SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V11I9P127

Volume 11 Issue 9, 294-315, September 2024
© 2024 Seventh Sense Research Group®

Original Article

Intelligent Fault Diagnosis of Rotating Machinery Using Deep Learning Algorithms: A Comparative Analysis of MLP, CNN, RNN, and LSTM

Vijayalakshmi K[1], Amuthakkannan Rajakannu[2], Mohsina Kamarudden[3], Ramachandran KP[4]
Sri Rajkavin A V[5]

[1,3]*Department of Electrical and Communication Engineering, National University of Science and Technology, Muscat 130, Oman.*
[2]*Department of Mechanical and Industrial Engineering, National University of Science and Technology, Muscat 130, Oman.*
[4]*Deanship of Graduate Studies and Research, National University of Science and Technology, Muscat 130, Oman.*
[5]*College of Design and Engineering, Department of Electrical and Computer Engineering, National University of Singapore, Singapore.*

[1]*Corresponding Author : vijayalakshmi@nu.edu.om*

Received: 12 July 2024 Revised: 15 August 2024 Accepted: 12 September 2024 Published: 28 September 2024

Abstract - *Health management in industrial systems is crucial for maintenance management, and it plays an important role in productivity, fault diagnosis, safety, efficiency, and economy in manufacturing industries. Early detection of faults in machinery may increase the effectiveness of maintenance actions and will avoid unwanted consequences in process operations and maintenance. Existing fault diagnosis methods have limitations such as insufficient accuracy, slow detection rate, and handling large and complex data sets. In this digital age, Industry 4.0 techniques have been applied across all fields, including the condition monitoring of machines. This research addresses the gaps in traditional fault diagnosis by using deep learning, a modern AI technique effective for diagnosing faults in various machines. In this research work, vibration signals are collected using the National Instruments- Data Acquisition (NI-DAQ) system, accelerometer, and LabVIEW software. These signals are processed using a series of steps, including sampling strategy, shuffling, standardization, and reshaping data augmentation. Deep learning algorithms Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) are tested for fault diagnosis using vibration datasets collected from Spectra Quest Machinery Fault Simulator (SQMFS). The result shows that MLP accuracy in the fault prediction is 0.9, CNN reached 0.95, and RNN and LSTM with 0.57 and 0.45, respectively. The high performance of CNN is due to its ability to effectively capture spatial patterns in vibration data, which is crucial for fault diagnosis in rotating machinery, followed by MLP due to its faster convergence during training. When the data is scaled, MLP performs better than CNN, demonstrating its adaptability to increased data complexity and volume. RNN and LSTM resulted in lower accuracy due to the need for larger datasets and temporal patterns in the vibration data, which they are designed to handle. This study shows that CNN has given better results than other deep learning algorithms, such as MLP, RNN, and LSTM, in fault diagnosis of rotating machinery. Future research could explore applying these techniques to different types of machinery and fault conditions.*

Keywords - *Condition monitoring, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Machine fault simulator, Multilayer Perceptron (MLP), Recurrent Neural Network (RNN).*

1. Introduction

Implementing Artificial Intelligence in manufacturing and process industries is becoming mandatory for sustainable and reliable failure-free operation on shop floors. Therefore, real-time fault diagnosis and precision health assessment systems are needed for effective maintenance programs in the industrial sector [1]. However, despite AI advancements, there is still a gap in developing real-time, robust, and accurate fault diagnoses capable of handling unbalanced datasets and complex fault conditions in rotating machinery. Rapid advancements in digital technologies and AI techniques have invited more attention from research and development in industrial sectors, particularly in the condition monitoring of machines and related industrial processes [2]. Existing methods struggle to reach the required precision and adaptability for online monitoring in various industrial environments, which leads to the need for better AI based solutions.

In this modern world, there are many technological digital revolutions with a focus on cyber-physical and biological systems such as Artificial Intelligence, Machine learning, Deep learning, Robotics, the Internet of Things, Virtual Reality, etc., In machining-based industries, identifying a reliable condition monitoring system would give a clear picture of the health of a machine, which is a mandatory requirement for adaptive or corrective actions [3]. A significant challenge lies in developing a condition monitoring system that can accurately diagnose faults under various machine conditions.

There is always a high demand for an accurate online machine condition or fault diagnosis system using modern technologies to reduce downtime and improve the effective utilization of machinery in the production system [2]. A well-structured predictive maintenance system will reduce economic stresses and give necessary precautions to avoid a stoppage in the production line [4]. Maintenance engineers typically follow three traditional strategies: breakdown, planned, and predictive maintenance [5]. Among the three maintenance methods, predictive maintenance is commonly used in industries for intelligent diagnosis. The intelligent diagnosis procedure can be applied offline or online [6]. Unlike offline techniques, online condition monitoring will detect continuous faults using sensors and data acquisition. The predictive maintenance methods can be broadly categorized as model-based predictive maintenance and data-based predictive maintenance. To indicate a fault, the mathematical model is used with empirical data in model-based maintenance, whereas the data-driven method uses intelligent models like fuzzy logic, machine learning, deep learning, etc.

In recent days, data-driven models have been aligned with machine learning, deep learning, and digital twins. Machine learning and deep learning are the powerful tools of artificial intelligence that push the boundaries of innovation. Machine learning uses algorithms and learns independently but needs human assistance to correct errors. Deep learning uses advanced computing, which uses much more data than machine learning. There is little or no human intervention in deep learning while doing advanced computing. However, there remains a gap in understanding the comparative strengths of these approaches when applied to rotating machinery fault diagnosis. Additionally, the effectiveness of condition monitoring systems can vary depending on the algorithms used for fault diagnosis, highlighting the importance of \electing the appropriate model.

Machine learning uses thousands of data points in condition monitoring, while deep learning-based condition monitoring uses millions of data points. Machine learning-based fault diagnosis methods use explicit programming, but deep learning algorithms solve the problems based on the layers of neural networks. Deep learning is a machine learning algorithm that uses deep (more than one layer) neural networks to analyze data and provide output accordingly.

Koutsoupakin j et al. discussed machine learning-based condition monitoring for gear transmission systems using data generated by optimal multibody dynamic models. Even though the title indicates machine learning, the author's research with CNN will generally come under deep learning techniques [7]. Zhang. L et al. discussed an imbalanced fault diagnosis method based on TFFO and CNN for rotating machinery, and the authors claimed that deep learning-based fault diagnosis usually requires a rich supply of data [8]. The lack of sufficient data or imbalance in datasets presents a significant challenge to existing methods, demanding more robust approaches.

In this study, a comparative analysis of deep learning models MLP, CNN, RNN, and LSTM is made for intelligent fault diagnosis of rotating machinery, contributing several insights to the field. This research investigates how different DL models perform across various scaled datasets. This approach enables a comprehensive understanding of the scalability of the models and robustness in real-world applications.

The paper is organized as follows: Section 2 presents a review of relevant literature and a review of deep learning techniques, including MLP, CNN, RNN, and LSTM. Sections 3 and 4 detail the methodology used for fault diagnosis in rotating machinery, describing the machine fault simulator and experimental setup. Section 5 explains data collection and signal pre-processing, with data pre-processing described in Section 6. The application of deep learning techniques is discussed in Section 7. Section 8 discusses the results of the comparative analysis of the deep learning techniques, followed by a thorough evaluation of model performance. Finally, Section 9 concludes the paper by summarizing key findings, discussing limitations, and suggesting future research directions.

2. Literature Review

Recently, various techniques have been developed for intelligent fault diagnosis in rotating machinery. While there has been considerable progress, achieving high accuracy in real-time monitoring and improving fault diagnosis systems remains challenging. This section reviews key developments in machine fault diagnosis, focusing on using deep learning algorithms such as MLP, CNN, RNN, and LSTM, which are essential to address the gaps.

2.1. Review Related to the AI Model

Numerous methods of machine condition monitoring are investigated by fault diagnosis researchers, which includes both direct and indirect methods. Direct methods like visual inspection, machine vision, and infra-red methodology are used to monitor the machine or machine tools, which will give less accuracy than indirect methods [9]. Indirect methods include using signals like force, vibrations, acoustic emission, etc., in which the signal features have a relationship with fault conditions/parameters [10]. Rui Zhao et al. have surveyed deep learning and its application to machine health monitoring. He has reviewed 108 technical articles and concluded: "It is believed that deep learning will have more and more prospective future impacting machine health monitoring, especially in the age of big machinery data". So, this survey indicates that deep learning can be used precisely to monitor tool wear, and the paper concluded that deep learning is a promising technique for assessing tool wear [11].

Lang Dai developed an improved deep learning model for online tool condition monitoring using output power signals. The output power from the sensor, which is mounted on the cutting tool holder during its operation, is used for analysis. This data is analyzed using Wider first-layer kernels (WCONV) and Long Short-Term Memory (LSTM) available in the deep learning algorithms. The weakness of the paper is addressing the output power signal. This paper focuses more on the output power signals and its analysis of deep learning algorithms, which needs more study on the condition monitoring of the tool [12]. Qun wang et al. have done an overview of tool wear monitoring methods based on CNN. The authors concluded that applying CNN in tool wear and condition monitoring is feasible and reliable. They have added that CNN can improve prediction accuracy [13].

CNN, coming under deep learning, are a typical data-driven fault diagnosis method that extracts features from images using convolutional layers and then pools and fully connects layers for tasks like image classification [14]. In condition monitoring, researchers have applied CNNs to diagnose faults in rotating machinery. Janssens et al. proposed a feature learning model for condition monitoring based on CNN [15]. Yao et al. used sound signals predicted using an acoustic emission sensor. They used CNN based on a multiscale dialog learning structure and attention mechanisms for gear fault prediction [16].

Zhang, W et al. applied Deep Convolutional Neural Networks (DCNN) for bearing fault diagnosis under different operating loads [17]. Abdelmaksoud M et al. proposed a CNN model to diagnose induction motor faults at the motor's starting time. The model detects faults, such as locked rotor, overload, voltage imbalance, overvoltage, and Undervoltage, under three loading levels: light, regular, and heavy. The authors concluded that DCNN is an effective tool for diagnosing multi-signal induction motor faults at its starting period with different versions of datasets [18].

2.2. Review Related to MLP

Marwala, T. used MLP for condition monitoring in a mechanical system. This paper uses MLP to identify false information in a population of cylindrical shells [19]. Zanic, D. and Zuban, A. discussed the monitoring of transformers using the MLP machine learning model. This paper describes the multilayer perceptron class of artificial intelligence to predict the temperature in a transformer by giving three input features in MLP (oil temperature, winding current, and outside temperature) [20]. The conventional MLP has been applied in machine health condition monitoring for many years [11].

Nguyen, V.Q et al. applied the MLP mixer model for bearing fault diagnosis. The authors concluded that the evaluation results show that the proposed MLP mixer model obtains high accuracy when the number of training samples is reduced. The authors added that the performance of the proposed model, when compared with other states of the art, proved the advantages of the MLP approach [21]. Tarek K et al. proposed an optimized multi-layer perceptron for fault diagnosis of induction motors using vibration signals. The authors concluded that the obtained results show the capability of detecting faults in the induction motor under different operating conditions [22].

2.3. Review Related to CNN

Neupane, D et al. discussed convolutionary neural network-based fault deduction for intelligent manufacturing. The result of this paper shows that one dimensional CNN is more efficient in terms of computational complexity for time series data [23]. With AI advancements, researchers without expertise can solve fault identification problems with bearings and gearboxes [24]. Artificial neural networks, support vector machines, particle filters [25], and extreme learning machines [26] were used to extract the failure features from the signals. Deep learning algorithms have recently been applied in rotating machine fault diagnosis due to their nonlinear regression ability [26].

Some of these are Convolution Neural Networks [27, 28], Recurrent Neural Networks [29], Deep encoders, and Generative Adversarial Networks [30]. Nonlinearities are common in real-world time series data [31], making applying conventional prediction techniques complex [32]. To address this issue, advanced techniques such as CNN, LSTM, and quantile regression were applied to predict failures. One-dimensional convolutional neural network is utilized for time series prediction [33].

Zhang et al. stated that higher accuracy is achieved using DNN for time sequence prediction [34]. Ruan, D et al. used CNN for bearing fault detection by giving fault periods under different fault types, and shaft rotation frequency was used to determine the size of CNN's input. In the paper, the authors confirmed that Physics-Guided CNN (PGCNN) with a rectangular convolution kernel works better than the baseline CNN with more accuracy and less certainty [35]. Kothuru A. et al. applied a deep visualization technique to gain knowledge on the inner workings of the deep learning models in tool wear prediction for end milling. The authors concluded that the

CNN model focuses more on the signal's frequency features than the features concerning the time period of the signal [36]. However, CNN suffers from the weakness of not considering the imbalanced distribution of machinery health conditions, and what CNNs have learned in predicting fault diagnosis is unclear. Jia F et al. suggested a framework called Deep Normalized Convolutional Neural Network (DNCNN) for imbalanced fault classification of machinery to overcome the first weakness. He proposed a Neuron Activation Maximization (NAM) algorithm to handle the weakness of what CNNs have learned in predicting fault diagnosis [37].

2.4. Review Related to RNN and LSTM

Elman's RNN is used by Serhat Seker et al. for condition monitoring in nuclear power plants and rotating machining. The first part of this work is a prediction of anomalies in gas-cooled nuclear reactors. The second part is detecting motor bearing damage in induction motors [38]. Halliday C et al. applied RNN for condition monitoring and predictive maintenance of pressure vessel components. The authors found that RNN is well suited to tackle the shortfalls in RNN-Residual Order Models (ROM) [39]. Swetha R Kumar and Jayaprasanth Devakumar used RNN with ten hidden neurons to estimate state variables with the inputs and outputs of the process. The authors studied the effectiveness of RNN in fault detection and isolation for different scenarios of sensor output, such as drift, erratic, hard-over, spike, and stuck [40].

Yahui Zhang et al. proposed a Gated Recurrent Unit (GRU) and MLP-based model for fault diagnosis to detect fault types. This work converts one-dimensional time-series vibration signals into two-dimensional images in the first phase. Then, GRU is introduced to exploit sequential information of time-series data and learn representative features from constructed images. Then, MLP is used to implement fault recognition. The results show that the proposed method gives the best result on two public datasets compared with existing work and shows robustness against the noise [41].

Zhuang Ye and Jianbo Yu applied LSTM units to capture sequential information from multi-sensor time series data. The authors used convolution calculation for noise reduction and feature extraction. Ultimately, the Health Index (HI) is generated based on reconstruction errors in run-to-fail data [42]. Zhao, H, Sun, S and Jin, B applied LSTM neural network-based fault diagnosis and indicated that LSTM can directly classify the raw process data without specific feature extraction and classifier design. This work evaluates LSTM to fault identification and analysis in the Tennessee Eastman benchmark process. The authors used LSTM to capture sequential information from multi-sensor time series data. This work uses convolution calculation for noise reduction and feature extraction. Multivariate Gaussian distribution generates a health index based on reconstruction errors of long short-term memory convolutional auto-encoder [43]. Afridi, Y.S. et al. have developed a fault prognostic system using LSTM for rolling element bearing because of the vital component involved and the highest fault diagnosis [44]. So, Deep learning architectures include deep belief networks, MLP, Autoencoder (AE), CNN, and RNN [45]. With the rapid development of DL techniques in recent years, many new architectures have been proposed and introduced into intelligent industrial fault diagnosis tasks. Similarly, CNN is prospering again due to recent progress in computer vision and enhanced visualization techniques [46].

The literature review demonstrates that while deep learning algorithms, particularly MLP, CNN, RNN, and LSTM, have shown promising results in intelligent fault diagnosis, challenges such as handling data imbalance generalization to different machine conditions and real-time fault diagnosis persist. This study aims to address these challenges by comparing the performance of these algorithms, which contribute to developing a more robust fault diagnosis system for rotating machinery.

3. Spectra Quest Machinery Fault Simulator - An Overview

In this research work, the Spectra Quest Machinery Fault Simulator (SQMFS) is utilized to simulate both balanced and imbalanced conditions of a rotary shaft. The Machine Fault Simulator (MFS) is a highly versatile and modular test platform designed to facilitate the study and simulation of various mechanical faults in rotating machinery. It is extensively used in academia, industry training, and research to provide a controlled environment for replicating real-world fault conditions, enabling comprehensive analysis and predictive maintenance. The simulator allows for the monitoring and analysis of fault conditions, providing valuable insights into vibration analysis and fault diagnostics [47]. Figure 1 shows the photograph of SQMFS with the critical components involved in it.

The heart of the MFS is the motor-driven shaft, the primary rotating element where faults can be introduced and studied. The motor allows for variable speed control, which is crucial for observing the mechanical behavior of the system under different operating conditions. Precise speed adjustments enable examining how faults manifest across various frequencies and loads, providing a comprehensive understanding of the system's dynamics.

The MFS includes multiple adjustable disks, and couplings mounted onto the shaft. These components are specifically designed with various tapped holes to accommodate the attachment of screws, weights, and other accessories. This flexibility is vital for simulating fault conditions, such as unbalance, misalignment, or looseness. For instance, introducing an imbalance involves asymmetrically adding weights to the disks, which leads to uneven mass distribution and the characteristic vibration patterns of unbalanced systems. Strategically placed tapped holes on the disks allow for the precise positioning of weights, enabling the controlled introduction of imbalances. By varying the size and placement of these weights, researchers can replicate a wide range of imbalance severities, from mild to extreme, providing insights into the effects of different levels of imbalance on machine behavior. These couplings connect various rotating components and can be adjusted to introduce misalignments and common faults in rotating machinery. This feature is essential for simulating both angular and parallel misalignments, which can cause significant changes in vibration patterns and lead to mechanical failure if left undetected.

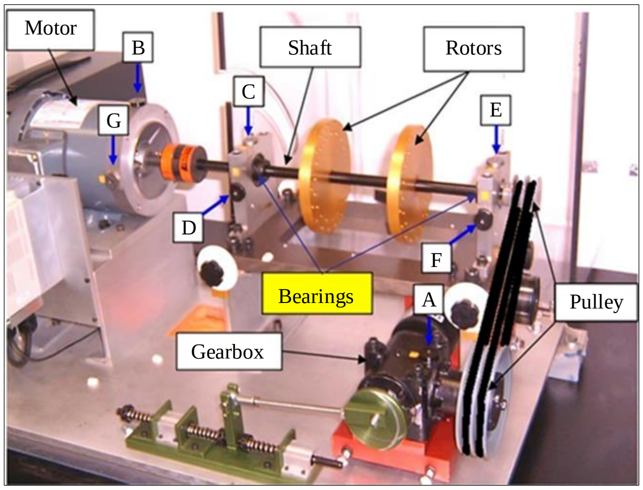


Fig. 1 Key components and structure of MFS [40]

The shaft is supported by high-precision bearings that ensure smooth rotation while also allowing for the introduction of bearing faults. The MFS's support structure is designed to be sturdy and minimize external vibrations that could interfere with the data, ensuring that observed faults are due solely to the simulated conditions. The MFS has various sensors, such as accelerometers, proximity probes, and load cells, to monitor vibrations, forces, and rotational speeds. These sensors are strategically placed to capture the data required for fault analysis. The placement of these sensors is configurable, allowing for targeted measurements aligned with the specific fault being studied.

- Accelerometers: These sensors, typically mounted on or near the shaft, measure radial and axial vibrations. They are susceptible to changes in vibration patterns, making them ideal for detecting imbalances and misalignments [48].
- Proximity Probes: Non-contact sensors that measure shaft displacement are critical for studying shaft misalignment and eccentricity.

The setup incorporates several critical components, including a motor-driven shaft, adjustable disks, a Lenze controller, and a data acquisition system to facilitate accurate data collection and analysis. Before each test, the system is configured to simulate the specific fault condition under study. For example, weights are attached to the rotor disks at predetermined locations to examine imbalance. The motor speed, managed by the Lenze Controller, is set to the desired level, ranging from low speeds (to observe subtle faults) to high speeds (to assess fault severity under real-world conditions).

When the shaft rotates, the faults generate vibrations captured by the attached accelerometer. In imbalance studies, uneven mass distribution causes centrifugal forces that result in periodic vibrations, exhibiting frequency spikes in the spectrum analysis. The sensors provide real-time data on these vibrations, allowing researchers to identify fault characteristics.

The MFS is integrated with a Lenze Controller, which provides precise control over motor speeds and torque during the simulation. This controller enables real-time adjustments of motor parameters such as speed, acceleration, and torque limits, which are essential for accurately replicating various fault conditions. The fine control provided by the Lenze Controller enhances the reliability and repeatability of fault simulations.

4. Experimental Setup of Unbalancing in Machine Fault Simulator

Figures 2 and 3 show the complete experimental setup of the Machinery Fault Simulator. As seen on the left in Figure 2, a computer monitor displaying the LabVIEW interface is used for data acquisition and analysis. In the center, there is a blue Lenze Controller responsible for precise motor speed control. The MFS on the right is enclosed in a transparent safety case. This setup allows for safe operation while providing clear visibility of the rotating components during experiments.

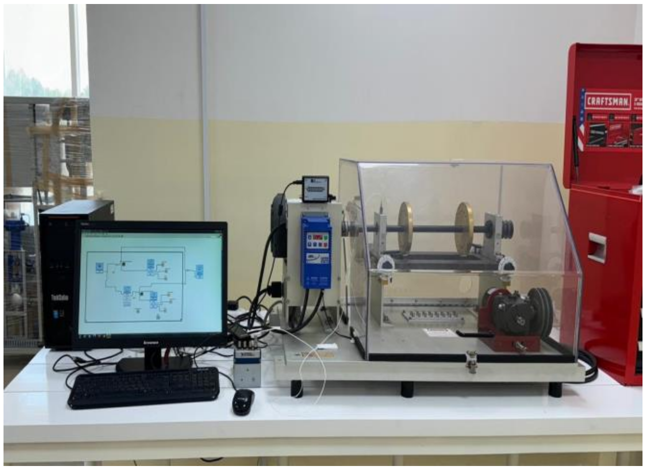


Fig. 2 Integration of MFS with LabVIEW software

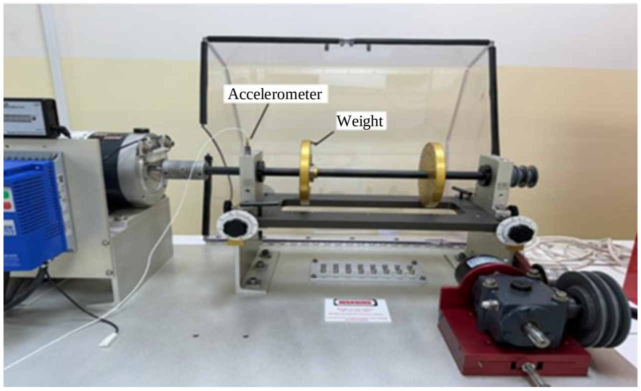
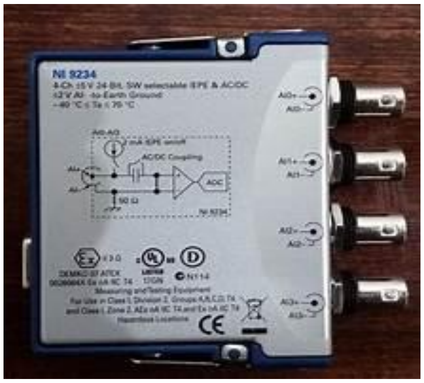


Fig. 3 MFS with weights on the discs create an imbalance

This close-up image of the MFS rotor provides a detailed view of the critical components. The motor-driven shaft, adjustable disks with tapped holes for attaching weights, and the accelerometer mounted capture vibration data. The golden disk on the shaft is one of the adjustable disks for introducing imbalance (Figure 3). This configuration allows for precise control over the introduction of faults and the measurement of resulting vibrations. The data acquisition system, consisting of components like the National Instruments (NI) cDAQ-9174 (Figure 4(a)) chassis and NI modules, captures vibration, displacement, and speed data from the sensors. The NIcDAQ9174 is a modular chassis that serves as the central data acquisition hub, with slots for various C Series I/O modules, such as the NI 9234 module (Figure 4(b)) for high-precision dynamic signal acquisition.

The accelerometer, strategically mounted on the rotary shaft, measures radial and axial vibration data. The collected data is transmitted to specialized software like LabVIEW for visualization, allowing researchers to interpret and diagnose fault conditions accurately. The NI 9234 module, inserted into one of the slots of the DAQ-9174 chassis, is designed explicitly for high-precision dynamic signal acquisition. It features four analog input channels with built-in anti-aliasing filters, capable of sampling up to 51.2 kS/s per channel. This module is essential for capturing subtle variations in vibration patterns and detecting early signs of mechanical faults. It supports many sensor types, including accelerometers, making it suitable for vibration analysis in rotating machinery.

(a)



(b)

Fig. 4 (a) Data acquisition components (NI Cdaq9174), and (b)NI 9234 – Signal conditioning device

The accelerometer captures both radial and axial vibration data, providing insights into the dynamic behavior of the shaft under balanced and imbalanced conditions. The vertical placement of the accelerometer is strategic, as it enhances sensitivity to vibrations caused by imbalances, thereby improving fault detection accuracy, as shown in Figure 5.

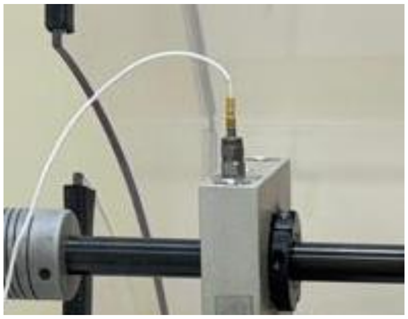


Fig. 5 Vertical mounting of the accelerometer on the rotating shaft

This integrated experimental setup enables in-depth analysis and understanding of machine behavior under simulated fault conditions, enhancing the reliability of fault detection methods and contributing to developing more robust machinery monitoring systems.

5. Data Collection and Signal Pre-processing

The experimental analysis focused on two primary operating conditions of the rotary shaft:

- Balanced Condition
- Imbalanced Condition

These conditions were carefully established to study the effects of imbalance on the dynamic behavior of rotating machinery, thereby providing critical insights for developing diagnostic tools.

5.1. Balancing Condition in MFS

Balancing plays a crucial role in operating the MFS, particularly when examining the effects of imbalance on machinery health. Balancing ensures uniform mass distribution in rotating components, minimizing vibration, reducing bearing wear, and enhancing overall machinery performance. The MFS facilitates the study of these effects by allowing users to adjust the balance state of the shafts and observe the resulting changes in machine behavior. The MFS uses an accelerometer to capture frequency and time-domain data, providing real-time insights into the impact of imbalance conditions. Weights are added or removed to the shaft disks to simulate unbalanced scenarios, replicating real-world situations like unbalanced rotors. This setup enables the detailed study of vibration patterns and highlights the significance of balancing in rotating machinery.

The rotary shaft was operated without additional weights or defects in the balanced condition. This state represented a perfectly balanced system, where the shaft's mass was uniformly distributed across its entire length. By operating the machinery under this condition, baseline vibration data was collected. This baseline data served as a critical reference point for all subsequent analyses, providing a standard against which the effects of various imbalances could be measured. Critical aspects of the balanced condition include:

- No added weights or defects: Ensuring that no external weights, screws, or other defects were present on the rotary shaft, resulting in minimal vibration levels.
- Uniform mass distribution: The rotary shaft's weight was evenly distributed along its axis, resulting in smooth operation with low amplitude vibrations.
- Data collection as a baseline: Vibration data collected under these conditions provided a control sample, essential for differentiating between normal operational vibrations and those caused by faults or imbalances.

Balancing condition monitoring is essential in fault diagnosis to compare with the unbalanced condition dataset.

5.2. Unbalancing Condition in MFS

To simulate real-world conditions, an imbalanced condition was introduced by adding screws and weights of varying magnitudes to the rotary shaft disks on both the left and right sections. The controlled introduction of these weights was designed to replicate common mechanical faults, such as unbalance, often encountered in rotating machinery. Critical aspects of the imbalanced condition include:

- Addition of Weights: Controlled weights, ranging from light screws to heavier attachments, were mounted on one side of the rotary shaft disks to create an intentional imbalance. This allowed for the simulation of different fault magnitudes, from minor to severe.
- Variable Weight Magnitudes: Different levels of imbalance were achieved by varying the magnitude and distribution of the weights, enabling the study of their effects on vibration patterns and machine behavior.

As for the experiment, 18 different weights were taken, as shown in Table 1, and were placed randomly in either the right or left rotary shaft. The added weights generated distinct vibration signatures in the data, particularly at specific harmonic frequencies. These signatures were analyzed to determine how imbalances influenced the machine's operational dynamics.

Table 1. Sample weights used for the experiment

Sample No.	Weight	Sample No.	Weight	Sample No.	Weight	Sample No.	Weight
1	4.34gm	6	5.86gm	11	4.91gm	16	9.29gm
2	4.93gm	7	4.38gm	12	5.48gm	17	11.03gm
3	5.56gm	8	4.37gm	13	9.25gm	18	8.74gm
4	5.55gm	9	4.49gm	14	10.2gm		
5	4.93gm	10	4.36gm	15	10.05gm		

In the signal collection method, National Instruments' LabVIEW was used to visualize the accelerometer data in real-time by collecting both time and frequency-domain data were collected using the National Instruments devices. Laboratory Virtual Instrumentation Engineering Workbench (LabVIEW) is an Instrumentation software in which the acquired signal/data can be stored and analyzed. The motor was operated at 30 HZ and had a gear or shaft frequency of 216 HZ to collect the data. Once the motor stabilized at 30Hz, data acquisition is started. This setup ensured that the data collected was representative and consistent, enhancing the reliability of further analysis. This visual representation helps us understand the complete data flow from acquisition to storage, highlighting the parallel processing of time and frequency domain data. Baseline data was recorded from a balanced system with no added weights, as shown in Figures 6 and 7. In the imbalance condition, screws of different weights were added to the rotary shaft disks to simulate various levels of imbalance, and the signals were collected, which are shown in Figures 8 and 9. Both frequency-domain and time-domain acceleration data were captured during each test run. This approach provided a comprehensive dataset for analysis, focusing on capturing the effects of imbalances through systematic simulations. In this experiment, the focus was exclusively on frequency-domain data to detect shaft imbalances in rotating machinery. This decision was driven by the superior ability of frequency-specific responses to indicate shaft imbalances compared to time-domain analysis. Imbalances in rotating machinery manifest as distinct peaks and harmonics within specific frequency ranges, making frequency-domain analysis particularly effective at capturing these characteristics. Analyzing the frequency domain, critical features such as resonant frequencies and harmonic distortions directly linked to mechanical imbalances are isolated. This offers a more precise diagnosis than time-domain analysis alone, as noted by Nossier et al. [49]. Frequency-domain analysis is a well-established mechanical fault detection method supported by theoretical research and practical applications. Yi et al. highlighted its effectiveness in isolating vibration signals corresponding to specific fault frequencies, often indicative of the condition of rotating components. The inherently periodic nature of shaft vibrations makes frequency-domain analysis particularly suited for identifying imbalances in such systems [50]. As Mali et al. points out, this approach allows for a more targeted examination of the periodic behaviors that are hallmarks of shaft imbalances [51]. Furthermore, Hertel et al. demonstrated that this method emphasizes crucial frequency components related to imbalance, making the detection process more reliable and accurate under simulated conditions [52]. Acceleration data for frequencies ranging from 1 to 100 Hz for this experiment are collected. This dataset forms the basis for training the algorithms to distinguish between balanced and imbalanced conditions. While frequency-domain analysis offers significant advantages for detecting shaft imbalances, it is important to note its limitations. Focusing solely on frequency-domain data may miss transient or dynamic features of imbalances that time-domain or wavelet analysis could capture. In practical applications where conditions change rapidly, additional analysis techniques might be required for a comprehensive diagnosis, as suggested by Yan [53].

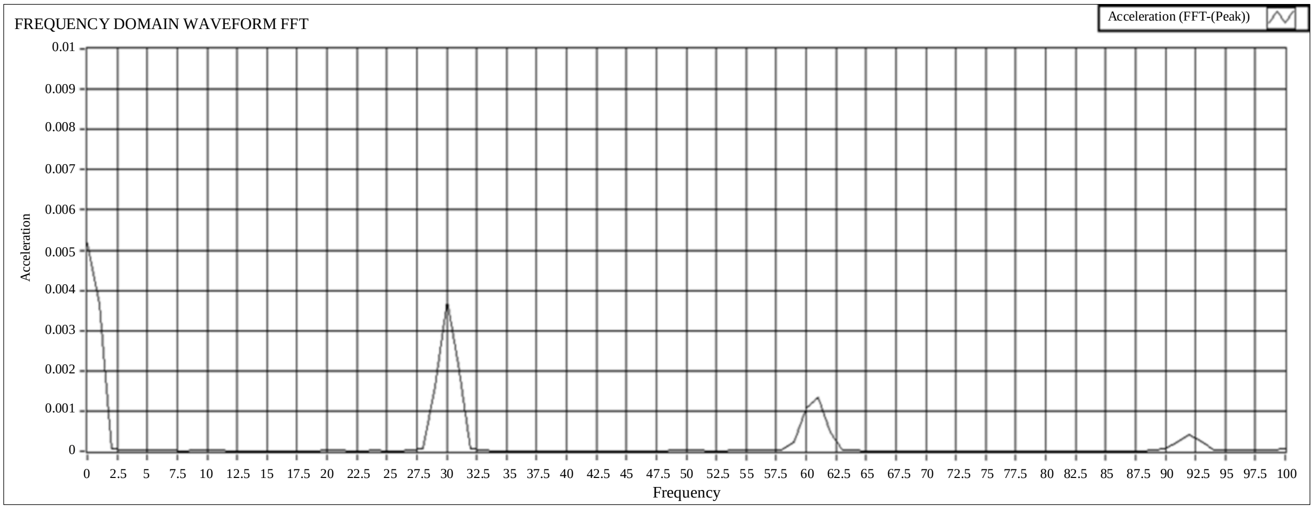


Fig. 6 Frequency domain chart for balanced data

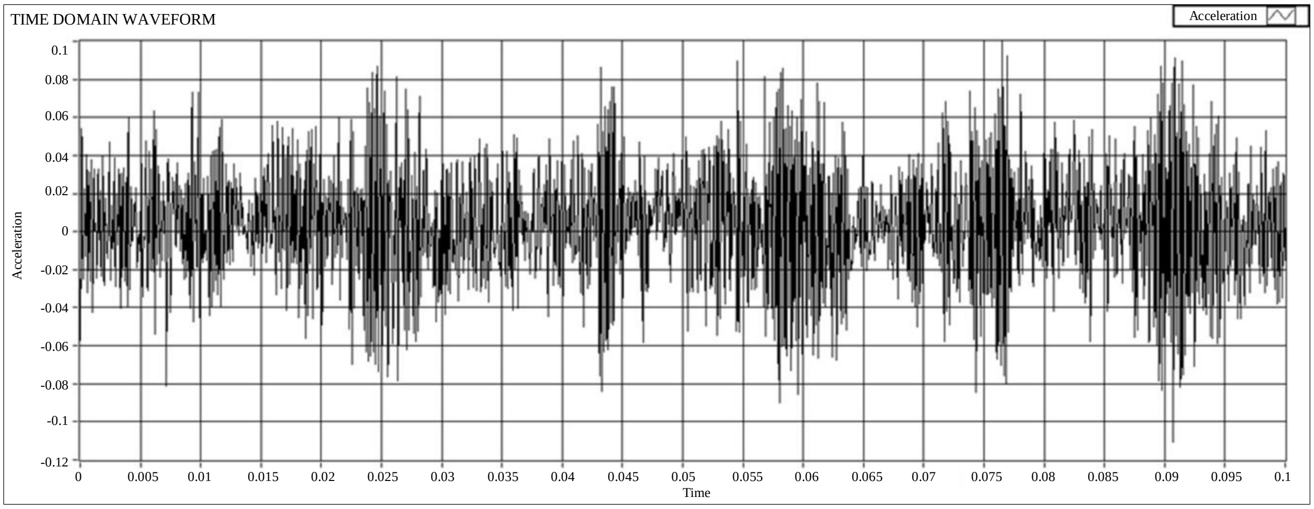


Fig. 7 Time domain chart for balanced data

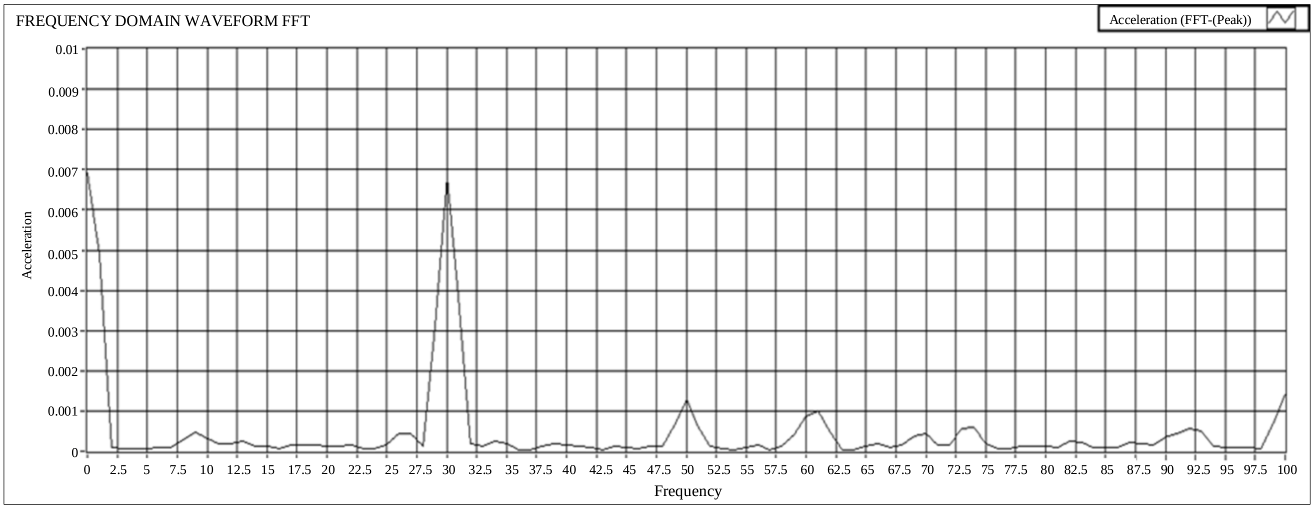


Fig. 8 Frequency domain chart for imbalanced data

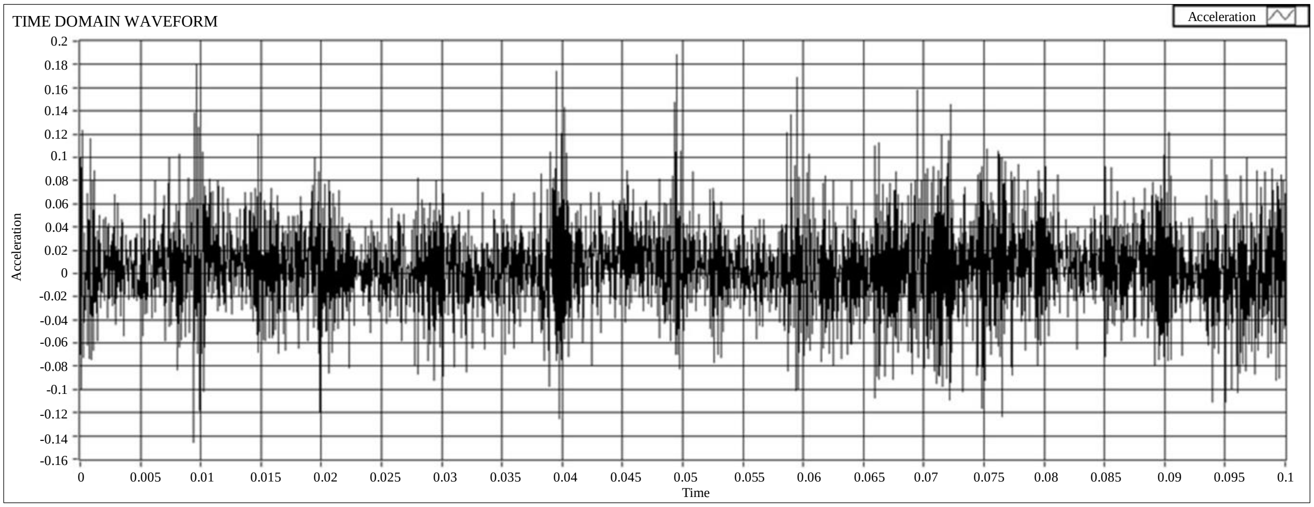


Fig. 9 Time domain chart for imbalanced data

This comprehensive setup and data collection process provides a solid foundation for applying deep learning techniques to condition monitoring. By leveraging the strengths of frequency-domain analysis, the aim was to develop more accurate and reliable diagnostic models for rotating machinery. The nature of shaft vibrations is inherently periodic, and frequency-domain analysis directly targets this periodicity, making it the preferred choice for identifying imbalances in such systems. This dataset, designed for a condition monitoring task, contains 110 entries across 102 columns. It includes 101 numerical features (F1 to F101), likely indicative of various measurements relevant to the monitoring process.

The numerical features exhibit low mean values, suggesting that the data has likely been standardized or normalized. Although most features show low variability, some have higher standard deviations, indicating varying fluctuation levels in the monitored parameters. Kurtosis measures peakedness; hence, it is a fine indicator of signal impulsiveness in fault detection for rotating components, especially for drill bits. Kurtosis is expressed as,

$$\text{kurtosis}\,(x) = (E\,\{(x\text{-}\mu)/\,\sigma^{4}) - 3 \qquad (1)$$

Where μ = mean of time series x
σ = standard deviation of time series x
E{.} is the expectation operation

The minus 3 is to make kurtosis of the normal distribution of the normal distribution equal to zero.

So, kurtosis analysis reveals diverse distribution characteristics among the features. For example, features like F1 to F4 have extremely high kurtosis values, potentially indicating the presence of outliers or rare events. In contrast, F97, F98, F100, and F101 exhibit negative kurtosis, implying a more consistent and flat distribution.

The collected data underwent several pre-processing steps to prepare it for input into the deep learning models. These steps ensured that the models received data in a format conducive to effective learning. Many research papers use signal pre-processing techniques, such as discrete wavelet transform, wavelet packet transform, etc., before implementing ML and DL algorithms. In this work, it is not mandatory due to the following reasons.

- Direct Feature Extraction: The raw data's inherent characteristics are sufficient for the deep learning models to learn and detect anomalies without additional pre-processing.
- Model Robustness: Modern deep learning models are robust enough to handle raw data, reducing the need for extensive pre-processing.
- Computational Efficiency: Avoiding complex pre-processing steps can save computational resources and time, making the monitoring system more efficient.

6. Data Pre-Processing

The collected data underwent several pre-processing steps to prepare it for input into the deep learning models, ensuring that the models received data in a format conducive to effective learning, as shown in Table 2. While many research papers use signal pre-processing techniques such as discrete wavelet transform and wavelet packet transform before implementing machine learning and deep learning algorithms [41], this work found it unnecessary for several reasons.

Firstly, the raw data's inherent characteristics were sufficient for the deep learning models to learn and detect anomalies without additional pre-processing. Secondly, modern deep-learning models are robust enough to handle raw data, reducing the need for extensive pre-processing [54]. Lastly, avoiding complex pre-processing steps can save computational resources and time, making the monitoring system more efficient.

6.1. Sampling Strategy to Balance Classes

Given the potential imbalance in the data, a sampling strategy was implemented to balance the dataset. This involved oversampling the minority class to ensure the deep learning models received a balanced dataset. This approach is crucial in preventing the models from becoming biased toward the majority class, leading to more accurate and generalizable predictions.

6.2. Shuffling

The dataset was shuffled to randomize the order of the data points. This step prevents the models from learning any order-based biases, which could lead to overfitting and poor generalization of new data.

6.3. Standardization

Features were standardized using Standard Scaler, a pre-processing technique that transforms the data such that each feature has a mean of 0 and a standard deviation of 1. This process is vital for deep learning models, as it ensures that all features contribute equally to the learning process, avoiding biases toward features with more extensive numerical ranges.

6.4. Reshaping

The data was reshaped to match the input requirements of different deep-learning models. For instance:

- MLP: The data was formatted into 2D arrays where each row represents a sample, and each column represents a feature.
- CNN and LSTM: The data was reshaped into 3D arrays, with dimensions corresponding to the number of samples, timesteps, and features. This reshaping is crucial for CNNs and LSTMs to process spatial and temporal patterns effectively.

Table 2. Final dataset after balancing the classes or types

	F1	F2	F3	F4	F5	…	F98	F99	F100	F101	Type
0	0.004744	0.003381	0.000049	0.000010	0.000021	…	0.000048	0.000007	0.000019	0.000050	Balanced
1	0.007321	0.005198	0.000182	0.000145	0.000149	…	0.000141	0.000171	0.000690	0.001385	Balanced
2	0.005246	0.003740	0.000084	0.000055	0.000030	…	0.000017	0.000017	0.000037	0.000055	Balanced
3	0.007143	0.005072	0.000056	0.000129	0.000120	…	0.000175	0.000081	0.000673	0.001540	Balanced
4	0.005172	0.003660	0.000065	0.000044	0.000034	…	0.000033	0.000027	0.000025	0.000048	Balanced
…	…	…	…	…	…	…	…	…	…	…	…
195	0.005098	0.003610	0.000041	0.000030	0.000025	…	0.000040	0.000017	0.000032	0.000053	Balanced
196	0.007185	0.005023	0.000079	0.000042	0.000086	…	0.000071	0.000142	0.000527	0.001304	Balanced
197	0.007388	0.005169	0.000173	0.000168	0.000096	…	0.000171	0.000173	0.000702	0.001523	Balanced
198	0.008096	0.005748	0.000186	0.000234	0.000286	…	0.000326	0.000223	0.001016	0.001922	Imbalanced
199	0.007111	0.005128	0.000217	0.000289	0.000254	…	0.000300	0.000158	0.000691	0.001426	Imbalanced

6.5. Data Augmentation via Duplication

To increase the dataset's size and improve model robustness, the data was duplicated two times and four times in separate experiments. This duplication aimed to give the models more examples, helping them generalize better to unseen data. However, care was taken to ensure this augmentation did not introduce redundancy that could lead to overfitting, especially in models like LSTM.

7. Application of Deep Learning in Condition Monitoring

Deep learning is a machine learning branch involving neural networks with multiple layers. It enables the automatic extraction of complex patterns and features from large datasets. These models learn representations directly from raw data through layers of interconnected neurons, making them robust for tasks that involve classification, prediction, and pattern recognition [55].

This project used deep learning to analyze frequency-domain data from a rotary shaft for condition monitoring. Deep learning models allowed for automatically detecting imbalances and other faults in the machinery without manual feature engineering. By leveraging deep learning's ability to handle complex, nonlinear data, the models effectively differentiated between balanced and imbalanced states, demonstrating the practical application of these techniques in monitoring conditions based on frequency-domain signals.

Rationale for Algorithm Selection

The multilayer perceptron, convolution neural network, recurrent neural network, and long short-term memory models were selected based on their unique strengths in rotating machinery fault diagnosis. MLP is known for its robustness in structured data scenarios; CNN excels in spatial feature extraction, making it suitable for vibration analysis. RNN and LSTM are particularly effective for temporal sequences, which are essential for understanding dynamic operational states.

7.1. Multilayer Perceptron

MLP is a feedforward artificial neural network consisting of at least three layers of nodes: an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, in one layer is connected to every node in the next layer with an associated weight. These weights are adjusted during training using backpropagation, a supervised learning technique that minimizes the error between the predicted and actual outputs [56].

A connection between two nodes is assigned a weight value representing their relationship, as shown in Figure 10. A hierarchical connection has a weight property, and the node function can perform summation and activation operations. The summation function is:

$$S_j = \sum_{i=1}^{n} w_{i,j} I_i + \beta_j \qquad (2)$$

Where, n is the amount of input data, I_i is the input data, β_jIs the deviation and $w_{i,j}$ is the connection weight.

The output is obtained in the hidden layer using the activation function as,

$$f_j(x) = \frac{1}{1+e^{-s_j}} \qquad (3)$$

The output of the output layer cell in the MLP can be obtained by combining Equations (2) and (3),

$$y_i = f_j\left(\sum_{i=1}^{n} w_{i,j} I_i + \beta_j\right) \qquad (4)$$

MLPs are particularly effective for tasks where the relationship between input features and output labels is complex but does not involve spatial or temporal dependencies. They are widely used in pattern recognition, classification, and regression [57].

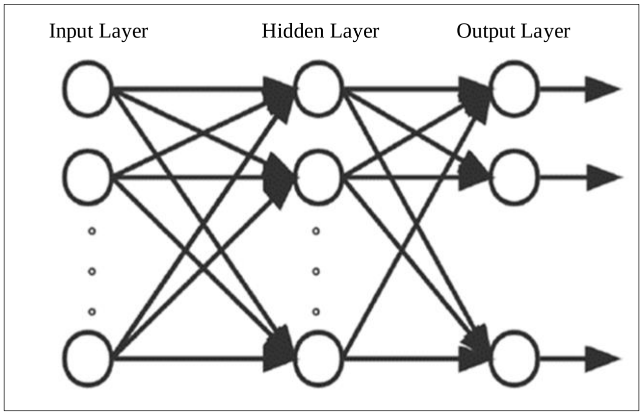


Fig. 10 Multilayer perceptron [56]

7.2. Convolutional Neural Network

CNN is a deep neural network designed to process data with a grid-like structure, such as images. CNNs utilize convolutional layers that filter the input data, identifying local patterns like edges and textures. These features are then aggregated and classified by fully connected layers. CNNs are highly effective in image recognition tasks because they can capture spatial hierarchies in data [58]. These networks have revolutionized computer vision tasks, achieving unprecedented performance in image classification, object detection, and various signal processing applications [59].

At the heart of CNNs lies the convolution operation, from which they derive their name, as shown in Figure 11. This mathematical operation enables the network to capture local patterns and spatial hierarchies within input data. In the context of image processing, the discrete convolution operation can be expressed as:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \quad (4)$$

Where:
S (i, j) is the output of the convolution operation at position (i, j).
I is the input matrix.
K is the kernel (filter) matrix.
(i, j) are the coordinates of the output matrix.
(m, n) are the coordinates of the input matrix.

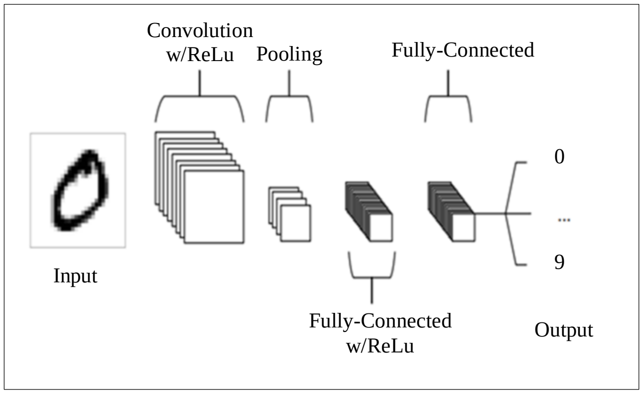


Fig. 11 Simple CNN architecture [54]

Equation (4) represents the convolution operation, where the kernel (K) is applied to the input (I) to produce the output (S). For each position (i, j) in the output matrix, the value is computed by summing the products of the overlapping elements of the input matrix and the kernel [60].

The architecture of a CNN typically comprises several specialized layers: Convolutional layers apply filters to the input data, detecting local patterns. Activation layers introduce nonlinearity. A common activation function is the Rectified Linear Unit (ReLU): The mathematical equation of CNN pooling, classification, and weight update rule is given in Equations (5), (6), (7), and (8).

$$f(x) = max(0, x)(2) \quad (5)$$

Pooling layers reduce spatial dimensions. Max pooling, a common operation, can be expressed as:

$$y_{ij} = max((p,q) \in R_{ij})x_{pq} \quad (6)$$

R_i,j represents a local neighborhood around position (i,j) [61].

Fully connected layers perform final classification based on extracted features:

$$y = \sigma(\sum_i w_i P_i + b) \quad (7)$$

Here,σ is an activation function, w_i are weights, x_i Re inputs, and b is a biased term.

The learning process in CNNs is facilitated by backpropagation, adjusting network parameters to minimize error. The weight update rule can be expressed as:

$$w_{ij} = w_{ij} - \eta \frac{\partial E}{\partial w_{ij}} \quad (8)$$

Where, η is the learning rate, and E is the error function [59].

CNNs offer several advantages over traditional neural networks in image processing tasks. They employ parameter sharing and local connectivity, significantly reducing the number of learnable parameters. Combined with translation invariance, these properties make CNNs exceptionally effective in capturing spatial hierarchies in visual data [55]. The power of CNNs lies in their ability to learn hierarchical representations automatically. As data progresses through the network, it transforms from raw pixel values to increasingly abstract features. This hierarchical learning enables CNNs to capture intricate patterns within images, often surpassing human-level accuracy in specific visual recognition tasks [61].

CNNs represent a significant advancement in machine learning and artificial intelligence, particularly in computer vision. Their unique architecture, built upon convolution and nonlinear activations, enables effective processing of grid-like data structures. As research progresses, CNNs continue to shape the future of AI and its applications across various domains.

7.3. Recurrent Neural Network

Recurrent Neural Networks (RNNs) are a class of neural networks where connections between nodes form a directed graph along a sequence, allowing them to maintain a memory of previous inputs. This means that RNN is a special artificial neural network adapted to work with time series or data involving sequences. RNN can be used for Language translation, Speech recognition, Time series forecasting, Text generation, Sentiment analysis, etc.

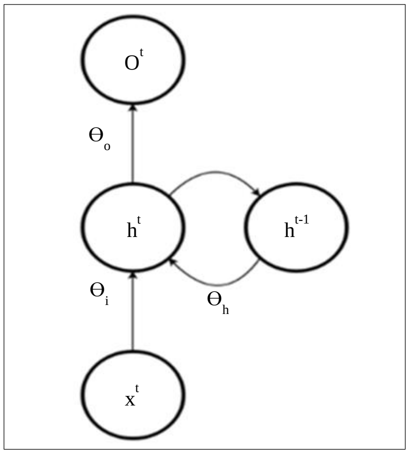


Fig. 12 RNN graphical model

In Figure 12, the values of θ_i, θ_h, and θ_o represent the parameters associated with the inputs, previous hidden layer states, and outputs. Equations (9) and (10) define how an RNN evolves over time.

$$O^t = f\ (\ h^t;\theta) \tag{9}$$

$$h^t = g\ (h^{t-1}, x^t;\theta) \tag{10}$$

Where O^t is the output of the RNN at time t, x^t is the input to the RNN at time t, and h^t is the state of the hidden layer(s) at time t. The image in Figure 12 shows a simple graphical model to illustrate the relation between these three variables in an RNN's computation graph.

Equation (9) says that for the given parameter θ, the output at the time t, depends on the hidden layer at the time t, in the feed forward neural network. Equation (10) says that, for the given parameters, θ, the hidden layer at time t depends on the hidden layer at time $t-1$ and the input at time t. Equation (10) demonstrates that the RNN can remember its past by allowing set computations h^{t-1} to influence the present computations h^t.

The aim of training the RNN is to get the sequence $o^{t+\tau}$ to match the sequence y_t, where τ represents the time lag (that y= 0) between the first meaningful RNN output $o^{\tau+1}$ and the first target output y_t. A time lag is sometimes introduced to allow the reach of an informal hidden state $h^{\ \tau+1}$ before it starts producing elements of the output sequence [62].

7.4. Long Short-Term Memory

Long Short-Term Memory (LSTM) units are a special RNN designed to capture long-term dependencies in sequential data by incorporating memory cells that store information across time steps. LSTMs overcome the vanishing gradient problem that affects standard RNNs, making them more effective in learning long-term dependencies [63]. LSTM has three gates: (i) Input Gate, (ii) Forget Gate, and (iii) Output Gate.

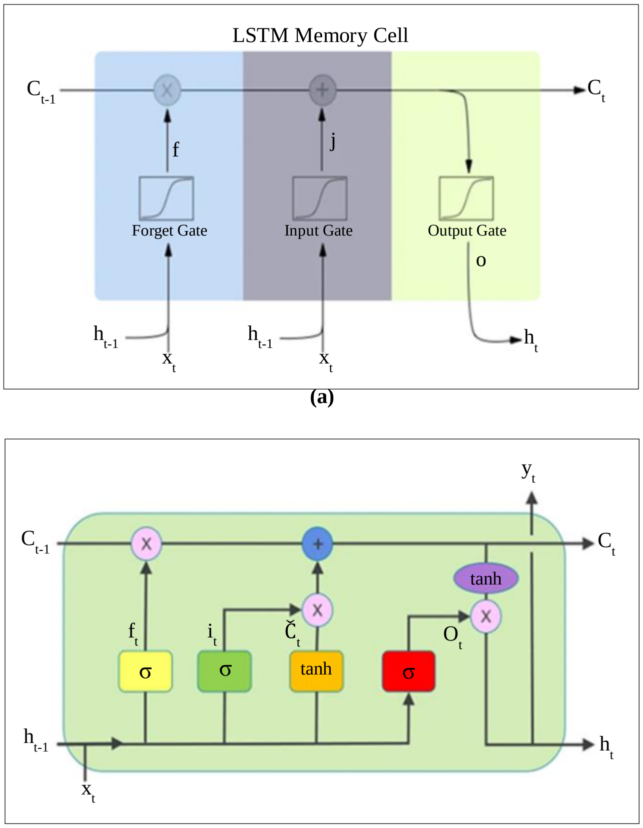
Gates in LSTM are the sigmoid activation functions, which means the output value is between 0 and 1, in which 0 means gates are blocked and 1 means gates are allowed. The gate comprises a sigmoid (σ) neural network layer. Equations (11), (12), and (13) are for the gates.

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \tag{11}$$

$$f_t = \sigma\big(w_f[h_{t-1}, x_t] + b_f\big) \tag{12}$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \tag{13}$$

i_t → represents the input gate.
f_t → represents the forget gate.
o_t → Represents the output gate.
σ → Represents a sigmoid function.
w_x → Weight for the respective gate (x) neurons.
h_{t-1} → the the the the the the the Output of the previous lstm block (at timestamp -1).
x_t → Input at current timestamp.
b_x → Biases for the respective gates (x).



(a)

(b)

Fig. 13 (a) LSTM Memory cell, and (b) Block of LSTM at any timestamp {t} [64].

Equation (11) tells what new information will be stored in the cell state. Equation (12) is for the forget gate, which throws the information away from the cell state. Equation (13) is the output gate, which activates the final output of the LSTM block at timestamp 't', as shown in Figure 13. Equations (14), (15), and (16) are for the cell state, candidate cell state, and the final output, respectively.

$$\tilde{c_t} = \tanh(w_c[h_{t-1}, x_t] + b_c) \tag{14}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c_t} \tag{15}$$

$$h_t = o_t * \tanh(c^t) \tag{16}$$

c_t → cell state(memory) at timestamp (t).
$\tilde{c_t}$ → represents a candidate for cell state at timestamp (t).
Note* others are the same as above.

The candidate is calculated to get the memory vector for the current timestamp (c_{t}). Now, the cell state knows what it needs to forget from the previous state (i.e. f_{t} * c_{t-1}) and what it needs to consider from the current timestamp (i.e. I_{t} * c`_{t}). Filtering in the cell state and then passing through the activation function predicts what portion should appear as the output of the current LSTM unit at timestamp t. Let us look at a block of LSTM at any timestamp {t}. h_{t} the output from the current LSTM block is passed through the softmax layer to get the predicted output(y_{t}) from the current block, as shown in Figure 13.

LSTMs are particularly useful for tasks involving sequential data, such as time series forecasting, natural language processing, and speech recognition. Their ability to retain information over long periods makes them ideal for modeling sequences with essential order and duration of events [65].

7.5. Experimental Design for Deep Learning

The following parameter settings were configured for each deep learning model applied in this study to optimize the performance.

7.5.1. Multilayer Perceptron

The MLP model is structured as follows:
Input Layer: Accepts input with dimension X
First Hidden Layer: 64 neurons with ReLU activation
Second Hidden Layer: 32 neurons with ReLU activation
Output Layer: 1 neuron with sigmoid activation
MLP is a feedforward neural network with two hidden layers. It can be described mathematically as:

Layer 1: $h1=ReLU(W1x + b1)$
Layer 2: $h2=ReLU(W2h1 + b2)$
Output: $y=\sigma(W3h2 + b3)$

Where, ReLU Activation: Rectified Linear Unit $f(x) = \max(0, x)$is used in the hidden layers. ReLU helps mitigate the vanishing gradient problem and allows for faster training.

Sigmoid Activation: The output layer uses the sigmoid function $\sigma(x) = 1 + e - x1$, which maps the output to a probability between 0 and 1, suitable for binary classification tasks.

Learning Rate: This is not explicitly specified in the code, but typically, the Adam optimizer with a default learning rate of 0.001 is used.

7.5.2. Convolutional Neural Network

The CNN architecture is structured as follows:
Conv1D Layer: 64 filters, kernel size of 3, ReLU activation.
MaxPooling1D Layer: Pool size of 2.
Flatten Layer: Converts 2D feature maps to a 1D feature vector.
Dense Layer: 64 neurons with ReLU activation.
Output Layer: 1 neuron with sigmoid activation.
The CNN applies convolution operations followed by pooling:
Conv Layer: $a = ReLU(W * x + b)$, where * denotes convolution.
Pooling: $p = \max pool(a)$ reduces spatial dimensions and provides translation invariance.
Flatten: Converts the pooled features into a 1D vector f=flatten(p).
Dense Layer: $d = ReLU(W1f + b1)$
Output: $y = \sigma(W2d + b2)$
Convolutional Layer: Applies 64 filters of size 3 to extract local features from the input sequence.

Pooling Strategy: Max pooling with a size of 2 reduces the spatial dimensions and helps achieve translation invariance.

Filter Sizes: The kernel size of 3 captures local patterns in the input data, which can be essential in sequence analysis.

7.5.3. Recurrent Neural Network

The RNN model consists of:
SimpleRNN Layer: 64 units with tanh activation.
Output Layer: 1 neuron with sigmoid activation.
The SimpleRNN can be described as:

$$h_t = \tan h\,(W_{hh}h_{t-1} + W_{xh}x_t + b_h$$

$$y = \sigma(W_{h_y}h_t + b_y)$$

Where: h_t is the hidden state at time t, and x_t is the input at time t.

RNN Type: Simple RNN. This architecture is known for its limitations in capturing long-term dependencies due to vanishing/exploding gradient problems.

Hidden Units: 64 recurrent units in the RNN layer.

Activation: tanh is used, which is standard for RNNs because it maintains better stability across long sequences compared to ReLU.

7.5.4. Long Short-Term Memory

The LSTM architecture is structured as follows:
LSTM Layer: 64 units with tanh activation.
Output Layer: 1 neuron with sigmoid activation.

- LSTMs are designed to address the vanishing gradient problem by using memory cells and gating mechanisms.
- 64 LSTM units are stored and updated in the hidden state based on the input and past hidden state.

- tanh is used, which is the standard activation for LSTM units. It's responsible for squashing values into the range [−1,1][-1, 1][−1,1] inside the LSTM cell and helps in handling long-term dependencies better.

The performance of each model was evaluated using the following metrics:

- Accuracy: Indicates the overall correctness of the model's predictions.
- Precision: Reflects the model's ability to identify true positives among predicted positives.
- Recall: Measures the model's ability to detect all actual positives.
- F1 score: Provides a balance between precision and recall, especially important in imbalanced datasets.

Before delving into the detailed comparison of results, it is essential first to explain the rationale behind the experimental setup, which involves evaluating the models across three sections-one with the original data, one with doubled data, and one with quadrupled data.

This structure is designed to assess how well different deep learning models MLP, CNN, RNN, and LSTM perform with varying data volumes, with the primary goals being to:

- Evaluate Model Scalability: Testing with increased data allows us to observe how effectively each model scales as the dataset grows. Some models may perform better with small datasets but struggle as the data volume increases, while others might improve in performance with more data.
- Analyse Generalization and Overfitting: By comparing the training and validation accuracy and loss across different dataset sizes, it can be identified whether a model is overfitting (performing well on training data but poorly on validation data) or generalizing well to new, unseen data.
- Test Model Stability: Larger datasets can reduce the effect of noise and randomness, leading to smoother learning curves. This comparison helps determine which models remain stable and consistent under varying conditions.

To assess the impact of data scaling on model performance, the dataset was processed in three distinct variations: original, doubled, and quadrupled. This approach allows us to evaluate how each model adapts as the complexity and volume of data increase:

- Normal Data: This baseline comparison focuses on the model's performance with the original dataset given in Table 5. The results provide insight into how well the model can learn patterns from the initial data configuration.
- Doubled Data: The dataset is artificially increased by duplicating the data to simulate a scenario with more data available, as shown in Table 6. This approach simulates a scenario with more data, helping assess how model performance changes with larger datasets.
- Quadrupled Data: Quadrupling the dataset given in Table 7 tests the model's behaviour under even more extensive data volumes. The results highlight whether the model continues to improve, stabilizes, or begins to exhibit issues such as overfitting.

7.6. Observations on Model Performance across Scaled Datasets

The following subsections break down the performance of each model based on different dataset scales:

7.6.1. Model Performance for the Normal Dataset

This subsection evaluates each model's performance using the original dataset, focusing on critical metrics like accuracy, precision, recall, and F1-score, as shown in Table 3.

Table 3. Model performance for normal dataset

Model	Accuracy	Precision	Recall	F1-Score
MLP	0.9	0.917391	0.9	0.899499
CNN	0.95	0.954762	0.95	0.95
RNN	0.575	0.646094	0.575	0.533908
LSTM	0.45	0.453419	0.45	0.43312

7.6.2. Model Performance for the Doubled Dataset

In this section, the dataset is doubled to analyze whether the increased volume affected each model's learning capability and generalization. The performance metrics are compared with the normal dataset to identify changes, as shown in Table 4.

Table 4. Model performance for doubled dataset

Model	Accuracy	Precision	Recall	F1-Score
MLP_2	1.0	1.0	1.0	1.0
CNN_2	0.9125	0.928618	0.9125	0.913573
RNN_2	0.65	0.816102	0.65	0.634444
LSTM_2	0.6375	0.812708	0.6375	0.619083

7.6.3. Model Performance for the Quadrupled Dataset

The dataset size was quadrupled to assess how each model handles a significant increase in data volume and complexity. The metrics given in Table 5 are reviewed to determine whether models can effectively scale and maintain robust performance. The comparative analysis highlights that models like MLP and CNN adapted well to data scaling, consistently maintaining high accuracy and stable learning, while RNN and LSTM struggled with larger datasets. The ranking of the models based on their scalability and stability underscores the adaptability of specific models over others when faced with increasing data complexity.

Table 5. Model performance for quadrupled dataset

Model	Accuracy	Precision	Recall	F1-Score
MLP_4	1.0	1.0	1.0	1.0
CNN_4	0.98125	0.981944	0.98125	0.981252
RNN_4	0.90625	0.921371	0.90625	0.905643
LSTM_4	0.55	0.550312	0.55	0.55007

7.7. Ranking the Performance of All AI Models

This section ranks the models based on their overall performance across all datasets, considering the consistency of accuracy, precision, recall, and F1-score, as provided in Table 6.

Table 6. DL algorithms ranked on performance

Model	Accuracy	Precision	Recall	F1-Score
MLP	0.9	0.917391	0.9	0.899499
CNN	0.95	0.954762	0.95	0.95
RNN	0.575	0.646094	0.575	0.533908
LSTM	0.45	0.453419	0.45	0.43312
MLP_2	1.0	1.0	1.0	1.0
CNN_2	0.9125	0.928618	0.9125	0.913573
RNN_2	0.65	0.816102	0.65	0.634444
LSTM_2	0.6375	0.812708	0.6375	0.619083
MLP_4	1.0	1.0	1.0	1.0
CNN_4	0.98125	0.981944	0.98125	0.981252
RNN_4	0.90625	0.921371	0.90625	0.905643
LSTM_4	0.55	0.550312	0.55	0.55007

8. Comparison between MLP, CNN, and RNN and its Analysis

8.1. Multilayer Perceptron

The MLP model consistently delivered near-perfect performance across all scaled datasets: normal, doubled, and quadrupled, as shown in Figures 14, 15 and 16. This consistency is reflected in the model's high accuracy, precision, recall, and F1 score across all scales, making it the most reliable and stable model for this application. The accuracy curves demonstrate close alignment between training and validation sets, indicating strong learning capability with minimal overfitting. The loss curves decrease smoothly and converge closely in all three dataset variations, highlighting effective learning and robustness. Among the datasets, the quadrupled set performed best, achieving near-perfect accuracy with smoothly converging loss curves, reinforcing the model's scalable, stable performance capacity.

8.2. Convolutional Neural Network

The Convolutional Neural Network model performed exceptionally well across all scaled datasets, showing only slight variations in performance while maintaining vital accuracy and convergence, as shown in Figures 14, 15, and 16.

The CNN models demonstrated robust adaptability as data was scaled, with closely aligned training and validation accuracy curves that stabilized early across all datasets. This indicates strong generalization and effective learning. The loss curves consistently decreased and converged smoothly, reflecting minimal overfitting. The doubled dataset stood out as the best, with smooth accuracy and well-converged loss curves, striking an ideal balance between learning speed and stability. Notably, the CNN_4 model in the quadrupled dataset performed very closely to the MLP, showcasing CNN's ability to maintain high performance under larger data volumes.

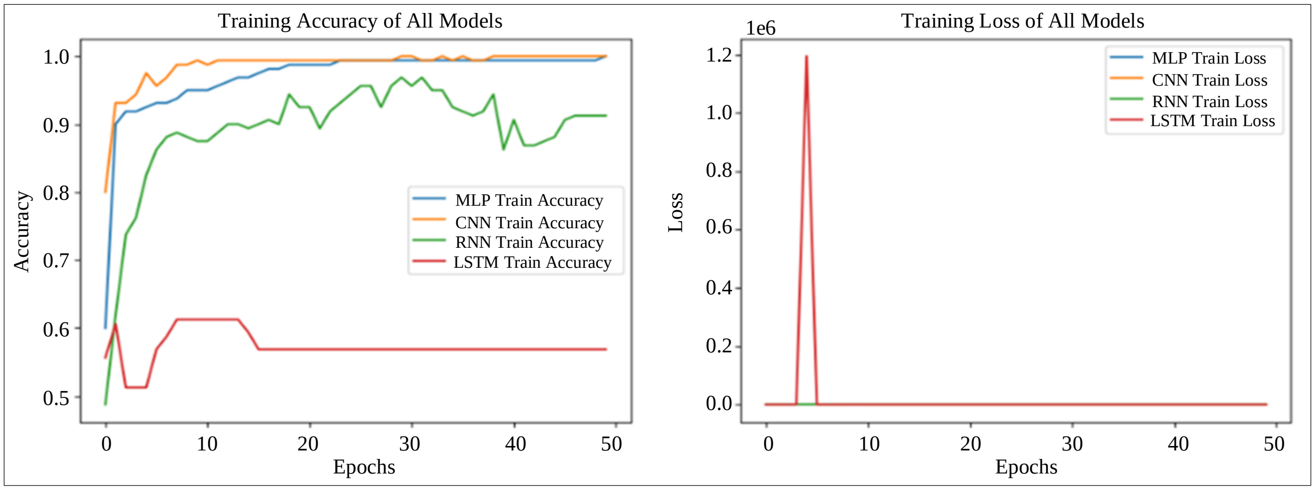


Fig. 14 Training and validation metrics for normal dataset

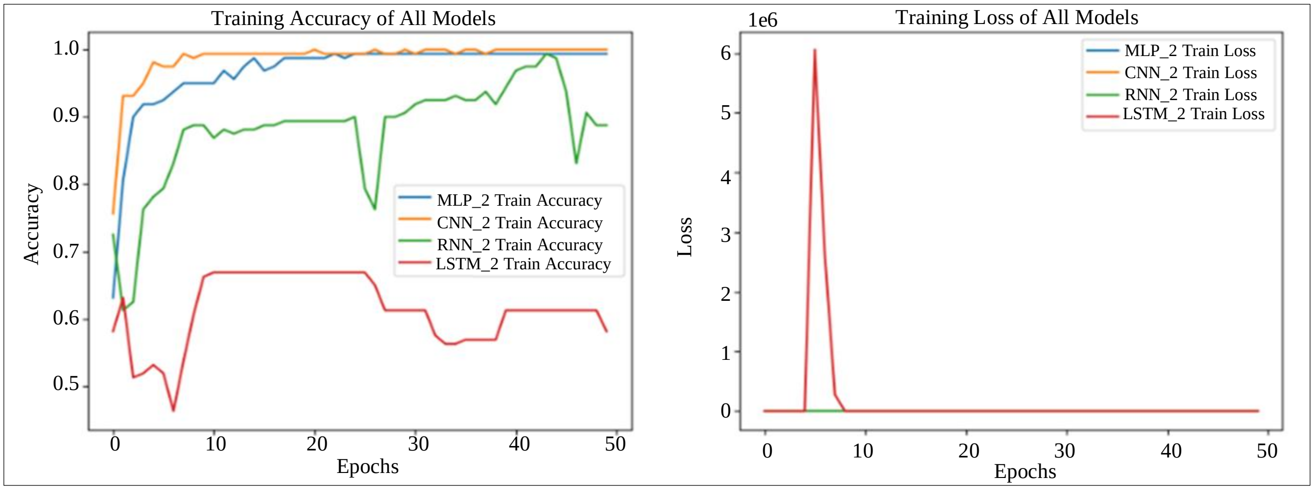


Fig. 15 Training and validation metrics for doubled dataset

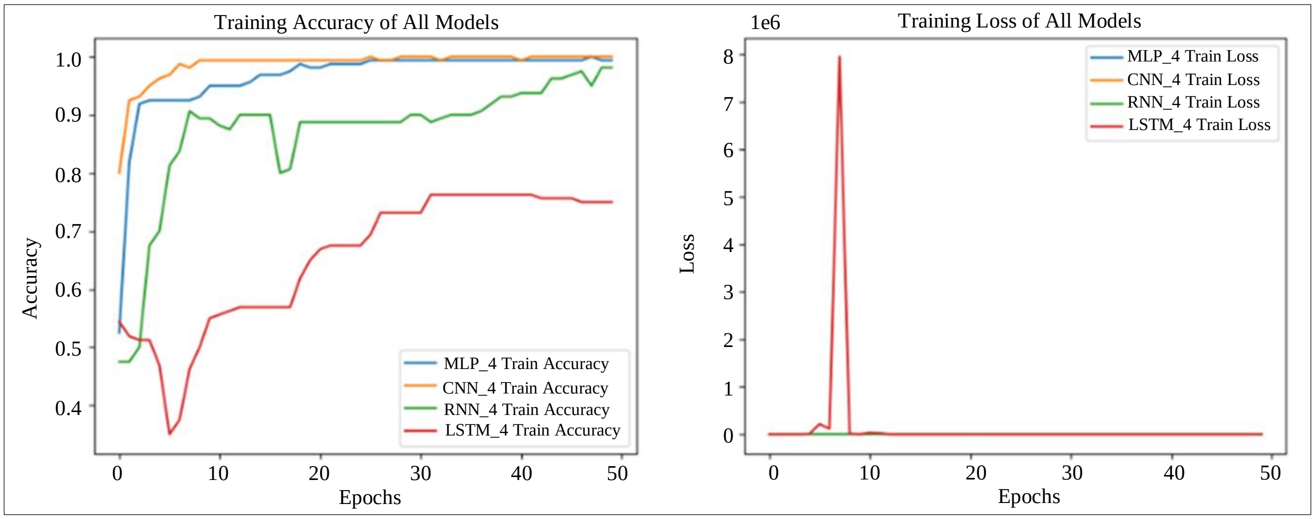


Fig. 16 Training and validation metrics for the quadrupled dataset

8.3. Recurrent Neural Network

The Recurrent Neural Network model displayed fluctuating performance as data was scaled, with accuracy and generalization stability less consistent than the MLP and CNN models. While the RNN models performed reasonably well, they showed more variation between training and validation accuracy curves, mainly as dataset size increased. This fluctuation suggests that RNNs struggle to learn across different data volumes consistently.

The loss curves for RNNs revealed greater volatility, indicating less stable learning, especially with larger datasets. Despite these challenges, the RNN models achieved decent alignment between training and validation curves, particularly in the quadrupled dataset, the best-performing set among the RNN models. Although capable, RNNs were less reliable in scaling effectively than simpler architectures like MLP and CNN.

8.4. Long Short-Term Memory

The Long Short-Term Memory models encountered the most difficulty when dealing with data scaling, with significant drops in accuracy, precision, recall, and F1-score as the dataset size increased. The models showed notable fluctuations in accuracy across all datasets, struggling to maintain stable learning and generalization. This issue was particularly pronounced in larger datasets, where the models had trouble capturing patterns effectively.

The loss curves were highly erratic, featuring sharp spikes, especially in validation loss, signaling overfitting and poor generalization. The most problematic performance was observed in the quadrupled dataset, where the model exhibited highly unstable behavior with substantial variation in accuracy and loss, making it the weakest performer overall. This inconsistency indicates that LSTM models were the least reliable when scaling data in this experiment.

8.5. Factors Influencing Performance

Understanding the reasons behind the performance variations among the models provides the following insights:

- Data characteristics: MLP and CNN models are well suited for structured data with clear patterns that contribute to their high performance. RNN and LSTMs designed for sequential data struggled due to the nature of the dataset, which lacks temporal dependencies.
- Model Complexity: The simplicity of MLP led to practical learning with minimal overfitting. Feature extraction from vibration data was better due to CNN's convolutional layers. On the other hand, RNN and LSTM introduced challenges when scaling due to input size and structure.
- Task complexity: Diagnosing faults in rotating machinery involves recognizing distinct patterns and features in the data. MLPs and CNN excelled at this due to their architectures, while RNN and LSTMs faced difficulties capturing these patterns in larger datasets.

The performance analysis across scaled datasets reveals that the Multilayer Perceptron consistently outperformed other models, demonstrating near-perfect accuracy, stability, and generalization across all dataset sizes. The Convolutional Neural Network also performed strongly, showing robust accuracy and effective learning with minimal overfitting even as data was scaled. While the Recurrent Neural Network achieved decent results, it exhibited more fluctuations in accuracy and stability, especially in larger datasets.

However, the Long Short-Term Memory model struggled the most, with significant drops in performance, erratic loss curves, and pronounced overfitting issues as the dataset size increased. Overall, the analysis highlights that simpler models like MLP and CNN were better suited for this specific problem, while LSTM struggled to handle the complexity introduced by larger datasets.

9. Conclusion and Future Recommendation

By systematically varying fault conditions, the experimental setup in this work aimed to replicate realistic operational scenarios that machinery might experience in the manufacturing machine.

The results derived from both balanced and imbalanced conditions offer a robust foundation for condition monitoring and analysis. The controlled introduction of weights provides a comprehensive dataset for training and validating diagnostic models, which is essential for detecting and predicting faults.

According to the analysis and discussions in the previous section, the methods of DL, Which include MLP, CNN, and RNN, adaptively extract the fault information and overcome the disadvantages of many traditional methods. The findings from this study contribute to the development of more accurate and reliable condition monitoring techniques, enhancing predictive maintenance strategies. This, in turn, can lead to extended operational life, reduced downtime, and improved overall efficiency of industrial machinery. Compared to the algorithms between MLP, CNN, RNN, and LSTM, CNN has given 95% accuracy compared to the MLP of 90%.

The RNN and LSTM are not providing promising results on the dataset taken for analysis. This does not mean RNN and LSTM will not give accurate results in all cases. The lower performance of RNN and LSTM may be attributed to their sensitivity to noise or the specific temporal dynamics of the data not aligning well with these models' strengths. CNN's superior performance is primarily due to its ability to capture spatial patterns and localized features in the data, making it well-suited for tasks involving structured or spatially dependent information [47].

However, MLP's robust performance, despite its simpler architecture, demonstrates its adaptability and consistent learning across scaled datasets, giving an accuracy of 100%. MLP's fully connected structure allows it to utilize all input features equally without prioritizing spatial hierarchies, which can be advantageous when the task does not rely heavily on spatial dependencies. So, it is concluded that the DL is the future of maintenance engineering, and fault diagnosis will be more accurate in this modern industrial revolution.

Although the proposed methods in this research have achieved some promising results in rotary machinery, there are still many challenges in the current research in selecting the right DL technique and future research directions. There are some limitations in this study.

1. Hyperparameter Tuning: The algorithms were not fine-tuned through hyperparameter optimization, which may have limited their potential to achieve the highest possible accuracy.
2. Limited Data Volume: The dataset was relatively small, which constrained the model's ability to generalize, particularly in predicting fault weights accurately.
3. Minimal Pre-processing: Data pre-processing was intentionally kept minimal, which, while simplifying the model pipeline, may have restricted the model's ability to leverage the data's informative features fully.
4. Limited Generalizability: The findings are based on the Spectra Quest Machinery Fault Simulator data. Performance may vary when these models are applied to real-world scenarios with different machinery, operating conditions, or fault types.

The following research directions are proposed to address these limitations.

- Hybrid Models: Investigating the potential of hybrid models that combine the strengths of various deep learning techniques to enhance performance.
- Incorporating additional data types: Exploring the integration of various data types, such as thermal data and historical maintenance records, to provide a more comprehensive view of machinery health.
- Application to different machinery and fault conditions: Applying the methods proposed to diverse machinery and varying fault conditions to assess the generalizability in real-world settings.
- Hyperparameter optimization: Implementing hyperparameter optimization techniques to get the full potential of the chosen algorithms for improved accuracy and robustness.

Author Contributions

Conceptualization, K.V., and R.A .; Literature collection, K.V., R.A., M.K and S.RK.A.V; Formal analysis, R.A, K.V. and M.K.; Funding acquisition, R.A., K.V. and KPR; Investigation, R.A., K.V and M.K; Methodology, R.A., KPR, K.V., M.K., and S.RK.A.V; Validation, R.A. and K.V.; Writing-original draft, R.A.; Review and editing, R.A., M.K., KPR, K.V. and S.RK.A.V. All authors have read and agreed to the published version of the manuscript

Funding

This work was performed as part of the research work titled" Deep learning based real-time drill condition monitoring using acoustic emission sensor", funded by MOHERI, OMAN, grant number BFP/RGP/EI/22/433.

Institutional Review Board Statement

This study was conducted according to the National University of Science and Technology, Oman guidelines.

Data Availability Statement

The original contributions presented in the study are included in the article/supplementary material. Further inquiries can be directed to the corresponding author/s.

References

[1] H.S. Kumar, and Gururaj Upadhyaya, "Fault Diagnosis of Rolling Element Bearing Using Continuous Wavelet Transform and K- Nearest Neighbour," *Materials Today: Proceedings*, vol. 92, pp. 56-60, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Anirbid Sircar et al., "Application of Machine Learning and Artificial Intelligence in Oil and Gas Industry," *Petroleum Research*, vol. 6, no. 4, pp. 379-391, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] Karali Patra, "Acoustic Emission Based Tool Condition Monitoring System in Drilling," *Proceedings of the World Congress on Engineering*, London, pp. 1-5, 2011. [Google Scholar] [Publisher Link]

[4] Shuping Cao et al., "Research on Fault Diagnosis Technology of Centrifugal Pump Blade Crack Based on PCA and GMM," *Measurement*, vol. 173, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Onur Surucu, Stephen Andrew Gadsden, and John Yawney, "Condition Monitoring Using Machine Learning: A Review of Theory, Applications, and Recent Advances," *Expert Systems with Applications*, vol. 221, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] G. Serin et al., "Review of Tool Condition Monitoring in Machining and Opportunities for Deep Learning," *The International Journal of Advanced Manufacturing Technology*, vol. 109, no. 3-4, pp. 953-974, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] Josef Koutsoupakis, Panagiotis Seventekidis, and Dimitrios Giagopoulos, "Machine Learning Based Condition Monitoring for Gear Transmission Systems Using Data Generated by Optimal Multibody Dynamics Models," *Mechanical Systems and Signal Processing*, vol. 190, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] Long Zhang et al., "An Imbalanced Fault Diagnosis Method Based on TFFO and CNN for Rotating Machinery," *Sensors*, vol. 22, no. 22, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Erkki Jantunen, "A Summary of Methods Applied to Tool Condition Monitoring in Drilling," *International Journal of Machine Tools & Manufacture*, vol. 42, no. 9, pp. 997-1010, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[10] Pushkar Dehspande et al., "Acoustic Emission and Machine Learning Based Classification of Wear Generated Using A Pin-on-Disc Tribometer Equipped with A Digital Holographic Microscope," *Wear*, vol. 476, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Rui Zhao et al., "Deep Learning and Its Applications to Machine Health Monitoring," *Mechanical Systems and Signal Processing*, vol. 115, pp. 213-237, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[12] Lang Dai et al., "An Improved Deep Learning Model for Online Tool Condition Monitoring Using Output Power Signals," *Shock and Vibration*, vol. 2020, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[13] Qun Wang et al., "Overview of Tool Wear Monitoring Methods Based on Convolutional Neural Network," *Applied Sciences*, vol. 11, no. 24, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[14] Mariela Cerrada et al., "A Review on Data-Driven Fault Severity Assessment in Rolling Bearings," *Mechanical Systems and Signal Processing*, vol. 99, pp. 169-196, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[15] Olivier Janssens et al., "Convolutional Neural Network Based Fault Detection for Rotating Machinery," *Journal of Sound and Vibration*, vol. 377, pp. 331-345, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[16] Yong Yao et al., “Learning Attention Representation with a Multi-Scale CNN for Gear Fault Diagnosis under Different Working Conditions,” *Sensors*, vol. 20, no. 4, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Wei Zhang et al., “A Deep Convolutional Neural Network with New Training Methods for Bearing Fault Diagnosis under Noisy Environment and Different Working Load,” *Mechanical Systems and Signal Processing*, vol. 100, pp. 439-453, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[18] Manar Abdelmaksoud et al., “Convolutional-Neural-Network-Based Multi-Signals Fault Diagnosis of Induction Motor Using Single and Multi-Channels Datasets,” *Alexandria Engineering Journal*, vol. 73, pp. 231-248, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[19] Tshilidzi Marwala, “Multi-Layer Perceptron for Condition Monitoring in a Mechanical System,” *Condition Monitoring Using Computational Intelligence Methods*, pp. 53-69, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[20] Dino Zanic, and Alan Zupan, “Monitoring Transformer Condition with MLP Machine Learning Model,” *The Journal of Energy*, vol. 72, no. 2, pp. 3-7, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[21] Van-Quang Nguyen et al., “A Deep Learning Approach Based on MLP-Mixer Models for Bearing Fault Diagnosis,” *2023 International Conference on System Science and Engineering (ICSSE)*, Ho Chi Minh, Vietnam, pp. 16-21, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] Khoualdia Tarek et al., “Optimized Multi-Layer Perceptron Artificial Neural Network-Based Fault Diagnosis of Induction Motor Using Vibration Signals,” *Diagnostyka*, vol. 22, no. 1, pp. 65-74, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[23] Dhiraj Neupane et al., “CNN-Based Fault Detection for Smart Manufacturing,” *Applied Sciences*, vol. 11, no. 24, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[24] Andreas Lundgren, and Daniel Jung, “Data-Driven Fault Diagnosis Analysis and Open-Setclassification of Time-Series Data,” *Control Engineering Practice*, vol. 121, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[25] Guoqiang Li et al., “Deep Reinforcement Learning-Based Online Domain Adaptation Method for Fault Diagnosis of Rotating Machinery,” *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 5, pp. 2796-2805, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[26] Yongxiang Lei et al., “Processes Soft Modeling Based on Stacked Autoencoders and Wavelet Extreme Learning Machine for Aluminumplant-Wide Application,” *Control Engineering Practice*, vol. 108, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[27] Jinghui Tian et al., “A Multi-Source Information Transfer Learning Method with Subdomain Adaptation for Cross-Domain Fault Diagnosis,” *Knowledge-Based Systems*, vol. 243, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[28] Daoguang Yang, Hamid Reza Karimi, and Len Gelman, “A Fuzzy Fusion Rotating Machinery Fault Diagnosis Framework Based on the Enhancement Deep Convolutional Neural Networks,” *Sensors*, vol. 22, no. 2, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[29] Yongxiang Lei, Hamid Reza Karimi, and Xiaofang Chen, “A Novel Self-Supervised Deep LSTM Network for Industrial Temperature Prediction in Aluminum Processes Application,” *Neurocomputing*, vol. 502, pp. 177-185, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[30] Funa Zhou et al., “The Deep Learning Fault Diagnosis Method is Based on Global Optimization GAN for Unbalanced Data,” *Knowledge-Based Systems*, vol. 187, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[31] Haobo Wang et al., “Approach to the Quantitative Diagnosis of Rolling Bearings Based on Optimized VMD and Lempel-Ziv Complexity under Varying Conditions,” *Sensors*, vol. 23, no. 8, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[32] Liyue Chen et al., “Temperature Prediction of Seasonal Frozen Subgrades Based on CEEMDAN-LSTM Hybrid Model,” *Sensors*, vol. 22, no. 15, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[33] Levent Eren, Turker Ince, and Serkan Kiranyaz, “A Generic Intelligent Bearing Fault Diagnosis System Using Compact Adaptive 1DCNN Classifier,” *Journal of Signal Processing Systems*, vol. 91, pp. 179-189, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[34] Wei Zhang, Xiang Li, and Qian Ding, “Deep Residual Learning-Based Fault Diagnosis Method for Rotating Machinery,” *ISA Transactions*, vol. 95, pp. 295-305, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[35] Diwang Ruan et al., “CNN Parameter Design Based on Fault Signal Analysis and Its Application in Bearing Fault Diagnosis,” *Advanced Engineering Informatics*, vol. 55, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[36] Achyuth Kothuru, Sai Prasad Nooka, and Rui Liu, “Application of Deep Visualization in CNN-Based Tool Condition Monitoring for End Milling,” *Procedia Manufacturing*, vol. 34, pp. 995-1004, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[37] Feng Jia et al., “Deep Normalized Convolutional Neural Network for Imbalanced Fault Classification of Machinery and Its Understanding via Visualization,” *Mechanical Systems and Signal Processing*, vol. 110, pp. 349-367, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[38] Serhat Seker, Emine Ayaz, and Erduinc Turkcan, “Elman's Recurrent Neural Network Applications to Condition Monitoring in Nuclear Power Plant and Rotating Machinery,” *Engineering Applications of Artificial Intelligence*, vol. 16, no. 7-8, pp. 647-656, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[39] Chris Halliday et al., “A Recurrent Neural Network Method for Condition Monitoring and Predictive Maintenance of Pressure Vessel Components,” *Pressure Vessels & Piping Conference*, Las Vegas, Nevada, USA, pp. 1-5, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[40] Swetha R. Kumar, and Jayaprasanth Devakumar, “Recurrent Neural Network Based Sensor Fault Detection and Isolation for Nonlinear Systems: Application in PWR,” *Progress in Nuclear Energy*, vol. 163, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[41] Yahui Zhang et al., “Fault Diagnosis of Rotating Machinery Based on Recurrent Neural Networks,” *Measurement*, vol. 171, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[42] Zhuang Ye, and Jianbo Yu, “Health Condition Monitoring of Machines Based on Long Short-Term Memory Convolutional Autoencoder,” *Applied Soft Computing*, vol. 107, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[43] Haitao Zhao, Shaoyuan Sun, and Bo Jin, “Sequential Fault Diagnosis Based on LSTM Neural Network,” *IEEE Access*, vol. 6, pp. 12929-12939, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[44] Yasir Saleem Afridi et al., “LSTM-Based Condition Monitoring and Fault Prognostics of Rolling Element Bearings Using Raw Vibrational Data,” *Machines*, vol. 11, no. 5, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[45] Jiaqi Xie et al., “An End-to-End Model Based on Improved Adaptive Deep Belief Network and Its Application to Bearing Fault Diagnosis,” *IEEE Access*, vol. 6, pp. 63584-63596, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[46] Shaohua Qiu et al., “Deep Learning Techniques in Intelligent Fault Diagnosis and Prognosis for Industrial Systems: A Review,” *Sensors*, vol. 23, no. 3, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[47] SpectraQuest, Inc., User Operating Manual for Machinery Fault Simulator™, Richmond, 2020. [Online]. Available: https://spectraquest.com/machinery-fault-simulator/details/mfs/

[48] Selina S. Y. Ng, Peter W. Tse, and Kwok L. Tsui, “A One-Versus-All Class Binarization Strategy for Bearing Diagnostics of Concurrent Defects,” *Sensors*, vol. 14, no. 1, pp. 1295-1321, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[49] Soha A. Nossier et al., “A Comparative Study of Time and Frequency Domain Approaches to Deep Learning Based Speech Enhancement,” *2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, UK, pp. 1-8, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[50] Kun Yi et al., “Frequency-Domain MLPs are More Effective Learners in Time Series Forecasting,” *37th Conference on Neural Information Processing Systems*, pp. 1-24, 2023. [Google Scholar] [Publisher Link]

[51] Ankur Mali et al., “Neural JPEG: End-to-End Image Compression Leveraging a Standard JPEG Encoder-Decoder,” *2022 Data Compression Conference (DCC)*, Snowbird, UT, USA, pp. 471-471, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[52] Lars Hertel, Huy Phan, and Alfred Mertins, “Comparing Time and Frequency Domain for Audio Event Recognition Using Deep Learning,” *2016 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, Canada, pp. 3407-3411, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[53] Hamid Karimi, *Fault Diagnosis and Prognosis Techniques for Complex Engineering Systems*, Springer, Berlin, 2021. [Google Scholar] [Publisher Link]

[54] Keiron O'Shea, and Ryan Nash, “An Introduction to Convolutional Neural Networks,” *arXiv Preprint*, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[55] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, “Deep Learning,” *Nature*, vol. 521, no. 7553, pp. 436-444, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[56] Chaochun Zhong et al., “Improved MLP Energy Meter Fault Diagnosis Method Based on DBN,” *Electronics*, vol. 12, no. 4, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[57] Kurt Hornik, Maxwell Stinchcombe, and Halbert White, “Multilayer Feedforward Networks are Universal Approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359-366, 1989. [CrossRef] [Google Scholar] [Publisher Link]

[58] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097-1105, 2012. [Google Scholar] [Publisher Link]

[59] Kaiming He et al., “Deep Residual Learning for Image Recognition,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, pp. 770-778, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[60] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, Cambridge, MA: MIT Press, 2016. [Google Scholar] [Publisher Link]

[61] Kaiming He et al., “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, pp. 1026-1034, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[62] John McGonagle, Christopher Williams, and Jimin Khim, Recurrent Neural Network. [Online]. Available: https://brilliant.org/wiki/recurrentneural-network/

[63] Sepp Hochreiter, and Jurgen Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997. [CrossRef] [Google Scholar] [Publisher Link]

[64] Divyanshu Thakur, LSTM and its Equations, Median, 2018. [Online]. Available: https://medium.com/@divyanshu132/lstm-and-its-equations-5ee9246d04af

[65] Alex Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, Springer Science & Business Media, 2012. [CrossRef] [Google Scholar] [Publisher Link]