Research Article GU J Sci 35(3): 1200-1210 (2022) DOI: 10.35378/gujs.937169

Gazi University

Journal of Science

http://dergipark.gov.tr/gujs

Using Long-Short Term Memory Networks with Genetic Algorithm to Predict Engine Condition

Semre ERPOLAT TASABAT[1], Olgun AYDIN[2, *]

[1]*Mimar Sinan Fine Arts University, 34380, Istanbul, Turkey*
[2]*Gdansk University of Technology, 80233, Gdansk, Poland*

Highlights

- Novel LSTM architecture.
- Different hyper parameter optimization technique.
- Efficient prediction.

Article Info

Received: 14 May 2021
Accepted: 03 Sep 2021

Keywords

Deep learning
Genetic algorithm
Artificial neural networks
Predictive maintenance
Cost efficient maintenance

Abstract

Predictive maintenance (PdM) is a type of approach for maintenance processes, allowing maintenance actions to be managed depending on the machine's current condition. Maintenance is therefore carried out before failures occur. The approach doesn’t only help avoid abrupt failures but also helps lower maintenance cost and provides possibilities to manufacturers to manage maintenance budgets in a more efficient way. A new deep neural network (DNN) architecture proposed in this study intends to bring a different approach to the predictive maintenance domain. There is an input layer in this architecture, a Long-Short term memory (LSTM) layer, a dropout layer (DO) followed by an LSTM layer, a hidden layer, and an output layer. The number of epochs used in the architecture and the batch size was determined using the Genetic Algorithm (GA). The activation function used after the output layer, DO ratio, and optimization algorithm optimizes loss function determined by using grid search (GS). This approach brings a different perspective to the literature for finding optimum parameters of LSTM. The neural network and hyperparameter optimization approach proposed in this study performs much better than existent studies regarding LSTM network usage for predictive maintenance purposes

1. INTRODUCTION

Usage of PdM has increased gradually with the definition of Industry 4.0. It has become an important investment area for manufacturers and asset managers. PdM helps develop analytical methods to monitor conditions of machines, optimize maintenance plans, and generate real-time alerts. PdM allows manufacturers to reduce service costs, maximize operating time and increase production efficiency.

The two most common approaches for predictive maintenance are rule-based and ML-based approaches. In the rule-based approach, which is also called status monitoring, data is continuously collected from the system through sensors. The system generates alerts when predefined rule-based threshold values are observed. In this approach, product teams work together with engineering and customer service teams to identify the factors causing machine malfunctions. The rules are created as "if it is so, take this action." For example, suppose the temperature and rotation speed are above a threshold. In that case, the system sends alerts to the operators responsible for avoiding malfunction. These rules provide automatic PdM capability at a certain level. However, it is still critical that a product team decide which data points need to be collected carefully.

*Corresponding author, e-mail: olgun.aydin@pg.edu.pl

Artificial Intelligence (AI) can be used for PdM purposes. With ML and AI-based systems, it is possible to make complex analyses using large amounts of historical data collected from the production process. It is possible to use ML algorithms to run different scenarios and predict failures before they occur [1]

System reliability is one of the most critical points for engineering applications. Failure of some parts of the system can affect the entire system. For example, problems in the units such as turbine engines, power supplies, batteries, etc., can cause failures in the entire system. To prevent the entire system's failure, it is essential to conduct maintenance for certain parts of the system or the entire system. [2] Jardine et al. recommended condition-based maintenance (CBM) to predict system units' current state. The primary purpose of CBM is to avoid unnecessary maintenance operations and carry out maintenance processes when an anomaly is detected in any part of the system. [3]

To develop an effective CBM strategy, estimating the remaining useful life (RUL) of the system with high accuracy is very important. It is possible to estimate RUL by processing the data collected via sensors. In 2008, Felix et al. proposed a time-frequency-specific method for estimating RUL. The method measures the complexity of time-frequency surfaces. A data-driven algorithm was developed to estimate RUL. C-MAPSS data set provided by NASA was used in the study [4]. In the competition held within the IEEE PHM 2012 Conference's scope, Porotsky developed a new model to control parameter optimization based on cross-validation procedure and was granted a prize. The C-MAPSS data set provided by NASA was used in the study [5]. In 2014, Jain et al. developed artificial neural networks (ANN) to estimate RUL. In this study, an ANN-based approach was proposed to make a more effective RUL estimation using NASA's C-MAPPS data set. The proposed model is built based on time series features [6]. Sateesh et al. proposed a new approach called Meta-Cognitive Regression Neural Network (McRNN) to estimate RUL. McRNN uses Extended Kalman Filter (EKF) to obtain optimum network parameters. In the study, the C-MAPSS data set was used. A more efficient RUL estimation methodology was proposed compared to the previous studies [7]. Santos et al. applied the PdM approach for hard disk drives based on SMART parameters that can predict failures at both long and short-term intervals with the help of LSTM networks [8]. In a study conducted by Aydin et al. in 2017, a decision-making process was developed using LSTM networks to estimate an engine's current state. The training process of LSTM networks was carried out using the C-MAPSS data set provided by NASA in a high-performance, large-scale data processing environment. In the study, a different RUL estimation approach was proposed. Instead of predicting RUL as a numeric value, predefined four classes were predicted [9]. Chen et al. proposed a predictive maintenance approach using LSTM. They compared LSTM performance with support vector regression machine (SVRM), and LSTM provided better results than support vector regression [10]. Das et al. proposed a solution called Desh for PdM methods using DL methods [11]. In 2018, Cachada et al. published a study stating the importance of predictive maintenance system architecture and smart decision-making systems compatible with Industry 4.0. The study considers the advanced analysis of data collected for the early detection of machine failures [12]. In another study conducted by Öztanır in 2018, DL techniques for PdM were discussed. The performance of DL techniques was compared with other machine learning methods. Experiments were performed using the C-MAPSS data set provided by NASA [13]. In 2018, Zhang et al. designed an LSTM architecture to analyze equipment's working status based on the collected data and estimate the operating status [14].

In this study, a new DNN architecture was proposed using the Genetic Algorithm and Grid Search as hyperparameter optimization approaches. All calculations were conducted using Apache Spark, a big data processing framework. The approach presented in the study brings a different perspective to literature for finding optimum parameters of LSTM networks. The neural network and hyperparameter optimization approach proposed in this study was compared with less complex networks and approaches proposed in other studies regarding LSTM network usage for predictive maintenance purposes, and the proposed approach out performed. For comparison accuracy metric was used based on confusion matrix, as it is standard approach in the literature about predictive maintenance.

2. MATERIAL AND METHOD

Nowadays, the Internet of Things (IoT) is developing rapidly. Engine manufacturers can now observe engine components by processing the data collected through sensors placed on the engine components. Manufacturers also build smart systems for predicting engine conditions using data produced by IoT sensors. With the help of a well-designed prediction system, the system's components can be replaced before the failure occurs. The life of the components prolongs with effective maintenance processes, the equipment's usability increases, and the maintenance costs reduce.

In the report published by General Electric, it is stated that PdM helps reduce unexpected downtime of the system and significantly contributes to the prevention of financial loss. The report also stated that the PdM method helps reduce the system's downtime due to unexpected reasons by more than 2% and helps prevent possible financial losses of 40 million dollars [15]. In another study conducted in 2018 by General Electric, results represent the readiness of manufacturers in European countries for PdM. Within the study's scope, senior managers of the big industrial manufacturers in European Countries were interviewed. The companies' approaches to PdM were examined. 93% of the company representatives participating in the study stated that their current maintenance processes were not sufficient, not very adequate and their maintenance processes needed improvement. 55% of the company representatives stated that they started to implement PdM processes, and 23% stated that PdM processes contributed positively to their operations. [16]

Research studies conducted in recent years show that the researchers have been working on Hidden Semi-Markov Models, Artificial Neural Networks (ANN), DNNs to propose different architectures for predictive maintenance purposes. These studies show that predictive models for PdM can also be developed using LSTM networks and LSTM networks provide superiorities to the other ANNs and traditional techniques.

2.1. LSTM

LSTM, a type of Recurrent Neural Network (RNN), was proposed by Sepp Hochreiter and Bergen Schmidhuber in the late 90s to develop a network that learns long-term relationships in the dataset. LSTM was designed to avoid long-term dependency problems. All recurrent neural networks have the form of repeating modules. In standard RNNs, this repeating module is in a simple structure such as a tanh layer. LSTM networks also have chain-like repetitive modules as in RNNs. However, the repeating modules have a structure different from the structure in RNN. Instead of a single neural network layer, four layers interact in a particular way.

The unit that has a crucial role in LSTM networks is the unit reporting cells' status in LSTM networks. LSTM can pass information to the cell state, which is carefully arranged by the structures called gates. The gates allow the information to be transferred to the next cell state. They are composed of a sigmoidal neural network layer and point multiplication operation. The sigmoid layer generates values between 0 and 1. If the layer generates zero, it means "do not let any information pass to the next step," having the value of one as a result of the layer means "let all information pass to the next step" [17].

2.2. Hyperparameter Optimization Using Genetic Algorithm (GA)

Darwin's theory of evolution has been inspiring many researchers in various disciplines. Various evolutionary algorithms have been developed using basic terms such as gene, natural selection, crossover, and mutation proposed by Darwin [18].

One of the essential evolutionary algorithms is genetic algorithms. The evolution process was interpreted in a computer environment by Goldberg and Holland [19]. This step was essential for GAs. Goldberg proved that there are more than 80 examples of GAs in real life [20]. The primary purpose of GA is to randomly select individuals as parents and generate children to form the next generation. GA has three main steps: natural selection, crossover, and mutation [20]. The flow of GA can be expressed as in Figure 1.

GA is used for hyperparameter optimization for ML and DL in recent studies. Ananto et al. in 2018 used ML algorithms to determine whether the news on the Internet was popular or not. They used GA to obtain the parameters of ML algorithms. As stated in this study, optimum hyperparameters were obtained using GA in a shorter computation time than grid search [21].

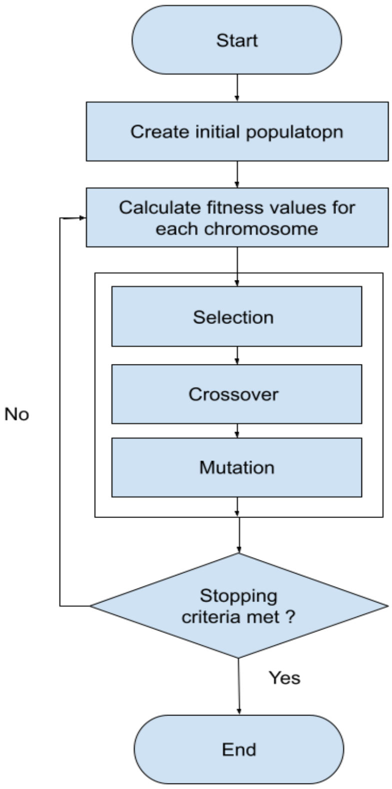
Start
Create initial populatopn
Calculate fitness values for each chromosome
Selection
Crossover
Mutation
No
Stopping criteria met ?
Yes
End

Figure 1. *Flow chart for GA*

Salah et al. used GA to determine the optimal time delays and how many layers of the LSTM network should be used [22]. In the study conducted by Chung et al., the number of hidden neurons of LSTM networks was determined using GA to develop a prediction model for stocks [23]. In these studies, GA is generally used to find specific hyperparameters that minimize the loss function.

2.3. Proposed Neural Network

The method proposed in this study is an LSTM network to predict the current state of an engine. The training process of the LSTM network was conducted on top of the big data framework Apache Spark. Thanks to this, researchers had lower computation time.

The data set provided by Prognostics CoE in NASA Ames was used to build the LSTM network. The data set is an open-source dataset and frequently used by researchers studying PdM.

Table 1. *Operational Settings*

Operational Settings
Altitude
Mach number
Throttle resolver angle

The dataset includes combinations of different engine operating conditions in a multivariate time series format. As detailed in Tables 1 and 2, the dataset consists of three operational settings and measurements of 21 sensors. Sensors include measures related to temperature, engine pressure, fuel, and coolant drain [24].

Before proceeding to the model development stage, the Ri, which presents the engine's remaining capacity, was calculated using the following equation [9]

$$R_i = \frac{(Time\ elapsed\ until\ the\ failure - Age\ of\ the\ engine)}{time\ elapsed\ until\ the\ failure}. \quad (1)$$

Having R_i equal to 1 means that the engine's remaining capacity is 100%. Having Ri equal to 0 means that the engine's remaining capacity is 0%.

Table 2. *Sensor measurements*

Sensor Measurements
Total temperature at fan inlet (°R)
Total temperature at LPC outlet (°R)
Total temperature at HPC outlet (°R)
Total temperature at LPT outlet (°R)
Pressure at fan inlet (psia)
Total pressure in bypass-duct (psia)
Total pressure at HPC outlet (psia)
Physical fa speed (rpm)
Physical cre speed (rpm)
Engine pressure ratio (P50/P2)
Engine pressure ratio (P50/P2)
Ratio of fuel flow to Ps30 (pps/psi)
Corrected fan speed (rpm)
Corrected core speed (rpm)
Bypass Ratio
Burner fuel-air ratio
Bleed Enthalpy
Demanded fan speed (rpm)
Demanded corrected fan speed (rpm)
HPT coolant bleed (lbm/s)
LPT coolant bleed (lbm/s)

Aydin et al. proposed to group RUL values into four groups, as shown in Table 3. "Healthy" indicates that the engine is at the beginning of its service life. "Critical condition" indicates that the engine should be repaired soon. "Repair engine" indicates that the engine will fail in a short time and should be repaired immediately, and "Engine failure" indicates that the engine is out of service [9].

Table 3. *Classes showing the condition of the engine*

R_i	Engine Condition
(0.85-1]	Healthy
(0.7-0.85]	Critical condition
(0.5-0.7]	Repair engine
[0-0.5]	Engine failure

In this study, the same approach has been used. The LSTM architecture has been proposed to predict engine conditions as "healthy", "critical condition", "repair engine", or "engine failure", as shown in Table 3.

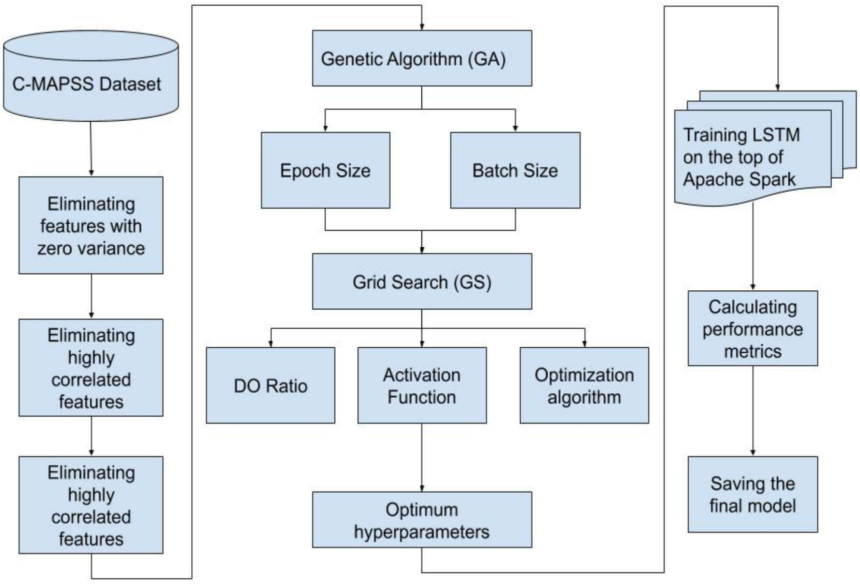
C-MAPSS Dataset
Eliminating features with zero variance
Eliminating highly correlated features
Eliminating highly correlated features
Genetic Algorithm (GA)
Epoch Size
Batch Size
Grid Search (GS)
DO Ratio
Activation Function
Optimization algorithm
Optimum hyperparameters
Training LSTM on the top of Apache Spark
Calculating performance metrics
Saving the final model

Figure 2. *Proposed methodology*

The proposed approach is shown in Figure 2. Since the goal was to provide an effective prediction model for understanding the patterns in the data coming from the sensors, feature engineering was performed before initiating the modeling part. The GA and GS were then performed. In these stages, DO ratio in the DO layer, activation functions used after LSTM layers, number of epochs, batch size to be used in each epoch, and the optimization method for optimizing the loss function were decided.

After deciding the parameters to be used in the network, DNN, which contains an input layer, an LSTM layer, a dropout layer, another LSTM layer, a hidden layer before the output layer and output layer, was trained. The input layer receives data coming from different sensors. This data is in the format of time series.

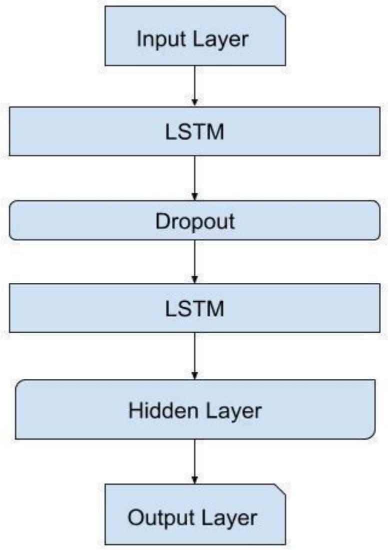

Figure 3. *Proposed LSTM Architecture*

In Figure 3, there is a DO layer after the first LSTM layer in the proposed architecture. DNNs having many parameters are very effective machine learning methods. However, when training such networks, overfit is often encountered. This layer helps mask information in each iteration and prevents the network from overfitting. In the proposed network, another LSTM layer following the DO layer learns the patterns in the data set by filtering the relevant and unrelated information in the data set. Both LSTM layers work on the same principle; they only function as filters that allow valuable information to be selected. There are seven variables as outputs of the second LSTM layer. The purpose of the additional hidden layer following the second LSTM layer is to take seven variables as input and reduce the number of those variables to four and pass four variables to the output layer. Thus, the output layer works on only four input variables instead of seven input variables. It is aimed to make the system more efficient and more accurate by reducing the number of variables entering the output layer to four. The hidden layer allows the information in each of the seven input variables to be randomly distributed to the four new meta input variables that will be sent to the final output layer. Since these networks contain many parameters, the network training process requires high computation power.

Data processing, training, and hyperparameter optimization steps were performed using a virtual machine with 16 GB ram and 1 NVIDIA K80 GPU on Google Cloud Platform. Since high computing power was required, the use of GPU was preferred. Python, one of the most commonly used programming languages in recent years, was chosen as the programming language. Python offers a variety of software libraries to users. Keras, which is a comprehensive DL library, is one of them. Therefore, the training of the networks was carried out using the Keras library. In order to increase performance and provide flexibility, elephas, which is a Python library, was used to train the DNN using the Keras library on the top of Apache Spark.

The results showed that if the current engine condition is predicted as a "repair engine", it is crucial to take necessary actions in advance. Three different engine setup information variables were combined into one variable and divided into six sub-groups by applying the k-means method. k-means was used as a clustering algorithm because it provides fast and powerful results. The silhouette score is an approach to decide how many groups should be used to divide the dataset while using the k-means algorithm. Silhouette scores were calculated for different k values, and “k” giving the highest silhouette score is determined as the number of groups, in other words, "k". As seen in Figure 4, the silhouette score reached the highest value when the number of groups “k” was 6.

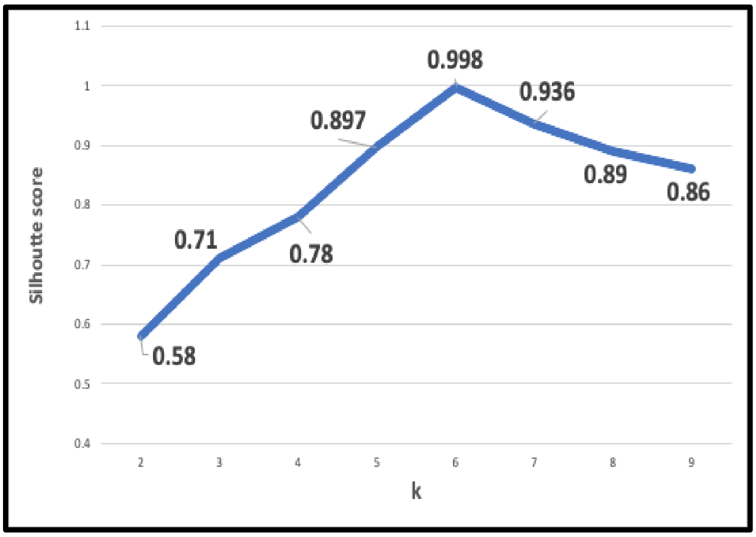

Figure 4. *Number of clusters and silhouette scores*

The variables with zero or very close to zero variance were determined and not included in the model. Sensors with zero variance can be interpreted as sensor data that did not produce different values over time. Because using such features can not contribute to the model positively, they were excluded together with highly correlated features from feature space. The number of epochs and batch size in each epoch were determined using GA. Table 4 shows the number of epochs, batch size obtained by using GA.

Table 4. *Number of epochs and Batch Sizes Obtained with GA*

Number of Epochs	Batch Size	Accuracy (Train set)	Accuracy (Validation set)
27	24	0.8397	0.8309
6	20	0.8358	0.8304
18	22	0.8403	0.8097
31	**20**	**0.842**	**0.8341**
12	22	0.8389	0.8289
11	3	0.8352	0.8189
20	13	0.8401	0.831
27	30	0.8419	0.8271
12	16	0.8404	0.824
4	24	0.8357	0.8292

As seen in Table 4, the highest accuracy was obtained when the batch size was 20 and the number of epochs was 31. In the next step, DO ratio, optimization algorithm used to optimize loss function, and activation function to be used after output layer were obtained using GS. 120 different combinations of values consisting of

For DO ratio: 0.1, 0.2, 0.3, 0.5, 0.6, 0.7, 0.8, 0.9
For optimization algorithm: adadelta, adagrad, sgd, adam, adamax
For activation function: softmax, softplus, sigmoid

were tested. It was observed that the combination giving the highest validation set accuracy with 83.6% included the DO ratio of "0.1", optimization algorithm "adagrad", and activation function "softplus". Finally, in the light of all this information, DNN was trained. 20% of the dataset was selected as a test set, and 80% as a training set. In order to control overfitting, 10% of the training set was used as a validation set. Figure 5 shows the model's accuracy on the training set and validation set for each iteration.

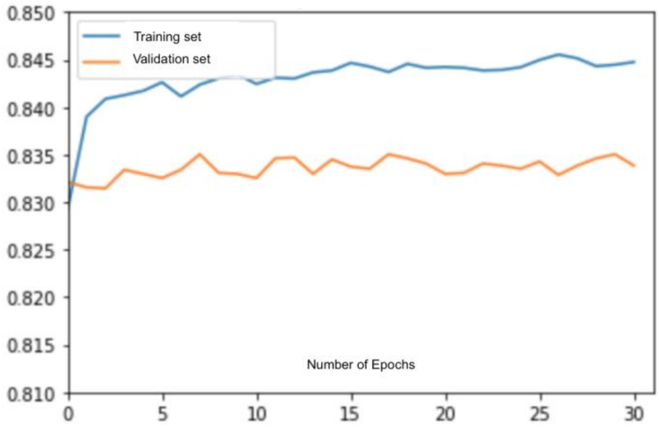

Figure 5. *Training and Validation Accuracy during training*

3. THE RESEARCH FINDINGS AND DISCUSSION

In order to observe the performance of predicting the classes ("healthy", "critical condition", "repair engine", "engine failure") by using the test data set, the confusion matrix was created. According to the confusion matrix shown in Table 5, "healthy" was predicted correctly with 90% accuracy, "critical condition" with 74% accuracy, "repair engine" with 65% accuracy, and "engine failure" with 91% success.

Table 5. *Confusion Matrix for the proposed network*

		Observed labels			
		Engine Failure	Repair	Critical Condition	Healthy
Predicted Labels	Engine Failure	0.91	0.22	0.004	0.002
	Repair	0.09	0.65	0.17	0
	Critical Condition	0	0.14	0.74	0.08
	Healthy	0	0	0.09	0.90

In order to compare the performance of the proposed network with the performances of:

- LSTM network containing input, a single LSTM layer, a hidden layer, and an output layer used commonly in the literature for classification purposes.
- LSTM network containing input, one hidden layer, two LSTM layers, and an output layer proposed by Aydin et al. for classification purposes.

Three networks, including the proposed network, were trained on the same dataset. The network consisting of a single LSTM layer was named LSTM1. The network consisting of two LSTM layers and one hidden layer was named LSTM2. The network proposed in this study was named LSTM3. As seen in Table 6, adding a second LSTM layer helps increase the training set's accuracy from 71% to 81%. The accuracy of the validation set from 72% to 81%.

Table 6. *Train and Validation accuracy for different LSTM architectures*

Architecture	Accuracy(Train set)	Accuracy(Validation set)
LSTM	0.7128	0.7242
LSTM2	0.8051	0.8137
LSTM3	**0.8449**	**0.8338**

With the help of hidden layers, dropout layers, and the proposed method to find optimum hyperparameters, the training set's accuracy increased from 81% to 84%, and the accuracy of the validation set from 81% to 83%. According to the results, the architecture proposed in the study (LSTM3) provided the best performance.

4. RESULTS

The machines are now equipped with sensors to make more effective decisions regarding systems' status. In the manufacturing field, IoT technology is an essential element of PdM. Data can be collected from systems or machines through sensors with IoT technology. With the help of predictive models developed using the data, errors/failures in the system or machine, which otherwise cannot be detected by the human, can be detected in advance.

Poorly planned maintenance scenarios can have a negative impact on production. Predicting failures in advance can help eliminate a machine or system's downtime while improving employees' safety working in production plants. The approach proposed in this study provides better performance compared to other LSTM networks presented in the literature for predictive maintenance purposes.

CONFLICTS OF INTEREST

No conflict of interest was declared by the authors.

REFERENCES

[1] Ran, Y., Xin, Z., Pengfeng, L., Yonggang, W., Ruilong, D., "A survey of predictive maintenance: Systems, purposes and approaches", arXiv, (2019).

[2] Zhou, Q., Junbo, S., Shiyu, Z., Xiaofeng, M., Mutasim, S., "Remaining useful life prediction of individual units subject to hard failure", IIE Transactions 46, (10): 1017-1030, (2014).

[3] Jardine, A. K.S., Daming, L., Dragan, B., "A review on machinery diagnostics and prognostics implementing condition-based maintenance", Mechanical Systems and Signal Processing, 20(7): 1483-1510, (2006).

[4] Heimes, F. O., "Recurrent neural networks for remaining useful life estimation", International conference on prognostics and health management, United States, 1-6, (2008).

[5] Porotsky, S., Zigmund, B., "Remaining useful life estimation for systems with non-trendability behaviour", IEEE Conference on Prognostics and Health Management, United States, 1-6, (2012).

[6] Jain, A. K., Pradeep, K., Bhupesh, K. L., "Prediction of Remaining Useful Life of an Aircraft Engine under Unknown Initial Wear", 5th international & 26th all India manufacturing technology, design and research conference, Guwahati, India, 1–5, (2014).

[7] Babu, G.S., Li, X. L., Suresh, S., "Meta-cognitive regression neural network for function approximation: application to remaining useful life estimation", International Joint Conference on Neural Networks (IJCNN), Canada, 4803-4810, (2016).

[8] dos Santos Lima, F.D., Amaral, G.M.R., de Moura Leite, L.G., Gomes, J.P.P., de Castro Machado, J., "Predicting failures in hard drives with lstm networks", Brazilian Conference on Intelligent Systems (BRACIS), Brazil, 222-227, (2017).

[9] Aydin, O., Guldamlasioglu, S., "Using LSTM networks to predict engine condition on large scale data processing framework", 4th International Conference on Electrical and Electronic Engineering (ICEEE), Turkey, 281-285, (2017).

[10] Zaifa, C., Yan-cheng, L., Siyuan, L., "Mechanical State Prediction Based on LSTM Neural Network", 36th Chinese Control Conference (CCC 2017), China, 3871-3876, (2017).

[11] Das, A., Mueller, F., Siegel, C., Vishnu, A., “Desh: deep learning for system health prediction of lead times to failure in HPC”, 27th International Symposium on High-Performance Parallel and Distributed Computing (HPDC ’18), United States, 40 - 51, (2018).

[12] Cachada, A., Barbosa, J., Leitño, P., Gcraldcs, C. A., Deusdado, L., Costa, J., Romero, L., “Maintenance 4.0: Intelligent and Predictive Maintenance System Architecture”, 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Portugal, 139-146, (2018).

[13] Öztanır, O., “Makine Öğrenmesi Kullanılarak Kestirimci Bakım”, Msc. Thesis, Hacettepe University, Turkey, (2018).

[14] Zhang, W., Wuwu, G., Xin, L., Yan, L., Jiehan, Z., Bo, L., Qinghua, L., Su, Y., “LSTM-based analysis of industrial IoT equipment.”, IEEE Access, (6): 23551-23560, (2018).

[15] Baker, H., “The Impact of Digital on Unplanned Downtime: An Offshore Oil and Gas Perspective”, https://www.ge.com/digital/sites/default/files/download_assets/ge-the-impact-of-digital-on-unplanned-downtime.pdf, (2016).

[16] Milojevic, D. M., Nassah, F., Digital Industrial Revolution with Predictive Maintenance, General Electrics, https://www.ge.com/digital/sites/default/files/download_assets/PAC_Predictive_Maintenance_GE_Digital_Executive_Summary_2018_1.pdf, (2018).

[17] Hochreiter, S., Schmidhuber, J., “Long Short-Term Memory”, Neural Computation, 9(8): 1735-1780, (1997).

[18] Koza, J. R., “Genetic Programming: on the Programming of Computers by Means of Natural Selection”, MIT Press, Cambridge, (1992).

[19] Goldberg, D. E., Holland, J. H., “Genetic algorithms and machine learning, Machine Learning”, 3(2): 95–99, (1998).

[20] Goldberg, D. E., “Genetic Algorithms in Search, Optimization, and Machine Learning”, Addison-Wesley, New York, (1989).

[21] Wicaksono, A. S., Supianto, A. A., “Hyper Parameter Optimization using Genetic Algorithm on Machine Learning Methods for Online News Popularity Prediction”, International Journal of Advanced Computer Science and Applications, 9(12): 263-267, (2018).

[22] Bouktif, S., Fiaz, A., Ouni, A., Serhani, M.A., “Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches”, Energies, 11(7): 1636, (2018).

[23] Chung, H., Kyung-shik, S., “Genetic algorithm-optimized long short-term memory network for stock market prediction”, Sustainability, 10(10): 3765, (2018).

[24] Saxena, A., Goebel, K., Simon, D., Eklund, N., “Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation”, International Conference on Prognostics and Health Management, United States, 1-10, (2008).