct a map between the acquired input signals and a fault diagnosis as an output, ML methods typically require four main stages, as illustrated in figure 1 [5], [6]. *Data Acquisition* is the first stage of diagnosing any machinery. This first step converts, amplifies, corrects the measurement acquired from multiple sensors, and finally stores them in a computer. Sensors are designed to convert the physical environmental inputs into electrical signals. Physical characteristics of the industrial asset are acquired utilizing sensors installed on the equipment. Followingly, *data processing* is utilized to achieve

Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications IV,
edited by Tien Pham, Latasha Solomon, Proc. of SPIE Vol. 12113, 121130R
 · 0277-786X · doi: 10.1117/12.2618913

higher test accuracy and faster training time, data processing is a vital procedure for prognosis purposes, where it is responsible for carrying out operations on data to translate them into useful shape [7]. The accuracy of the decision algorithm is highly dependent on the quality of the dataset. Feature extraction and selection methods are standard approaches to achieve the optimum dataset from the raw signals. Feature extraction is responsible for highlighting degradation indicators, handling missing values, and correcting irregularities in the acquired data. Feature selection reduces the extracted features by removing the input vectors' redundant attributes. Thus, the learning algorithm can provide a diagnostic result as an outcome without being exposed to any deceptive or false data. *RUL estimation* is the last process, where a ML model predicts the machinery's well-being by monitoring its emitted signals.

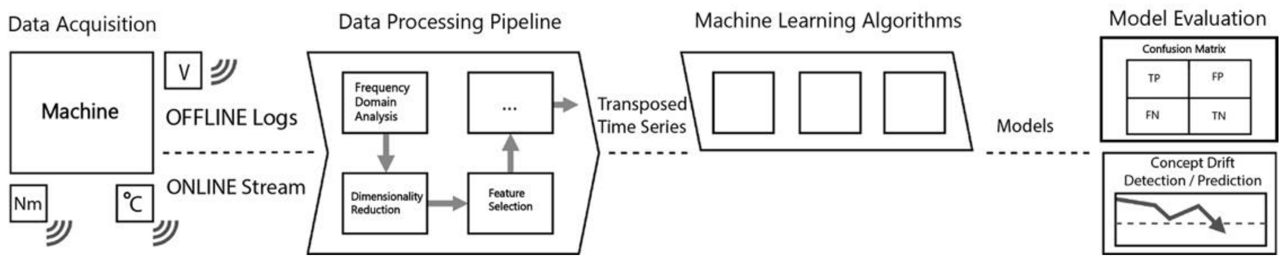

Figure 1. An example of ML-driven condition monitoring procedure from raw data to model's output.

Condition monitoring principles and techniques have been widely adopted in the arms industry. With fast and continuous improvements in military equipment technology, the tools that monitor and maintain this equipment are also expected to improve and evolve. Machine learning and data-driven techniques have thus become attractive options to automatize the process of condition monitoring. An example from the literature to the combination of machine learning and condition monitoring is a study that applied these principles to detect physical impairments in Unmanned Aerial Vehicle (UAV) rotor blades [8].The convenience of data-driven techniques is that they do not require a mathematical model of a system as opposed to many of the existing methods. In [8], finding out whether a rotor has degradation is simply a matter of observing the onboard sensor (IMU) measurements. To enrich the feature space, once signals are received, they are analyzed using Fast Fourier Transform (FFT), Wavelet Packet Decomposition (WPD) and by measuring signal power. The processed signals are then inputted to a Support Vector Machine (SVM) classifier to predict whether a fault has occurred. Another study [9] has implemented a Convolutional Neural Network (CNN) to monitor the states of supersonic combustor by taking in raw pressure data. The model was successful in classifying the combustion process into four modes. Another [10] paper has analyzed the Exhaust Gas Temperature data of an aircraft, to identify whether an anomaly has occurred, using a Relevance Vector Machine (RVM).

2. PROGNOS: THE PROPOSED MODEL

2.1 Overview of the proposed model

In this technical paper, we propose PROGNOS: a fully automatic ML-driven model that estimates the RUL of any machinery that emits capturable signals. Due to our search results, PROGNOS is the only model that is fully automatic and suitable to all signal emitting military equipment. In order to test the feasibility and viability of the proposed model, we utilized a well-known complex dataset: NASA aircraft gas turbine engine degradation dataset [11]. We chose this turbofan degradation dataset as it is a predominantly applied scenario within the prognostic community. It is a competition dataset, and no prior information was provided about the specific engines or the acquired signals. Hence, the lack of information creates an ideal scenario for validating the generalizability of the model. In this way, we realized PROGNOS's feasibility and viability by analyzing its performance on this sophisticated and accepted dataset.

The proposed model consists of two primary steps: feature engineering (feature decomposition and selection) and a machine learning model (RUL predictor), as shown in Figure 2. Filtering aims to remove unwanted components from the contaminated signal. Feature selection is a process of eliminating redundant input vectors to reduce the computational complexity and improve the model's performance. Feature extraction's objective is decomposing underlying characteristics of the given signal that indicate performance degradation of the technical asset. Therefore, this series of processing methods allow learning algorithms to pick up performance degradation indications, which ultimately increases the model's success rate. As the last layer of our proposed model, an LSTM model is trained to predict the RUL value of the given military equipment.

The process flow of the proposed model is illustrated in figure 2. The filtering process is carried out using a Kalman filter tuned with an expectation maximization algorithm. The filtered signals are populated using time and frequency domain analysis tools. These two approaches, used in tandem, make up the feature decomposition portion of the process.

Following this, we utilize a Principle component analysis (PCA), maximum relevance minimum redundancy (mRMR), and variance inflation factor (VIF), to properly select the most relevant features for predicting the RUL. These methods are carefully harmonized to optimize our choice of features among the decomposed features. Lastly, a machine learning model uses the selected features as its input, and is trained to predict the RUL of the given military equipment.

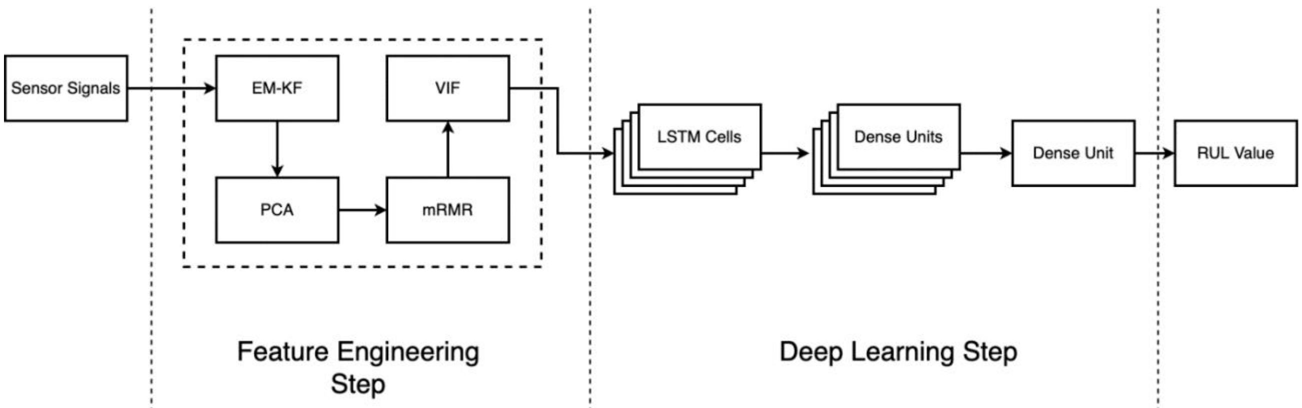

Figure 2. The process flow diagram of PROGNOS.

2.2 Theory of PROGNOS

In a complex military equipment, a filtering process is required when maintaining a large group of sensors that may be affected by unmeasured disturbances, where signal contamination can mislead a monitoring system. In 1960, R. E. Kalman proposed a recursive solution to discrete-data linear filtering problem, known as Kalman filter [12]. This model can serve as a denoising method or a model-based information extraction method in the condition monitoring industry. Kalman filter (KF) addresses the general problem of predicting the true state $x \in R^n$ of a discrete time series. KF is a set of mathematical equations that estimates $P(x_k|z_{0:k})$, as shown in (1):

$$x_{k+1} = A_k x_k + Bu_k + w_k \quad (1)$$

At time step k, with a measurement matrix of $z \in R^m$ can be computed using (2):

$$z_k = H_k x_k + v_k \quad (2)$$

Here, x_{k+1} denotes the KF's estimation, *A* indicates the state transition matrix, *B* is the transition offset, *H* is the observation model, v_k and w_k are the process and observation noises (respectively), where they were assumed as linear Gaussian model, as shown in (3) and (4):

$$p(w_k) = Normal(0, Q_k) \quad (3)$$

$$p(v_k) = Normal(0, R_k) \quad (4)$$

Where Q and R are transition and observation covariances. There is also a backward variant of KF, known as Kalman smoother (KS). The smoother refines estimates of previous states by observing later observations, in which it was designed to estimate $P(x_k|z_{0:K-1})$ [13]. In order to formulate a KF and integrate it into a dynamic system, the state space (mathematical model) of the system must be known. However, the parameters of a linear dynamical system can be approximated using the Expectation-Maximization (EM) algorithm. The EM algorithm has been used in the prediction of Gaussian mixture model parameters. Hence, the EM algorithm iteratively selects the KF parameters without user intervention. However, we must stress that the initial selection of the parameters significantly influences the final parameters, since this optimization procedure is considered a non-convex problem. After selecting Kalman parameters, the filtering method becomes ready to filter out the noisy components from the given signal.

Raw signals are often in a complex shape, requiring a series of functions to pick up performance degradation features. Feature extraction refers to an idea of decomposing valuable information from data by breaking them down into simpler components. A general objective of any extraction method is to assist the machine learning model in perceiving the correlations between the different attributes. Thus, this decomposition methodology allowed us to transform the raw data into health indicators for the learning algorithm. We utilized time and frequency domain analysis tools to decompose each sensor into multiple extracted features. In our proposed model, eight time-domain signal processing methods and three frequency domain tools were selected for extraction purposes, as presented in table 1. In this way, one sensor signal is decomposed into 11 sub signals. In other words, the raw dataset consisting of sensor signals are augmented to a large feature space.

Table 1. Statistical time-domain analysis [14]-[16].

Feature No	Time-domain analysis tools	Feature No	Frequency-domain analysis tools
1	Mean	9	Fast Fourier Transform (peak value)
2	Variance	10	Spectral Skewness
3	Skewness	11	Spectral Kurtosis
4	Kurtosis		
5	Standard Deviation		
6	Autocorrelation		
7,8	Instantaneous phase and amplitude envelop of Hilbert Transform		

Methods in this section were designed to distinguish relevant features from redundant ones, in which the objective is to positively affect the performance of the estimating model. Feature selection methods can be broken down into three approaches: wrapper, filter, and embedded methods. In our proposed model, only two techniques were employed to perform the selection process: filter and wrapper methods. The filter methods perform the selection step based on local performance indicators (independent from the predictor) such as correlation or mutual information criteria. Contrarily, if the features are evaluated based on the learning algorithm's prediction performance, this feature identification process is known as a wrapper method. In our proposed model, the *Principal component analysis (PCA)* method was employed as a filter feature selection method, and the *Maximum relevance minimum redundancy (mRMR)*, and *variance inflation factor (VIF)* was assigned to carefully pick the most appropriate features based on the decomposed features.

Principal component analysis (PCA) has been widely applied in the field of computer science. The algorithm was proposed in 1901 by Pearson [17] and further developed by Hotelling in the late 1930s [18]. This dimensionality-reduction method projects high-dimensional data into a low-dimensional subspace component by maximizing the variance. It is an essential first step for many data processing tasks. As we know that, PCA has been utilized for dimensionality reduction, yet the algorithm performs well in feature elimination as well [19], since they share the same principle: extracting informative components. Hence, we will commence with PCA-based feature extraction. Assume that given dataset $X \in R^{1\times N}$ with N samples, and each column in X is scaled to zero and unit variance. The covariance matrix S of the given data (5):

$$S = \frac{XX^T}{N} \tag{5}$$

Let $V \in R^{1\times N}$ as the eigenvectors of the covariance matrix, where the eigenvectors are sorted in descending form according to their associated eigenvalues. In this way, we can calculate principal components of the given data, where they explain most of the variance of the signal. it is known that we can calculate the feature extraction result, with respect to V, of the given dataset X , is $X \in R^{M \times N}$ [20]:

$$z = x^T v = \sum_{i=1}^{N} x_i v_i \tag{6}$$

where $v = [v_i \ldots v_N]^T$, $X = [x_i \ldots x_N]^T$, and N is the dimensionality of sample vectors. Since it is known that as the absolute value of x_i becomes smaller, the contribution to the output becomes less important. So, computing only the first principal component gives us major information on the dataset, and it can be mathematically derived with the following equation (7) [21]:

$$Y_1 = v_{11}x_1 + v_{12}x_1 + \ldots v_{1N}x_N \quad (7)$$

Here v can be considered as the weights and known as loading vector. In order to perform a feature selection using the PCA method, we sort the loading vectors of the first principal based on their magnitude and select the highest class among them [21]. In this way, we achieved a filter method that automatically detects the attributes that explain most of the variance by utilizing the PCA transform strategy.

Maximum relevance minimum redundancy (mRMR) is a feature selection method aiming to find the most relevant features, while keeping the redundancy at minimum. Due to that reason, the assigned score to each data vector is the harmonization of the relevance and redundancy scores. We utilized the following quotient (relevance divided by redundancy) rule-based formulation (8) to assign score to each attribute in the dataset [22]:

$$f^{mRMR}(x_j) = \frac{MI(x_j, y_i)}{Corr(x_{i,} y_i)} \quad (8)$$

The maximum relevance score (MI) is chosen as mutual information of candidate features and RUL value, where the minimum redundancy score is the Pearson correlation ($Corr$) between selected features and the candidate features. In each iteration, the mRMR algorithm selects a feature from the feature space respect to the scoring function (8). Therefore, the number of features that we want to select is the same as the iteration number of the mRMR algorithm.

Variance Inflation Factor (VIF) is a measure of multicollinearity in a multiple regression model. In other words, it computes how much the variance of an estimated regression coefficient is affected due to collinearity [23]. The following equation (9) formulates the VIF:

$$VIF_j = \frac{1}{1 - R_j^2} \quad (9)$$

Where, j is referred to jth predictor and R_j^2 is the multiple correlation coefficient, which describes the proportion of jth predictor between the rest of the features (predictors). If $VIF_j > 5$, then it means that the predictor j has correlation with the remaining predictors.

Machine learning is an AI-based learning algorithm, which processes data to recognize the hidden patterns by imitating human perceiving methods. *Long-Short Term Memory (LSTM)* is an alternative variation of recurrent neural networks (RNN). In order to deal with the gradient problem of RNNs, specially crafted memory cells are introduced to the RNN structure, and the model's name was proposed as Long Short-Term Memory (LSTM), which later became one of the most popular RNN. There are various LSTM memory cell architectures: LSTM without a Forget Gate, LSTM with a Forget Gate, LSTM with a Peephole Connection [24]. However, the most typical memory type is the LSTM memory cell with a Forget Gate, which was introduced by Ger Schimudhuber Cummins in 2000. The cell can be broken down into three main structures: Forget gate, Input gate, and Output gate. Unlike RNN, the information is being overwritten by the current state. Indeed, the gates provide the functionality of writing, reading, and resetting memory cells. The internal structure of an LSTM cell with a forget gate is expressed as follows [24]:

$$f_t = \emptyset_1(W_f\, h_{t-1} + W_{fx}\, x_t + b_f) \quad (10)$$

$$\widetilde{c_t} = tanh\left(W_f h_{t-1} + W_{\tilde{c}x}\, x_t + b_{\tilde{c}}\right) \quad (11)$$

$$i_t = \emptyset_2(W_{ih}\, h_{t-1} + W_{ix}\, x_t + b_i) \quad (12)$$

$$c_t = f_t . c_{t-1} + i_t . \widetilde{c_t} \quad (13)$$

$$o_t = \emptyset_3(W_{oh}\, h_{t-1} + W_{ox}\, x_t + b_o) \quad (14)$$

$$h_t = o_t . tanh(c_t) \quad (15)$$

Where f_t represent the forget gate, which decides whether to erase the cell state (0) or keep the information (1). The new information is being decided by the input gate i_t. The candidate memory cell ($\widetilde{c_t}$) describes the current input. Later, the memory cell (c_t) is computed using the input gate and the forget gate. In the final section, the output gate decides the final output by considering the previous information h_{t-1} and the current input x_t. As a response to that, the hidden state h_t can be computed as the last part of the LSTM cell structure. Similar to ANN dynamics, the LSTM method uses backpropagation to optimize the weights of the hyperparameters. Fundamentally, backpropagation is a method of propagating the total loss by evaluating the expression for the derivative of the cost function as a product of derivatives between the adjacent gates and units [25]. Therefore, minimizing the cost function will ultimately decrease the model error, and Gradient descent is one of the dominant optimizers to train deep networks today. This optimization method is a first-order iterative optimization strategy for detecting a local minimum of the given differentiable function [26]. Therefore, the cost function is minimized by taking opposite steps against the direction of the gradient. RMSPROP is an extension of Gradient descent, which uses adaptive sized gradients by utilizing an exponential average of its recent magnitude, and this optimizing strategy is becoming to be adopted by the industry [27].

3. EXPERIMENT

3.1 Dataset Overview

In the following experimental study, PROGNOS was compared with conventional machine learning regression methods. The training and validation set is NASA's challenge dataset, where it contains diagnostics and prognostics of equipment faults from the first conference of PHM'08 [11]. NASA modeled a damage propagation simulation for a fleet of similar aircraft gas turbine engines, to test prognostic algorithms. The synthetic dataset generation was carried out using the C-MAPSS system simulator, and it consists of multi-variate contaminated signals (26 features) that indicate performance degradation. Each engine starts with a different degree of initial wear and manufacturing variation and develops a fault at some point during its life cycle. The authors have provided two datasets for training and testing purposes. The training set consists of the complete life cycle of 100 turbofan engines, in which they are run until failure, while the test set is composed of partial life of turbofan engines.

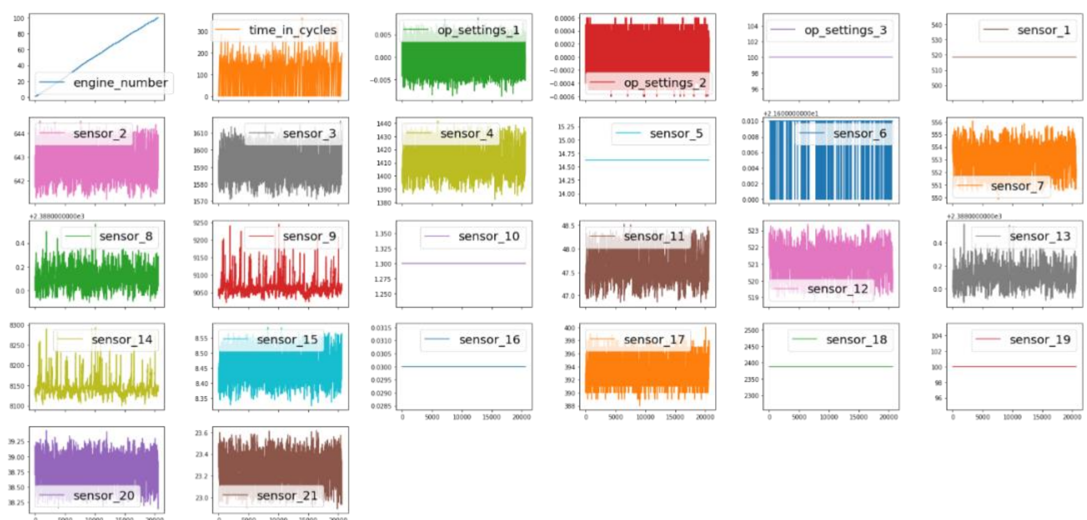

Figure 3. Visual representation of the raw training set (except the RUL value).

3.2 Feature engineering

Before starting the feature engineering process, the sensor signals are standardized, meaning that the unit variance is kept as one and the mean is set to zero for each data vector. The KF model was created using predetermined initial values, either a matrix of zeros or a matrix of ones. Table 2 presents the initial value for some of the used KF parameters. After initializing the Kalman parameters in this denoising process, the optimal parameters were iteratively estimated using the

EM algorithm, which was developed utilizing Python's pykalman library. Kalman Filter and Smoother methods predicted the true values of each selected sensor, where EM optimized the following parameters of KF: transition covariance, observation covariance, initial state mean, and initial state covariance.

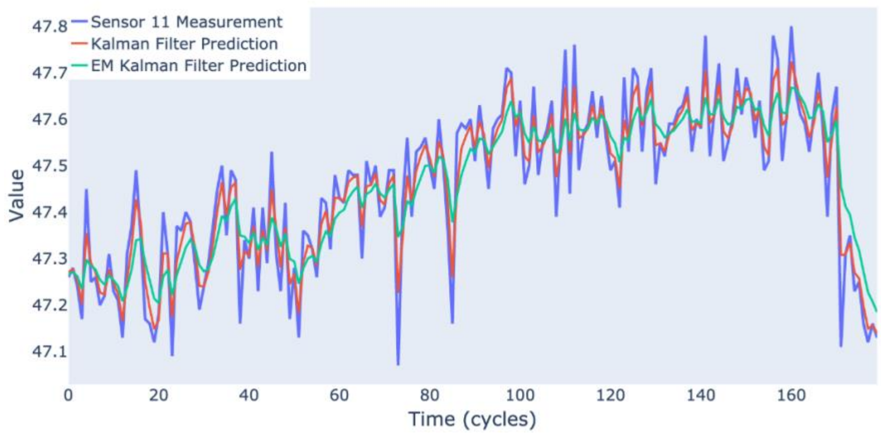

Figure 4. Sample measurements of sensor 11, with EM-KF's, and KF's corresponding estimation.

Table 2 shows the denoising results for the measurement of Sensor 11, the corresponding estimation of the KF with its initial parameters, and the EM optimized KF. As it can be seen, even though the noise was reduced without tuning the parameters of the KF algorithm, the noisy components were still present in the filtered signal. To address this issue, the KF's parameters were estimated in accordance with the given signal's behaviors. In other words, a further convergence to the true value was achieved by optimizing the KF parameters based on the EM algorithm. As a response to that, the algorithm became more reliable and robust against noisy components. The EM algorithm was run for 30 iterations to converge on an optimal value for each parameter, where they are presented in table 2. Consequently, the EM tuned KF successfully filtered out the spikes of the given data and reduced the baseline fluctuations in the signal without changing overall characteristics.

Table 2. Kalman filter parameter values, before and after EM optimization.

Kalman Parameter	**KF with initial values**	**EM tuned KF**
Transition covariance	1	1.22×10^{-3}
Observation covariance	1	1.01×10^{-2}
Initial state mean	0	47.32
Initial state covariance	0	1.07×10^{-4}

The filtered signals were decomposed using both time and frequency domain extraction techniques to highlight vital features that indicate degradation in the engine's remaining life. In the dataset, sensors did not have any associated identity and thus the suitable extraction method could not be determined easily. Yet, optimal combination among a population of extracted features led the model to desirable results. Firstly, popular feature extraction methods in literature were selected to decompose the given signals. Among extracted features for all signals, the ones that maximized the performance were considered as the selected features. In order to apply that, 11 frequency and time domain extraction methods were applied to each feature. As a result, the dimension of the dataset was transformed from $X \in R^{M \times 24}$ to $X \in R^{M \times 264}$, where X is the dataset, M represents the data points, and 24 refers to the number of dimensions (features). Table 3 below lists each feature extraction method that was applied during the extraction process. For illustrative purposes, results of the extraction methods were individually visualized for sensor 11 measurements, as shown in figure 5. Some decomposed features (i.e., spectral skewness or spectral kurtosis) were redundant for this sensor measurement, where they did not contain any prognostic value, seeing as the RUL value varies. On the other hand, a few features (i.e., variance) react as the remaining cycles decrease. The next stage's objective is to identify the features that explain the characteristic of the engine's life cycle.

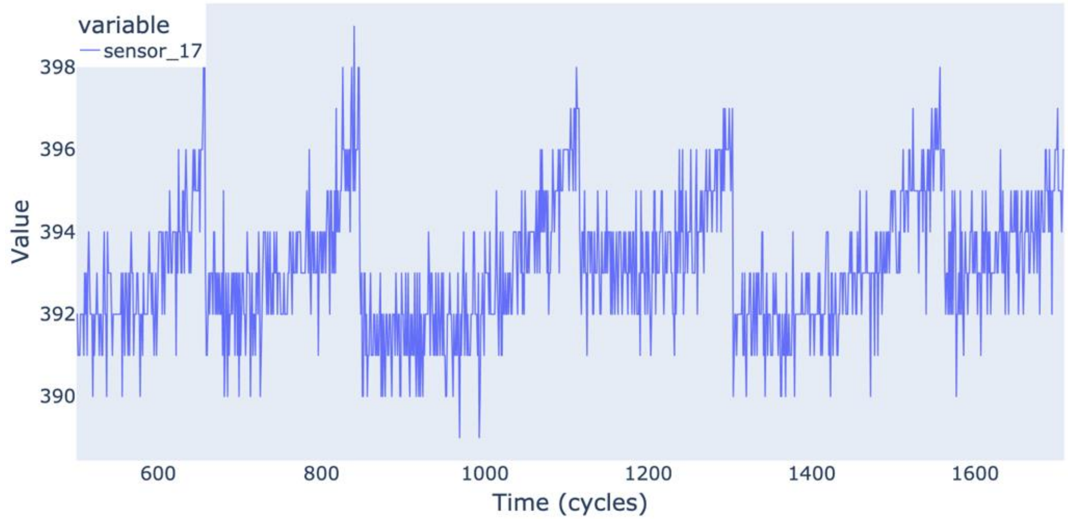

Figure 5. Sensor 11 values for six different engines run.

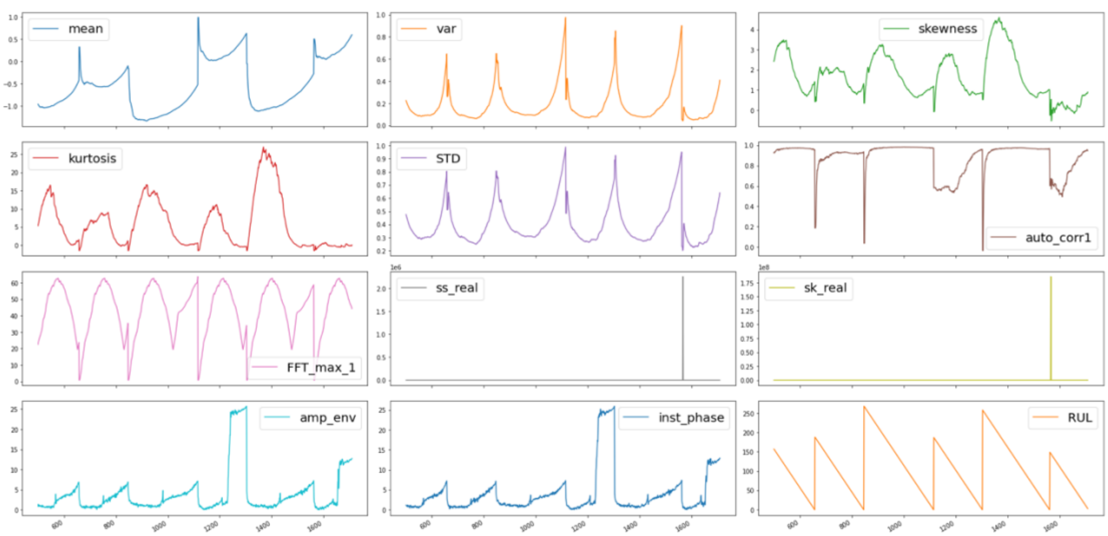

Figure 6. Visual representation of extracted features from the sensor 11's measurements for six engines run and the bottom down figure illustrates the corresponding RUL value of six engines run.

In this stage, we selected optimal features by utilizing sequentially three different feature selection methods: PCA, mRMR, and VIF. As the first selection process, PCA assigned a score to each decomposed feature based on how much variation it explains in the dataset. A total of 88 features, making up one-third of the data set, was eliminated based on their low variance contribution to the overall dataset. Later, mRMR was employed to select the features that contained the maximum relevance with respect to the RUL, while keeping the similarity between selected features at a minimum. This technique is called minimum redundancy. The maximum relevance score was then chosen as mutual information of candidate features and RUL value, where the minimum redundancy score is the Pearson correlation between selected features and the candidate features. Then the mRMR algorithm was run to select the most optimal 40 features. One of the weaknesses of the mRMR is that the algorithm can only compare the similarities between the last selected and the candidate features. However, the similarity between the features that have been previously selected, and the current candidates are not considered. Due to that reason, a VIF algorithm was employed to detect further multicollinearity between selected features, and it eliminated the ones that contains collinearity. Among the selected 40 features, the VIF algorithm only further eliminated further 25 features in the dataset. After this analysis, there were 15 relevant features remaining. These features are shown in table 3 below.

Table 3. Final attributes (optimal features) of the dataset.

Feature No	Signal Name	Feature Extraction Methods
1	Sensor 4	Instantaneous phase
2,3	Sensor 6	Instantaneous phase, Kurtosis
4	Sensor 9	Instantaneous phase
5, 6	Sensor 11	Instantaneous phase, Kurtosis
7	Sensor 12	Mean
8, 9	Sensor 13	Skewness, amplitude envelope
10, 11	Sensor 14	Instantaneous phase, Autocorrelation (second value)
12, 13	Sensor 15	Instantaneous phase, Kurtosis
14	Sensor 17	Kurtosis
15	Sensor 20	FFT (maximum value)

3.3 RUL estimation

The PROGNOS model was trained using the 15 extracted optimal features, which contained 20631 data points. Later, the proposed model was tested with a test set that NASA had prepared for the competition in PHM'08 conference [11]. It is worthy to mention that the test set raises several complications, since the RUL predictor wouldn't know how long the jet engine had been used previously, and the jet engines may not fail, meaning their RUL is never known. On the other hand, the engines that were used to generate the training set were run until they fail.

In order to predict the health state of each turbofan engine, an LSTM model with an adaptive sliding window was used. The LSTM model consisted of an LSTM layer, a dense layer and an output layer. The first layer consisted of 10 cascaded memory cells with a 30% of dropout rate, where the hidden states were passed to the dense layer. The dropout rate for the dense layer was 10%. Between the LSTM layer and the dense layer, a batch normalization process was used to standardize the LSTM layer's output before passing them to the dense layer. Lastly, a single neuron with a Linear function is utilized to output the RUL value. Following the model's estimation, the hidden state of the second layer's last memory cell was passed to the output linear activation layer. The RMSPROP algorithm minimized the training loss of the predictive algorithm with regard to the given training set. Finally, the output layer provided the RUL life estimation of the LSTM structure. The model was constructed using Python's open-source software library Keras. In order evaluate the model performance against the true values, we utilized the following regression metrics: R^2 (16), $RMSE$ (17), and MAE (18).

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{N}(\bar{y}_i - y_i)^2} \tag{16}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \tag{17}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{18}$$

Here $\hat{y}$, and y, respectively are the value predicted by the model and the true value from the test set. $\bar{y}$ represents the variance in y output values. Lastly, N is the number of time step (cycles).

After training the LSTM layer with extracted 15 features and mentioned configurations. The proposed model estimated the health state of given aircraft engines. Figure 7 illustrates PROGNOS' prediction on overall test set. Due to the regression metrics, PROGNOS estimated the well-being of 100 engines with 22.93 MAE, 30.44 $RMSE$, and 0.63 R^2.

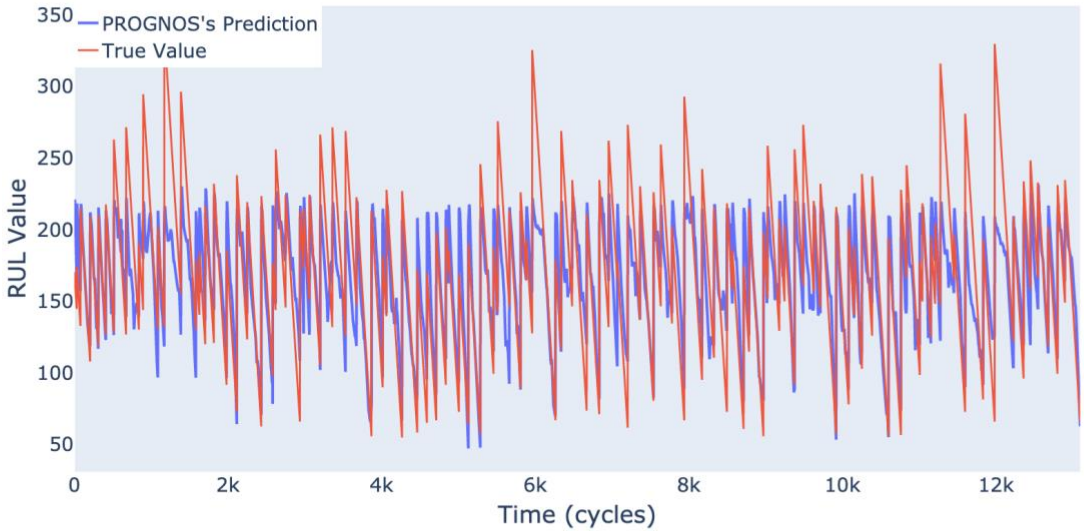

Figure 7. PROGNOS's RUL prediction on the test set.

In order to further evaluate the results of the model, we analyzed the model's performance on different cases: an aircraft engine with mode (most frequent engine running duration), and on extreme runs (the longest and shortest engines run). Six engines run were chosen for each case: mode and the extreme cases. The PROGNOS model's prediction for each case is illustrated in figures 8 & 9 and the evaluation metrics are displayed in Table 4.

In conclusion, PROGNOS was able to predict the RUL value of a jet engine by monitoring its sensor signals with impressive accuracy. Due to the mode cases, our proposed model performed well against the generalized dataset, and as a result, proves that it can be extended to predict the RUL of any military machine that produces capturable signals. However, if the cycle duration of an engine is too short or long (extreme cases), PROGNOS required some time to seek the RUL value of the engine. Though, at the end of the cycle, PROGNOS converged to the true RUL value.

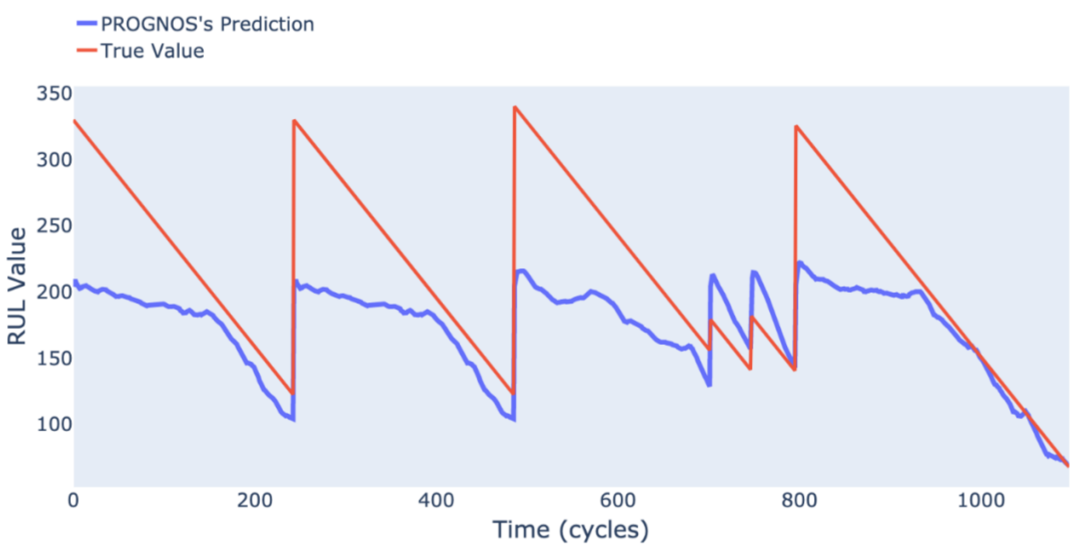

Figure 8. PROGNOS's RUL prediction on extreme (longest and shortest) engines run.

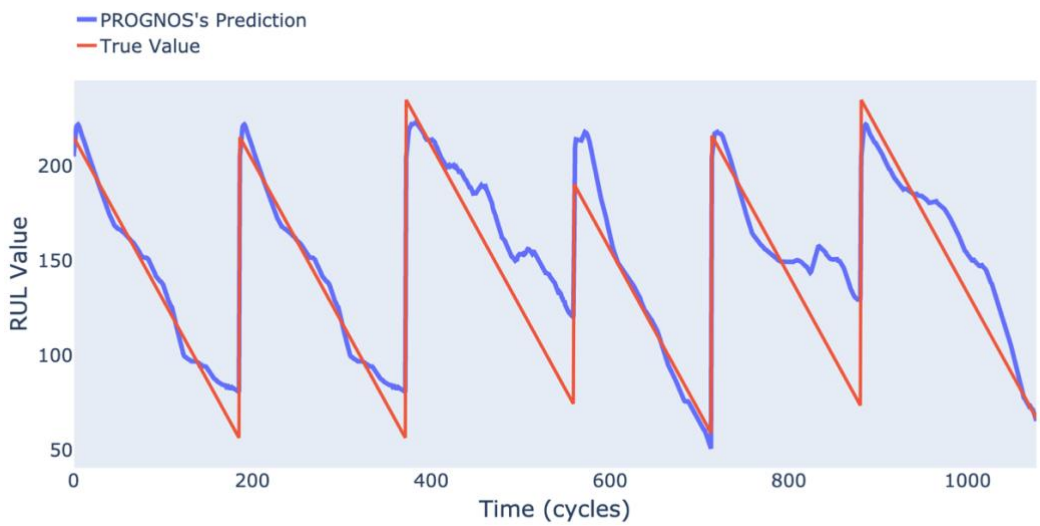

Figure 9. PROGNOS performance on mode (the most frequent) engines run.

Table 4. Performance Summary of PROGNOS model's prediction.

Performance Metrics	**Performance Summary of PROGNOS**		
	Extreme Run	Mode Run	Overall Test Set
MAE	46.79	13.56	22.93
RMSE	59.13	19.29	30.44
R^2	0.19	0.83	0.64

4. SUMMARY

In this technical paper, we proposed a fully automatic RUL estimator (PROGNOS) for military equipment. The main considerations of PROGNOS were feature extraction of the given sensor measurements, data-driven feature selection based on the extracted feature subset, and a deep learning model to estimate the RUL value of the system in study. The viability and performance of the proposed model were tested on a highly complex competition dataset: the NASA aircraft gas turbine engine degradation dataset. Without requiring any prior information, PROGNOS' feature engineering method successfully transformed noisy signals into meaningful data vectors. LSTM and dense layers intelligently recognized the pattern between the extracted features and engines' RUL value. In this way, PROGNOS model estimated the well-being of the aircraft engine in a fully autonomous way. According to the obtained test results, overall, the model yielded excellent performance in most of the cases. However, we noticed that the model was sensitive to the outliers (short or long runs). In such cases, the prediction error progressively got smaller as new data were introduced. Therefore, we can summarize that the proposed PROGNOS model was viable and robust overall, proving its general usefulness on all military equipment that emit signals.

REFERENCES

[1] B. K. N. Rao, "Condition monitoring and the integrity of industrial systems BT - Handbook of Condition Monitoring: Techniques and Methodology," A. Davies, Ed. Dordrecht: Springer Netherlands, 1998, pp. 3–34.

[2] R. K. Mobley, *An Introduction to Predictive Maintenance*. Oxford, England, UK: Butterworth-Heinemann, 2002.

[3] H. Wang, C. Ma, and L. Zhou, "A brief review of machine learning and its application," *Proc. - 2009 Int. Conf. Inf. Eng. Comput. Sci. ICIECS 2009*, pp. 9–12, 2009, doi: 10.1109/ICIECS.2009.5362936.

[4] Z. M. Çinar, A. A. Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, and B. Safaei, "Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0," *Sustain.*, vol. 12, no. 19, 2020, doi: 10.3390/su12198211.

[5] P. Tavner, L. Ran, and C. Crabtree, *Condition Monitoring of Rotating Electrical Machines (3rd Edition)*. Stevenage, UK, 2020.

[6] T. Marwala, "Condition monitoring using computational intelligence methods: Applications in mechanical and electrical systems," *Cond. Monit. Using Comput. Intell. Methods Appl. Mech. Electr. Syst.*, vol. 9781447123, no. 2001, pp. 1–235, 2012, doi: 10.1007/978-1-4471-2380-4.

[7] G. P. Stachowiak, P. Podsiadlo, and G. W. Stachowiak, "A comparison of texture feature extraction methods for machine condition monitoring and failure analysis," *Tribol. Lett.*, vol. 20, no. 2, pp. 133–147, 2005, doi: 10.1007/s11249-005-8303-1.

[8] A. Bondyra, P. Gasior, S. Gardecki, and A. Kasiński, "Fault diagnosis and condition monitoring of uav rotor using signal processing," in *2017 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, 2017, pp. 233–238.

[9] X. Zhu, Z. Cai, J. Wu, Y. Cheng, and Q. Huang, "Convolutional neural network based combustion mode classification for condition monitoring in the supersonic combustor," *Acta Astronaut.*, vol. 159, pp. 349–357, 2019.

[10] L. Wang, M. Li, L. Liu, and D. Liu, "Exhaust gas temperature sensing data anomaly detection for aircraft auxiliary power unit condition monitoring," in *2018 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*, 2018, pp. 426–430.

[11] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," *Int. Conf. Progn. Heal. Manag.*, Oct. 2008, doi: 10.1109/PHM.2008.4711414.

[12] R. E. KALMAN, "A New Approach to Linear Filtering and Prediction Problems," *Atmos. Environ.*, vol. 43, no. 1, pp. 1895–1898, 1960, [Online]. Available: http://fluidsengineering.asmedigitalcollection.asme.org/pdfaccess.ashx?ResourceID=4506257&PDFSource=13%0Ahttp://dx.doi.org/10.1016/j.atmosenv.2010.06.034%250Ahttp://dx.doi.org/10.1016/j.atmosenv.2010.04.013%250Ahttp://dx.doi.org/10.1016/j.atmosenv.2010.03.

[13] S. P. B. S. T. Barrat, *Fitting a kalman smoother to data*. 2020.

[14] L. S. Bearing and W. Caesarendra, "A Review of Feature Extraction Methods in Vibration-Based Condition Monitoring and Its Application for Degradation Trend Estimation of," doi: 10.3390/machines5040021.

[15] W. Caesarendra, B. Kosasih, K. Tieu, and C. A. S. Moodie, "An application of nonlinear feature extraction - A case study for low speed slewing bearing condition monitoring and prognosis," in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Jul. 2013, pp. 1713–1718, doi: 10.1109/AIM.2013.6584344.

[16] V. K. Garg and Y.-C. Wang, "1 - Signal Types, Properties, and Processes," W.-K. B. T.-T. E. E. H. CHEN, Ed. Burlington: Academic Press, 2005, pp. 951–956.

[17] K. Pearson, "On Lines and Planes of Closest Fit to Points in Space," *Philos. Mag.*, vol. 2, pp. 559–572, Nov. 1900, doi: 10.1080/14786440109462720.

[18] H. Hotelling, "Analysis of a complex of statistical variables into principal components.," *J. Educ. Psychol.*, vol. 24, no. 6, pp. 417–441, 1933, doi: 10.1037/h0071325.

[19] F. Song, Z. Guo, and D. Mei, "Feature Selection Using Principal Component Analysis," in *2010 International Conference on System Science, Engineering Design and Manufacturing Informatization*, 2010, vol. 1, pp. 27–30, doi: 10.1109/ICSEM.2010.14.

[20] B. Wang, X. Yan, and Q. Jiang, "Loading-Based Principal Component Selection for PCA Integrated with Support Vector Data Description," *Ind. Eng. Chem. Res.*, vol. 54, no. 5, pp. 1615–1627, Feb. 2015, doi: 10.1021/ie503618r.

[21] S. M. Holland, "PRINCIPAL COMPONENTS ANALYSIS (PCA)," 2008.

[22] Jo, K. Lee, and Oh, "Improved Measures of Redundancy and Relevance for mRMR Feature Selection," *Computers*, vol. 8, p. 42, May 2019, doi: 10.3390/computers8020042.

[23] C. Thompson, R. Kim, A. Aloe, and B. Becker, "Extracting the Variance Inflation Factor and Other Multicollinearity Diagnostics from Typical Regression Results," *Basic Appl. Soc. Psych.*, vol. 39, pp. 1–10, Feb. 2017, doi: 10.1080/01973533.2016.1277529.

[24] Y. Yu, X. Si, C. Hu, and J. Zhang, “A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures,” *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019, doi: 10.1162/neco_a_01199.

[25] H. Zhao, Z. Lai, H. Leung, and X. Zhang, “Neural-Network-Based Feature Learning: Recurrent Neural Network BT - Feature Learning and Understanding: Algorithms and Applications,” H. Zhao, Z. Lai, H. Leung, and X. Zhang, Eds. Cham: Springer International Publishing, 2020, pp. 253–275.

[26] E. Bisong, “Optimization for Machine Learning: Gradient Descent BT - Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners,” E. Bisong, Ed. Berkeley, CA: Apress, 2019, pp. 203–207.

[27] S. J. Reddi, S. Kale, and S. Kumar, “On the Convergence of Adam and Beyond,” *ArXiv*, vol. abs/1904.0, 2018.