

Article

# Estimation of Excitation Current of a Synchronous Machine Using Machine Learning Methods

Matko Glučina <sup>1</sup>, Nikola Anđelić <sup>2</sup>, Ivan Lorencin <sup>2,\*</sup> and Zlatan Car <sup>2</sup><sup>1</sup> University of Rijeka, Trg Braće Mažuranića 10, 51000 Rijeka, Croatia<sup>2</sup> Faculty of Engineering, University of Rijeka, Vukovarska 58, 51000 Rijeka, Croatia

\* Correspondence: ilorencin@riteh.hr; Tel.: +385-51-505715

**Abstract:** A synchronous machine is an electro-mechanical converter consisting of a stator and a rotor. The stator is the stationary part of a synchronous machine that is made of phase-shifted armature windings in which voltage is generated and the rotor is the rotating part made using permanent magnets or electromagnets. The excitation current is a significant parameter of the synchronous machine, and it is of immense importance to continuously monitor possible value changes to ensure the smooth and high-quality operation of the synchronous machine itself. The purpose of this paper is to estimate the excitation current on a publicly available dataset, using the following input parameters:  $I_y$ : load current; PF: power factor;  $e$ : power factor error; and  $d_f$ : changing of excitation current of synchronous machine, using artificial intelligence algorithms. The algorithms used in this research were: k-nearest neighbors, linear, random forest, ridge, stochastic gradient descent, support vector regressor, multi-layer perceptron, and extreme gradient boost regressor, where the worst result was elasticnet, with  $R^2 = -0.0001$ ,  $MSE = 0.0297$ , and  $MAPE = 0.1442$ ; the best results were provided by extreme boosting regressor, with  $\overline{R^2} = 0.9963$ ,  $\overline{MSE} = 0.0001$ , and  $\overline{MAPE} = 0.0057$ , respectively.

**Keywords:** artificial intelligence algorithms; excitation current; regression algorithms; synchronous machine



**Citation:** Glučina, M.; Anđelić, N.; Lorencin, I.; Car, Z. Estimation of Excitation Current of a Synchronous Machine Using Machine Learning Methods. *Computers* **2023**, *12*, 1. <https://doi.org/10.3390/computers12010001>

Academic Editor: Paolo Bellavista

Received: 15 November 2022

Revised: 1 December 2022

Accepted: 6 December 2022

Published: 20 December 2022



**Copyright:** © 2022 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

Electrical machines are electromechanical devices that can convert electrical energy into mechanical energy and vice versa [1]. According to the type of input, EMs can be classified into single-phase and three-phase machines [2], where the most common types of three-phase electric machines are synchronous (SM) and asynchronous machines (AM). Electrical machines (EM) are built from a stator, which is the static part of the EM consisting of phase-shifted coiled poles, and a rotor, which is the rotary part constructed depending on the tasks for which the machine is intended. AMs are most often used as motor-driven machines [3–5] in the automotive industry [6,7], construction [8], elevators [9], etc., while SMs are most often used to produce electricity in fossil fuel power plants [10–12] or in renewable energy plants, such as hydroelectric power [13,14] and wind power plants [15,16]. For energy to be transmitted from the producer to the final consumer, it must be of an alternating current (AC) [17]. The reasons for choosing AC over direct current (DC) are as follows [17,18]:

- Easy maintenance and change of AC voltage for transmission and distribution;
- AC transmission plant costs (switches, transformers, etc.) are much lower than equivalent DC transmission;
- The power plant produces AC power, so it is better to use AC than DC instead of converting;
- In the case of major faults in the network, it is easier to disconnect an AC system because the sinusoidal current tends to zero at a certain moment.

An SM has two characteristic parts, the armature on the stator and the excitation on the rotor where the armature winding (most often three-phase) is symmetrically distributed in slots around the circumference of the machine and indicates the part of the machine in which the changes of the magnetic flux induce a voltage [19]. DC flows through the exciting winding element, which is located on the rotor of the machine and creates the exciting flow, i.e., magnetic flux. Without the excitation current on the rotor, it would not be possible to produce the induced voltage on the armature windings of the synchronous generator, which means that there would be no electricity [19]. This type of excitation is used with synchronous generators that either have electromagnets on the rotor of the machine connected to a slip ring [20] or a rotor made with permanent magnets [21]. Estimation of the excitation current on the rotor is one of the key factors for the control of the SM. The excitation current maintains the efficient production of the induced voltage and is one of the most important factors while regulating the plant's power factor (maintaining the capacitive or inductive character of the power grid network) [22].

AC electrical machine loads draw reactive power from the electrical power grid. Reactive power is a disadvantage, i.e., a problem, because it causes overloading in the power grid, switches, transformers, and relays and, unfortunately, reactive power cannot be converted into useful, i.e., mechanical energy. To normalize the amount of reactive power in the power grid, SMs are used as reactive power compensators [23]. To achieve high-quality regulation and compensate for reactive power, it is necessary to regulate an SM's power factor parameter; this can be accomplished with proper regulation of the excitation current. The value of the excitation current dictates the operation mode of the synchronous motor (capacitive or inductive character) and affects the stability of the system [24].

The main problem is that most EM manufacturers do not provide enough information about their machines, which reduces the possibility of achieving highly efficient control. In general, manufacturers provide information on rated output power, rated maximum speed, rated input voltage, rated current, protection level, dimensions, and weight. However, a minority of manufacturers provide more valuable information, such as the speed-torque curve (most often at the request of the customer). Often, the nominal parameters of a synchronous motor are available and sufficient for designing the regulation, but the problem is the nonlinearity of the parameters, which is noticeable in an unadjusted operating environment (for example, high or low ambient temperature) or is not adapted for unique operating conditions (for example, speed or load torque) [25,26].

Apart from the nonlinearity, external and internal conditions affect an EM. The main problem is that there is no clear relationship between the parameters of the synchronous motor [27–30]. The parameters of an SM are mostly complex and nonlinear; thus, modeling SM parameters, such as excitation current, power factor, and load current, when the synchronous motor is running in a lagging, leading, or unity condition for reactive power compensation is a difficult task [24,27]. With the aim of improving the quality of SM modeling and a more precise estimation of parameters, many researchers use artificial intelligence (AI) estimation methods. The estimation of model parameters using techniques for linear systems has been perfected. However, an increasing number of systems require nonlinearities when increasing the dynamic range of high-performance equipment. Nonlinearities are usually ignored, but only under the assumption that linear system theories are sufficient for the retrieval of less accurate approximate solutions. For industrial applications, these solutions are most often within acceptable limits, but for applications in high-efficiency machines, linear systems are inadequate. There is therefore a need for further development of nonlinear systems, such as the synchronous motor. After defining the obstacles that are typical for highly coupled nonlinear system, such as synchronous motors, the authors of [27] used particle swarm optimization (PSO) to obtain a high-quality model with a low error rate that is robust and generally applicable to similar nonlinear systems. Various optimization algorithms, such as the evolutionary algorithm, ANN, artificial bee colony (ABC), immune method (IM), whale optimization method (WOM), particle swarm optimization (PSO) method, flower pollination (FP) method, cuttlefish optimization algorithm,

and genetic algorithm (GA), were used to optimize a permanent magnet SM (PMSM), and they are analyzed in detail in the state-of-the-art study in [25,31]. The parameters of the PMSM tend to change; in other words, there is an influence due to the nonlinearity of the parameters. This can occur due to the change in temperature or aging of SM, and for this reason, online techniques are used to identify more up-to-date and precise parameters [32].

In addition to potential optimization, AI is used to obtain improved waveforms at the output and reduce oscillations (such as output speed, torque, and current variations for three phases). For example, in [33], the authors used a fuzzy logic controller in combination with AI with certain conditions (in this case with 25 conditions and 49 conditions), which resulted in a much smoother rotation of the engine and thus less oscillation, which greatly contributed to the improvement of the system.

Using AI more precisely, fuzzy logic in combination with ANN represents an advanced method that is applied for AM control logic. AM is a nonlinear machine, where the influence of temperature, age, and additional vibration elements related to electromagnetism affect the operation of the machine. So, with this idea, the authors in [34], after mathematically modeling AM, defined a control strategy based on rotor flux, which ensures the robustness of the algorithm.

Regarding synchronous motors with electromagnets, the authors of [35] indicate the complexity and nonlinearity of the SM parameters. By applying the symbiotic organisms search (SOS) algorithm, gravitational search algorithm (GSA), ABC, and GA, the authors investigate the possibility of obtaining a high-quality algorithm with a small error of approximation, whereby the best results are achieved by SOS with a maximum error of 0.1703 A.

In the last decade, the implementation of AI algorithms has become an increasingly common strategy to solve parameter prediction, optimization, or control methods for electrical machines. Various studies are being carried out related to the optimization of excitation current losses. In the research of Kahraman et al. [24], the authors looked for an efficient solution to overcome the challenges in excitation current estimation. Implementing the k-nearest neighbor (kNN) algorithm led to optimal results of the excitation current, with an estimation error rate of 4.5% and standard deviation ( $\sigma$ ) of 1.5%. Inter-turn faults in field winding can also affect the operation of the synchronous generator, which in this case is a turbogenerator. According to Guillen et al. [36], the authors used AI to obtain a model based on machine learning (ML) regression algorithms, where the excitation current was estimated using AI, and the algorithm estimated the field current, which was compared with the actual measured current in several lifetime fluctuations. From three different models (Potier, ASA, and SVR), SVR performed the best with mean average error (MAE) and root mean square error (RMSE) metrics of 0.448 and 0.009, respectively. With the use of artificial neural networks (ANN), it is possible to achieve a model for estimating the parameters of an SM; in [37], the authors use an adaptive artificial neural network (AANN). The purpose of this research was to estimate the excitation current and help designers with the modulation of the excitation current while developing sophisticated software with a low degree of programming and improving the efficiency of the classic ANN-based approach. The results were evaluated with an average error percentage,  $\sigma$ , and the arithmetic average of all error rates of 3.507958, 2.305857851, and 2.305858, respectively. Temperature monitoring of permanent magnets in SM, which is applied to the automotive industry, is a complex challenge. The heating of the machine can cause deterioration and impact the performance of SM itself. Monitoring and reacting to high-temperature variation values inside of an SM is a challenging task. Kirchgässner et al. [38] trained several ML models that are empirically evaluated based on their accuracy for the given task of predicting. The ML algorithms used for predicting high-dynamic latent magnet temperature profiles were as follows: linear ordinary least square regression, support vector regression, kNN, and neural networks. The best result on the test set for predicting the permanent magnet temperature was achieved with the convolutional neural network (CNN) with an  $R^2$  of 0.99, MAE of 0.85 and MSE of 1.52; kNN and multi-layer perceptron

(MLP) achieved  $R^2 = 0.98$ , MAE = 1.32 and MSE = 3.20. Further proof that AI and ML can be applied in the modeling and simulation of complex phenomena is provided by surrogate models [39]. The authors presented a workflow for developing data-driven surrogate models including data generation with a physics-based simulation, designed experiments with the training data, and trained and tested the surrogates to compare ANN and the gradient boosting decision tree (GBDT) algorithm for estimation the torque behavior in SM with permanent magnets. The accuracy of the proposed model with the ANN algorithm was better, compared to the competitor GBDT, and was quite close to the finite element simulation in which the best result for negative root mean square (NMRSE) obtained with the non-hybrid model was 3.8%, while for the hybrid model it was 1.76%. Mukherjee et al. [40] estimated the speed and torque of a PMSM using tree-based algorithms. Several tree-based algorithms were used where the fine tree algorithm achieved a much lower error rate compared to the other tested algorithms, namely medium tree and coarse tree. The RMSE for the fine tree algorithm was 0.029224 and 0.052538 for the prediction of speed and torque, respectively. Traue et al. [41] developed a reinforcement learning environment toolbox for intelligent electric motor control using the open-source Python package. The obtained model showed better results in control tasks compared to conventional methods. Li et al. [42] conducted a detailed analysis for the optimization of electromagnetic devices. By using various algorithms such as support vector machines, MLP, k-NN, and CNNs, the authors proved that ML algorithms have greater robustness, high speed, and accuracy by applying a fitting algorithm in different scenarios. Bayindir et al. [43] showed the importance of the excitation current in synchronous motors in the case of reactive power compensation. By using the k-NN classifier and load current, power factor, power factor error, and change of excitation current as input values, the following results are obtained for MAE, mean absolute percentage error (MAPE), and NRMSE: 0.059, 13.146, and 8.167, respectively. The given metrics are referred to as three tuple input values, which were as follows: load current, power factor, and change of excitation current, which showed the importance of certain parameters when modeling the AI algorithm and the viability of using AI for optimizing and predicting the parameters in EM.

Considering the high accuracy rate of AI algorithms in prediction and regression, the following hypothesis questions are raised regarding SM:

- Is it possible to estimate the excitation current of SM using AI algorithms with a high precision rate and a small evaluation error?
- Is it possible to optimize the model and confirm the obtained results with 5 k-fold cross-validation using the randomized hyperparameter search?
- Which algorithm provides the best results with the possibility of implementation in a real-life situation?

This research was conducted to obtain an optimal AI algorithm that, with its properties and characteristics, can estimate the excitation current of the SM (in this case, synchronous motor drive), which contains nonlinear characteristics and phenomena. The best AI algorithm is determined to predict the future values of the excitation current with a high accuracy, small standard deviation error ( $\sigma$ ), and almost non-existent regression metrics error. By comparing the best algorithm with the related work, i.e., comparing the extreme gradient boosting regressor with small hyperparameter variations and robustness, a solution for the given problem can be obtained. In this way, precautions can be taken for plants with these machines and additional challenges can be prevented in time. The structure of this paper, along with the introduction, is divided into the following four sections: Materials and Methods, Results and Discussion, and Conclusions. The Materials and Methods section describes the entire dataset, including the input values along with targeted value, an analysis of the dataset distribution using histogram bar plots is provided, and the SM used in the creation of the dataset is described. The analysis of the dataset used for training ML algorithms is shown, and the algorithms used for the prediction of the excitation current are stated alongside the hyperparameters used in randomized hyperparameter search. All models were validated with 5 k-fold cross-validation. The Results and Discussion section

shows the results obtained from the selected/used AI algorithm, the parameters used for research, and the training of the best obtained AI model. The achieved results were described and we elaborate on which algorithm provided the best results, and which one was the worst according to the obtained metrics. In the Conclusion, the final results of this research are outlined, including which algorithm had the best regression performance for the given challenge, and the initial hypotheses questions are answered.

## 2. Materials and Methods

In the Materials and Methods section, the SM used for data collection is presented. The dataset is described as well as the input and output variables. The input data and statistical analysis before the actual use of ML algorithms are presented. Additionally, the potential challenges are indicated, and the importance of AI implementation is highlighted.

### 2.1. Potential Challenges When Modeling a Synchronous Motor

In modern EM modeling, the contribution of Štumberger et al. [44], Kron [45] is important, in which the general theoretical background for EM modeling is outlined. The generality of Kron's tensor-based approach diminishes when using matrices. By using matrices, EMs are mostly treated as magnetic (deterministic) linear systems, while nonlinear properties are ignored. However, to create positive ratios between measured and calculated results, nonlinear theory must be included in the model of the SM.

In addition to neglected factors when modeling a synchronous machine, there are several losses in SM that make the machine complex and nondeterministic and require additional power (iron losses, winding losses, and ventilation losses) [46]. Of course, these losses depend on the situation in which the machines are used; this makes it difficult to estimate parameters with regular measuring methods. Certain calculations can be made to gather information; however, due to the complexity of thermal effects in EM, numerical calculation methods such as electromagnetic and thermal finite element modeling (FEM) or computational fluid dynamics (CFD) are only used for the specific analysis of EM (part of the stator core, for example). A complete thermal analysis would require full CFD/FEM coupling. As for the modeling of the SM, the turn insulation in the field winding in the case of the silent pole synchronous machine (generator) is thin, so it is often neglected in calculations. However, surface insulation is of similar dimensions, almost equal to the surface of the conductor, so it would affect the result of the heat calculation. On the other hand, the calculation of the temperature field of a synchronous machine with silent pole rotor insulation in terms of the insulating layer is often neglected; even with little insulation there is a temperature drop, which can affect the modeling of the EM itself [47].

This subsection highlights the limitations of calculations when modeling or estimating EM. This highlights the importance of using AI algorithms for the optimization or estimation of synchronous machine parameters. AI algorithms are based on the data contained in the dataset, and the more data the dataset (algorithm) has, the greater the robustness and readiness, as well as the accuracy of the obtained model. In this work, the parameters of the used synchronous motor and the statistical analysis and distribution of data for input into the AI model are described.

### 2.2. General Information about SM

As presented in the introduction of this paper, all AC machines consist of a stator and a rotor. The main difference between AM and SM is the speed of rotation of the rotor, which in the case of SM, as the name indicates, rotates synchronously. This means that there is no delay in the rotating magnetic field of the rotor relative to the stator. In terms of the design of the rotor, SM has two versions. The first one is the salient pole [48] shown in Figure 1, and the other one is the non-salient pole (round/cylindrical) rotor [49] shown in Figure 2.



Figure 1. Salient-pole type rotor SM.



Figure 2. Round rotor type SM.

Figure 3 shows the section of the rotor and stator (armature) inside the SM. The figure shows the position of the rotor field winding for both configurations and the position of the polarization for both designs.

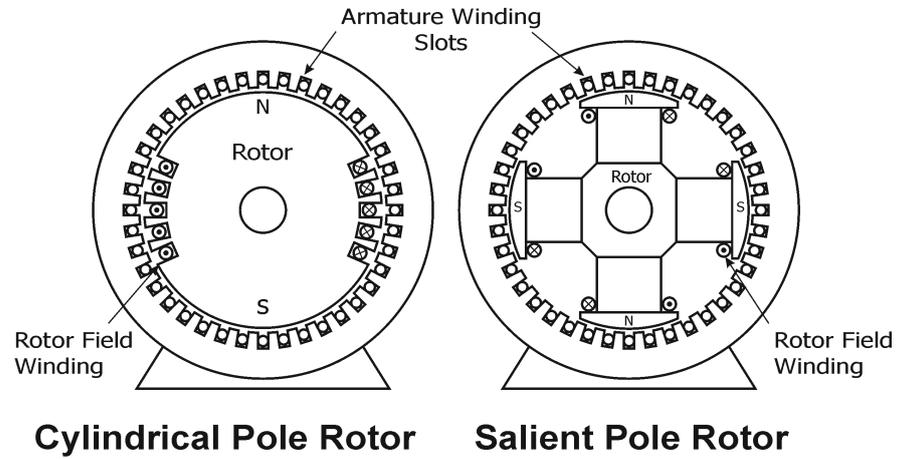


Figure 3. Representation of the salient pole and cylindrical pole rotor type SM.

The overall configuration is shown in Figure 4, which shows each component of the SM operation. In Figure 4, every part of SM is visible. To create a rotating magnetic field with a phase shift of 120 degrees, a three-phase voltage is required, with each phase marked as a, b, c and a', b', c'. The dotted dashed lines indicate the position of the phased coils, while the hatched part indicates the location of the conductors located in the armature SM.

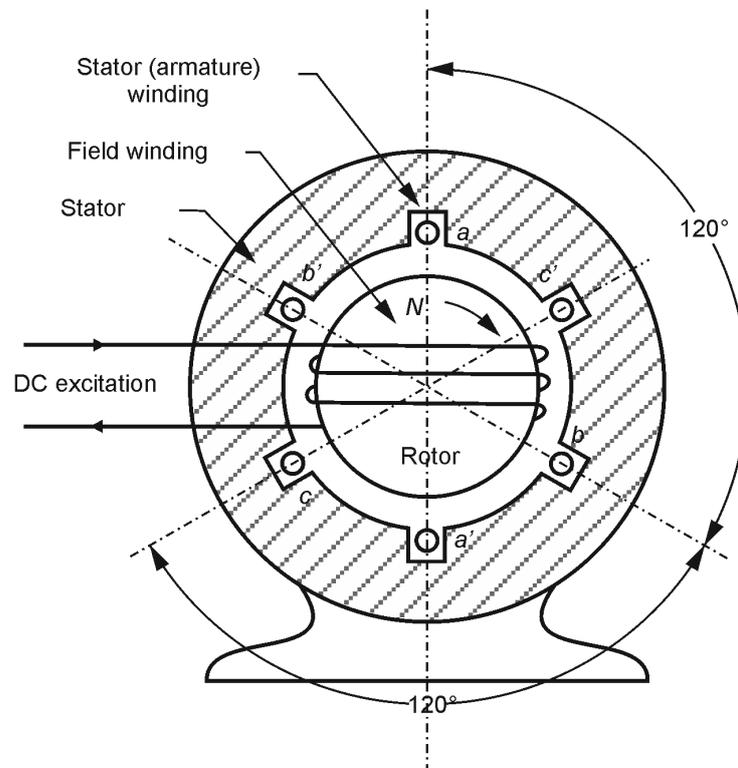
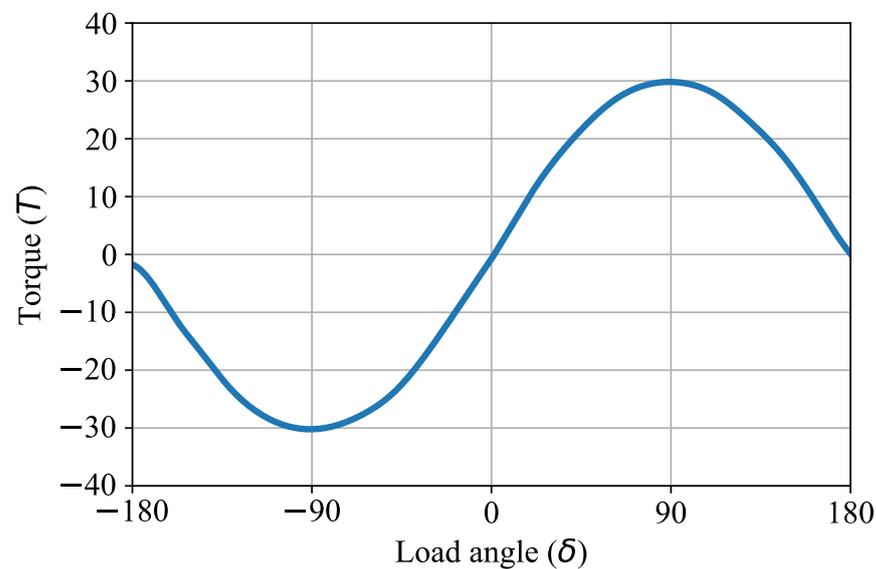


Figure 4. The basic operation of SM.

The working principle of SM is based on the establishment of a magnetic field that rotates at a synchronous speed, as follows:

$$n = \frac{120 \cdot f}{p}. \quad (1)$$

As shown in Equation (1), the speed of the rotating magnetic field of the stator depends on two factors  $f$  which are the frequency of the rotating magnetic field and  $p$  the number of poles. If the speed of the rotor (magnetic field caused by the DC component) [50], and the speed of the rotating magnetic field of the rotor are equal, under the condition that there is no load torque, both magnetic fields will tend to align against each other. When the load is applied, the rotor lags a little, i.e., "slips" by a certain degree, but at the same time adheres to the revolutions of the rotating magnetic field [49]. The achievement of the highest torque is due to the relationship between the magnetic field of the rotor and the stator, which attempt to maintain a value of 90 degrees, as can be seen from Figure 5.



**Figure 5.** The effect of load angle on produced torque.

Figure 5 shows the production of torque ( $T$ ) under the effect of load angle ( $\delta$ ); it is visible that when SM has an angle of 90 degrees, the maximum  $T$  is produced [49].

$$T = T_{max} \cdot \sin(\delta). \quad (2)$$

In addition to displaying the maximum torque at a rotor angle of 90 degrees [51], it is possible to calculate the current torque using Equation (2), where:

- $T$  is the calculated torque;
- $T_{max}$  is the maximum torque for SM;
- $\sin(\delta)$  is the sinus function of load angle.

### 2.3. Operation Conditions and Dataset Collection

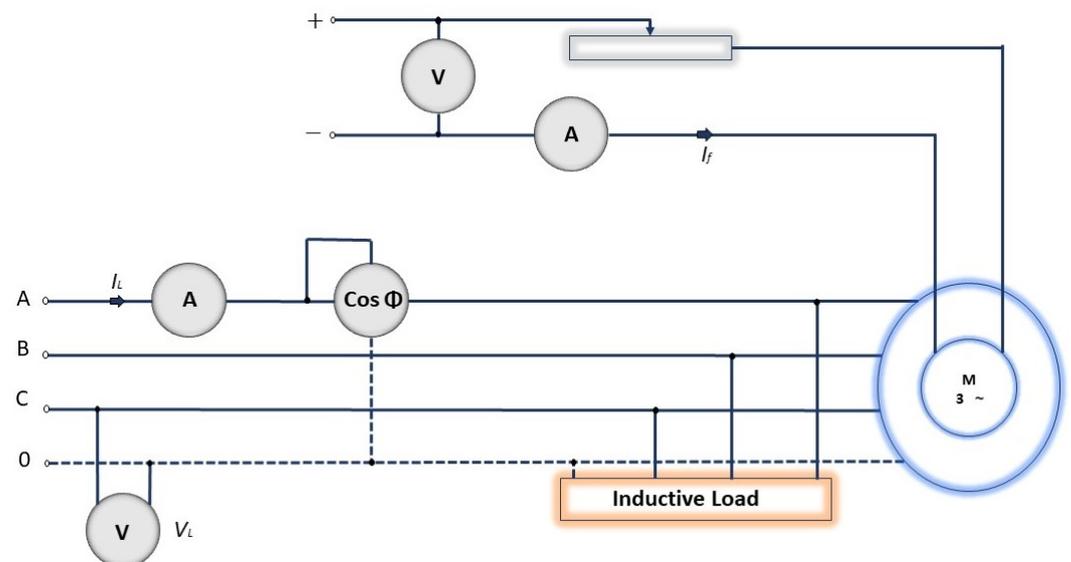
So far, the general view of SM has been described, but it is also important to describe the working conditions of the synchronous motor in which the measuring of the parameters is performed. The working conditions of the SM (synchronous motor regime of work) that were measured are shown and described in Table 1 [52].

**Table 1.** Operating properties of experimental synchronous motor used in this research.

Condition			
Star connected motor (Y) voltage [V]	Star connected motor (Y) current [A]	Triangle connected motor ( $\Delta$ )	Star connected motor (Y) current [A]
400	5.8	231	10
Power factor ( $\cos \varnothing$ )			
0.8			
Apparent power (kVA)			
4			
Revolutions per minute (rpm)			
1000			

An auxiliary motor is used to drive the synchronous motor in the test rig. In the star connection (Y), the motor has a voltage of 400 V with a 5.8 A current. In the delta connection ( $\Delta$ ), the values are different, and the voltage amount is 231 V with a current of 10 A. The values for the power factor ( $\cos \varnothing$ ), apparent power, and revolutions per minute (rpm) are the same, at 0.8, 4 kVA, and 1000, respectively [52].

The data collection procedure for the synchronous motor is shown in Figure 6 of this investigation, and it was conducted in the following four steps: firstly, the motor was used as an auxiliary motor in a laboratory environment, while a series rheostat was used to obtain information about the DC supply variable in the field circuit ( $I_f$ ). Secondly, using the AC voltage, the stator windings of the synchronous motor are powered to achieve a synchronous speed. In the third step, DC voltage is applied to the field winding of the synchronous motor, and the motor enters synchronism. In the last step, synchronous speed is realized, and the field current is modified to the minimum value (using a serial rheostat connected in series to the field circuit.) After performing all the steps, the motor takes the minimum current value from the power supply and obtains the maximum efficiency with a stable power factor [52].

**Figure 6.** The scheme of work for the experiment with an SM [52].

#### 2.4. Dataset Statistical Analysis

The previously described data are sufficient for the description of the dataset collection process and the dataset creation; however, it is also recommended to additionally analyze

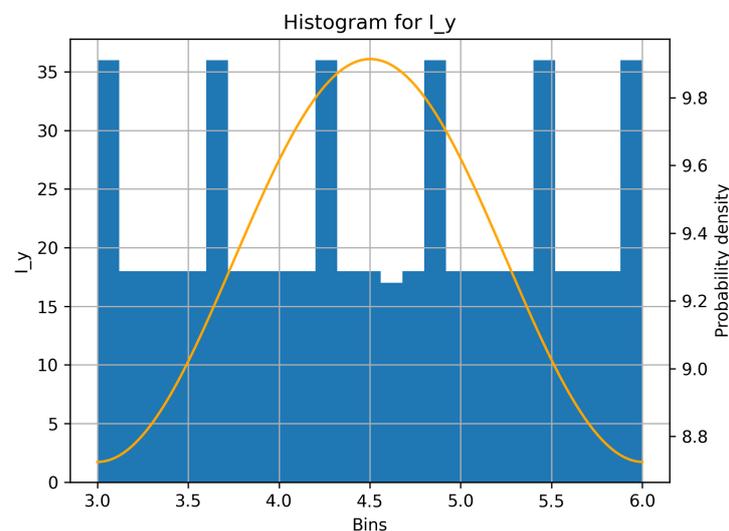
the dataset with the aim of obtaining better insight into the state of the variables and their interrelationships. In Table 2, there are four specific values that were analyzed for each individual variable, namely the mean value, minimum value, standard deviation, and maximum value.

**Table 2.** Dataset analysis used in research.

	$I_y$	PF	e	$d_f$	$I_f$
Mean value	4.499	0.825	0.174	0.350	1.530
Minimum value	3.0	0.650	0.0	0.037	1.217
Standard deviation	0.896	0.103	0.103	0.180	0.180
Maximum value	6.0	1.0	0.350	0.769	1.949

The mean value represents the mean value in each variable, the minimum represents the lowest measured value, while the maximum represents the highest measured value. It is especially important to analyze the standard deviation of the data. This is performed to consider the possibility of implementing the pre-processing methods for the input variables of the dataset, which were  $I_y$ , PF, e, and  $d_f$  for the estimation of the target variable  $I_f$ . Table 2 shows that the deviation of the data is not too high, while the mean value of the data is mostly above the ideal mean value (concerning the minimum and maximum value of the individual variable). This leads to the conclusion that it is not necessary to perform data pre-processing. Further on, the training of the AI model is performed using the original dataset values for each variable.

Furthermore, in addition to defining operating conditions and statistical analysis of data, it is necessary to consider the distribution of data in each variable, so starting from  $I_y$  shown in Figure 7, the distribution of data for each variable was analyzed.



**Figure 7.** Data distribution/histogram for Load current in the dataset, the histogram consists of an analyzed parameter as the number of inputs with the given value.

Figure 7 shows a histogram plot of the data distribution for the  $I_y$  parameter. The blue bars represent input bins while the orange curve represents the best-fitted distribution for the given data. In statistics and probability theory, this form of data distribution belongs to Von Mises lines, also known as a circular normal distribution, and is a continuous probability distribution of the circle. It is a close approximation of the wrapped normal distribution, which is the circular analogy of the normal distribution. The circular analogy can be seen

by the repeating peaks of individual values (interval maximum values of individual bar plots). The Von Mises probability density function can be presented as follows:

$$f(x | \mu, \kappa) = \frac{e^{\kappa \cos(x-\mu)}}{2\pi I_0(\kappa)}, \quad (3)$$

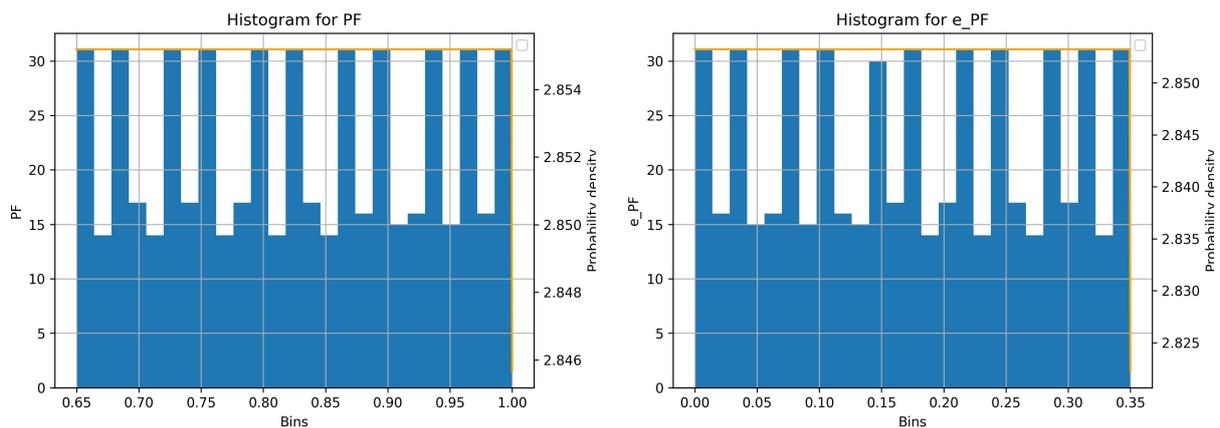
where is:

- $x$ , angle of the density function;
- $\mu$  is a representation of measure location (the given cluster distribution around  $\mu$ );
- $\kappa$  is a representation of the measure concentration;
- $I_0(\kappa)$  is the modified Bessel function with order zero.

Figure 8 shows the data distribution related to PF and e. Based on the entered data, the best possible distribution function corresponds to the Three-Parameter Kappa distribution. In statistics, K-distribution or Kappa 3 distribution is a family of continuous probability distributions that consists of three parameters and is constructed by combining two gamma distributions. The probability density function is presented as follows:

$$f(x, a) = a(a + x^a)^{-(a+1)/a}, \quad (4)$$

for  $x > 0$  and  $a > 0$  and  $a$  and  $x$  are the shape parameters of the function.



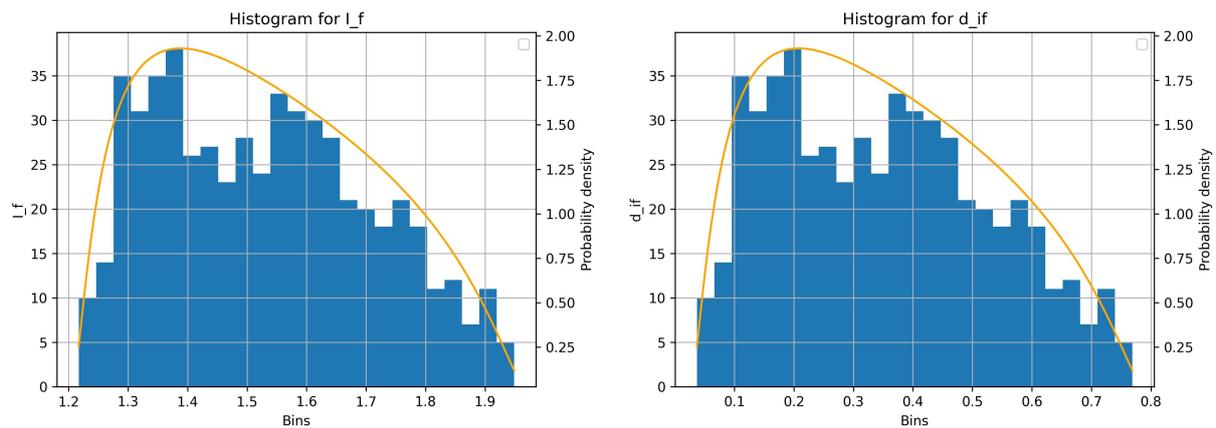
**Figure 8.** Data distribution/histogram for power factor PF and power factor error e in the dataset, the histogram consists of an analyzed parameter as the number of inputs with the given value.

In the case of an  $I_f$  and  $d_f$  of the SM, the distribution of data is slightly different than in the previous cases. By observing Figure 9, it is evident that the distribution of the data resembles a standard Gaussian distribution; however, due to the shift of the data for a certain parameter, the best-fitted distribution is Johnson's SB distribution. From Figure 9, it is evident that the highest ratio of data is at smaller values. The probability density function can be represented mathematically as follows:

$$f(x, a, b) = \frac{b}{x(1-x)} \phi\left(a + b \log \frac{x}{1-x}\right), \quad (5)$$

where:

- $x$ ,  $a$ , and  $b$  are real scalars;
- $b > 0$  and  $x \in [0, 1]$  is the probability density function of the normal distribution;
- $\phi$  is the cumulative distribution function of the normal distribution.



**Figure 9.** Data distribution/histogram for excitation current  $I_f$  and changing of excitation current  $d_f$  of synchronous machine; the histogram consists of an analyzed parameter as the number of inputs with the given value.

At the given distribution of data, shown in the histograms in Figures 7–9, the first logic attempt is to train the ML model without data pre-processing. The given dataset has favorable conditions for training ML algorithms, and in this paper, the prediction of  $I_f$  is performed with the non-preprocessed dataset.

### 2.5. Research Methodology

This section describes the research methodology of the various AI algorithms that were used in this investigation and their hyperparameters with the given table values of the selected hyperparameter. Linear models, ensembles, and even more complex ones such as artificial neural networks (ANN) and tree-based regression methods of value prediction were considered. Hyperparameter optimization was performed using randomly selected hyperparameters and by training the model using the grid-search (GS), and was validated using 5 k-fold cross-validations. During this research, the Python programming language and scikit learn library (known as sklearn) version 1.0.2 were used. The aforementioned Python library holds all of the used machine learning algorithms (the exception was the extreme gradient boosting regressor), whereby an initial investigation was conducted using default hyperparameters for individual algorithms. After the initial investigation, the hyperparameter optimization was conducted using random values that are presented in each table for every algorithm used, where the used algorithms are as follows:

- Extra trees regressor (ETR);
- Elasticnet regressor (EN);
- K-nearest neighbor regressor (k-NN);
- Linear regressor (LR);
- Random forest regressor (RFR);
- Ridge regressor (RR);
- Stochastic gradient descent regressor (SGD);
- Support vector regressor (SVR);
- MLP regressor;
- Extreme gradient boosting regressor (XGBoost).

Each algorithm will be described in a separate subsection where the hyperparameter values used for training the model will be shown. During the investigation, the following two search methods were applied: the initial search with default hyperparameters, and then the randomized hyperparameter search. After the initial search, all the algorithms were optimized using GS and cross-validation.

### 2.5.1. Extra Trees Regressor

ETR belongs to the group of ML ensemble methods that combines predictions from multiple decision trees. It belongs to a random forest algorithm although it is based on a simpler approach where each member is a decision tree. In this case, the mentioned algorithm often results in a similar or better result than the base algorithm, which is the random forest algorithm [53]. ETR trees differ from the usual decision trees in terms of the principle of building the tree itself. The search for the best division for separating the hub samples into two rough ones is drawn for a random division of the selected features, i.e., `max_features`, and the best division among them is selected [54]. The varied hyperparameters of the algorithm are visible in Table 3 and are varied based on previous research according to [55–57].

**Table 3.** An overview of the varied hyperparameters used for the ETR.

Parameter Name	Minimum Value	Maximum Value
<code>n_estimators</code>	1000	10,000
<code>criterion</code>	squared error, <code>friedman_mse</code>	
<code>max_depth</code>	None	
<code>min_samples_split</code>	2	10
<code>max_features</code>	auto, sqrt, log2	
<code>random state</code>	0	63

The criterion is a quality measurement of split data. For this kind of decision tree algorithm, supported criterion inputs are `squared_error` for the mean squared error (MSE) and `friedman_mse` for the improved Friedman mean squared error. The main difference between these two criteria are that the Friedman MSE calculates the impurity of the current node which leads to a reduction in the leaf impurity. The number of trees that ETR produces is regulated with `n_estimators`. `Max_depth` indicates the maximum depth of the tree. It was set to none where the main reason was to remove all of the leaf impurities from the tree. `max_features` is the number of considered features when looking for the optimal split. Input values can be sqrt, log2, integer, or float number. `Random_state` controls the randomness of the instance of bootstrapping samples while building trees and the feature sampling for consideration when looking for optimal node split, and draws the splits for `max_features`. The lowest value of the sample number for splitting the internal node is defined with `min_samples_split`. If the given value is an integer, then `min_samples_split` is considered as the minimum number; in the case of the float value, the `min_samples_split` is considered as a fraction.

### 2.5.2. Elasticnet Regressor

EN belongs to the group of linear models of the scikit library where the regression process uses the penalties from both lasso and ridge techniques. This technique provides a learning spares model, where several weights are not 0 as is the case in lasso, while still maintaining the regularization characteristic from the ridge. The combined technique from lasso and ridge regression methods helps to overcome the deficiency and improve the statistical regulation model [54,58]. During the investigation of this AI algorithm, the parameters shown in Table 4 were varied according to [59,60].

**Table 4.** An overview of the varied hyperparameters used for the EN.

Parameter Name	Minimum Value	Maximum Value
alpha	0.1	10
l1_ratio	0.1	10
max_iter	1000	10,000
selection	random, cyclic	

Alpha constantly multiplies the penalty terms if its value is set to 0.0; then, the mathematical meaning of this parameter is equivalent to a linear ordinary least square regression. The elasticnet l1\_ratio is the mixing parameter between 0.0 and 1.0. The maximum number of iterations is max\_iter set to 10,000. The selection parameter is the coefficient updated for every iteration by choosing the right coefficient of selection, and the results often converge faster.

### 2.5.3. K-nearest Neighbour Regressor

The k-NN algorithm is one of the decision tree supervised ML algorithms based on determining the local minimum of the target function that is used to learn the unknown function with precision and accuracy. The decision system essentially calculates the distance between data points in space. It relies on observable data similarity and near-distance metrics for accurate prediction generation [38,54]. When modelling k-NN based on research in [35,54,61], the parameters shown in Table 5 were varied.

N\_neighbours is the number of neighbor data in the decision tree. There are three ways in which the weights of the prediction function are distributed, which are uniform, inverse, and callable. The computation of the closest neighbor is achieved with an algorithm. The eaf\_size can be used in BallTree or KDTree. It can be used to determine the memory of a given leaf of the tree as well as speed up the construction of the building process.

**Table 5.** An overview of the varied hyperparameters used for the k-NN.

Parameter Name	Minimum Value	Maximum Value
n_neighbours	1	1000
weights	uniform, distance	
leaf_size	1	1000
algorithm	auto, ball_tree, kd_tree, brute	
p	2	50

### 2.5.4. Linear Regressor

LR is better known as the ordinary least squares method. The method consists of establishing a correlation between the dependent variable and the independent variable, where the best approximation fits a straight linear line [62]. The variation of parameters is presented in Table 6.

**Table 6.** An overview of the varied hyperparameters used for the LR.

Parameter Name	Minimum Value	Maximum Value
fit_intercept	True, False	
normalize	True, False	
positive	True, False	

Fit\_intercept calculates the intercept for the given model. Normalize is used for normalization of the input parameter, and positive is used if the True forces modulation coefficients are positive.

#### 2.5.5. Random Forest Regressor

The RFR algorithm belongs to the class of machine learning algorithms that, in combination with random decision trees, trains a model on sub-datasets [63]. The use of multiple trees contributes to the stability of the algorithm and reduces the variance of the result itself. Each tree created by this algorithm uses a different sample of input data; furthermore, different features are taken into account at each node where, after training each tree, the mean value of the unique result is calculated [63,64]. The varied hyperparameters are shown in Table 7 and are based on [55,65,66].

**Table 7.** An overview of the varied hyperparameters used for the RFR.

Parameter Name	Minimum Value	Maximum Value
n_estimators	100	5000
criterion	squared_error, absolute_error, Poisson	
max_features	sqrt, log2	

Given that the ETR in section 2.5.1 originated from the basic RFR, the parameters that were varied in this investigation will not be described in detail. When investigating this algorithm, the guiding principle was that differences between the two algorithms exist in terms of ETR [67]. In this case, RFR uses bootstrap replicas, i.e., it uses subsamples of the input data and replaces it, whereas ETR uses the original complete samples. In addition, RFR uses an optimal node split point, while ETR takes a random one.

#### 2.5.6. Ridge Regressor

RR belongs to the linear model method of the scikit-learn python library. The specialization of RR lies in the fact that it analyzes multiple regressions that are multicollinear. Because of the complex science behind this ML algorithm, it is not the most used algorithm; however, because of its regularization methods, it is sometimes the best fit. RR is a linear least squares method with an additional regularization parameter. The selection of the varied parameters presented in Table 8 was carried out according to previous [68–70] articles on related issues.

**Table 8.** An overview of the varied hyperparameters used for the RR.

Parameter Name	Minimum Value	Maximum Value
alpha	1.0	100.0
max_iter	1000	50,000
tol	$1 \times 10^{-5}$	$1 \times 10^{-1}$
fit_intercept	True, False	
solver	svd, cholesky, lsqr, sparse_cg, sag, saga	

The parameters alpha and max\_iter are explained in the previous subsections, tol indicates the precision of the solution, while the novelty is the solver where the solver is the principle of model optimization.

#### 2.5.7. Stochastic Gradient Descent Regressor

The SGD regressor belongs to the group of linear models of the scikit-learn python library. It is a generic optimized algorithm that can find a high-quality and optimal solution for a wide range of situations. The idea for altering the parameters is an iterative process

of selecting hyperparameters to reduce the cost function. The proportions of the step are one of the most crucial parameters determined by varying the learning rate, which in the case of setting too small a value can lead to slow convergence and the opposite of that can exceed the optimal value [71,72]. In line with to [73,74], the hyperparameters in Table 9 are as follows.

**Table 9.** An overview of the varied hyperparameters used for the SGD.

Parameter Name	Minimum Value	Maximum Value
alpha	0.0001	10.0
max_iter	1000	10,000
validation_fraction		0.15
power_t	0.1	0.5
learning_rate		invscaling, optimal, constant
shuffle		True, False
l1_ratio	0.0001	0.5

The validation set is a proportion of training data used for early stopping defined with validation\_fraction. The power\_t parameter is an exponent for the invscaling learning\_rate module which defines the result of the value of the learning rate parameter. The role of the shuffle parameter is to shuffle data points for each training epoch. The elasticnet mixing parameter is varied with the l1\_ratio, and its value is in the range from 0 to 1.

#### 2.5.8. Support Vector Regressor

SVR is an ML regression algorithm that is compatible with both linear and polynomial regression. This algorithm works on the principle support vector machine (SVM), but in the case of SVR, it differs in the way that SVM is a classifier for the prediction of discrete categorical values, while SVR is a regressor used for continuous order predicting variables [75,76]. In this investigation, several hyperparameters are varied, as shown in Table 10.

**Table 10.** An overview of the varied hyperparameters used for the SVR.

Parameter Name	Minimum Value	Maximum Value
kernel		linear, poly, rbf, sigmoid
degree	1	10,000
gamma		scale, auto
coef0	0.0	10.0
tol	$1 \times 10^{-5}$	$1 \times 10^{-10}$
C	0.5	20.0
epsilon	0.05	20.0
max_iter	100	10,000

According to previous research, [77–79], the varied hyperparameters are as follows. The kernel hyperparameter specifies a data-mapping function. The degree is the degree of the polynomial kernel function, and gamma is the kernel coefficient but only in the case of rbf, poly, and sigmoid functions. Coef0 is an independent term in kernel function and can only be significant in the poly or sigmoid kernel function, C is a regularization parameter where the amplitude of regularization is inversely proportional and must be strictly positive. The last hyperparameter is epsilon, which defines the margin of tolerance.

### 2.5.9. Multi-Layer Perceptron Regressor

An MLP is a feed-forward ANN that generates a set of defined outputs from an input value. Specifically, it is characterized by a node that is connected to several layers as a direct link between input, hidden, and output layers where a backpropagation logic is used for training the neural network itself. Its design allows it to predict any continuous function and it can solve nonlinearly separable problems. In this investigation, several hyperparameters are modified according to research shown in [54,80–82].

Table 11 shows the varied hyperparameters used in this investigation. In addition to the previously described hyperparameters, such as `olver`, `alpha`, `power_t`, `max_iter`, and `tol`, new parameters unique to the MLP ML algorithm have been changed. Activation stands for the activation function which decides whether a neuron contained in MLP should be activated or left in stasis. Using simple mathematical operations, it determines the importance of the selected neuron in the ANN network for the prediction process. `Hidden_layer` is a number of hidden layers inside the MLP ANN, and the solver is the algorithm specified for weight optimization along all of the nodes contained in MLP.

**Table 11.** An overview of the varied hyperparameters used for the MLP.

Parameter Name	Minimum Value	Maximum Value
<code>hidden_layer_sizes</code> (2,3 and 4 hidden layers)	5	600
<code>activation</code>	tahn, relu, identity, logistic	
<code>solver</code>	adam, lbfgs	
<code>alpha</code>	0.02	0.5
<code>power_t</code>	0.5	2.0
<code>max_iter</code>	1000	10,000
<code>tol</code>	$1 \times 10^{-5}$	$1 \times 10^{-1}$
<code>max_iter</code>	1000	10,000

### 2.5.10. Extreme Gradient Boosting Regressor

XGBoost is an AI algorithm that is efficient and scalable for implementation based on the gradient boosting framework. The algorithm includes a linear model solver and a tree-based algorithm that supports several objective functions, including regression, classification, and ranking. Its special features are the possibility to regularize learning, the gradient tree boosting functionality, and the ability to shrink subsample columns [83,84]. Due to its performance and previous research, it was taken into account during this investigation. So, the varied hyperparameters used in this research are shown in Table 12.

**Table 12.** An overview of the varied hyperparameters used for the XGBoost.

Parameter Name	Minimum Value	Maximum Value
<code>learning_rate</code>	0.2	0.02
<code>max_depth</code>	6	64
<code>min_child_weight</code>	1	10
<code>gamma</code>	0.0	1.5
<code>colsample_bytree</code>	0.1	1.0
<code>max_delta_step</code>	0.0	1.5

`Learning_rate` in XGBoost is slightly different than in the previously mentioned algorithms. In this case, the mentioned hyperparameter represents a reduction for each step that the algorithm performs. `Min_child_weight` is a parameter that defines the minimum

sum of instance weights needed for generating a child. The pseudo-regularization hyperparameter specific to gradient boosting and XGBoost is gamma. It is vital because the higher the gamma is, the better the regularization process. Subsampling is performed using `colsample_bytree`, which takes a portion of the defined dimensions. `Max_delta_step` is the maximum step allowed in the algorithm for each tree's weight estimation.

### 3. Results and Discussion

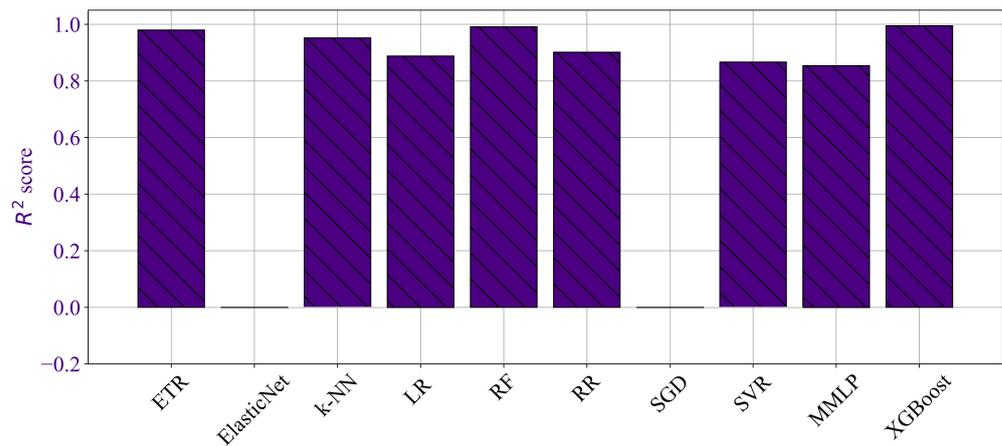
In the results section, the results of this research are presented. The following evaluation metrics are presented for each algorithm: The coefficient of determination ( $R^2$ ), MSE, and MAPE. When defining the results, the best-performing algorithms were selected, which were additionally evaluated using the  $\sigma$  for all three individual metrics.

The first part of the research refers to the defined algorithms in Section 2.5 from Sections 2.5.1–2.5.10. Algorithms were trained on default parameters from the scikit-learn Python library. Each algorithm model was evaluated with three previously defined metrics, and the results can be found in Table 13.

**Table 13.** Table of the results and evaluation metrics of all algorithms used in this research for default hyperparameters.

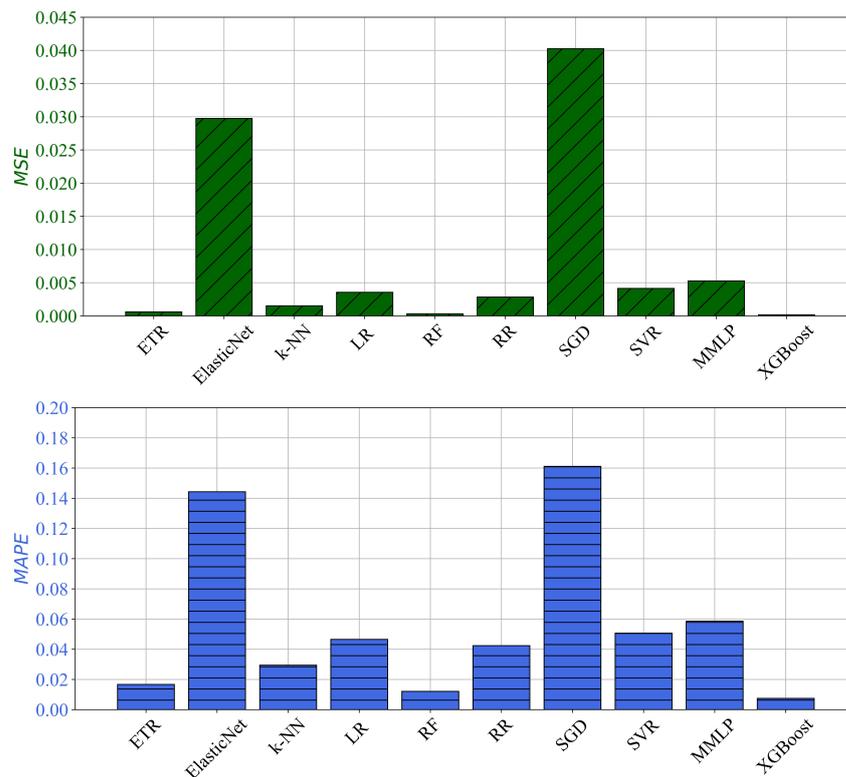
Regressor Name	$R^2$	MSE	MAPE
Extra Trees	0.9795	0.0006	0.0166
ElasticNet	0	0.0297	0.1442
k-Nearest Neighbour	0.9517	0.0015	0.0294
Linear	0.8875	0.0035	0.0465
Random Forest	0.9909	0.0003	0.0121
Ridge	0.9012	0.0028	0.0423
Stochastic Gradient Descent	0	0.0402	0.1611
Support Vector Machines	0.8662	0.0041	0.0506
Multi-layer Perceptron	0.8534	0.0052	0.0585
Extreme Gradient Boosting	0.99433	0.0001	0.0074

Table 13 shows the results of the AI algorithm used in this research with default hyperparameters for each algorithm. The goal of the initial research was to determine algorithms suitable for estimating the excitation current of SM. This part of the research indicates the oscillations of the results for each algorithm, which can be seen from Table 13. Several acceptable algorithms satisfy the given tasks including the tree-based algorithms, which had the best results in the initial research. Regarding  $R^2$ , the best result is achieved with XGBoost, of 0.9943, followed by 0.99 with RFR. Furthermore, results above 0.9 are given by RFR, ETR, and RR, as shown in Figure 10.



**Figure 10.** The  $R^2$  results from the initial investigation.

Regarding other evaluation metrics, XGBoost again has the best result; in this case, the smallest MSE and MAPE are 0.0001 and 0.0074, respectively. The second-best algorithm with the best performance is RF with 0.003 and 0.0121 MSE and MAPE. The third best algorithm is ETR with 0.006 and 0.016. Other algorithms also have a low error rate in both metrics, although with at least one potential higher than the mentioned algorithms, as shown in Figure 11.



**Figure 11.** The MSE and MAE results from the initial investigation.

The most frequent problem faced while training ML algorithms is overfitting of the training data. Overfitting can be avoided by using the cross-validation method, as is shown in the second part of this investigation. Parameters presented in individual tables in the subsections from Sections 2.5.1–2.5.10 were used, and the optimization was achieved using a randomized hyperparameter search method with 5 k-fold cross-validation for each algorithm. The results of optimized and cross-validated algorithms can be seen in Table 14.

**Table 14.** Table of results and evaluation metrics of all algorithms used in this research for varied hyperparameters and cross-validated models.

Regressor Name	$\overline{R^2}$	$\overline{MSE}$	$\overline{MAPE}$
Extra Trees	0.9784	0.0006	0.0164
ElasticNet	0	0.0297	0.1442
k-Nearest Neighbour	0.4484	0.0173	0.1127
Linear	0.8881	0.0035	0.0462
Random Forest	0.9746	0.0008	0.0223
Ridge	0.4833	0.01487	0.1054
Stochastic Gradient Descent	0.8731	0.0046	0.0535
Support Vector Machines	0.8844	0.0045	0.0523
Multi-layer Perceptron	0.9303	0.0025	0.0392
Extreme Gradient Boosting	0.9963	0.0001	0.0057

Table 14 shows optimized and cross-validated results for each algorithm. Several similarities and differences are visible and will be explained in the following part of this research. In this part of the research, Tree-based algorithms still dominate in all evaluation values. XGboost still has the most ideal  $R^2$  metrics of 0.9963,  $\overline{MSE}$  of 0.0001, and  $\overline{MAPE}$  of 0.0057. Furthermore, ETR, RFR, and MLP have  $R^2$  values above 0.9 and the smallest evaluation errors of  $\overline{MSE}$  and  $\overline{MAPE}$ . Other algorithms have minimal improvement or have a deterioration of metrics, and the SGD demonstrated the largest increase in  $R^2$ , with an increase 0.96 compared to the default hyperparameters, while LR had the smallest shift in results. The largest decreases in  $\overline{MSE}$  and  $\overline{MAPE}$  errors for the MLP algorithm were stark, while k-NN's performance was worst. Furthermore, as mentioned in the previous section based on the given results from Tables 13 and 14, three AI algorithms were selected for the final estimation of the excitation current in SM, namely: ETR, RFR, and XGBoost. The obtained ratios of  $\sigma$  for  $R^2$ ,  $\overline{MSE}$  and  $\overline{MAPE}$  are shown in the figures below.

Figure 12 shows the  $R^2$  and  $\sigma$  for ETR, RFR, and XGBoost algorithms. As shown in Table 14, XGboost has the highest  $R^2$ , followed by ETR and finally RFR; however, in this case, the main focus is on the  $\sigma$  of all three algorithms, so the best result is achieved by XGBoost, which has the smallest  $\sigma$  value out of all three algorithms of 0.0011, which is up to eight times lower than for ETR and RFR.

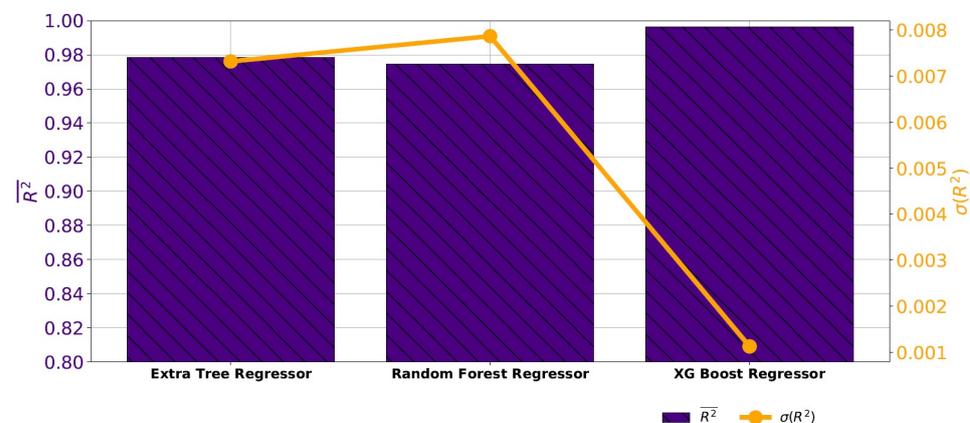
**Figure 12.** The  $\sigma$  of  $R^2$  results for ETR, RFR and XGBoost algorithms.

Figure 13 represents  $\overline{MSE}$  and  $\sigma$  for the ETR, RFR, and XGboost algorithms. It can be seen from the figure that the XGBoost again has the smallest  $\overline{MSE}$  of 0.0001, while the largest in this part of the research is ETR, valued at 0.000035. The  $\sigma$  for XGBoost again

achieves the best results, with a  $\sigma$  result of 0.0000033, while the ETR again had the worst performance.

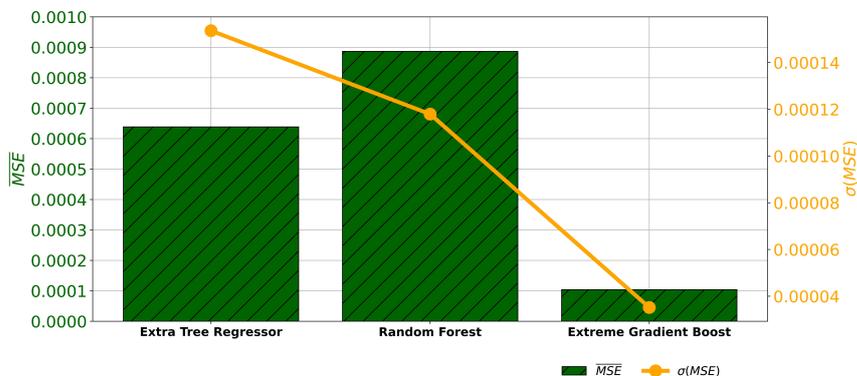


Figure 13. The  $\sigma$  of MSE results for ETR, RFR, and XGBoost cross-validated algorithms.

In the last part of the performance evaluation of the selected algorithms, the value of MAPE and  $\sigma$  of individual algorithms is shown in Figure 14. The value of the  $\overline{MAPE}$  result favors (as in the previous two cases) of the XGBoost algorithm with a value of 0.0057, while the worst performance was achieved by RFR with a value of 0.0223.  $\sigma$  for  $\overline{MAPE}$ , in this case, was 0.0003, while for RFR it was 0.0012 and for ETR it was 0.001869.

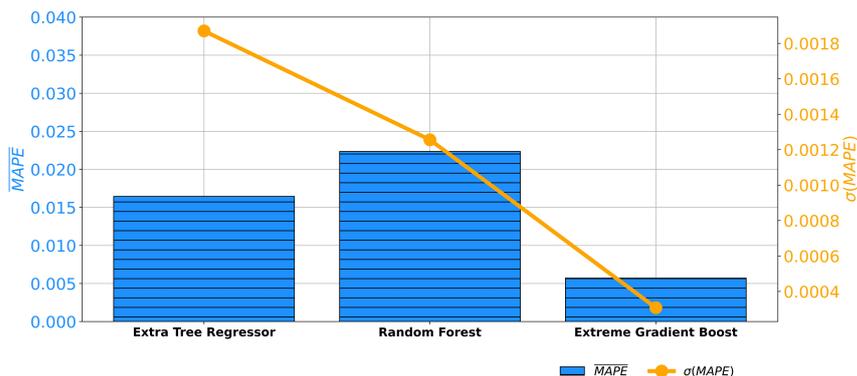


Figure 14. The  $\sigma$  of MAPE results for ETR, RFR, and XGBoost cross-validated algorithms.

#### 4. Conclusions

In this paper, different AI algorithms were used to estimate the excitation current of SM. During the investigation, three different approaches were taken to obtain a solution. The first approach was to use linear, ensemble, tree-based, and neural network algorithms to obtain favorable results for further consideration. At the same time, the hyperparameters that were taken for training the defined algorithms were set to default values. It can be seen from the results that a larger number of algorithms were considered, which shows the possibility of optimal selection for the estimation of the excitation current, but the question arose of whether the obtained model led to overfitting, i.e., whether this was the case for the affected samples in the train/test ratio. After that, 5 k-fold cross-validation of the data with randomized hyperparameters was performed. The results varied from algorithm to algorithm, with the XGBoost algorithm excelling with the best result in all three metrics. As the last part of this research, an investigation of the  $\sigma$  was carried out for all three metrics; however, in this case, only the tree-based AI algorithms were considered (ETR, RFR, and XGBoost). When looking at the results for all three metrics, XGBoost again demonstrated the best results for all validation metric areas assessed. Compared to other control techniques in the literature, this model provides the following benefits: simpler and more precise results because it takes into account only the highlighted weight parameters

of the synchronous motor, and the possibility to determine the relationship between the synchronous motor parameters and the excitation current. Finally, it provides a novel approach to excitation current estimation using a different type of algorithm compared to the related research for categories of smaller motors (characteristics shown in Table 1).

Based on the conducted research, the answers to the hypothetical questions in the Introduction are as follows:

- It is possible to estimate the excitation current of a synchronous motor using an AI algorithm with high precision and accuracy. Based on the given research it was shown that the most optimal algorithm was XGBoost.
- Using GS and cross-validation, the values were validated, and the parameters of the AI model were optimized, which provides suitable evaluation metrics for the estimation of the excitation current.
- From the larger number of presented algorithms in this paper, the best possible algorithm that provides optimal results and the smallest  $\sigma$  is XGboost with a high value of  $R^2$  and small values of MSE and MAPE.

To conclude, and in terms of future research plans, the following points should be made. First, the given dataset should be further expanded with more up-to-date points, for example for millisecond or microsecond ratios, i.e., changes in these ratios. As a second step, the influence of heat on SM should be incorporated, as it is well known that every electrical machine, device, etc., under the influence of heat experiences changes in the given parameters. For example, a synchronous motor in low temperatures has a different resistance in relation to electrical conductors on the stator or the rotor (depending on the version). The endpoint of additional research would be to implement a given model to control logic so as to evaluate the obtained results in a real experiment. When all the mentioned points are implemented, a dataset will be created and the optimal AI model that approximates the excitation current will be obtained again; a more precise and realistic solution will be obtained concerning the research carried out in this article.

**Author Contributions:** Conceptualization, N.A., I.L. and Z.C.; methodology, N.A., M.G. and Z.C.; software, N.A., I.L. and Z.C.; validation, I.L., M.G. and Z.C.; formal analysis, N.A., M.G. and I.L.; investigation, N.A. and I.L.; resources, Z.C.; data curation, I.L. and M.G.; writing—original draft preparation, N.A., I.L. and Z.C.; writing—review and editing: N.A., M.G. and I.L.; visualization, I.L. and Z.C.; supervision, Z.C.; project administration, Z.C.; funding acquisition, Z.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The dataset is available at: <https://www.kaggle.com/datasets/fedesoriano/synchronous-machine-dataset> (accessed on 14 November 2022).

**Acknowledgments:** This research received support from the CEEPUS network CIII-HR-0108, European Regional Development Fund under the grant KK.01.1.1.01.0009 (DATACROSS), project, CEKOM under the grant KK.01.2.2.03.0004, Erasmus+ project WICT under the grant 2021-1-HR01-KA220-HED-000031177, and the University of Rijeka scientific grant uniri-tehnic-18-275-1447.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Kazim, M.; Aliyeva, L. Development of Electromechanic Power Control System. p. 74. Available online: <http://ieeacademy.org/wp-content/uploads/2022/06/Ecoenergetics-N2-2022-papers.pdf#page=75> (accessed on 14 November 2022).
2. Rajput, R. *Alternating Current Machines*; Firewall Media: New Delhi, India, 2002.
3. Demir, U.; Akuner, M. Design and optimization of in-wheel asynchronous motor for electric vehicle. *J. Fac. Eng. Archit. Gazi Univ.* **2018**, *33*, 1517–1530.

4. Omelchenko, E.; Khramshin, T.; Tanich, V.; Kozhevnikov, I. Dynamic computer model of traction asynchronous motor. In Proceedings of the 2019 IEEE Russian Workshop on Power Engineering and Automation of Metallurgy Industry: Research & Practice (PEAMI), Magnitogorsk, Russia, 4–5 October 2019; pp. 59–63.
5. Enache, M.A.; Campeanu, A.; Enache, S.; Vlad, I.; Popescu, M. Optimal Design of Asynchronous Motors used for Driving Coal Mills. In Proceedings of the 2019 International Conference on Electromechanical and Energy Systems (SIELMEN), Craiova, Romania, 9–11 October 2019; pp. 1–6.
6. Crelerot, O.; Bernot, F.; Kauffmann, J. Study of an electrical differential motor for electrical car. In Proceedings of the 1993 Sixth International Conference on Electrical Machines and Drives (Conf. Publ. No. 376), Oxford, UK, 8–10 September 1993; pp. 416–420.
7. Migal, V.; Arhun, S.; Hnatov, A.; Dvadnenko, V.; Ponikarovska, S. Substantiating the criteria for assessing the quality of asynchronous traction electric motors in electric vehicles and hybrid cars. *J. Korean Soc. Precis. Eng.* **2019**, *36*, 989–999. [[CrossRef](#)]
8. Durantay, L.; Velly, N.; Pradurat, J.F.; Chisholm, M. New testing method for large high-speed induction motors. *IEEE Trans. Ind. Appl.* **2016**, *53*, 660–666. [[CrossRef](#)]
9. Mutoh, N.; Ohnuma, N.; Omiya, A.; Konya, M. A motor driving controller suitable for elevators. *IEEE Trans. Power Electron.* **1998**, *13*, 1123–1134. [[CrossRef](#)]
10. Syahputra, R.; Nugroho, A.W.; Purwanto, K.; Mujaahid, F. Dynamic Performance of Synchronous Generator in Steam Power Plant. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 12. [[CrossRef](#)]
11. Setiyoso, A.; Purwadi, A.; Halimi, B.; Rizqiawan, A. Design of synchronous generator 10625kVA for small power-plant in Indonesia. In Proceedings of the 2016 3rd Conference on Power Engineering and Renewable Energy (ICPERE), Yogyakarta, Indonesia, 29–30 November 2016; pp. 81–86.
12. Khodadadi, A.; Pishkesh, M.N.; Zaker, B.; Karrari, M. Parameters identification and dynamical modeling of excitation system and generator in a steam power plant. In Proceedings of the 2018 6th International Conference on Control Engineering & Information Technology (CEIT), Istanbul, Turkey, 25–27 October 2018; pp. 1–5.
13. Rasal, R.S.; Shinde, S.M. Modeling and Simulation of Hydro Power Plant with Reversible Turbine and Synchronous Generator. In Proceedings of the 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), Pune, India, 19–21 September 2019; pp. 1–5.
14. Glavan, B.; Hanić, Z.; Kovačić, M.; Vražić, M. Condition-Monitoring System for Identification and Representation of the Capability Diagram Limits for Multiple Synchronous Generators in a Hydro Power-Plant. *Energies* **2020**, *13*, 3800. [[CrossRef](#)]
15. Celikdemir, S.; ÖZDEMİR, M. Permanent Magnet Synchronous Generator Wind Power Plant Study. In Proceedings of the 2019 4th International Conference on Power Electronics and their Applications (ICPEA), Elazig, Turkey, 25–27 September 2019; pp. 1–4.
16. Celikdemir, S.; ÖZDEMİR, M. Wind Power Plant Application with Permanent Magnet Synchronous Generator. In Proceedings of the 2019 4th International Conference on Power Electronics and their Applications (ICPEA), Elazig, Turkey, 25–27 September 2019; pp. 1–4.
17. Newbold, F.; Perkins, T. Wellbore transmission of electrical power. *J. Can. Pet. Technol.* **1978**, *17*. [[CrossRef](#)]
18. Wilberforce, T.; Olabi, A.; Sayed, E.T.; Elsaid, K.; Maghrabie, H.M.; Abdelkareem, M.A. A review on zero energy buildings—Pros and cons. *Energy Built Environ.* **2021**, in press. [[CrossRef](#)]
19. Li, J. *Design and Application of Modern Synchronous Generator Excitation Systems*; John Wiley & Sons: Singapore, 2019.
20. Weber, J.N.; Ponick, B. Berührunglose Übertrager als Alternative zu synchronen oder Gegendrehfeld-Erregermaschinen. *E I Elektrotechnik Und Informationstechnik* **2018**, *135*, 204–212. [[CrossRef](#)]
21. Eriksson, S. Permanent Magnet Synchronous Machines. *Energies* **2019**, *12*, 2830. [[CrossRef](#)]
22. Nakamura, Y.; Kudo, T.; Ishibashi, F.; Hibino, S. High-efficiency drive due to power factor control of a permanent magnet synchronous motor. *IEEE Trans. Power Electron.* **1995**, *10*, 247–253. [[CrossRef](#)]
23. Colak, I.; Bayindir, R.; Bay, Ö.F. Reactive power compensation using a fuzzy logic controlled synchronous motor. *Energy Convers. Manag.* **2003**, *44*, 2189–2204. [[CrossRef](#)]
24. Kahraman, H.; Bayindir, R.; Sagiroglu, S. A new approach to predict the excitation current and parameter weightings of synchronous machines based on genetic algorithm-based k-NN estimator. *Energy Convers. Manag.* **2012**, *64*, 129–138. [[CrossRef](#)]
25. Rifaq, M.S.; Jung, J.W. A comprehensive review of state-of-the-art parameter estimation techniques for permanent magnet synchronous motors in wide speed range. *IEEE Trans. Ind. Inform.* **2019**, *16*, 4747–4758. [[CrossRef](#)]
26. Fernandez, D.; Hyun, D.; Park, Y.; Reigosa, D.D.; Lee, S.B.; Lee, D.M.; Briz, F. Permanent magnet temperature estimation in PM synchronous motors using low-cost hall effect sensors. *IEEE Trans. Ind. Appl.* **2017**, *53*, 4515–4525. [[CrossRef](#)]
27. Liu, L.; Liu, W.; Cartes, D.A. Particle swarm optimization-based parameter identification applied to permanent magnet synchronous motors. *Eng. Appl. Artif. Intell.* **2008**, *21*, 1092–1100. [[CrossRef](#)]
28. Leon, A.; Solsona, J.; Figueroa, J.; Valla, M. Optimization with constraints for excitation control in synchronous generators. *Energy* **2011**, *36*, 5366–5373. [[CrossRef](#)]
29. Dehghani, M.; Karrari, M.; Rosehart, W.; Malik, O. Synchronous machine model parameters estimation by a time-domain identification method. *Int. J. Electr. Power Energy Syst.* **2010**, *32*, 524–529. [[CrossRef](#)]
30. Senjyu, T.; Kinjo, K.; Urasaki, N.; Uezato, K. High efficiency control of synchronous reluctance motors using extended Kalman filter. *IEEE Trans. Ind. Electron.* **2003**, *50*, 726–732. [[CrossRef](#)]
31. Arellano-Padilla, J.; Sumner, M.; Gerada, C. Winding condition monitoring scheme for a permanent magnet machine using high-frequency injection. *IET Electr. Power Appl.* **2011**, *5*, 89–99. [[CrossRef](#)]

32. Sebastian, T. Temperature effects on torque production and efficiency of PM motors using NdFeB magnets. *IEEE Trans. Ind. Appl.* **1995**, *31*, 353–357. [[CrossRef](#)]
33. Raj, C.T.; Srivastava, S.; Agarwal, P. Energy Efficient Control of Three-Phase Induction Motor—A Review. *Int. J. Comput. Electr. Eng.* **2009**, *1*, 61. [[CrossRef](#)]
34. Khammar, F.; Debbache, N. Application of artificial intelligence techniques for the control of the asynchronous machine. *J. Electr. Comput. Eng.* **2016**, *2016*, 8052027. [[CrossRef](#)]
35. ÇELİK, E. Estimation of synchronous motor excitation current using multiple linear regression model optimized by symbiotic organisms search algorithm. *Mugla J. Sci. Technol.* **2018**, *4*, 210–218. [[CrossRef](#)]
36. Guillen, C.E.G.; Cosano, A.M.D.P.; Tian, P.; Diaz, J.C.; Zarzo, A.; Platero, C.A. Synchronous Machines Field Winding Turn-to-Turn fault severity estimation through Machine Learning Regression Algorithms. *IEEE Trans. Energy Convers.* **2022**, *37*, 2227–2235.
37. Bayindir, R.; Colak, I.; Sagiroglu, S.; Kahraman, H.T. Application of adaptive artificial neural network method to model the excitation currents of synchronous motors. In Proceedings of the 2012 11th International Conference on Machine Learning and Applications, Boca Raton, FL, USA, 12–15 December 2012; Volume 2, pp. 498–502.
38. Kirchgässner, W.; Wallscheid, O.; Böcker, J. Data-driven permanent magnet temperature estimation in synchronous motors with supervised machine learning: A benchmark. *IEEE Trans. Energy Convers.* **2021**, *36*, 2059–2067. [[CrossRef](#)]
39. Tahkola, M.; Keränen, J.; Sedov, D.; Far, M.F.; Kortelainen, J. Surrogate modeling of electrical machine torque using artificial neural networks. *IEEE Access* **2020**, *8*, 220027–220045. [[CrossRef](#)]
40. Mukherjee, D.; Chakraborty, S.; Guchhait, P.K.; Bhunia, J. Application of machine learning for speed and torque prediction of pms motor in electric vehicles. In Proceedings of the 2020 IEEE 1st International Conference for Convergence in Engineering (ICCE), Kolkata, India, 5–6 September 2020; pp. 129–133.
41. Traue, A.; Book, G.; Kirchgässner, W.; Wallscheid, O. Toward a reinforcement learning environment toolbox for intelligent electric motor control. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *33*, 919–928. [[CrossRef](#)]
42. Li, Y.; Lei, G.; Bramerdorfer, G.; Peng, S.; Sun, X.; Zhu, J. Machine learning for design optimization of electromagnetic devices: Recent developments and future directions. *Appl. Sci.* **2021**, *11*, 1627. [[CrossRef](#)]
43. Bayindir, R.; Yesilbudak, M.; Colak, I.; Sagiroglu, S. Excitation current forecasting for reactive power compensation in synchronous motors: A data mining approach. In Proceedings of the 2012 11th International Conference on Machine Learning and Applications, Boca Raton, FL, USA, 12–15 December 2012; Volume 2, pp. 521–525.
44. Štumberger, G.; Štumberger, B.; Marčič, T. Magnetically nonlinear dynamic models of synchronous machines and experimental methods for determining their parameters. *Energies* **2019**, *12*, 3519. [[CrossRef](#)]
45. Kron, G. *Equivalent Circuits of Electric Machinery*; Wiley: New York, NY, USA, 1951.
46. Traxler-Samek, G.; Zickermann, R.; Schwery, A. Cooling airflow, losses, and temperatures in large air-cooled synchronous machines. *IEEE Trans. Ind. Electron.* **2009**, *57*, 172–180. [[CrossRef](#)]
47. Dong, Z.; Weili, L.; Feng, Z.; Yunpeng, H. Numerical calculation of air gap magnetic field of a salient synchronous generator with the consideration of the effect of turn insulation. In Proceedings of the International Conference on Power System Technology, Kunming, China, 13–17 October 2002; Volume 2, pp. 774–778.
48. Rossi, C.; Casadei, D.; Pilati, A.; Marano, M. Wound rotor salient pole synchronous machine drive for electric traction. In Proceedings of the Conference Record of the 2006 IEEE Industry Applications Conference Forty-First IAS Annual Meeting, Tampa, FL, USA, 8–12 October 2006; Volume 3, pp. 1235–1241.
49. Lipo, T.A. *Analysis of Synchronous Machines*; CRC Press: New York, NY, USA, 2017.
50. Doherty, R.; Nickle, C. Synchronous machines I—an extension of Blondel’s two-reaction theory. *Trans. Am. Inst. Electr. Eng.* **1926**, *45*, 912–947. [[CrossRef](#)]
51. De Mello, F.; Hannett, L. Representation of saturation in synchronous machines. *IEEE Trans. Power Syst.* **1986**, *1*, 8–14. [[CrossRef](#)]
52. Kahraman, H.T. Metaheuristic linear modeling technique for estimating the excitation current of a synchronous motor. *Turk. J. Electr. Eng. Comput. Sci.* **2014**, *22*, 1637–1652. [[CrossRef](#)]
53. Jayaraman, B.; Mirnalinee, T. Multi Regressor Based User Rating Predictor for ImageCLEF Aware 2022. In Proceedings of the CLEF 2022: Conference and Labs of the Evaluation Forum, Bologna, Italy, 5–8 September 2022.
54. Kramer, O. Scikit-learn. In *Machine Learning for Evolution Strategies*; Springer: London, UK, 2016; pp. 45–53.
55. Al-Gabalawy, M.; Elmetwaly, A.H.; Younis, R.A.; Omar, A.I. Temperature prediction for electric vehicles of permanent magnet synchronous motor using robust machine learning tools. *J. Ambient. Intell. Humaniz. Comput.* **2022**, 1–18. [[CrossRef](#)]
56. Tang, M.; Chen, Y.; Wu, H.; Zhao, Q.; Long, W.; Sheng, V.S.; Yi, J. Cost-sensitive extremely randomized trees algorithm for online fault detection of wind turbine generators. *Front. Energy Res.* **2021**, *9*, 686616. [[CrossRef](#)]
57. Kudelina, K.; Asad, B.; Vaimann, T.; Rassölkin, A.; Kallaste, A.; Khang, H.V. Methods of condition monitoring and fault detection for electrical machines. *Energies* **2021**, *14*, 7459. [[CrossRef](#)]
58. Zou, H.; Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2005**, *67*, 301–320. [[CrossRef](#)]
59. Ferreira, J.G.; Warzecha, A. An application of machine learning approach to fault detection of a synchronous machine. In Proceedings of the 2017 International Symposium on Electrical Machines (SME), Miami, FL, USA, 21–24 May 2017; pp. 1–6.
60. Maulud, D.; Abdulazeez, A.M. A review on linear regression comprehensive in machine learning. *J. Appl. Sci. Technol. Trends* **2020**, *1*, 140–147. [[CrossRef](#)]

61. Hassan, M.K. Optimal power factor of synchronous motors in operating conditions. In Proceedings of the 2022 8th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE), Ruse, Bulgaria, 30 June–2 July 2022; pp. 1–4.
62. Lakshmi, K.; Mahaboob, B.; Rajaiyah, M.; Narayana, C. Ordinary least squares estimation of parameters of linear model. *J. Math. Comput. Sci.* **2021**, *11*, 2015–2030.
63. Segal, M.R. *Machine Learning Benchmarks and Random Forest Regression*; Center for Bioinformatics and Molecular Biostatistics, University of California: San Francisco, CA, USA, 2004.
64. Smith, P.F.; Ganesh, S.; Liu, P. A comparison of random forest regression and multiple linear regression for prediction in neuroscience. *J. Neurosci. Methods* **2013**, *220*, 85–91. [[CrossRef](#)] [[PubMed](#)]
65. Savant, R.; Kumar, A.A.; Ghatak, A. Prediction and analysis of permanent magnet synchronous motor parameters using machine learning algorithms. In Proceedings of the 2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAEECC), Bengaluru, India, 11–12 December 2020; pp. 1–5.
66. Kucukyildiz, G.; Yolacan, E.; Ocak, H.; Aydin, M. Detection of Structural Magnet Defects for Permanent Magnet Synchronous Motors. *IEEE Trans. Energy Convers.* **2021**, *37*, 665–674. [[CrossRef](#)]
67. Fernández-Delgado, M.; Sirsat, M.S.; Cernadas, E.; Alawadi, S.; Barro, S.; Febrero-Bande, M. An extensive experimental survey of regression methods. *Neural Netw.* **2019**, *111*, 11–34. [[CrossRef](#)] [[PubMed](#)]
68. Gonzalez-Abreu, A.D.; Osornio-Rios, R.A.; Jaen-Cuellar, A.Y.; Delgado-Prieto, M.; Antonino-Daviu, J.A.; Karlis, A. Advances in Power Quality Analysis Techniques for Electrical Machines and Drives: A Review. *Energies* **2022**, *15*, 1909. [[CrossRef](#)]
69. Liu, Z.; Zhang, P.; He, S.; Huang, J. A Review of Modeling and Diagnostic Techniques for Eccentricity Fault in Electric Machines. *Energies* **2021**, *14*, 4296. [[CrossRef](#)]
70. Anđelić, N.; Lorencin, I.; Glučina, M.; Car, Z. Mean Phase Voltages and Duty Cycles Estimation of a Three-Phase Inverter in a Drive System Using Machine Learning Algorithms. *Electronics* **2022**, *11*, 2623. [[CrossRef](#)]
71. Goswami, M.; Sabata, P. Evaluation of ML-Based Sentiment Analysis Techniques with Stochastic Gradient Descent and Logistic Regression. *Trends in Wireless Communication and Information Security*; Springer: Singapore, 2021; pp. 153–163.
72. Newton, D.; Yousefian, F.; Pasupathy, R. Stochastic gradient descent: Recent trends. *Recent Advances in Optimization and Modeling of Contemporary Problems*; INFORMS: Catonsville, MD, USA, 2018; pp. 193–220.
73. Xu, X.; Qiao, X.; Zhang, N.; Feng, J.; Wang, X. Review of intelligent fault diagnosis for permanent magnet synchronous motors in electric vehicles. *Adv. Mech. Eng.* **2020**, *12*, 1687814020944323. [[CrossRef](#)]
74. Lang, W.; Hu, Y.; Gong, C.; Zhang, X.; Xu, H.; Deng, J. Artificial Intelligence-based Technique for Fault Detection and Diagnosis of EV Motors: A Review. *IEEE Trans. Transp. Electrif.* **2021**, *8*, 384–406. [[CrossRef](#)]
75. Brereton, R.G.; Lloyd, G.R. Support vector machines for classification and regression. *Analyst* **2010**, *135*, 230–267. [[CrossRef](#)]
76. Çevik, A.; Kurtoglu, A.E.; Bilgehan, M.; Gülşan, M.E.; Albegmprli, H.M. Support vector machines in structural engineering: A review. *J. Civ. Eng. Manag.* **2015**, *21*, 261–281. [[CrossRef](#)]
77. Deng, W.; Zuo, S. Electromagnetic vibration and noise of the permanent-magnet synchronous motors for electric vehicles: An overview. *IEEE Trans. Transp. Electrif.* **2018**, *5*, 59–70. [[CrossRef](#)]
78. Liu, R.; Yang, B.; Zio, E.; Chen, X. Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mech. Syst. Signal Process.* **2018**, *108*, 33–47. [[CrossRef](#)]
79. Sobbouhi, A.R.; Vahedi, A. Transient stability prediction of power system; a review on methods, classification and considerations. *Electr. Power Syst. Res.* **2021**, *190*, 106853. [[CrossRef](#)]
80. Del Angel, A.; Geurts, P.; Ernst, D.; Glavic, M.; Wehenkel, L. Estimation of rotor angles of synchronous machines using artificial neural networks and local PMU-based quantities. *Neurocomputing* **2007**, *70*, 2668–2678. [[CrossRef](#)]
81. Memon, A.P.; Memon, A.S.; Akhund, A.A.; Memon, R.H. Multilayer perceptrons neural network automatic voltage regulator with applicability and improvement in power system transient stability. *Int. J. Emerg. Trends Electr. Electron.* **2013**, *9*, 30–38.
82. Car, Z.; Baressi Šegota, S.; Anđelić, N.; Lorencin, I.; Mrzljak, V. Modeling the spread of COVID-19 infection using a multilayer perceptron. *Comput. Math. Methods Med.* **2020**, *2020*, 5714714. [[CrossRef](#)]
83. Azmi, S.S.; Baliga, S. An Overview of Boosting Decision Tree Algorithms utilizing AdaBoost and XGBoost Boosting strategies. *Int. Res. J. Eng. Technol.* **2020**, *7*, 529.
84. Kadiyala, A.; Kumar, A. Applications of python to evaluate the performance of decision tree-based boosting algorithms. *Environ. Prog. Sustain. Energy* **2018**, *37*, 618–623. [[CrossRef](#)]

**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.