Received: January 28, 2025 | Revised: February 12, 2025 | Accepted: February 14, 2025

JOURNAL OF APPLIED RESEARCH ON SCIENCE AND TECHNOLOGY
JARST
ISSN ONLINE: 2773-9473
https://ph01.tci-thaijo.org/index.php/rmutt-journal
INSTITUTE OF RESEARCH AND DEVELOPMENT
RAJAMANGALA UNIVERSITY OF TECHNOLOGY THANYABURI

A predictive model for highly efficient helicopter maintenance in the Royal Thai Air Force using deep learning

Somkait Hoonsakul*, Prasong Praneetpolgrang and Payap Sirinam

Graduate School, Navaminda Kasatriyadhiraj Royal Thai Air Force, Saraburi 18180, THAILAND
*Corresponding author: Som.hoonsakul@gmail.com

ABSTRACT

The Royal Thai Air Force has helicopters in service to support tactical transport missions. Over time, helicopters deteriorate, making maintenance essential to maintain mission capabilities. Regular and timely maintenance helps to maintain operational readiness and safety, reduces the risks associated with unexpected failures, and ensures the continuous availability of critical resources. In this research, the ultimate goal is to use the results of this research as a guideline for improving the Royal Thai Air Force's helicopter maintenance plans to be good and efficient, where efficient maintenance planning can significantly reduce costs and enhance safety. This research provides valuable insights for academia and aviation industry professionals. The researcher has proposed a model to predict helicopter maintenance in the Royal Thai Air Force to improve maintenance efficiency and increase the accuracy of spare parts calculations. Using a helicopter maintenance dataset from January 2017 to September 2020, a total of 3,819 datasets covering a variety of maintenance scenarios and operating conditions, the researchers applied deep learning (DL) techniques to make predictions. The algorithms used in this research include fully connected neural networks (FCNN), long-term short-term memory (LSTM), and convolutional neural networks (CNN). FCNN is suitable for general numerical data that are not related in sequence or space, making it effective for linear or simple numerical datasets. On the other hand, LSTM is ideal for analyzing time-sequence data because it can capture past trends to predict future outcomes. CNN excels in handling spatially correlated data, especially those related to helicopter maintenance patterns that require analyzing multiple related factors. The results show that FCNN achieves an accuracy and precision of 1.00, while both LSTM and CNN achieve an accuracy and precision of 0.94. The results of this study clearly highlight the potential of DL-based models to improve prediction accuracy. However, the study's limitations may lie in the accuracy of deep learning model in predicting the Royal Thai Air Force's helicopter maintenance. The future direction could be to develop more accurate predictive maintenance guidelines for the Royal Thai Air Force's helicopters. The specific research gap this study, by improving deep learning algorithms and collecting more diverse data from the Royal Thai Air Force's helicopters maintenance, resulting in increased accuracy. To use the results of this research as a guideline for improving The Royal Thai Air Force's helicopter maintenance plans to be good and efficient. This will indirectly result in reducing the helicopter maintenance budget of the Royal Thai Air Force and increasing the reliability of the operations of the Royal Thai Air Force.

Keywords: Predictive maintenance, Helicopter, Deep learning

INTRODUCTION

Application of deep learning techniques for helicopter maintenance [1-2] of the Royal Thai Air Force. Helicopters are essential aircraft for tactical transport and public rescue missions. Due to the long service lifetime of helicopters in the Royal Thai Air Force, maintenance is crucial to maintain operational capability.

Research indicates that maintenance personnel are the most important factor, they must have comprehensive knowledge, experience, and certification. However, most experienced technicians are nearing retirement age. In addition, helicopters are composed of many parts with varying service life, while the Royal Thai Air Force operates under a limited maintenance budget.

Currently, the government lacks comprehensive support for the aircraft maintenance industry, including production, operation, and improvement. This gap extends to capital management, infrastructure, and competitiveness development. The use of artificial intelligence as a management and decision-support tool is a practical solution.

This research explores the use of deep learning to predict helicopter maintenance needs [3-4] using datasets from spare parts requests, flight records,

DOI: 10.60101/jarst.2024.260451

maintenance schedules, instrument calibration history, and expert technician level to develop the accurate predictive models. Policy development should focus on aligning Thailand's aircraft maintenance standards with international standards (ICAO, EASA, and FAA); while developing specific regulations and enhancing personnel capabilities. This approach aims to make Thailand a regional maintenance hub, enhance national security, and reduce dependence on foreign technologies through sustainable development.

MATERIALS AND METHODS

This research presents a literature review on the application of deep learning techniques in predictive maintenance, covering models such as FCNN, LSTM, and CNNs, as well as case studies in various industries such as aviation and manufacturing.

Deep learning is a part of machine learning, which is an algorithm used for learning that can enable machines to make decisions like humans. Machine learning is the application of statistical knowledge to analyze data and create models to predict the results from the data.

The starting point of deep learning is the artificial neural network, which is an algorithm invented by imitating the functionals of the human brain. The human brain has complex functions and can analyze a large number of things efficiently. Artificial neural networks simulate the functioning of neurons, each of which has connections to send information to each other for decision-making. The highlight of the human brain is that each neuron can be connected thoroughly and has a clear distribution of data analysis for each cell [5].

Artificial neural networks are designed to work similarly to the human brain. Behind the scenes of an artificial neural network, some subunits work similar to human neurons called nodes. Each node can be combined into several layers called layers. Each node has a working process divided according to the function of each layer, such as Input Layer, Hidden Layer, and Output Layer, as shown in Figure 1.

From Figure 1, the operation of Nodes is based on both Linear Regression and can be represented as in Equation (1). Each Node has small components called Weight, which is comparable to the Intercept value from Linear Regression. It is used to determine the weight of each variable used in the analysis and Bias, which is comparable to the Coefficient in Linear Regression, which compares Node to the work of Linear Regression. In the training process, training will be done to find parameter values that are appropriate to the data used for used prediction. Then the obtained parameters are used to build a model for use in predicting results by training. The Neural Network must be trained in rounds. In one round, it must be trained with every data used for coach in 1 round, it must be trained with all data used for training, which is called the number of rounds in training called Epoch.

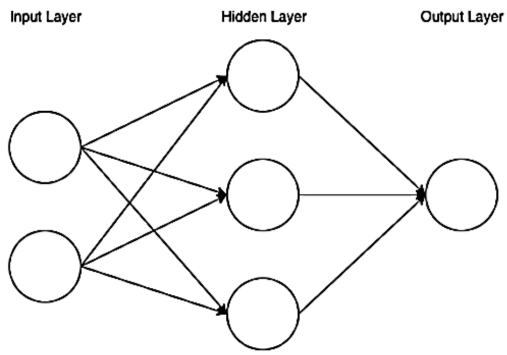

Figure 1 The structure of an artificial neural network has a sub-computing part called a Node and the arrangement of each Node in layers called Layer.

$$\text{Neural Network } (m,n) = \text{activation } (W_{mn} i_n + b_m) \quad (1)$$

Where

Neural Network (m,n) = Represents the results of the neural network topology model.

Activation = Represents the function used to convert the values obtained from the calculation of the weights of the neural network structure.

Mn^i = Represents the weights of nodes in the neural network structure.

N = Represents the input value of the artificial neural network structure.

B_m = Represents the bias of the neural network structure.

There are many types of artificial neural networks, which can be divided by the training process. Examples of popular types of artificial neural network structures include feedforward neural network, which feeds data from the front to the back, such as perception, etc., and backpropagation neural networks, which feed data and learn by feeding the results in reverse. Deep learning is one of the algorithms that is developed from artificial neural networks, which imitates the learning of the human nervous system. It is developed by laying down a structure in the form of stacking many layers, both in the form of stacking in the form of the same type in every layer and in the form of stacking in each layer, which works differently. It is caused by applying the capabilities of each structure used together to increase the efficiency of analysis [6].

Deep learning is currently providing better analysis and prediction efficiency than the use of traditional machine learning algorithms. Both in terms of the amount of data that has increased dramatically and the more efficient computer processors, deep learning can use a large number of variables in analysis to increase the efficiency of analysis. In addition to

DOI: 10.60101/jarst.2024.260451

increasing the number of layers, deep learning has adjusted the learning ability of the algorithm by taking learning as Backpropagation is improved by separating the learning algorithm into 2 functions: Loss Function and Optimize Function. The Loss Function is used to calculate the error value obtained by comparing the results from the model and the results from training. Then, the resulting Loss value is used with the Optimize Function, which is a function for adjusting the parameters used in learning the created model. There are various Loss Functions and Optimize Functions available for use today. Currently, deep learning has invented various algorithms and structures to respond to the needs of various forms of analysis. The currently popular algorithms are as follows:

Fully Connected Neural Networks (FCNN) or Dense Neural Networks (DNN), Long Short-Term Memory (LSTM) Convolutional Neural Networks (CNNs), etc. The increasing popularity of deep learning is due to the higher analysis efficiency and the rapid development of hardware technology, both high-performance processors and the application of a Graphic Processing Unit (GPU) in deep learning, which can process faster than the Central Processing Unit (CPU) because the GPU processes in the form of Matrix Parallels, while the CPU processes in the form of Serial. [7] Fully Connected Neural Network (FCNN) or Dense Neural Network is the simplest implementation of deep learning algorithms. It is an extension of the neural network by adding several layers, resulting in more variables for feature extraction. To increase the efficiency in analysis and prediction [8], the operation of the structure is similar to the artificial neural network, as shown in Figure 2.

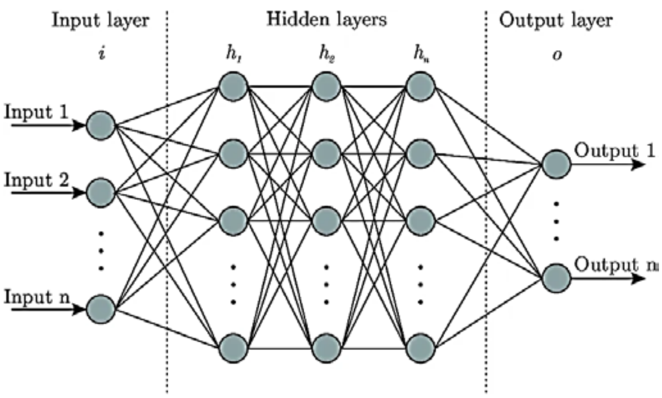

Figure 2 Example structure of the Fully Connected Neural Network (FCNN) algorithm.

Fully Connected Neural Network (FCNN) or Dense Neural Network has the most similar function and structure to artificial neural networks, but has added more layers for analyzing data features. The larger number allows for a more detailed analysis of features and training using the principle of Backpropagation in training.

Fully Connected Neural Network (FCNN) or Dense Neural Network has been used to analyze data in a structured form. In analyzing FCNN or DNN data, the analysis is performed without destroying the original data, allowing for accurate and highly efficient analysis.

Fully Connected Neural Network (FCNN) [9-11] is one of the most important technologies in the world of AI, especially in Deep Learning. The highlight of FCNN is the completed connection between neurons in each layer. That is, neurons in one layer can be completely connected to all neurons in the next layer. This connection allows FCNN to process a wide range of data, covering everything from basic and easy-to-understand data to complex data that requires multidimensional analysis. FCNN works from the input layer, which is the first layer that receives raw data into the system, such as numerical data, statistics, or data related to equipment maintenance. This data is then passed to the hidden layers, which are the core of FCNN. These layers transform the data into a format that the system can easily process. There may be many hidden layers, and each layer contains many processing units.

The more layers or processing units there are, the more complex the system can learn and capture. However, adding hidden layers or the number of processing units in each layer must be done carefully. If the system is too complex, it may lead to overfitting, which means that the system learns too well from the sample data and cannot be used to learn new data effectively. Once the data has passed through all the hidden layers, this data is then forwarded to the Output Layer, which is the final layer of the system. This layer is responsible for combining all the results from the previous layers and converting the results into a format that meets the desired goals, such as data classification or outcome prediction. For example, in the helicopter maintenance analysis, FCNN is used to classify the status of equipment into three categories: equipment that cannot be repaired, equipment that can be repaired but is not worth it, and equipment that can be repaired but is worth investing in. What makes FCNN outstanding and interesting is its flexibility. The system can design hidden layers that are appropriate for each type of data. For example, for data that does not have a clear pattern or structure, FCNN can help find patterns or relationships in the data that may not be obvious at first. This makes FCNN an important tool in analyzing numerical data, such as sales forecasts, financial risk assessments, or analysis of various factors in the industry [12-15].

Long Short-Term Memory (LSTM) is an algorithm that is developed from RNN by solving the problem of Gradient Vanishing by designing a new Cell that can store the state of the computation [16]. In the Cell of LSTM, there is a sub-computation unit called a Gate, which consists of an input Gate, Forget Gate,

DOI: 10.60101/jarst.2024.260451

Memory Cell State Gate, and Output Gate as shown in Figure 3.

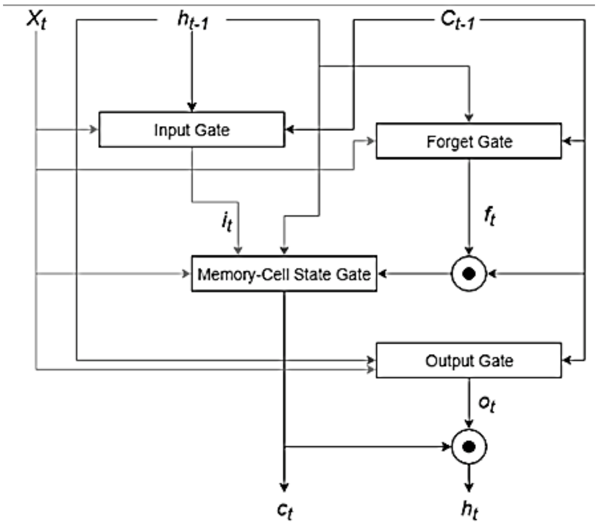

Figure 3 Example of working in a cell of the LSTM neural network algorithm.

Input Gate is a sub-unit to determine the data to be analyzed in Cell by receiving data to write values into each Cell as in Equation (2) [17].

$$i_t = \sigma\,(W_{xi}x_t + W_{hi}\,h_{t-1} + W_{ci}c_{t-1} + b_i) \qquad (2)$$

Where

i_t = represents the result obtained from the Input Gate.
σ = represents the Sigmoid function.
W_{xi} = represents the weight value for calculating the Input in the Input Gate.
x_t = represents the input value that is entered for calculation.
W_{hi} = represents the weight value for calculating the Hidden State in the Input Gate.
h_{t-1} = represents the Hidden State value obtained from the calculation in the previous time unit.
W_{ci} = represents the weight value for calculating the Memory Cell State in the Input Gate.
c_{t-1} = represents the Memory Cell State value obtained from the calculation in the previous time unit.
b_i = represents the Bias value used in the calculation of the Input Gate.

Forget Gate is a subunit used to determine the data to be analyzed in Cell by determining whether the data should be recorded or forgotten, it can be defined as Equation (3) [17].

$$f_t = \sigma\,(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \qquad (3)$$

where

f_t = represents the result obtained from Forgot Gate.
σ = represents the Sigmoid function.
W_{xf} = represents the weight value for calculating the Input in Forgot Gate.
x_t = represents the input value that is entered for calculation.
W_{hf} = represents the weight value for calculating the Hidden State in Forgot Gate.
h_{t-1} = represents the Hidden State value obtained from the calculation in the previous time unit.
W_{cf} = represents the weight value for calculating the Memory Cell State in Forgot Gate.
c_{t-1} = represents the Memory Cell State value obtained from the calculation in the previous time unit.
b_f = represents the Bias value used in the calculation in Forgot Gate.

Memory Cell State Gate is a sub-unit to define the data that is entered for analysis in the Cell and calculating the state value to be used in the next calculation, with equation (4) [17].

$$c_t = f_t{\cdot}c_{t-1} + i_t{\cdot}tanh\,(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \qquad (4)$$

where

C_t = represents the Memory Cell State in the time interval.
f_t = represents the result obtained from Forget Gate.
c_{t-1} = represents the Memory Cell State from the previous time interval.
i_t = represents the result obtained from the Input Gate.
$tanh$ = represents the Hyperbolic tangent function.
W_{xc} = represents the weight for calculating the Input from the Memory Cell State Gate.
x_t = represents the Input value that is entered into the calculation.
W_{hc} = represents the weight for calculating the Hidden State in Memory Cell State Gate.
h_{t-1} = represents the Hidden State obtained from the calculation in the previous time interval.
b_c = represents the Bias value used in the calculation in Forgot Gate.

The Output Gate is a sub-unit for calculating the Output of the Cell, which has two results: Output and Hidden State for use in the next calculation, with equations as Equations (5) and (6) respectively [17].

$$o_t = \sigma\,(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o) \qquad (5)$$

$$h_t = o_t \cdot \tanh(c_t) \qquad (6)$$

DOI: 10.60101/jarst.2024.260451

Journal of Applied Research on Science and Technology

where

o_t = represents the result obtained from the Output Gate.

σ = represents the Sigmoid function.

W_{xo} = represents the weight value for calculating the Input in the Output Gate.

x_t = represents the input value that is entered into the calculation.

W_{ho} = represents the weight value for calculating the Hidden State in the Output Gate.

h_{t-1} = represents the Hidden State value obtained from the calculation in the previous time unit.

W_{co} = represents the weight value for calculating the Memory Cell State in the Output Gate.

c_{t-1} = represents the Memory Cell State value obtained from the calculation in the previous time unit.

b_o = represents the Bias value used in the calculation of the Output Gate.

h_t = represents the Hidden State value from the calculation.

Convolutional Neural Networks (CNNs) [18-20] are an innovation that plays a major role in the world of artificial intelligence, especially deep learning, which is currently very popular. CNNs are designed to support the analysis of spatial data such as images, videos, audio, or data with physical structures, especially in tasks that require detecting patterns or relationships in the data that humans may not be able to see. The structure of CNNs makes them versatile tools that can systematically explore and transform raw data into insights. The process begins with a convolutional layer, which is the heart of CNNs. Small filters in this layer act like microscopes that scan the original data piece by piece, whether it is an image region, a specific outline, or a shape. Each filter has a specific role. Some filters look for image regions, some detect color intensity; or even small characteristics that are the basis of further analysis. Once the data has been scanned and has important features, it is passed to the dimensionality reduction layer, which compresses the data while preserving the important features of the data. This process allows the system to process faster and reduce unnecessary complexity. A popular example of dimensionality reduction is Max Pooling, which selects only the largest values in each subset of data. As with filtering the essence of the data, the data that passes through this layer is reduced in dimensionality; but still has clarity in important aspects.

Once the data has been processed and reduced in dimensionality, it enters a fully connected layer, which is like the center of all the data in the system. This layer is responsible for collecting the results from the previous steps and analyzing them to get the final answer, whether it is image classification, outcome prediction, or complex decision-making. This layer is responsible for connecting the scattered data to the big picture, allowing the system to make accurate decisions.

The highlight of CNNs is the ability to extract features hidden in the data efficiently without having to rely on manual feature design as in the past. This makes CNNs very flexible and can be used for specific tasks that require specific accuracy, such as diagnosing diseases from X-rays, interpreting languages from images, and detecting objects in videos, or for general tasks that require intelligent systems that can learn by themselves.

The great thing about CNNs is their ability to learn and understand data at a level that is more complex than humans can do in a short period, bringing our world closer to innovations such as self-driving cars, intelligent robots, or accurate and deep language translation systems.

In an era where data is becoming increasingly complex, CNNs are an ally in translating that complexity into simple, actionable answers. Here's why CNNs are a key part of driving the rapid advancement of AI.

Convolutional Neural Networks (CNNs) are deep learning algorithms that work similarly to human eye scanning by dividing the features into groups for analysis and using the newly obtained features to predict the results. CNNs are outstanding in Feature Extraction from data sets, focusing on finding features from data sets in the form of groups of data [21].

The CNN algorithm is divided into 2 parts: Feature Extraction and Classification. Feature Extraction is a work to select features for use in predicting the results in the Classification step, which is the next step for Feature Extraction of CNNs. It is the use of Filters to select Features by defining the size of the Filter used for data selection. This Filter is in the form of a Matrix, working by placing it on the data set to define the area used for analysis and processing, as shown in Figure 4.

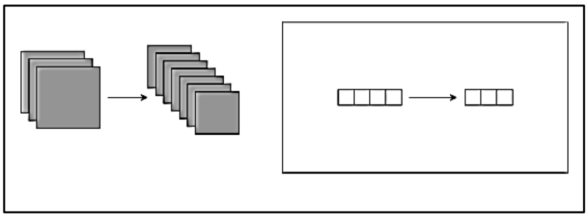

Figure 4 Example of working of CNNs algorithm.

From Figure 1, the CNN algorithm uses filters to create new feature sets. Once we have new features, we can reduce the size of the features and still maintain the identity of the original data without distortion. There are 2 algorithms to choose from: Max Pooling and Average Pooling. Max Pooling is creating another filter to use in data analysis. Then, the highest value

DOI: 10.60101/jarst.2024.260451

in the filter is extracted for use, as shown in Figure 4. Average Pooling creates a filter similar to Max Pooling, but it extracts the average of the values of the filter, as shown in Figures 5 and 6.

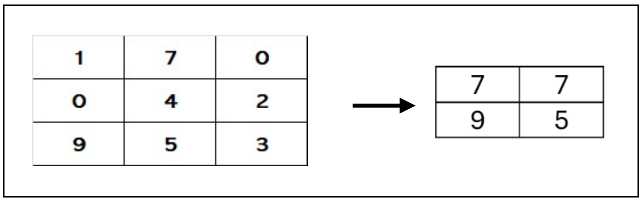

Figure 5 Data obtained from the Filter is passed through a 2x2 Max Pooling.

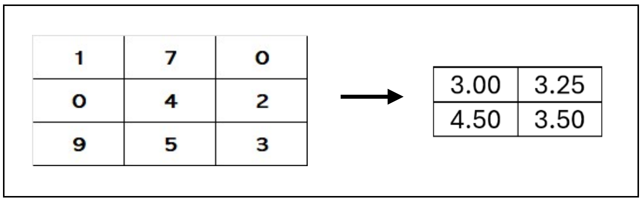

Figure 6 Data obtained from the filter is passed through a 2x2 Average Pooling.

This research uses three models to cover all data characteristics, where FCNN is suitable for basic data, LSTM is suitable for temporal data, and CNNs are suitable for spatial data. All of these make the development of the helicopter maintenance prediction model the most comprehensive and accurate. This research aims to develop and test deep learning models in various algorithms to predict helicopter maintenance of the Royal Thai Air Force. The research process consists of steps as shown in Figure 7.

From Figure 7, the research process starts with data collection. Then, the data is processed into a form suitable for model use, starting with the selection of important features [22-23] to create the necessary data framework for input into the model. Then, models are selected for testing, model construction, and training, model suitability is checked, and the performance of the prediction model is measured to obtain the most accurate and suitable model. For data collection, researchers collect various types of data, such as helicopter maintenance history, spare parts list, flight data, maintenance plans, history and calibration list of maintenance tools and equipment, and information about maintenance technicians who can repair helicopters. This data is made into a dataset for training a deep-learning model to help predict future helicopter maintenance trends. The details of each feature are shown in Table 1. Consider and analyze the prediction model. The researchers conducted further research on three more datasets. The remaining 80% is used to verify the diagnosis results. If the results are not good, a new fiber will be created. The new algorithm will optimize the internal structure of the prediction model before doing it again until the best prediction model is obtained. The remaining 20% of the data will be used to test and compare the three models.

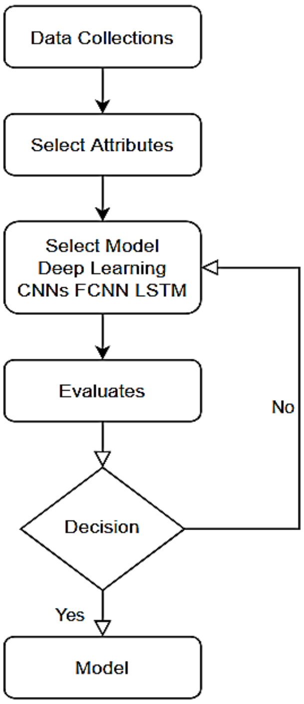

Figure 7 Research process.

Table 1 List of attributes and data attributes.

No.	Attribute	Data Property	Description
1	Car_Number	Discrete	Number of Aircraft/Helicopters.
2	Working_Status	Discrete	Maintenance status.
3	Skill_Level	Discrete	Technically skill level.
4	Years_Left	Discrete	The number of remaining years before a part or device expires.
5	Manual_Available	Discrete	Is there a manual for use?
6	Tool_Name	Discrete	Name of the tool used in the maintenance process.
7	Tool_Size	Discrete	Size of tools used.
8	Calibrated	Discrete	Instrument calibration status.
9	Operational	Discrete	Operational status of parts or equipment.

DOI: 10.60101/jarst.2024.260451

No.	Attribute	Data Property	Description
10	Part_Name	Discrete	The name of the part in use or under Maintenance.
11	Installation_Date	Discrete	Date of installation of the part.
12	Hours_Used	Discrete	Number of hours the part or equipment has been used.
13	Minutes_Used	Discrete	Number of usage times.
14	Expiration_Date	Discrete	The expiration date of the part or device.
15	Date_Now	Discrete	Current date.
16	leDay	Discrete	Forecast data
17	Management	Discrete	Situation management
18	statusMaintenance	Discrete	Maintenance situation

Data Collection The researchers collected data on helicopter maintenance history, spare parts list, helicopter flight data, helicopter maintenance plan, instrument calibration history, maintenance equipment calibration list, and helicopter mechanics who can perform helicopter maintenance. This data was used as the baseline data for creating a dataset used to train a deep learning prediction model, which enables the prediction of future helicopter maintenance trends. The details of each feature are shown in Table 1.

To measure the performance of the model in predicting helicopter maintenance with three outcomes (0, not repairable; 1, repairable but poor performance; 2, repairable and good performance), the researchers used a Confusion Matrix [24], which shows the number of correct and incorrect predictions in each category. The calculation Equations (7)-(10) are as follows:

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+FP+TN+FN)} \tag{7}$$

$$\text{Precision} = \frac{TP}{(TP+FP)} \tag{8}$$

$$\text{Recall} = \frac{TP}{(TP+FN)} \tag{9}$$

$$\text{F1-Score} = \frac{(TP+TN)}{(TP+FP+TN+FN)} \tag{10}$$

Where TP = The predicted data is correct and the actual result is correct.
TN = The predicted data is incorrect and the actual result is incorrect.
FP = The predicted data is correct but the actual result is wrong.
FN = The predicted data is incorrect but the actual result is correct.

The selection of evaluation metrics in Accuracy, Precision, Recall, and F1-Score in this research is important because the nature of maintenance prediction problems involves imbalanced data and the consequences affect the success or failure of decision-making. For example, if the forecasting system fails to identify helicopters that need urgent maintenance, it may lead to safety risks. Therefore, it is necessary to consider metrics that not only measure general accuracy but also focus on the ability to handle unequally distributed data in the data set (Class Imbalance).

Precision and Recall are important metrics in this context because Precision measures the accuracy of the prediction in the group that the system indicates as "maintenance required", which reduces false positives that may lead to unnecessary maintenance. On the other hand, Recall helps assess the system's ability to find cases that are "really maintenance required", which is crucial to avoid false negatives. Neglecting a helicopter that needs maintenance may lead to system failure or mission risk. Using both Precision and Recall ensures that the system can fully meet these requirements.

The F1-Score is used to balance Precision and Recall, especially in cases of imbalanced data, for example, if the prediction is correct in a large group (e.g. maintenance-free helicopters) but fails in a small group (maintenance helicopters). Accuracy may look good, but it does not reflect the true performance of the system. The F1-Score provides a combined measure of Precision and Recall, without focusing on one or the other, allowing for a comprehensive evaluation of the system that is most suitable for the research goal of predicting maintenance accurately and efficiently.

RESULTS AND DISCUSSION

In this study, the researchers trained all the models for 50 iterations, dividing the prediction output into three types of inputs to examine the difference between the features added to the three prediction models to see how these features affect the accuracy of the prediction models. In the first case, the model input is determined based on SEM, with the input features being Working Status, Operational, leDay, and Management as shown Figure 8.

From the Heat Map Diagram, we can see the relationship between different variables. Related to equipment maintenance in the Air Maintenance Practice Dataset, for example, in the picture there is an interesting point: Skill Level and Years Left are negatively

DOI: 10.60101/jarst.2024.260451

related (-0.96). This means that people with higher skills tend to have shorter careers. Moreover, the relationship between the number of hours used (Hours Used) and the length of time before maintenance (IeDay) have a value of 0.62, indicating that if the equipment is used more, it will be time for maintenance sooner. This information can help us predict and plan maintenance more accurately. Another interesting point is Management has a relationship with the duration of equipment use (IeDay), a correlation value of 0.62, this means that good management can help control and extend the life of your equipment. Meanwhile, Management and Status Maintenance have a negative correlation (-0.36), indicating that good management may help reduce the frequency of maintenance. This information can be used to improve planning and increase the efficiency of maintenance systems. Helps reduce costs and increase equipment availability for use.

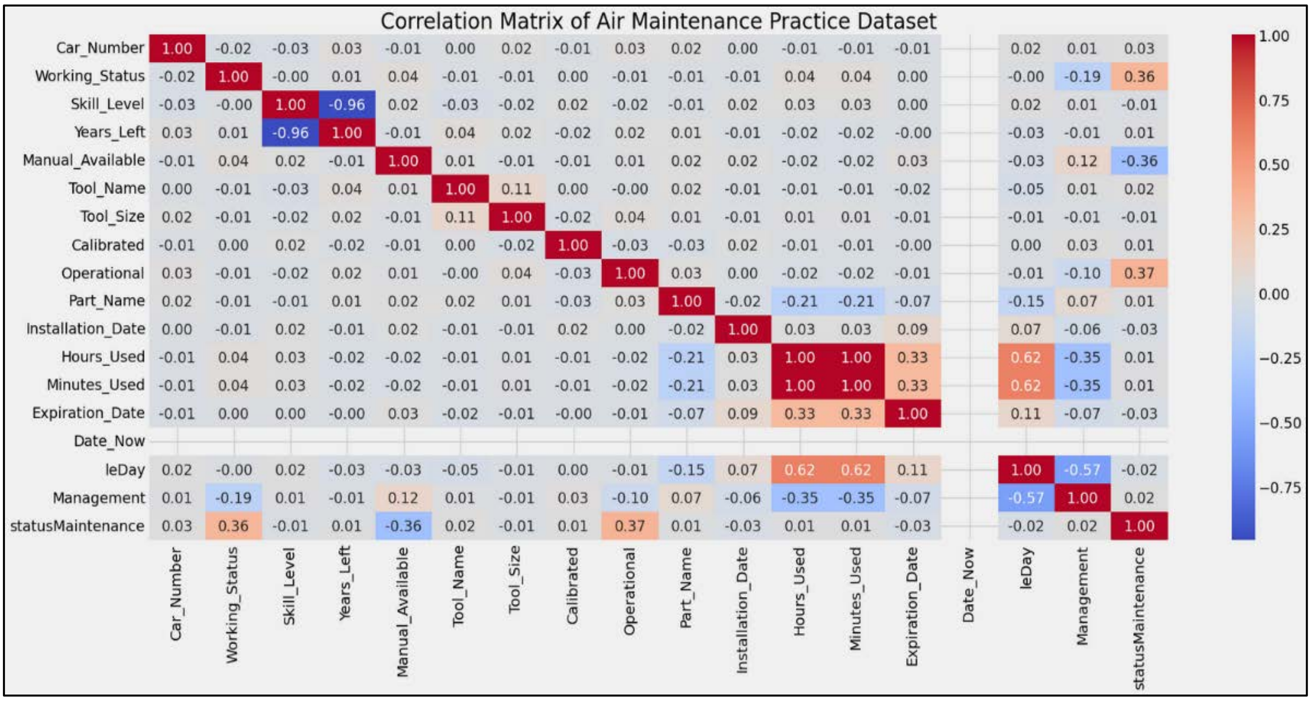

Figure 8 Heat Map diagram shows the relationship of each attribute.

Table 2 Hyperparameters, or values that are set before training the model without being learned by the model itself.

Algorithm	Learning Rate	Batch Size	Number of Layers	Epochs
FCNN	0.001	32	Dense 64, relu Dense 32, relu Dense 1, sigmoid	50
CNNs	0.001	32	LSTM 64, Droput (0.2) Dense 32, relu Dense 1, sigmoid	50
LSTM	0.001	32	Conv2D (32, (1,1)), relu MaxPooling1D (2), Flatten Dense 1, sigmoid	50

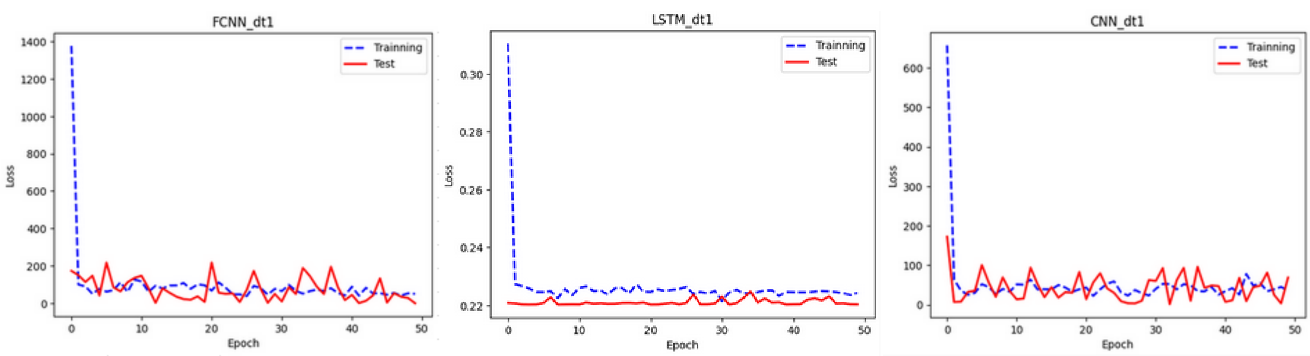

Figure 9 Loss of SEM-based feature-based models Case 1.

DOI: 10.60101/jarst.2024.260451

This research has set Hyperparameters, or values that are set before training the model without being learned by the model itself. Details are as shown in Table 2.

The loss test results are shown in Figure 9 From Figure 10, it can be seen that the loss value result is not very good. In each model, the test results fluctuate, indicating that the model still needs a large amount of training data, and the internal structure of the model still needs to be less complex. The experimental results are shown in Table 3.

From Table 3, it can be seen that all the models have the value of 1, indicating that the model is overfitting, or the model has learned too much and cannot be used in practice. This experimental result indicates that if we introduce features based on SEM, it will result in overfitting of the model.

Table 3 In case 1 results of the predictive model by specifying the input of the reference model according to SEM.

Algorithm	Time	Loss	Accuracy	Precision	Recall	F1-Score
FCNN	18	3.2284	1	1	1	1
CNNs	26	1.113	1	1	1	1
LSTM	19	0.0818	1	1	1	1

The heatmap [25] shown here helps us to understand the relationship between data in the equipment maintenance dataset more easily. The correlation value in the heatmap tells us how much two variables are related. If the value is close to 1, there is a clear positive correlation between the two variables, i.e. when the value of one variable increases, the other variable tends to increase. On the contrary, if the value is close to -1, there is a negative correlation between the two variables, i.e. when the value of one variable increases, the other variable tends to decrease. If the value is close to 0, there is no clear correlation between the two variables.

From the heatmap, we can see that some variables have a significant impact on the maintenance (Status Maintenance). For example, Working Status and Operational have a positive correlation, meaning that the working status and usage of the equipment are important factors in determining the need for maintenance. In addition, the variable Minutes Used, which represents the time the equipment is used, is also highly relevant, meaning that equipment that is used more often requires more maintenance. At the same time, some variables, such as Skill Level and Management, have a negative correlation, meaning that when employees are more skilled or better managed, the need for maintenance may be reduced.

In addition, some variables do not have a clear impact on maintenance, such as Tool Name and Part Name, which have correlation values close to zero, so it can be considered that these variables may not be necessary in the forecasting model.In case 2, the features are input including Operational, leDay, Management, Working_Status, Car_Number, Tool_Name, Management, Years_Left, Calibrated, Part_Name, Hours_Used, and Minutes_Used, which are combined features from SIM and Heat Map with values greater than 0, with the following Loss values as shown in Figure 10.

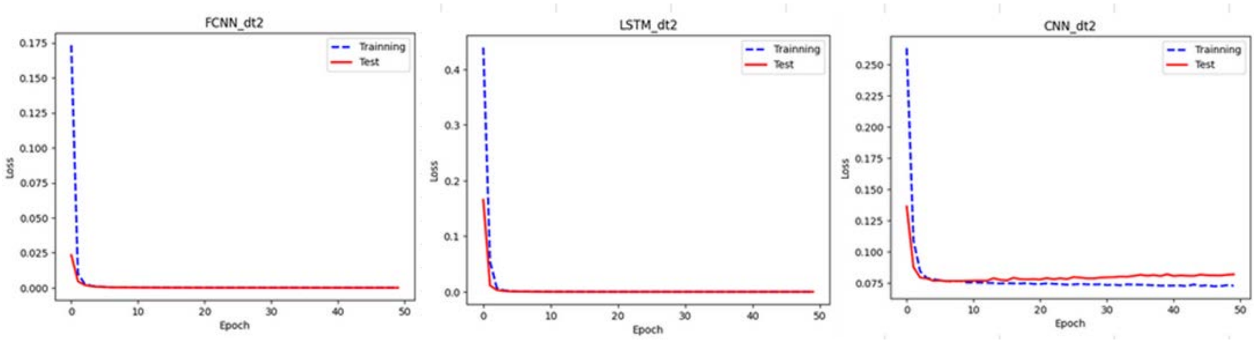

Figure 10 Loss of SEM-based feature-based models.

From Figure 11, it can be seen that all prediction models have Test Loss values that are very close to the Training values.

In the FCNN model, the Loss values start to be close to each other from the beginning of training and remain constant throughout, while LSTM has lower values at the beginning and starts to be close to each other in 5-10 training rounds and remains constant throughout. This characteristic shows that the prediction model does not need a very high number of training rounds. Meanwhile, CNNs in the beginning are similar to LSTM; but will start to overfit more when training reaches 10 rounds or more. This characteristic shows that the prediction model is suitable for training no more than 10 rounds. If it exceeds this, the model will not work. The details of the experimental results are shown in Table 4.

DOI: 10.60101/jarst.2024.260451

Table 4 In case 1 results of the predictive model by specifying the input of the reference model according to SEM.

Algorithm	Time	Loss	Accuracy	Precision	Recall	F1-Score
FCNN	23	0.0424	1	1	1	1
CNNs	26	0.2203	0.94	0.94	0.97	1
LSTM	18	68.9337	0.94	0.94	0.97	1

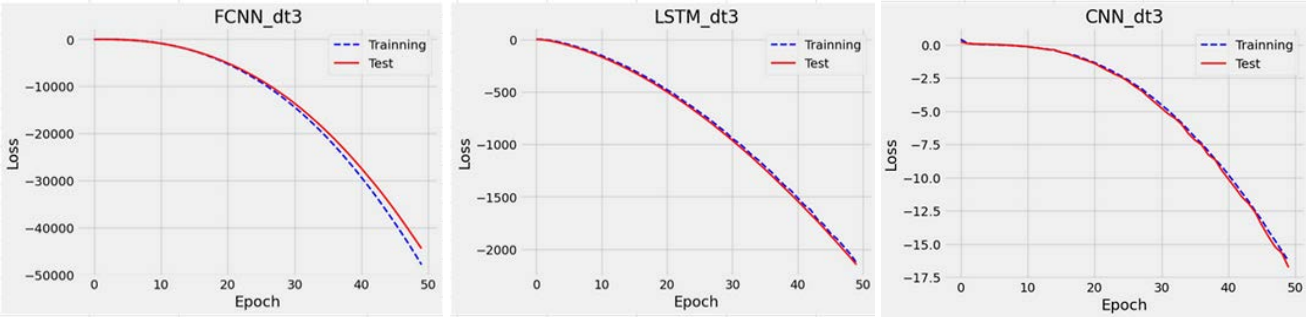

Figure 11 Loss of SEM-based feature-based models.

Table 5 In case 1 results of the predictive model by specifying the input of the reference model according to SEM.

Algorithm	Time	Loss	Accuracy	Precision	Recall	F1-Score
FCNN	20	-75058.70	1	1	1	1
CNNs	36	-3183.13	0.94	0.94	0.97	1
LSTM	40	-2135.13	0.94	0.94	0.97	1

Case 3 will insert all the feature values, which will consist of Car_Number, Working_Status, Skill_Level, Years_Left, Manual_Available, Tool_Name, Tool_Size, Calibrated, Operational, Part_Name, Installation_Dete, Hours_Used, Minutes_Used, Expiration_Date, Date_Now, leDey and Management, with the model loss values as shown in Figure 11.

From Figure 11, it can be seen that all models have the Loss value of Underfitting or the negative Loss value, which indicates that the prediction model receives too much Input value, but the internal structure is too complex, resulting in a negative Loss value and the model cannot be used. The details of the experimental results are shown in Table 5.

From Table 5, it can be seen that the Loss value of all models is very high. Although the test results of Accuracy, Precision, Recall, and F1-Score show that this model works, in reality, this model cannot work with such a variety of data. If all the features are needed, the internal structure of the prediction model should be developed to be more complex, which will result in better training Loss value.

From Table 3-5 of this experiment, the researcher has divided the data set into 3 parts: Data set 1, 60%, is used for training. Data set 2, 20%, is used for data tuning. And the third set of data, another 20%, is used for testing. From picture 10-12, the results of Training will be blue lines and the results of Testing will be red lines. For data tuning, it is a model tuning process where the three models above have been fine-tuned on the data set used for 20% tuning. The researcher selects the model with the best value from the results of data tuning to test and compare with the data set to divide the test again. Therefore, the values have shown in the graphs in Figure 10-12 are the results of a comparison between the training data sets and those used for testing. In the case where the red curve and the blue line run further apart according to the number of training rounds, it means that the model is underfitting or the model is not predicting accurately. This must be fixed by increasing the training cycle or adjusting the architecture of the new model. In the case of the red graph lines running parallel to each other and keep running closer to each other. When increasing the rounds of training, it means that the model has good prediction performance, and in the case where the red and blue line graphs intersect and run away from each other or run parallel but keep getting negative values according to the number of training cycles, it means that the model is overfitting or the model can only remember the training results when new data is encountered, the model is unable to predict or makes incorrect predictions. This will require adjusting the architecture of the new model to solve the problems.

Link Dataset :
https://drive.google.com/drive/folders/1q5SpnN-WycJEuzs53DdvbXM7IoeVzseS?usp=sharing.

CONCLUSIONS

The analysis of the experimental results in each case can clearly show the advantages and

DOI: 10.60101/jarst.2024.260451

disadvantages of the used models and indicate the suitability of the data features used in each situation. In this study, the researcher used Anaconda and Google Colab to run the Python program. The advantage of using FCNN, LSTM, and CNNs in selecting such models for comparison in this study. FCNN has the ability to analyze the relationship among features. Especially good at classifying damage and predicting worn out spare parts. LSTM is suitable for serial data, it has the ability to predict the lifespan of spare parts and detect trends in failure and CNNs is capable of analyzing patterns in spatial data and automatically extracting features. These three models are therefore suitable to be used in this study. In the first case, using only four features, namely Working Status, Operational, leDay, and Management, which are from the SEM model, all FCNN, CNNs, and LSTM models show the highest Accuracy and Precision values of 1.0, which are excellent results in numerical terms. However, considering the high and fluctuating Loss values throughout the training, it is clear that these models have the problem of Overfitting. The models can perfectly remember the training data, but they cannot properly adapt to new, never-seen data. This problem comes from the fact that the input data is too limited. Therefore, the models cannot sufficiently learn the complexity of the data in real situations. In the second case, which increases the variety of data features, such as Operational, Tool Name, and Hours Used, the models perform better. The training Loss values are significantly reduced and are more stable than in the first case, especially in the FCNN and LSTM models, which show Accuracy values close to 0.94 throughout the training process. While CNNs, despite their good initial performance, However, the trend of overfitting begins when the training exceeds the 10th iteration. The experimental results in this case show that increasing the number and variety of data features helps the model to better understand the relationship between variables. However, the problem of overfitting CNNs can be solved by reducing the number of training iterations or reducing the complexity of the model structure. In the last case, when all data features are combined, all models show very high, sometimes negative, loss values, indicating an overfitting problem where the model cannot learn enough about the complexity of the data. Although the CNN and LSTM models still maintain an accuracy of 0.94, the very high loss values make the results unreliable. This problem is caused by the fact that the data contains too many features, making the model unable to process the relationships efficiently. In addition, the internal structure of the model may not be suitable for the increasing complexity of the data. Adding features without deeply analyzing whether each feature is important for forecasting may hurt the model's ability. When comparing the three cases, it can be seen that Case 2 is the most balanced, both in terms of the number of features and model performance. The low and constant loss value reflects the model's ability to learn the data efficiently. While cases 1 and 3 suffer from overfitting and underfitting, respectively, this analysis shows that the diversity and appropriateness of the input data are key to the model's success. Moreover, optimizing the model structure to suit the data characteristics, such as reducing the number of training iterations or using regularization techniques, can help the model adapt and perform better in different scenarios with optimal performance.

Future suggestions

Future research: the most effective method for future research should be focused on developing hybrid models that combine the advantages of FCNN, LSTM, and CNNs to improve the ability to analyze data with different characteristics. The scope of the dataset used in the research should be expanded, such as integrating temporal data in real work processes and variables directly related to the environment, to increase the accuracy and depth of the results. In addition, developing deep learning techniques that can be adapted to the characteristics of data in each field, such as preventive maintenance in various industries, will improve efficiency and reliability. Testing and improving the model on different machines or equipment will help to understand the suitability of the model in different contexts more deeply.

REFERENCES

1. Carvalho TP, Soares FA, Vita R, Francisco RdP, Basto JP, Alcalá SG. A systematic literature review of machine learning methods applied to predictive maintenance. Computers & Industrial Engineering. 2019;137;106024.
2. Hoonsakul S, Pajayakrit P, Khaemkham S, Kaiyawann Y. Causal factors influencing the maintenance efficiency of Thai Air Force helicopters. Kasetsart Engineering Journal. 2016;29(98):43-54.
3. Khosantia N, Kaewsawan C, Friestad P. Strategic guidelines for developing Thailand's aircraft maintenance industry. Journal of MCU Peace Studies. 2021;9(4):1397-406.
4. Reaboot S. Enhancing fraud detection in financial systems using machine learning and network analysis techniques [thesis]. Bangkok: Dhurakij Pundit University, College of Innovative Technology and Engineering; 2022.
5. Hecht-Nielsen R. Theory of the backpropagation neural network. Neural networks for perception: Elsevier; 1992. p. 65-93.
6. LeCun Y, Bengio Y, Hinton G. Deep learning. nature. 2015;521(7553):436-44.

DOI: 10.60101/jarst.2024.260451

7. Dean J, Corrado G, Monga R, Chen K, Devin M, Mao M, et al. Large scale distributed deep networks. Advances in neural information processing systems. 2012;25.
8. Han S, Liu X, Mao H, Pu J, Pedram A, Horowitz MA, et al. EIE: Efficient inference engine on compressed deep neural network. ACM SIGARCH Computer Architecture News. 2016;44(3):243-54.
9. Akhtar MZ, Kreiner C, Schmid M, Zippelius A, Tetzlaff U, Elger G, editors. Fully Connected Neural Network (FCNN) Based Validation Framework for FEA Post Processing to Improve SAC Solder Reliability Analysis. 2024 IEEE 10th Electronics System-Integration Technology Conference (ESTC); 2024: IEEE.
10. Song Y, Wang D, Xiong X, Cheng X, Huang L, Zhang Y. The Prediction and Dynamic Correction of Drifting Trajectory for Unmanned Maritime Equipment Based on Fully Connected Neural Network (FCNN) Embedding Model. Journal of Marine Science and Engineering. 2024;12(12): 2262.
11. De Filippi FM, Ginesi M, Sappa G. A fully connected neural network (FCNN) model to simulate karst spring flowrates in the Umbria region (Central Italy). Water. 2024;16(18):2580.
12. Han S, Pool J, Tran J, Dally W. Learning both weights and connections for efficient neural network. Adv Neur In. 2015;28.
13. Qian Z, Hayes TL, Kafle K, Kanan C. Do we need fully connected output layers in convolutional networks? [Internet]. 2020 [cited 2025 Jan 19]; Available from: arXivpreprintarXiv:200413587.
14. Sze V, Chen YH, Yang TJ, Emer JS. Efficient processing of deep neural networks: A tutorial and survey. Proc IEEE. 2017;105(12):2295-329.
15. Gao H, Liu Z, Pleiss G, Laurens van der M, Weinberger KQ. Convolutional Networks with Dense Connectivity. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2019;44(12):8704-16. doi: 10.1109/TPAMI.2019.2918284.
16. Hochreiter S. Long Short-term Memory. Neural Computation MIT-Press. 1997.
17. linPremjith B, Soman K, Kumar MA. A deep learning approach for Malayalam morphological analysis at character level. Procedia Comput Sci. 2018; 132:47-54.
18. Upreti A. Convolutional neural network (cnn). a comprehensive overview. 2022.
19. Purwono P, Ma'arif A, Rahmaniar W, Fathurrahman HIK, Frisky AZK, ul Haq QM. Understanding of convolutional neural network (cnn): A review. International Journal of Robotics and Control Systems. 2022;2(4):739-48.
20. Wu Z. Application of CNN Classic Model in Modern Image Processing. Journal of Advances in Engineering and Technology. 2024;1(3):1-6.
21. Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences [Internet]. 2014 [cited 2025 Jan 19]; Available from: arXivpreprintarXiv:1404.2188.2014.
22. Khunsuk T, Thongkam J. Feature selection method for improving customer reviews classification. RMUTI J Sci Technol [Internet]. 2019 Oct 7 [cited 2025 Jan 19];13(1):129-43. Available from: https://ph01.tci-thaijo.org/index.php/rmutijo/article/view/188091.
23. Khiewwan K, Weeraphan P, Tantisontisom K, Ongate J. Application of Data Mining Techniques for Classification of Traffic Affecting Environments. J Renew Energy Smart Grid Technol [internet]. 2020 Jun. 28 [cited 2025 Jan. 19];15(1). Available from: https://ph01.tci-thaijo.org/index.php/RAST/article/view/240698.
24. Krstinić D, Braović M, Šerić L, Božić-Štulić D. Multi-label classifier performance evaluation with confusion matrix. Computer Science & Information Technology. 2020;1:1-14.
25. Jain A. Unveiling the power of heatmaps in data visualization [Internet]. Medium. 2023 [cited 2024 Dec 24]. Available from: https://medium.com/@abhishekjainindore24/unveiling-the-power-of-heatmaps-in-data-visualization8b370efca935.

DOI: 10.60101/jarst.2024.260451