technologies

MDPI

Article

Predicting the Temperature of a Permanent Magnet Synchronous Motor: A Comparative Study of Artificial Neural Network Algorithms

Nabil El Bazi [1,2,*], Nasr Guennouni [1], Mohcin Mekhfioui [1], Adil Goudzi [3], Ahmed Chebak [1] and Mustapha Mabrouki [2]

[1] Green Tech Institute (GTI), Mohammed VI Polytechnic University (UM6P), Benguerir 43150, Morocco; nasr.guennouni@um6p.ma (N.G.); mohcin.mekhfioui@um6p.ma (M.M.); ahmed.chebak@um6p.ma (A.C.)
[2] Laboratory of Industrial Engineering (LGIIS), Faculty of Science and Technology, University Sultan Moulay Slimane (USMS), Beni Mellal 23000, Morocco; m.mabrouki@usms.ma
[3] TechniX, Mohammed VI Polytechnic University (UM6P), Benguerir 43150, Morocco; adil.goudzi@um6p.ma
* Correspondence: nabilelbazi1994@gmail.com; Tel.: +212-666-343-887

Abstract: The accurate prediction of temperature in Permanent Magnet Synchronous Motors (PMSMs) has always been essential for monitoring performance and enabling predictive maintenance in the industrial sector. This study examines the efficiency of a set of artificial neural network (ANN) models, namely Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), and Convolutional Neural Network (CNN), in predicting the Permanent Magnet Temperature. A comparative evaluation study is conducted using common performance indicators, including root mean square error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2), to assess the predictive accuracy of each model. The intent is to identify the most favorable model that balances high accuracy with low computational cost.

Keywords: Permanent Magnet Synchronous Motor (PMSM); temperature prediction; machine learning (ML); artificial neural network (ANN)

check for updates

Academic Editors: Jianxiong Zhu, Xiaodong Yang, Juan Wei, Xiaoshun Zhang and Da Xu

Received: 10 February 2025
Revised: 7 March 2025
Accepted: 12 March 2025
Published: 17 March 2025

Citation: El Bazi, N.; Guennouni, N.; Mekhfioui, M.; Goudzi, A.; Chebak, A.; Mabrouki, M. Predicting the Temperature of a Permanent Magnet Synchronous Motor: A Comparative Study of Artificial Neural Network Algorithms. *Technologies* **2025**, *13*, 120. https://doi.org/10.3390/technologies13030120

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The condition monitoring and diagnosis of PMSMs are one of the most important and promising directions for the development of effective health index management [1]. Due to their critical roles as actuators, PMSMs have received widespread applications in important sectors such as advanced manufacturing, energy production [2], and electric transportation [3,4]. Their reliable operation plays a vital role in guaranteeing the efficiency, sustainability, and innovation of these industries, increasing the importance of robust diagnostic and prognostic strategies.

PMSMs face a range of problems like voltage defects, stator and rotor problems, overheating due to voltage problems, and similar issues, which require advanced monitoring and control methods, especially for temperature management. Predicting the thermal behavior of a PMSM is very difficult due to its complexity, which is characterized by nonlinear relationships between mechanical and electrical parameters [5]. Moreover, temperature significantly affects the efficiency and stability of PMSMs. It mainly causes overheating, one of the most frequent and harmful issues, which occurs because of the pressure caused by the mechanical part of the engine, the core iron, and the copper in the stator windings. Copper losses directly influence heating in the stator windings, leading to the degradation

of engine combustion, i.e., the most heat-generating component in it, after which comes an overall system breakdown. Due to this, the engine requires close monitoring and control so that its operation falls within the permissible limit of temperature.

In this regard, the integration of state-of-the-art technologies into the PMSM monitoring framework, such as machine learning models [6], would be a powerful method for fault prediction and diagnosis and for taking effective mitigations to ensure the general health and uninterruptible operation of the system. Our work investigates how ML-based ANN models could enhance monitoring and prediction capabilities in PMSM temperature control. In this work, four advanced machine learning algorithms are discussed in terms of performance: Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), and Multilayer Perceptron (MLP).

The key parameters are considered comprehensively by the evaluation methodology: prediction accuracy, computational efficiency, and RMSE are taken into account to enable full understanding of the potential of each parameter in real-time monitoring applications of PMSM. This systematic analysis of the models and pointing out their strengths and limitations are contributory to valuable insights as to how these parameters can be optimized for optimal lifecycle management of PMSMs.

Ultimately, this study contributes to advancing predictive maintenance strategies and ensuring the operational reliability of PMSMs, which are indispensable components in manufacturing, energy production, and electric transportation systems.

2. Related Works

PMSM temperature prediction has been the subject of a considerable number of related studies, which have used a variety of approaches, such as machine learning and hybrid models, to address these issues. A nine-layer deep neural network (DNN) model with a coefficient of determination (R^2) of 0.9439 was developed by Zhang et al. [7] to predict the stator winding temperature. The model demonstrates the effectiveness of deep learning in temperature monitoring systems by significantly outperforming conventional regression techniques. Similarly, Cen et al. [8] demonstrated the resilience of the approach under various operating conditions and outperformed conventional methods in terms of prediction accuracy by using a long short-term memory (LSTM) network to capture time dependencies in PMSM temperature data.

Chen et al. [9] proposed a hybrid system incorporating physics-based models and CNNs. Such a combination takes advantage of both physical laws and deep learning methods to make temperature predictions in real time and with high reliability for various operating conditions. Li et al. developed a Back Propagation Neural Network for temperature predictions of the stator and rotor, achieving an excellent correlation coefficient (R^2) between the predicted and actual temperatures. This underlines the importance of temperature monitoring, which allows maximum performance for PMSM and prevents overheating. Furthermore, Li et al. [10] proposed a hybrid model that combines neural networks with LPTN to give better temperature predictions in multiple nodes, especially under conditions of limited training data. LSTMs have been used to correct uncertainties and faults in the physical model, whereas LPTNs compartmentalize complex thermal interactions into a network of nodes and thermal resistances. Liu et al. [11] have also developed a method that combines machine learning and physical modeling, using the outputs of the physical model as inputs to a neural network. This method shows the effectiveness of real-time predictions by incorporating sensor data in the physical domain through data-driven methodologies.

Hence, the combined findings of these investigations mark a substantial step forward into the realm of PMSM temperature supervision and monitoring, suggesting that ML frameworks, including those using AI reinforcement learning, hold the potential of being very strong tools for better system safety and reliability.

3. Methodologies

This section presents the methodologies employed in this research. It begins with an overview of the prediction process based on the ANN inference model, followed by a detailed explanation of the functioning of each machine learning model among the four predefined algorithms. Finally, it introduces the performance indicators used to evaluate the results of each model.

3.1. ANN-Based PMSM Temperature Prediction Process

The temperature prediction of PMSMs using ANNs requires the preparation of this dataset, which was initially stored in a CSV file. Multiple preprocessing steps were performed to provide ready high-quality data and prepare this dataset for training our ML models (Figure 1). The first step is normalization, which will scale all the input variables within a common range. This step will eliminate the existence of value differences between features. This step improves model performance by preventing large variables from dominating. Next comes the cleaning stage, which is performed to remove any inconsistencies, outliers, or irrelevant data, so that only correct and relevant information is fed into the model.

After cleaning, imputation addresses missing values by replacing them with appropriate estimates, preventing data gaps from affecting the learning process.

After the preprocessing is performed on the data, it is standardized, resulting in a dataset with similar qualities. The standardized dataset is then divided into a training and testing set, usually an 80%/20% split. The training subset is for training the ANN models, and the test set is used to evaluate the models' performance and generalization capacity.

The data, once prepared, are then fed into the predictive model inference layer of the ANN framework. We separately implement and train four ANN models (MLP, RNN, LSTM, and CNN) for the temperature prediction of PMSM. Each model produces its temperature outputs, which are later subjected to three measures: the root mean square error (RMSE), the mean absolute error (MAE), and the coefficient of determination (R^2) of all models. These metrics encapsulate the accuracy, repeatability, and confidence of each model described. Ultimately, the performance evaluation determines which model will be used in the next stage for the reliable and efficient prediction of PMSM temperatures.

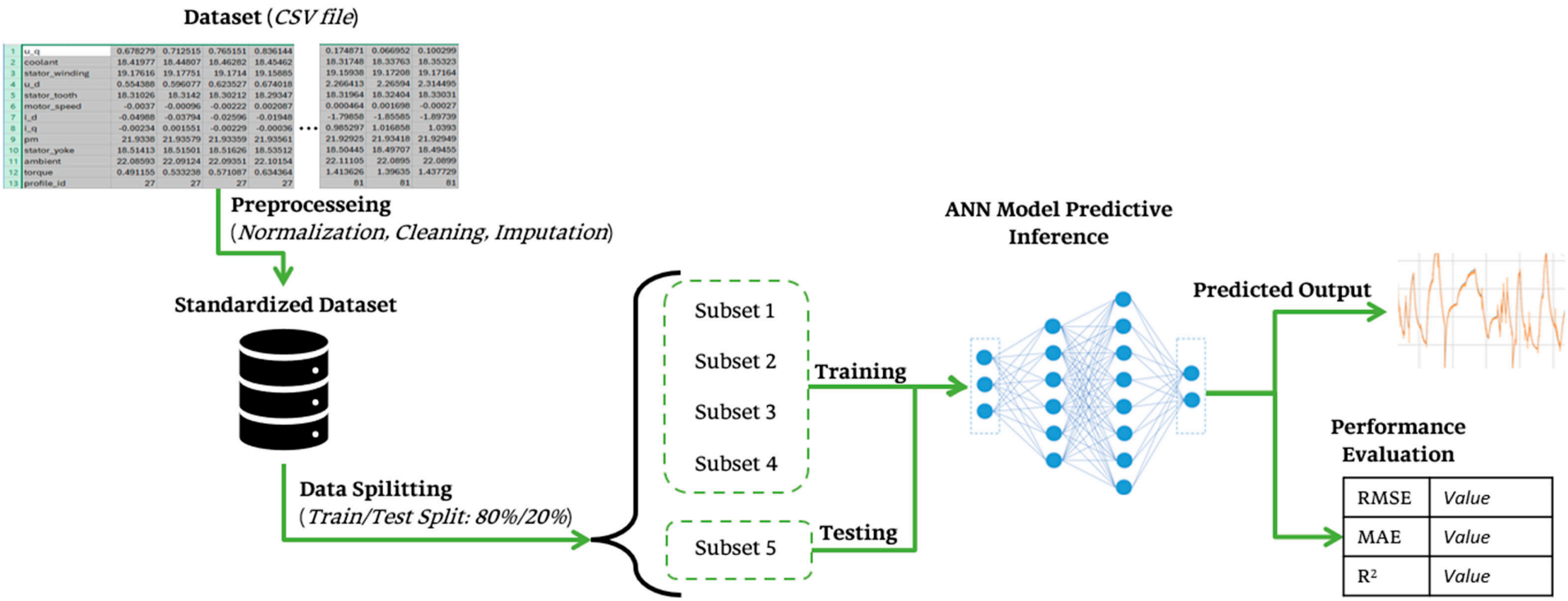


Figure 1. ANN-based PMSM temperature prediction process.

3.2. ANN-Based Machine Learning Models

3.2.1. Multilayer Perceptron (MLP)

The MLP is a versatile machine learning model suitable for a wide range of predictive tasks [12]. It receives a set of input variables that characterize the system being studied. The model's capacity to accurately identify essential associations in the data can be enhanced by thoroughly preprocessing and normalizing these input variables [13]. MLP is an inclusive family of models. The input layer is in charge of collecting the input variables passed to one or more hidden layers [14,15]. The standard architecture and design of these hidden layers are fundamental for the general performance of the model; there are many tuning parameters that may strongly influence the capability of the model to carry out tasks with high accuracy in a time-efficient way [16]. Thus, a weighted sum over the ingested input, followed by the ReLU-Rectified Linear Unit, Sigmoid, and GELU, among others, as activation functions introducing nonlinearity in the following steps, would enable the modeling of intricate patterns and relationships through back propagation, which is crucially embedded into these kinds of hidden layers of results [17]. Eventually, the hidden layer results in a combination that will present their output over an output layer. This acts in the development of the eventual prediction or classification.

A typical MLP model, as shown in Figure 2, consists of an input layer containing m input variables, one or more hidden layers with a number of nodes, and an output layer corresponding to the prediction target [18]. This flexible shape allows the MLP to model a wide range of functional forms (regression, as well as classification, and more specialized tasks) by learning the relevant input data corresponding to the output.

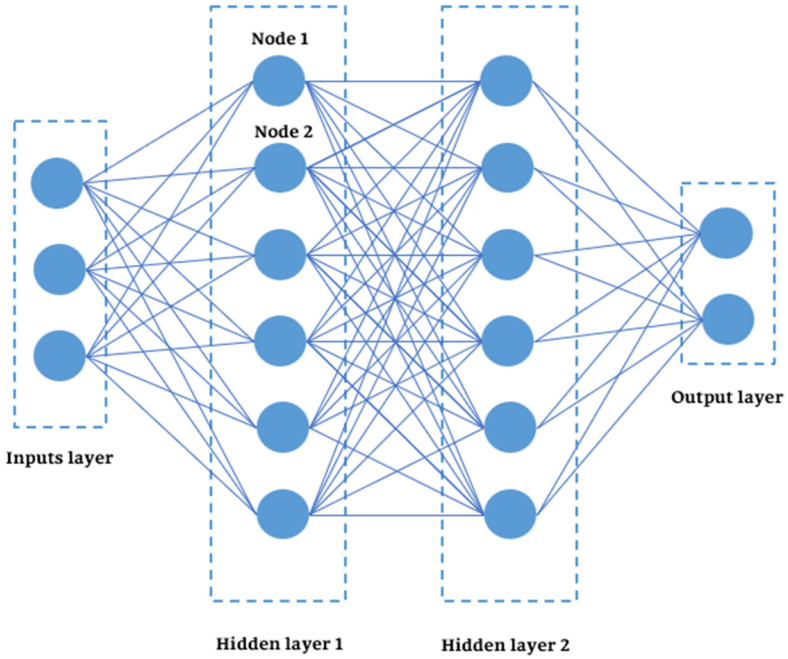


Figure 2. Multilayer perceptron diagram.

3.2.2. Recurrent Neural Network (RNN)

RNN refers to a class of artificial neural networks intended to process sequential data or temporal dependencies. This makes it an excellent fit for specific applications in which the order between the temporal relations or context counts. Unlike classical feedforward neural networks, it has recurrent links that enable an RNN to capture information and communicate it to all other time periods [19].

In an RNN, the input layer deals with sequence data and is then fed into recurrent hidden layers. Furthermore, every neuron in these hidden layers receives inputs from the

current time step as well as its previous state, thus allowing the network to have some sort of memory of previous inputs per neuron [20]. The hidden layers use either tanh or ReLU as an activation function for introducing nonlinearity so the network can learn complicated patterns from the input sequential data. Lastly, the output layer summarizes the information from these hidden layers in order to form predictions at each time step.

A standard RNN, as illustrated in Figure 3, processes a sequence of input vectors (x_1, x_2, . . ., x_n) throughout a set of iterative applications of the relevant Equations (1) and (2) [20]. It computes the hidden node states h_n and the outputs y_n.

$$h_n = f_h(x_n w_{hx} + h_{n-1} w_{hh} + b_h) \quad (1)$$

$$y_n = f_y(h_n w_{yh} + b_h) \quad (2)$$

where f_h and f_y introduce the activation functions, b_h and b_h represent bases, and w represents weights of the recurrence matrix between the hidden layers.

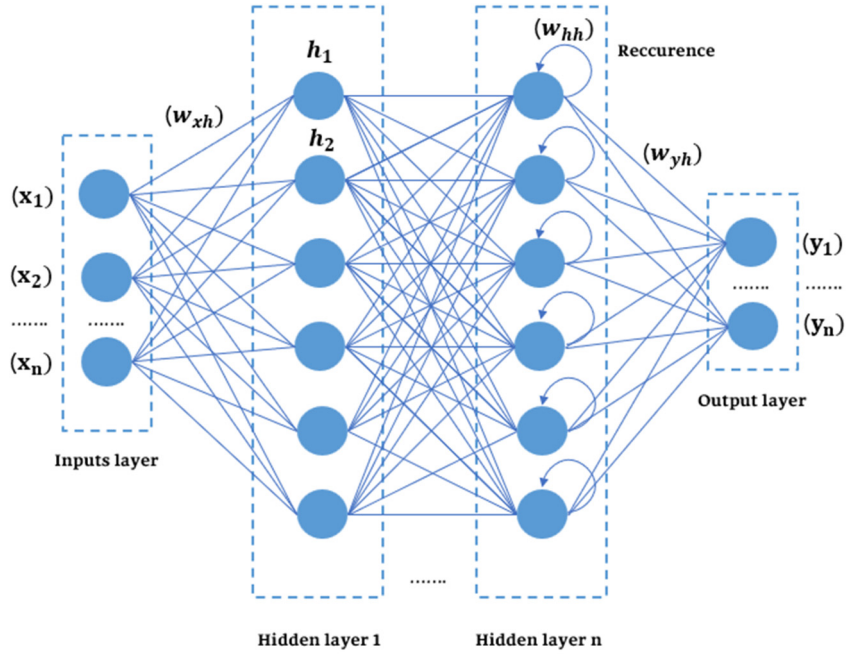


Figure 3. Recurrent neural network diagram.

3.2.3. Long Short-Term Memory (LSTM)

LSTM networks are a particular type of Recurrent Neural Network designed to handle the shortcomings of traditional RNNs in capturing long-term dependencies in sequential data. LSTM networks use an entirely new architecture (shown in Figure 4) to directly handle the vanishing gradient problem and, hence, learn long-term sequences directly [19,21].

The key concepts of the LSTM model are its memory cell and gating mechanism. There are three gates for each LSTM unit:

Forget Gate: It regulates what information from the prior time step will be removed from the cell state.

Input Gate: It controls what new information should be added to the cell state.

Output Gate: By using the sigmoid activation function, the output gate is responsible for controlling the output of the memory cell at the current time step, or in determining which information is given to the next layer or time step.

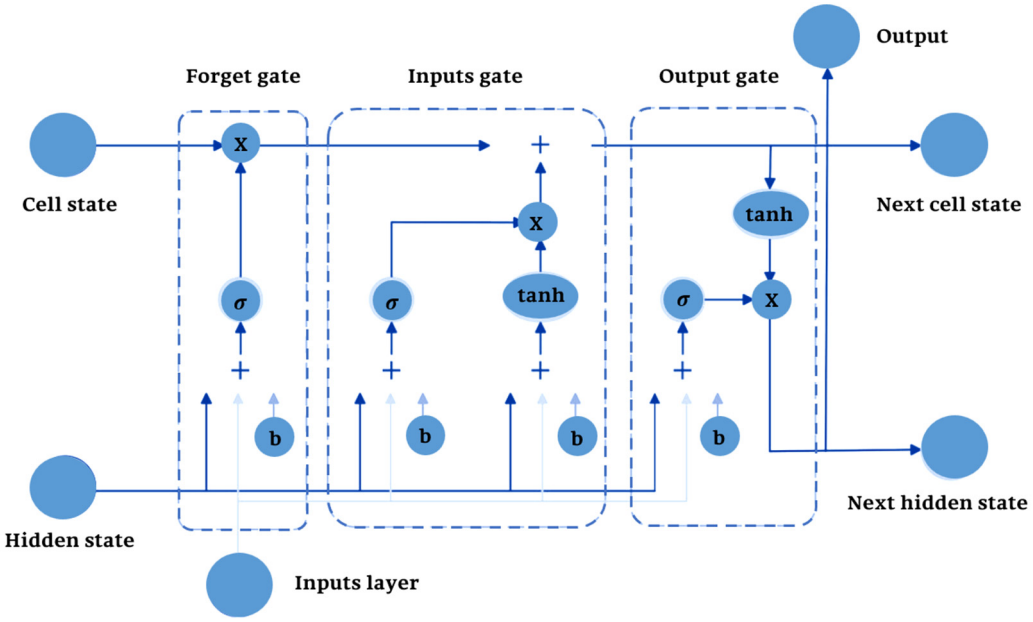


Figure 4. Long-Short-Term-Memory Diagram.

Due to the gating structure, the LSTM can effectively learn to retain or discard information, making it quite powerful for learning temporal dependencies at multiple time scales [22]. They add nonlinearity and determine how much information flows through the gates by having activation functions like the sigmoid function and the tanh function [23,24].

LSTMs are used in applications involving sequence prediction, including time series analysis, speech recognition, machine translation, and video analysis [22]. Their ability to capture temporal dependencies and adapt to varying sequence lengths makes them a robust tool for a broad range of applications involving complex temporal patterns [17].

3.2.4. Convolutional Neural Network (CNN)

CNNs are specific types of artificial neural networks that have strong capabilities to handle grid-structured data like images, time series, or structured sensor outputs [15,25].

One of the key advantages of CNNs is that, by employing convolutional layers, they learn to extract spatial and hierarchical features automatically, decreasing the dependence on handcrafted features that characterize traditional neural networks [19,25].

A CNN consists primarily of three sections (as seen in Figure 5):

Convolutional Layers: When given an input image, the CNN extracts the features from the image using several convolutional filters. These layers learn to recognize local patterns such as edges, textures, and shapes, which are essential for identifying defects or anomalies in manufacturing scenarios.

Pooling Layers: One of the beginning stages in the neural network, pooling layers, downsample the feature maps, reducing dimensionality by preserving most necessary features, which saves the computational load of the network.

Fully Connected Layers: The extracted features are concatenated to perform the final prediction, such as defect classification or machine failure prediction.

This type of CNN has broad applications for complex datasets when making higher-order decisions and optimizing processes [26,27]. The scalability and adaptability of a CNN to a wide range of manufacturing big datasets, multispectral images, vibration signals, or thermal maps are indispensable in industry applications [27].

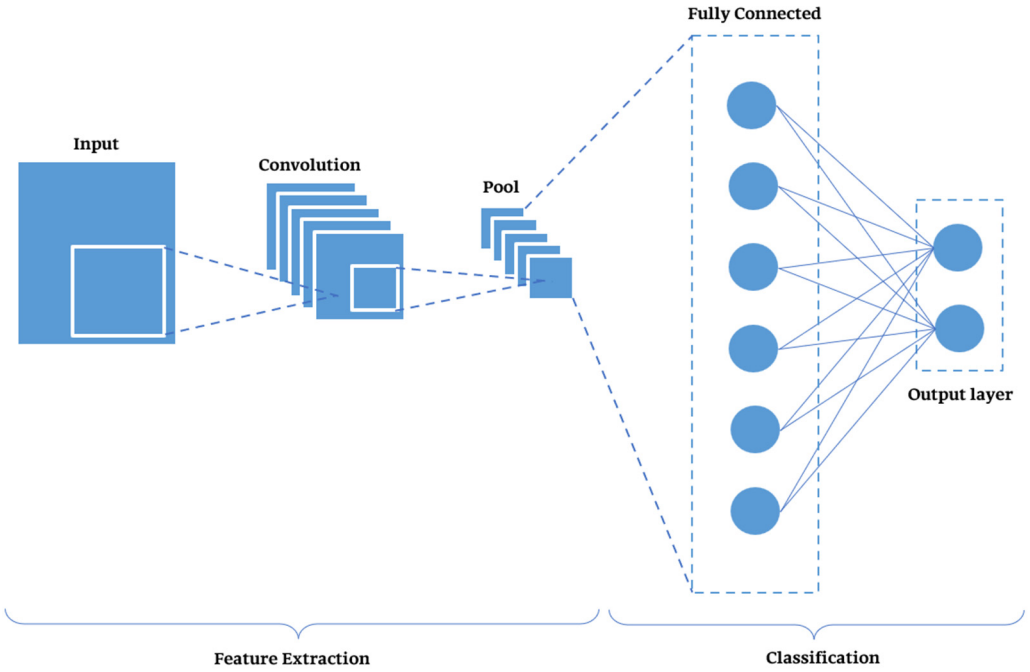


Figure 5. Convolutional neural network diagram.

3.3. Performance Indicators

Three key metrics have been used in this study to provide a comprehensive picture of the effectiveness and consistency of each of the models. The performance of the trained ML models is based on the analysis of the variance between the predicted and the true data for a variety of measures. The evaluation of the proposed models is carried out using the following indicators: root mean square error (RMSE), mean absolute error (MAE), and R-Squared (R^2).

3.3.1. RMSE

The root mean square error (RMSE) is a widespread indicator of measuring how much the predicted values from a model differ from the actual observed values. It is important to note that the RMSE depends on the scale of the data, so it is more useful for evaluating prediction errors within a particular dataset than for comparing different datasets [28]. The equation for the RMSE is given in (3):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y - y')2} \quad (3)$$

3.3.2. MAE

The mean absolute error (MAE) is a vital metric for evaluating discrepancies between paired observations related to the same event. The MAE is calculated using the following formula, as outlined in Equation (4):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y - y\prime| \quad (4)$$

The mean absolute error is calculated on the same scale as the data. However, it should be noted that, as this accuracy metric is dependent on the scale in question, it is not suitable for comparing series with different scales. In the field of time series analysis, the mean absolute error is a common metric used to assess forecast accuracy.

3.3.3. R^2

R^2 is the goodness-of-fit metric, which measures the model's ability to accurately predict data outcomes. It has a range between 0 and 1. Higher values indicate a stronger fit, suggesting that the model performs better [29]. The following Equation (5) calculates this indicator:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y - y')^2}{\sum_{i=1}^{n} (y - \overline{y})^2} \quad (5)$$

4. Experimentation and Results

This section presents the experimentation and results of the study. It begins with an overview of the dataset, detailing the dataset-associated features, followed by an analysis of the performance of each model. Finally, a comprehensive comparative evaluation is conducted to assess the overall efficacy of the models.

4.1. Dataset Overview

The dataset used in this study is sensor data over a prototype PMSM developed by a German original equipment manufacturer (OEM). The data were collected from a testing experiment performed in the LEA department at Paderborn University [30]. These recordings measured the motor performance for different operating conditions with a sampling frequency of 2 Hz. The measurements in sessions are between one and six hours long and can be identified by the column "*profile_id*".

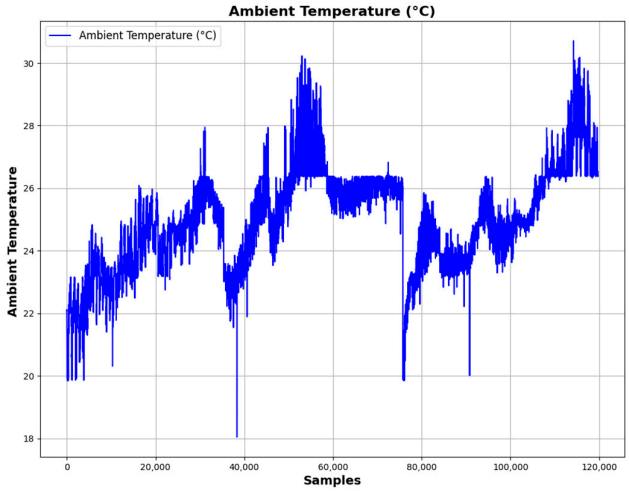
In the process, the motor was driven according to specifically designed cycles of reference values of motor speed and torque. Such driving cycles attempt to closely approximate those occurring in an on-road environment rather than simple ramp-up or steady-state excitations. The currents and voltages in the d/q-coordinates ("*i_d*", "*i_q*", "*u_d*", and "*u_q*") were varied in order to change the motor's speed and torque as an integral part of the control approach. This immediately reflects in the resulting outputs of "*motor_speed*" and "*torque*".

Table 1 depicts the key characteristics.

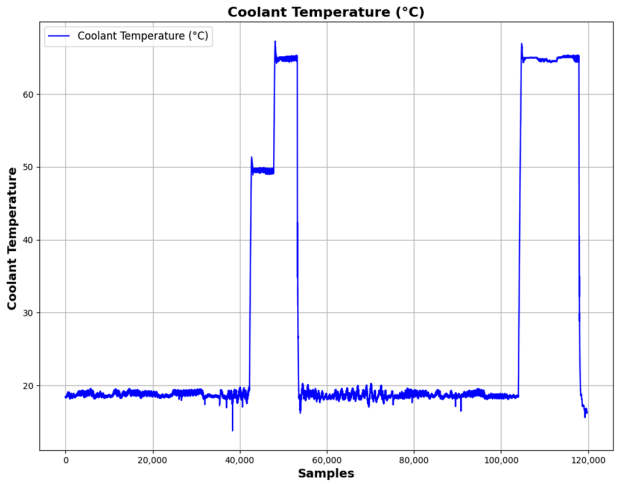
Table 1. Key features of dataset.

Feature	Type	Designation
ambient	Environmental	The ambient temperature measured near the stator
coolant	Environmental	The coolant temperature measured at the water outflow
u_d	Control	The voltage of the d-component
u_q	Control	The voltage of the q-component
motor_speed	Performance	The motor's rotational speed
torque	Performance	The torque induced by the currents
i_d	Control	The current of the d-component
i_q	Control	The current of the q-component
pm	Performance	The surface temperature of the rotor's permanent magnet
stator_yoke	Performance	The temperature of the stator yoke measured with a sensor
stator_tooth	Performance	The temperature of the stator tooth measured with a sensor
stator_winding	Performance	The temperature of the stator winding measured with a sensor

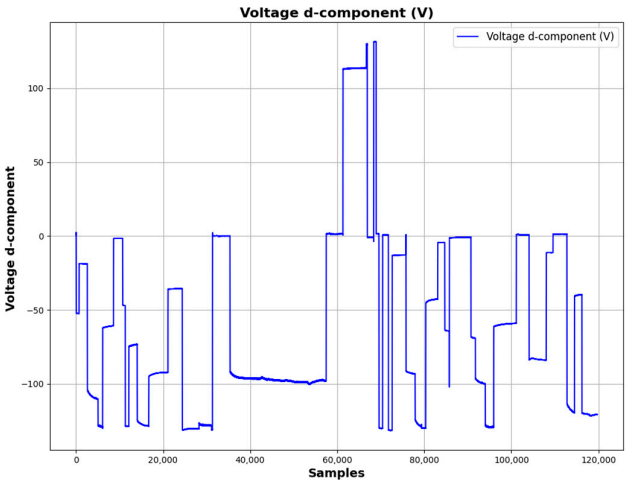
Figure 6 illustrates a random sample of 120,000 entries from this dataset.



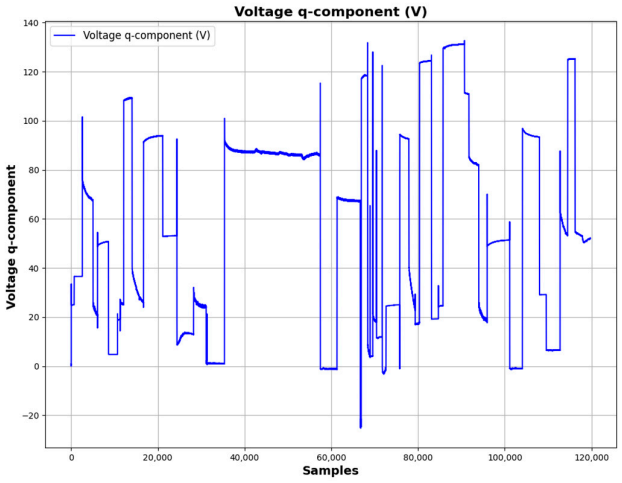
(**a**)



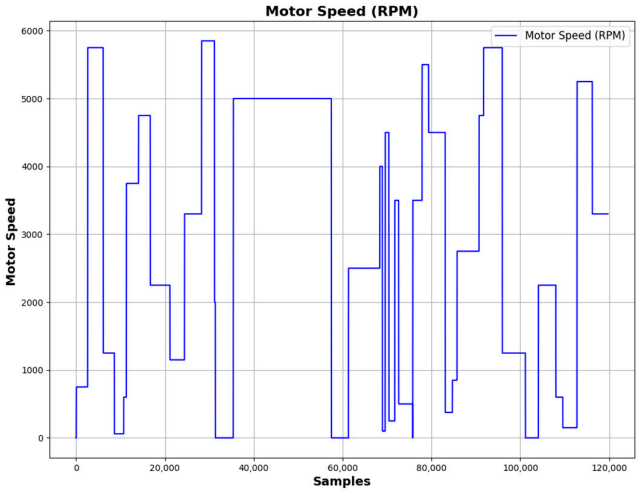
(**b**)



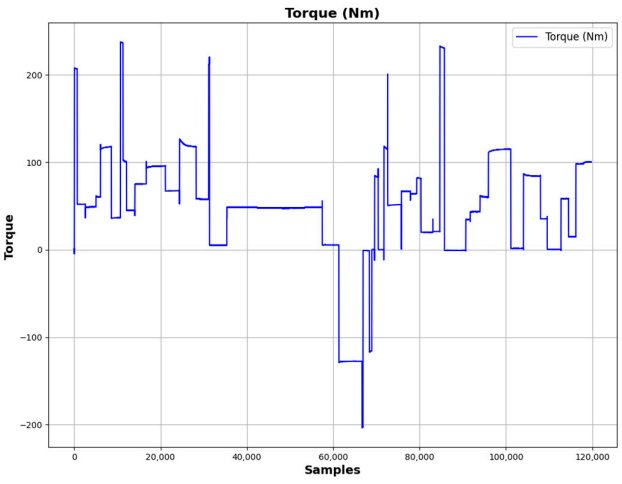
(**c**)



(**d**)



(**e**)



(**f**)

Figure 6. *Cont.*

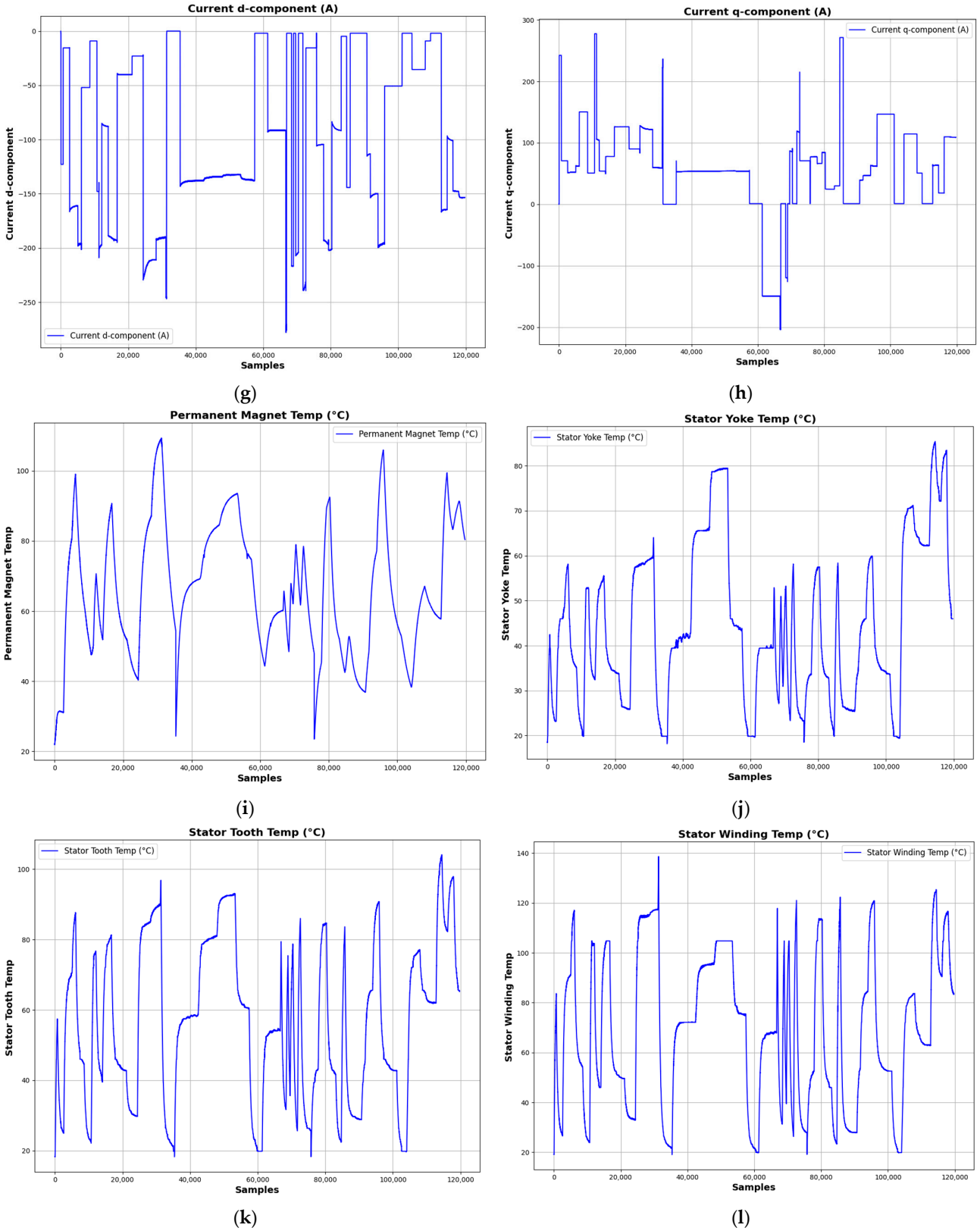


Figure 6. PMSM dataset overview. (**a**) Ambient temperature. (**b**) Coolant temperature. (**c**) Voltage d-component. (**d**) Voltage q-component. (**e**) Motor speed. (**f**) Torque. (**g**) Current d-component. (**h**) Current q-component. (**i**) Permanent magnet temp. (**j**) Stator yoke temp. (**k**) Stator tooth temp. (**l**) Stator winding temp.

4.2. Performance Analysis of Models

The specifications of the computer used in the experiment are listed in Table 2. Moreover, the programming language used was Python 3.8.3.

Table 2. Computer specifications.

Device Manufacturer	Lenovo
Processor	12th Gen Intel(R) Core (TM) i7-12800HX 2.00 GHz
Installed RAM	32.0 GB (31.7 GB usable)
Product ID	00355-61104-67914-AAOEM

The optimal model performance could be achieved only by a well-implemented, systematic grid search through all the combinations of the predefined hyperparameters, such as the number of layers, neurons in each layer, batch size, learning rate, and type of activation functions. In such a way, it was feasible to check every combination and make a comparison for the selection of the best configuration for each model. The struggles were in the training loss convergence, divergences, instability, overfitting, and the vanishing gradient problem during the training phase, mainly concerning the RNN and LSTM models. Advanced optimizers were employed to make sure that vanishing gradient issues were avoided, while techniques for regularization, means of dropout layers, and cross-validation prevented overfitting. For instability, the tuning of inappropriate learning rates was necessary. Minimizing the loss fluctuation could be achieved by normalizing data and optimizing the batch size. The whole process allowed for the enhanced model to be robust, accurate, and with improved generalization capability.

4.2.1. Results with MLP

An MLP model was used as a baseline for the prediction task (Table 3). The results of hyperparameter tuning showed that a 128-node, single-layer MLP with the learning rate set to 0.001 gives the best performance. Minimizing divergences with the Adam optimizer allowed the model to achieve a good trade-off of complexity versus generalization.

Table 3. MLP top combination prediction performance.

Combination	Hidden Layers	Optimizer	Learning Rate	MAE	RMSE	R^2
#16	**(128, 64)**	**adam**	**0.0010**	**0.889309**	**1.284423**	**0.995044**
#12	(64, 32)	adam	0.0010	1.082264	1.448051	0.993731
#17	(128, 64)	adam	0.0005	1.277344	1.591908	0.990635

4.2.2. Results with RNN

The RNN model had the best performance with 64 units by using the Adam optimizer and learning rate of 0.001 for a mean absolute error (MAE) of 1.58, a root mean square error (RMSE) of 2.27, and a coefficient of determination (R^2) of 0.98. Fewer unit configurations (like 32) or a lower learning rate of 0.0005 resulted in non-optimal performance. On the other hand, slow convergence and stability issues contributed toward the 1000-fold regression of stochastic gradient descent (SGD) method. The results showen in Table 4 depicts the effectiveness of Adam optimizer, highlighting the importance of fine-tuning hyperparameters to achieve the best performance of the model.

Table 4. RNN top combinations prediction performance.

Combination	Units	Optimizer	Learning Rate	MAE	RMSE	R^2
#4	**64**	**adam**	**0.0010**	**1.583786**	**2.268776**	**0.984503**
#5	64	adam	0.0005	1.932705	2.890273	0.974867
#0	32	adam	0.0010	2.403096	3.447521	0.964240

4.2.3. Results with LSTM

The LSTM model achieved its best performance with 128 units, a learning rate of 0.001, and a batch size of 32, resulting in the lowest RMSE (1.72) and highest R^2 (0.99). The configurations with the learning rates of 0.001 (128 and 64 units) performed comparably, with slightly higher RMSE and MAE values. These results as presented in Table 5, emphasize the LSTM model's strong predictive capability, particularly when tuned with an optimal balance of units and learning rates.

Table 5. LSTM top combination prediction performance.

Combination	Units	Batch Size	Learning Rate	MAE	RMSE	R^2
#8	**128**	**32**	**0.0010**	**1.173979**	**1.720373**	**0.991081**
#9	128	64	0.0010	1.491014	2.087330	0.986936
#4	64	32	0.0010	1.599378	2.161259	0.985922

4.2.4. Results with CNN

The CNN model performed best with 64 filters, a learning rate of 0.001, and a batch size of 64, achieving the lowest RMSE of 2.30 and the highest R^2 value of 0.98. A small reduction in the learning rate to 0.0005, while keeping the number of filters unchanged, resulted in a slight drop in performance. On the other hand, reducing the number of filters to 32 resulted in an increase in RMSE to 2.78 and a decrease in R^2 to 0.97, highlighting the importance of using an adequate number of filters and a well-tuned learning rate for optimal results.

Table 6 illustrates more insights about the CNN model performance.

Table 6. CNN top combinations prediction performance.

Combination	Filters	Batch Size	Learning Rate	MAE	RMSE	R^2
#6	**64**	**64**	**0.0010**	**1.732543**	**2.304708**	**0.984090**
#8	64	64	0.0005	1.887692	2.633485	0.979069
#3	32	64	0.0010	1.994809	2.787847	0.976564

4.2.5. Execution Time

After applying the grid search methodology to each model, the optimal hyperparameter combination was selected based on performance metrics. However, the execution time remains a cornerstone factor in evaluating the efficiency of our models. Table 7 presents the execution time corresponding to each model, providing a comparative assessment of computational efficiency.

Table 7. Execution time of ANN-based models.

Model	MLP	RNN	LSTM	CNN
Execution Time (s)	**248.46**	264.69	436.51	259.43

The performance of the models reveals significant differences in computational cost due to the different architectures. The MLP demonstrates the fastest execution time at 248.46 s, making it the most computationally efficient model in the study. Secondly, we find that the RNN performs slightly higher than the MLP, with an execution time of 264.69 s. The CNN records an execution time of 259.43 s, indicating that its feature extraction process does not significantly increase the computational cost. Conversely, the LSTM (Long Short-Term Memory) has the highest execution time at 436.51 s, reflecting its increased complexity due to memory cell operations and gating mechanisms designed for long-term dependencies.

The subsequent figure (Figure 7) illustrates the comparison between the actual and predicted data across a dataset of 120,000 samples, emphasizing the models' proficiency in approximating the target output: Permanent Magnet Temperature ("*pm*"). This graphical representation offers a comprehensive evaluation of prediction accuracy and demonstrates the degree to which each model aligns with the anticipated values.

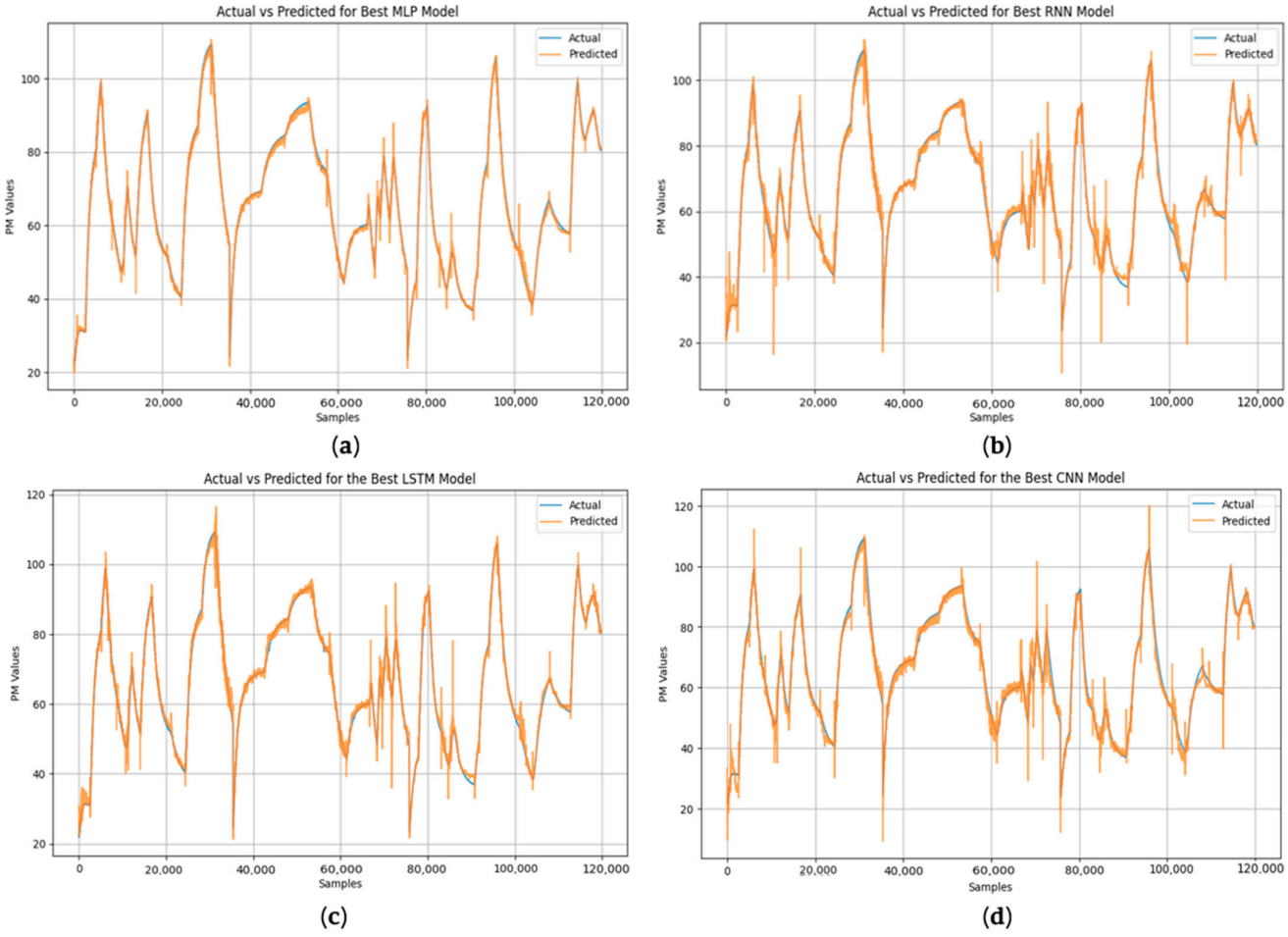


Figure 7. Actual vs. predicted plots. (**a**) Best combination of MLP model. (**b**) Best combination of RNN model. (**c**) Best combination of LSTM model. (**d**) Best combination of CNN model.

4.3. Comparative Evaluation

The assessment of the performance efficacy of our ANN-based models, namely MLP, RNN, LSTM, and CNN, employs three commonly recognized regression metrics (Section 3.3). The study utilizes optimized hyperparameter settings for each model to facilitate an equitable comparison of their predictive performance.

Of these, the MLP model gave the best performance for all metrics: lowest error rates, with RMSE = 1.284 and MAE = 0.889, and highest explanatory power, with R^2 = 0.995. These results support the fact that the feedforward architecture of the MLP, together with its

optimized configuration, effectively modeled the underlying patterns of the data, and the model predictions showed minimal deviation from the observed ones. Trained on temporal dependencies, the LSTM model ranked second, with a moderate error level (RMSE = 1.720, MAE = 1.174) but with a strong R^2 of 0.991. Although this model had slightly worse performance compared to MLP, its inherent ability to process sequential data allows it to stand out for the time-series or context-dependent tasks.

However, both the RNN and CNN architectures were relatively weaker. The RNN resulted in higher errors—a respective RMSE of 2.269 and MAE of 1.584 and a lower R^2 score of 0.984, probably due to challenges in modeling long-term dependencies, which is one of the well-known limitations of vanilla RNNs. Similarly, though efficient in handling spatial data, the CNN underperformed in this context: the RMSE was 2.305, while the MAE was 1.733, with an R^2 value of 0.984. This might reflect a mismatch between the grid-based feature extraction mechanism of CNNs and the structure of the data, which lacks inherent spatial hierarchies.

Figure 8 presents the relative performance of the four models, arranged from left to right on the X-axis, against the performance metrics, namely RMSE, MAE, and R^2, on the Y-axis.

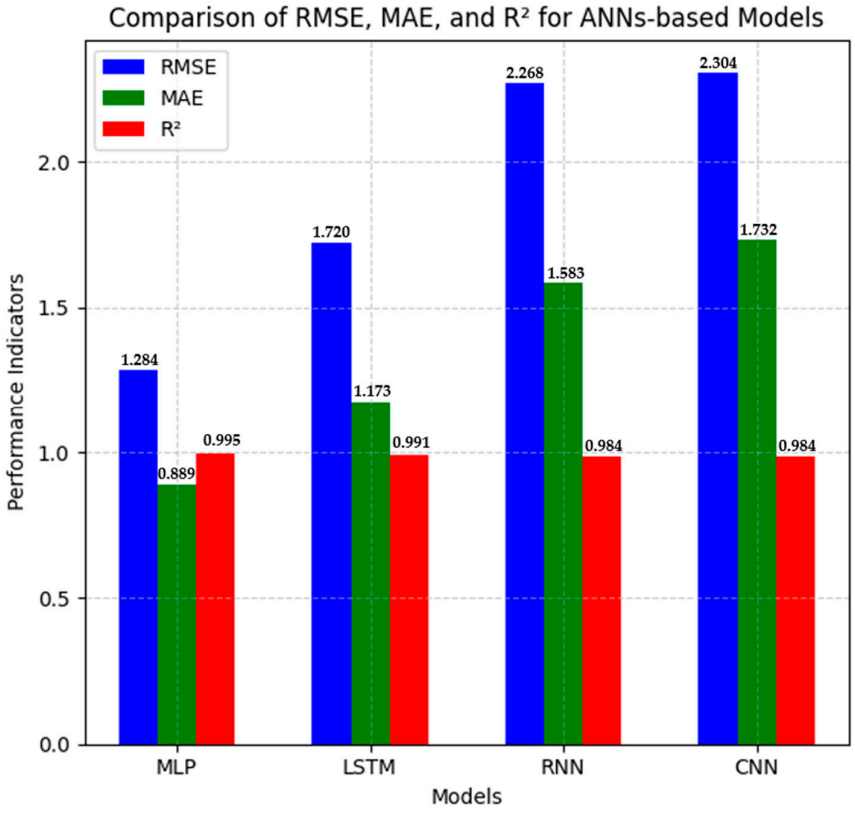


Figure 8. ANN-based models' performance comparison.

5. Discussion

This work investigates the predictive performance of four categories of ANN architectures, namely MLP, RNN, LSTM, and CNN. With the basic statistical measures of RMSE, MAE, and the R^2 coefficient of determination, an analysis is conducted to depict accuracy and computational efficiency and the practical relevance of the model.

From this comparison, the MLP model had the lowest RMSE and MAE results, suggesting its better capability to decrease error and improve the predictions. On the other hand, the CNN and RNN models had larger RMSE values, indicating relatively less accuracy in predicting temperature. However, their R^2 values were similar, indicating that these

models can still explain important trends in the data, even if they have larger error margins. Compared to other works published, the findings support well-documented research that has highlighted the high performance of MLP architecture when dealing with time-series forecasting problems for datasets of moderate complexities.

These results are in line with the literature; MLP architectures outperform other models tested on time-series prediction tasks with moderate complexity. In contrast, the results show that LSTM and RNN models are a preferred choice for sequential data modeling, but their performance relies upon hyperparameter tuning, dataset properties, and feature selection strategies. Commonly used in computer vision problems, the CNN model also exhibited relatively moderate prediction strength in this investigation, but its statistically higher RMSE reflects the need for a more sophisticated model in order to excel in predicting time-series outputs.

The MLP model not only obtained the best accuracy but also had the shortest computation time, making it a viable choice for use in real-world predictive maintenance frameworks.

On the other hand, training and inference by the LSTM and RNN models took an extensive amount of time, probably because of their sequentially dependent nature and dependence on past states, which increases their complexity. CNN gave moderately effective performance in regard to prediction but was very computationally expensive since convolutional operations are usually optimized for spatial data, not time series prediction. The results indicate a trade-off between model complexity and computational efficiency, which is an important factor for real-time industrial applications.

This study demonstrates the dominance of the MLP algorithm, presenting a consistent model to use when it comes to simpler scenarios in which temporal or spatial complexity is limited, with the superior abilities of both predictive accuracy and computational efficiency compared with RNNs and LSTMs.

6. Conclusions and Outlook

The promising performance of the MLP model in predicting Permanent Magnet Temperature opens new avenues toward the improvement of predictive maintenance strategies within digital twin frameworks for PMSMs. These low values of the metrics for the model, combined with its computational efficiency, make the model a strong candidate for real-time integration within industrial digital twins. If this model is embedded in the architecture of the digital twin, eventually, the operators will have the potential capability of continuous high-fidelity monitoring of PMSM thermal behavior, the early detection of anomalies such as overheating and risk of demagnetization, and proactive maintenance intervention. Coupling MLP-based temperature prediction with real sensor data streams at runtime, including current, voltage, and rotor speed, constitutes a very crucial step within a digital twin. In this regard, dynamic calibration under different operation conditions would, therefore, be possible, wherein the model attunes itself in case of big fluctuations in the load or sudden environmental stressors. Coupling temperature predictions with physics-based models of motor degradation, such as insulation wear and magnet aging, could further refine failure prognostics, thereby transforming the digital twin into a decision-support tool for maintenance scheduling or operational parameter optimization.

Nevertheless, scaling up the latter for industrial deployment still comes with challenges: first, seamless data synchronization across physical motors with their digital counterpart(s) will require strong IoT infrastructure with minimal latency; second, the validation of the model's generalizability to larger variations in both design and operating contexts to prevent the model from overfitting to a particular dataset will be necessary; third, building in explainability features—for instance, attention mechanisms or uncertainty

quantification into an MLP architecture—is indispensable for creating trust among the operators, more so in safety-critical applications.

Future research needs to be conducted in these hybrid architectures that will marry the efficiency of the MLP with domain-specific enhancements. For instance, federated learning can enable collaborative model training across distributed PMSM fleets to achieve better generalization without compromising data privacy, while edge computing frameworks can decentralize the inference tasks to reduce reliance on cloud infrastructure and improve real-time responsiveness.

From an industrial point of view, this meets the goals of Industry 4.0, such as predictive analytics and autonomous systems. Embedding MLP-driven insights into the digital twin will enable industries to shift away from reactive maintenance strategies toward condition-based ones, reduce unplanned downtime, and extend motor lifespan. Furthermore, the MLP model fits well in these resource-constrained environments, such as offshore wind turbines or electric vehicles, thanks to the smaller computational overhead; real-time processing becomes important here.

Accordingly, the integration of MLP-based temperature forecasting into the PMSM digital twin brings a tremendous leap toward intelligent condition-based predictive maintenance.

Author Contributions: Conceptualization, N.E.B.; methodology, N.E.B., N.G. and M.M. (Mustapha Mabrouki); software, N.E.B.; validation, M.M. (Mustapha Mabrouki) and A.C.; formal analysis, N.E.B., A.G. and M.M. (Mohcin Mekhfioui); investigation, N.E.B. and M.M. (Mohcin Mekhfioui); resources, N.E.B. and A.G.; data curation, N.E.B., N.G. and A.G.; writing—original draft preparation, N.E.B.; writing—review and editing, N.G. and M.M. (Mohcin Mekhfioui); visualization, N.E.B.; supervision, A.C.; project administration, M.M. (Mustapha Mabrouki); funding acquisition, A.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Green Tech Institute of UM6P.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. El Bazi, N.; Mabrouki, M.; Laayati, O.; Ouhabi, N.; El Hadraoui, H.; Hammouch, F.-E.; Chebak, A. Generic Multi-Layered Digital-Twin-Framework-Enabled Asset Lifecycle Management for the Sustainable Mining Industry. *Sustainability* **2023**, *15*, 3470. [CrossRef]
2. Laayati, O.; El Hadraoui, H.; El Magharaoui, A.; El-Bazi, N.; Bouzi, M.; Chebak, A.; Guerrero, J.M. An AI-Layered with Multi-Agent Systems Architecture for Prognostics Health Management of Smart Transformers: A Novel Approach for Smart Grid-Ready Energy Management Systems. *Energies* **2022**, *15*, 7217. [CrossRef]
3. El Hadraoui, H.; Zegrari, M.; Chebak, A.; Laayati, O.; Guennouni, N. A Multi-Criteria Analysis and Trends of Electric Motors for Electric Vehicles. *World Electr. Veh. J.* **2022**, *13*, 65. [CrossRef]
4. Guennouni, N.; Machkour, N.; Chebak, A. Model Predictive and Sliding Mode Control Hybridization for Voltage and Average Current Control of Dual Active Bridge DC-DC Converter in Battery Electric Vehicles Powertrain. In Proceedings of the 2024 6th Global Power, Energy and Communication Conference (GPECOM), Budapest, Hungary, 4–7 June 2024; pp. 92–97.
5. Jin, L.; Mao, Y.; Wang, X.; Lu, L.; Wang, Z. A Model-Based and Data-Driven Integrated Temperature Estimation Method for PMSM. *IEEE Trans. Power Electron.* **2024**, *39*, 8553–8561. [CrossRef]
6. Ledmaoui, Y.; El Fahli, A.; El Maghraoui, A.; Hamdouchi, A.; El Aroussi, M.; Saadane, R.; Chebak, A. Enhancing Solar Power Efficiency: Smart Metering and ANN-Based Production Forecasting. *Computers* **2024**, *13*, 235. [CrossRef]
7. Guo, H.; Ding, Q.; Song, Y.; Tang, H.; Wang, L.; Zhao, J. Predicting Temperature of Permanent Magnet Synchronous Motor Based on Deep Neural Network. *Energies* **2020**, *13*, 4782. [CrossRef]
8. Cen, Y.; Zhang, C.; Cen, G.; Zhang, Y.; Zhao, C. The Temperature Prediction of Permanent Magnet Synchronous Machines Based on Proximal Policy Optimization. *Information* **2020**, *11*, 495. [CrossRef]

9. Wang, P.; Wang, X.; Wang, Y. End-to-End Differentiable Physics Temperature Estimation for Permanent Magnet Synchronous Motor. *World Electr. Veh. J.* **2024**, *15*, 174. [CrossRef]
10. Sheng, K.; Li, Y.; Xu, X.; Ling, H. Stator and Rotor Temperature Prediction of Permanent Magnet Synchronous Motor Based on BP Neural Network. In Proceedings of the 2024 43rd Chinese Control Conference (CCC), Kunming, China, 28–31 July 2024; pp. 8832–8837.
11. Liu, Z.; Kong, W.; Fan, X.; Li, Z.; Peng, K.; Qu, R. Hybrid Thermal Modeling With LPTN-Informed Neural Network for Multinode Temperature Estimation in PMSM. *IEEE Trans. Power Electron.* **2024**, *39*, 10897–10909. [CrossRef]
12. Rajamanickam, R.; Baskaran, D. Chapter 18—Neural Network Model for Biological Waste Management Systems. In *Current Trends and Advances in Computer-Aided Intelligent Environmental Data Engineering*; Marques, G., Ighalo, J.O., Eds.; Intelligent Data-Centric Systems; Academic Press: Cambridge, MA, USA, 2022; pp. 393–415, ISBN 978-0-323-85597-6.
13. Mahrouch, A.; Ouassaid, M. Primary Frequency Regulation Based on Deloaded Control, ANN, and 3D-Fuzzy Logic Controller for Hybrid Autonomous Microgrid. *Technol. Econ. Smart Grids Sustain. Energy* **2022**, *7*, 1. [CrossRef]
14. Uzair, M.; Jamil, N. Effects of Hidden Layers on the Efficiency of Neural Networks. In Proceedings of the 2020 IEEE 23rd International Multitopic Conference (INMIC), Bahawalpur, Pakistan, 5–7 November 2020; pp. 1–6.
15. Cannizzaro, D.; Aliberti, A.; Bottaccioli, L.; Macii, E.; Acquaviva, A.; Patti, E. Solar Radiation Forecasting Based on Convolutional Neural Network and Ensemble Learning. *Expert Syst. Appl.* **2021**, *181*, 115167. [CrossRef]
16. Breen, K.H.; James, S.C.; White, J.D.; Allen, P.M.; Arnold, J.G. A Hybrid Artificial Neural Network to Estimate Soil Moisture Using SWAT+ and SMAP Data. *Mach. Learn. Knowl. Extr.* **2020**, *2*, 283–306. [CrossRef]
17. Ali, A.R.; Kamal, H. Time-to-Fault Prediction Framework for Automated Manufacturing in Humanoid Robotics Using Deep Learning. *Technologies* **2025**, *13*, 42. [CrossRef]
18. Sousa, A.L.; Ribeiro, T.P.; Relvas, S.; Barbosa-Póvoa, A. Using Machine Learning for Enhancing the Understanding of Bullwhip Effect in the Oil and Gas Industry. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 994–1012. [CrossRef]
19. Kumari, P.; Toshniwal, D. Deep Learning Models for Solar Irradiance Forecasting: A Comprehensive Review. *J. Clean. Prod.* **2021**, *318*, 128566. [CrossRef]
20. Graves, A. Generating Sequences with Recurrent Neural Networks. *arXiv* **2013**, arXiv:1308.0850.
21. Fkirin, M.A.; Gowaly, Z.M.; Elsheikh, E.A. Dynamic Controller Design for Maximum Power Point Tracking Control for Solar Energy Systems. *Technologies* **2025**, *13*, 71. [CrossRef]
22. Lim, S.-C.; Huh, J.-H.; Hong, S.-H.; Park, C.-Y.; Kim, J.-C. Solar Power Forecasting Using CNN-LSTM Hybrid Model. *Energies* **2022**, *15*, 8233. [CrossRef]
23. Kandi, K.; García-Dopico, A. Enhancing Performance of Credit Card Model by Utilizing LSTM Networks and XGBoost Algorithms. *Mach. Learn. Knowl. Extr.* **2025**, *7*, 20. [CrossRef]
24. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
25. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a Convolutional Neural Network. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–6.
26. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional Neural Networks: An Overview and Application in Radiology. *Insights Imaging* **2018**, *9*, 611–629. [CrossRef] [PubMed]
27. Coşkun, M.; Uçar, A.; Yildirim, Ö.; Demir, Y. Face Recognition Based on Convolutional Neural Network. In Proceedings of the 2017 International Conference on Modern Electrical and Energy Systems (MEES), Kremenchuk, Ukraine, 15–17 November 2017; pp. 376–379.
28. El Bazi, N.; Laayati, O.; Darkaoui, N.; El Maghraoui, A.; Guennouni, N.; Chebak, A.; Mabrouki, M. Scalable Compositional Digital Twin-Based Monitoring System for Production Management: Design and Development in an Experimental Open-Pit Mine. *Designs* **2024**, *8*, 40. [CrossRef]
29. Ghritlahre, H.K.; Prasad, R.K. Application of ANN Technique to Predict the Performance of Solar Collector Systems—A Review. *Renew. Sustain. Energy Rev.* **2018**, *84*, 75–88. [CrossRef]
30. Electric Motor Temperature. Available online: https://www.kaggle.com/datasets/wkirgsn/electric-motor-temperature (accessed on 7 February 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.