Classification and Severity Estimation of Eccentricity Faults in Salient Pole Synchronous Machine using Deep Learning

Latifa Yusuf, *Graduate Student Member, IEEE*, Ashwin Shejwalkar, Belaid Moa, and T. Ilamparithi, *Member, IEEE*

***Abstract*—The presented research work is focused on the classification and severity estimation of eccentricity faults in Salient Pole Synchronous Machines. Building on our comparative study of Artificial Neural Network and Convolutional Neural Network for eccentricity fault classification, we propose an end-to-end deep learning model, namely Hierarchical Convolutional Neural Network, for eccentricity classification and severity estimation. The deep learning model inherently consists of an eccentricity detection component for fault classification and a severity estimation component for fault quantification. The deep learning model is built using the experimental data of a 3-phase, 2-kW, salient pole synchronous machine. The machine is subjected to 20%, 40%, and 60% severities of static and dynamic eccentricity faults under different loading conditions. Stator line currents and line-to-line voltages obtained from different operating conditions are used to train, validate and test the proposed model. To enhance the model's performance, time delay construction was incorporated to augment the datasets and carefully evaluate the impact of selected raw input features, specifically stator currents and voltages, as well as the load. Among the evaluated scenarios, the use of voltage with time delay (V, TD) as input features produced the best results, achieving 100% classification accuracy and a root mean square error of 0.0046 for static eccentricity and 0.0188 for dynamic eccentricity estimation. Results indicate that the model performs excellently in fault classification and severity estimation. Compared to traditional machine learning models, the presented model is an end-to-end deep learning architecture devoid of manual feature extraction and is robust to load variations.**

***Index Terms*—Eccentricity, Faults, Hierarchical Convolutional Neural Network, Machine Learning, Salient Pole Synchronous Machine**

I. Introduction

SALIENT pole synchronous machines (SPSM) are used in plants and industries for power factor correction, voltage regulation, and high-starting torque applications. SPSMs can be affected by different types of faults. Common ones include bearing faults, broken rotor bars, eccentricity faults, and stator winding insulation breakdowns [1]. Eccentricity can occur when there is a misalignment in the rotor axis compared to that of the stator. This misalignment can be caused by various factors such as shaft bending, imprecise rotor positioning, worn-out bearings, movement in the stator core, or a deformed rotor shaft. There are two types of eccentricity faults: static and dynamic, as shown in Fig. 1 [2]. In static eccentricity (*SE*), the displacement between the rotor and stator remains constant over time, and the rotor's center line is aligned with its rotational axis (the rotor's center O_r and the center of rotation O_w coincide). *SE*, therefore, produces a non-uniform air gap between the rotor and stator that stays fixed over time. Conversely, dynamic eccentricity (*DE*) occurs when the rotor's center line doesn't match the rotational axis (the rotor's center O_r and the center of rotation O_w are different), causing the non-uniform air gap to rotate with the rotor [3]. Mixed eccentricity is a combination of both scenarios.

When eccentricity faults occur, depending on their severity, it is crucial that the rotor, bearing, and shaft inspection is carried out immediately by the maintenance crew. If this is not done promptly, considerable damage can occur, leading to undesirable outcomes. Most plants have a scheduled maintenance routine for their electrical machinery to avoid economic losses and efficiency reductions [4]. However, faults can happen at any time, which makes condition monitoring a very intricate part of maintenance. Newly manufactured SPSMs are inherently static eccentric (usually below 10%) because of their miniaturized air gap to stator bore diameter ratio. This makes it extremely difficult to achieve perfect rotor alignment within the stator's axis [5]. A considerable increase in electromagnetic, radial, and tangential forces are some of the outcomes of the effects of eccentricity on synchronous machines. These outcomes can lead to vibrations and oscillations, thereby reducing efficiency [6].

Generally, invasive and non-invasive techniques are the two main categories of eccentricity fault detection that have been documented in the literature over the last few decades. Search coils are the most common invasive technique used to detect eccentricity faults. They can be placed internally to detect magnetic flux changes in the stator of synchronous and induction machines. These flux changes can occur if the rotor is not centered correctly. The coils develop voltage signals from these flux changes, indicating the presence of SE or DE [7]–[10]. One of the major shortfalls of this kind of invasive technique is that it involves physically inserting or mounting the coil within the machine's stator or rotor structure. This process requires dismantling or partially disassembling the machine and may require specialized equipment, expertise, and additional costs. Multiple sensor techniques have been explored for detecting eccentricity faults in electrical motors. Accelerometers, hall-effect sensors, flux sensors, vibration sensors, and acoustic sensors have been used to detect and estimate static eccentricity faults under varying load conditions [11]–[13]. However, These sensors may produce inaccurate readings due to their susceptibility to environmental conditions

[14].

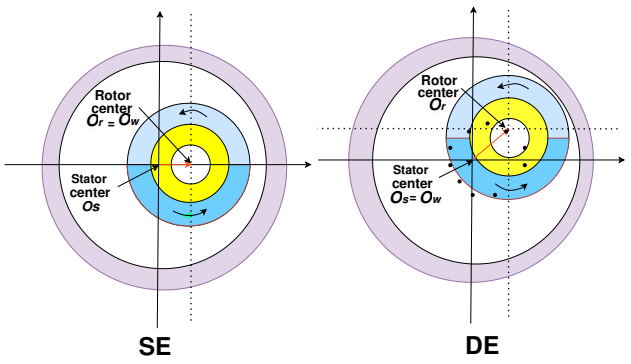

Fig. 1. A schematic illustrating static and dynamic eccentricity.

The non-invasive techniques include vibration analysis, motor current signature analysis (MCSA), and machine learning algorithms. Authors in [15]–[18] used vibration measurements and harmonic analysis to generate an eccentricity index to discriminate between SE and DE. The MCSA method relies on distinct harmonic components in the machine's line currents to detect faults [19]. Fault-specific frequency components synonymous with static, dynamic, and mixed eccentricity faults were analyzed in a reluctance synchronous machine [20]. Frequency spectral analyses such as Fast Fourier Transform (FFT) and Discrete Wavelet Transform (DWT) have also been combined with MCSA to detect eccentricity faults [21]–[23]. Some of the drawbacks of MCSA include the effect of noise, load change, and vibration on the line current harmonic components. There is also the issue of other motor faults generating similar signatures [24]–[26].

The application of machine learning (ML) algorithms for eccentricity detection has been explored in recent years. ML eliminates the impracticality of model-based methods in real-life scenarios and relies on data-driven intelligence-based approaches [27]. It can be categorized as supervised or unsupervised. Authors in [28] presented a study that showcased an analysis encompassing diverse time and frequency metrics derived from vibration signals of a Permanent Magnet Synchronous Machine (PMSM). It was concluded that time and frequency-domains signal processing techniques can be applied to extract crucial feature sets related to healthy and faulty cases in a machine. [29], [30] have used Decision Tree (DT) and k-Nearest Neighbor (k-NN) to diagnose SE using motor current and vibration signals. An accuracy of 58.4% and 100% were documented, respectively. Principal Component Analysis (PCA) reduces the dimensionality of a dataset by converting correlated variables into a smaller set of uncorrelated variables using a linear transformation. PCA was used in [31] to isolate DE in an SPSM. Meaningful features were extracted from the stator current irrespective of the machine's loading condition.

A range of studies have successfully utilized Artificial Neural Networks (ANN) to detect eccentricity faults in electrical machines. In fact, [32], [33] used discriminative features in the line current spectrum to train an ANN to classify and detect SE in induction motors. [34] considered using ANN in variant operating conditions in a closed-loop drive-connected induction motor. [35] used ANN to detect and estimate the severity of eccentricity faults in Line Start Permanent Magnet Synchronous Machine (LSPMSM). PCA was used to extract principal components from the stator line currents. Then, the network was trained on specific ranges of *SE*, and an accuracy of 95%-98% was achieved. The work in [36] showed how to monitor and detect eccentric faults in a synchronous reluctance motor from vibration signals using ANN.

The accuracy of a machine learning model for eccentricity classification and estimation relies heavily on high-quality input data and accurate feature selection. Manual feature engineering techniques must be carefully designed to build and train accurate machine learning models successfully [37]. End-to-end deep learning, however, offers significant benefits for classifying and estimating eccentricity faults without resorting to high-quality data or manual feature extraction. It can achieve high accuracy by automatically extracting features and learning complex patterns from large datasets, eliminating the need for manual feature mining [38], [39]. Convolutional Neural Networks (CNN) are the most common deep learning models used for fault classification and diagnosis, especially for potentially noisy time series data. [40]–[42] have demonstrated the potential of hierarchical CNN (HCNN) for fault detection in electrical machines. [43] introduces a Dislocated Time Series CNN (DTS-CNN) for intelligent fault diagnosis in electrical machines, particularly in industrial applications. The results obtained showed that the method was effective in both constant and non-stationary conditions in mechanical systems.

Researchers in [44] also used an HCNN for event classification on phasor measurement units (PMUs). The results obtained showed a higher performance than traditional classification models. [45] proposes a two-level hierarchical diagnosis network based on an HCNN for fault diagnosis. This HCNN models both fault patterns and fault severity as a hierarchical structure, allowing simultaneous estimation of both. The network design includes two classifiers and a two-stage training strategy to optimize these classifiers at once. [37] developed a Feature Inherited Hierarchical CNN (FIHCNN) for induction motor classification and severity estimation leveraging latent features using stator currents signals. Even though the model performed well in classifying and estimating the severity of the faults (eccentricity, broken rotor bar and unbalanced rotor), the paper was not explicit on the type of eccentricity and did not handle quantification of different kinds of eccentricities. This information is very important because we know that SE and DE exhibit different features in the stator current signal [20], which means that the model is not a size fit for all in terms of eccentricity. In addition, knowing the type of eccentricity is very essential when it comes to maintenance.

In [46], the authors performed a comparative study of ANN and CNN for the classification of SE and DE in SPSM. The study considered varying degrees of severity, unbalanced source voltage, and load conditions. Stator line currents $\mathbf{I_s}$ and field currents $\mathbf{I_r}$, as well as their combination, were used as inputs into both neural networks. The findings from the

study showed that CNNs outperformed ANNs. Therefore, we propose extending that work with the objective of estimating the severity of SE and DE using an end-to-end hierarchical convolutional network (HCNN). By doing so, we eliminate the need for invasive approaches, advanced feature engineering, and expensive sensors. Given that CNNs are resistant to noise (due to their convolutional layers), our architecture is suitable for real-life scenarios where data is greatly affected by noise, transient dynamics, and load and voltage imbalances. Moreover, in industrial applications where more than one type of fault can occur simultaneously in a synchronous machine, our model can easily be extended and trained to classify and quantify several faults simultaneously. By design, SPSMs have non-constant air gaps between the rotor and the stator, so their *SE* and *DE* air gaps are usually difficult to detect, let alone quantify. As such, and without loss of generality, we limit the testing of our model to SPSM SE and DE detection and quantification.

II. Experimental Setup and Data Collection for HCNN

A. Experimental Setup

The experimental setup for this study is shown in Fig. 2. A 2kW 3-phase SPSM (denoted by **M**) was used for the experiment. Details of the SPSM are shown in Appendix A. To introduce eccentricity in the SPSM, several modifications were made to the original setup of **M**. The original sheet metal enclosures were replaced with rugged ones that allowed air-gap measurements using a feeler gauge. *SE* was introduced by affixing specially cut bushings to the end plates and placing them between the bearings and the end plates on both sides of the rotor. The alignment of the eccentric points was carefully maintained to avoid inclined eccentricity, and the air gap was measured around the rotor periphery to ensure accuracy.

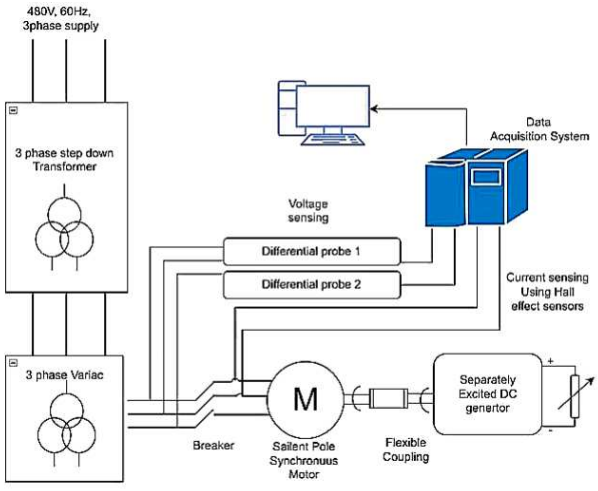

Fig. 2. The experimental setup without showing the field current sensor

DE was implemented by mounting eccentric sleeves on the shaft, requiring the replacement of the original bearings with larger ones to fit the *DE* sleeves. Inner concentric bushings were replaced with eccentric ones, while outer bushings remained concentric. The inner sleeves were shrunk to fit onto the shaft to prevent slipping. *SE* and *DE* bushings with offsets of 0%, 20%, 40%, and 60% of the air-gap length were used, along with new bearings ($6208R$ SR, 40 mm x 80 mm x 18 mm) to perform *SE* and *DE*. This is shown in Fig. 3.

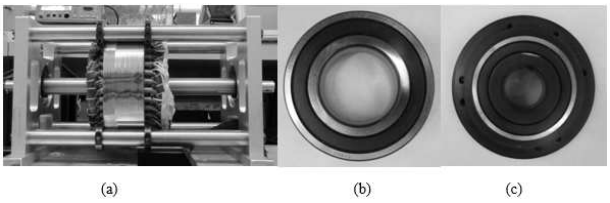

Fig. 3. An SPSM with a. modified enclosure b. bearing and modified 60% *SE* bushing. c. bearing and modified 60% *DE* bushings

A 208V, 60Hz 3-phase auto-transformer powered the SPSM. For experimental purposes, the machine's steady state line currents, as well as line-to-line voltages, were monitored and recorded using the National Instruments DAQ box. This data collection was conducted for 13 seconds for data training and validation, and 2 seconds for testing, with a sampling rate of 3600 Hz, ensuring precise and detailed capture of the current and the voltage waveforms. The SPSM was operated under various loading conditions, including no-load, 25% full load, 50% full load, 75% full load, and 100% full load.

B. Data Acquisition and Preprocessing

For each of the load conditions we mentioned above, data on the machine's stator line currents $\mathbf{I_s}$ and the voltages $\mathbf{V_s}$ between the phases of the machine were collected to establish a baseline for healthy operation. Afterward, data for 20%, 40% and 60% of *SE* and *DE* were collected for each loading condition. A total of $54,000$ samples were collected for each combination of fault and loading conditions for the end-to-end deep learning analysis. A summary of the data collected from the DAQ is shown in Table I.

TABLE I
Data used for Analysis

SPSM Condition	Load Condition
Healthy (**HL**)	25%, 50%, 75%, full load-(FL), no load (NL)
20% DE (**20DE**)	”
20% SE (**20SE**)	”
40% DE (**40DE**)	”
40% SE (**40SE**)	”
60% DE (**60DE**)	”
60% SE (**60SE**)	”

While our goal is to develop end-to-end deep learning models that require minimal preprocessing, selecting the right raw features to input into the models remains crucial. In our context, features such as stator line-to-line voltages $\mathbf{V_s} = (\mathbf{V_{ab}}, \mathbf{V_{ac}})$, stator line currents $\mathbf{I_s} = (\mathbf{I_a}, \mathbf{I_b})$, and load L directly influence model performance and predictive capability, and we, therefore, need to carefully evaluate which ones to

use as inputs to our model. Moreover, to build effective deep learning models, larger and well-balanced training data are required. For that, we rely on the time delay (TD) strategy to augment our data. TD is used in time series analysis and other data-driven approaches and involves generating time-delayed sequences to produce more overlapping sequences. To evaluate the importance of different raw features and the effect of TD, we considered the following different scenarios in our preprocessing step:

- **Scenario (I, L)**: $\mathbf{I_s}$ and L - the two stator line currents I_a and I_b, and the load L are used as inputs. Treated as time series, both currents were partitioned into sequences of size 256 and fed into our model.
- **Scenario (I, L, TD)**: $\mathbf{I_s}$ and L with time delays - the tuple (I_a, I_b, L) is used as an input to the model. The corresponding data is generated using a rolling window of size 256 and with delays of $(9, 9, 3)$. The delay $(9, 9, 3)$ means that there is a 9-sample delay for DE and SE fault condition data and a 3-sample delay for healthy data. This scenario is mainly to gauge the importance of using time delay construction to improve the performance of our models.
- **Scenario (I, TD)**: $\mathbf{I_s}$ with time delays. This scenario is similar to the above, except that we are not using the load L. This scenario allows us to validate the importance of including the load as part of the raw features.
- **Scenario (V, TD)**: $\mathbf{V_s}$ with time delay. The voltages V_{ab} and V_{ac} are used as inputs. This scenario enables us to compare the use of voltage compared to the current and the load. This scenario is also instrumental for setups monitoring the voltages only.
- **Scenario (V)**: $\mathbf{V_s}$ without time delay. This scenario is similar to the previous one but does not incorporate time delays. This scenario allows us to assess the significance of time delays in training models when using voltage inputs.

As we mentioned above, 13 seconds of data, equivalent to 46,800 samples and representing 87% of the total data set collected, were used for training and validation, while the remaining 2 seconds, equivalent to 13% of the total dataset (equivalent to 7,200 samples), were kept aside exclusively for testing. The performance reported is only for testing data that we did not use during training. No prior normalization of data is required because a batch normalization layer is a part of the end-to-end model itself (HCNN architecture). To have a visual glimpse of the input feature data used for our model and analysis, Fig. 4 shows the current lines and the voltages corresponding to no and full load under 40% DE. All analyses were carried out using MATLAB on the Arbutus cloud and the Cedar HPC cluster provided by the Digital Research Alliance of Canada (alliancecan.ca).

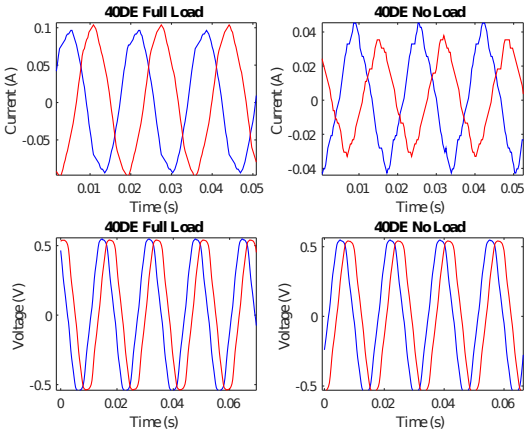

Fig. 4. Current lines $\mathbf{I_s}$ and line-to-line voltages $\mathbf{V_s}$ for no load and full load under 40% DE.

III. HCNN for Eccentricity Classification and Severity Estimation

In our previous work [46], the CNN model performed better than all the ANNs for eccentricity classification; hence, as an extension of that study, an HCNN model for better detection and severity estimation is proposed. HCNN model follows a hierarchical architecture to refine its classification and leverages probabilistic methods to integrate predictions from coarse and fine categories to achieve better accuracy and performance [47]. They enhance the basic CNN architecture by incorporating multiple levels of convolutional and pooling layers; each intended to capture increasingly abstract and complex features from the input data. The features learned by earlier layers of the network are subsequently passed on to the next layers, encouraging the network to learn more intricate features [37], [47].

These capabilities make HCNN a very suitable architecture to both detect and quantify faults. In our context and by design, the HCNN's *coarse* component can handle the classification of the different SPSM conditions, namely *HL* (for healthy), *SE* (for static eccentricity) and *DE* (for dynamic eccentricity). In contrast, the *fine* component is responsible for estimating the severity associated with the detected fault. Fig. 5 depicts the proposed HCNN for eccentricity classification and severity estimation. The first number in the square brackets in the figure represents the window size, while the second number represents the number of filters. As the figure shows, our HCNN is composed of three CNNs, which are divided into two primary components: the Eccentricity Detection (*ED*) component and the Severity Estimation (*SEV*) component. For this study, the experimental data collected, as shown in Table I, was used to train, validate and test the performance for both components of our HCNN models. Fig. 6 shows the HCNN workflow.

A. Eccentricity Detection (ED Component)

Depending on the scenarios discussed above, the appropriate raw features are fed into a batch normalization followed by the rest of the *ED* component, which consists of three convolutional layers, three normalization layers, two max pooling layers, one average pooling layer, and one fully connected layer. The *ED* component is responsible for the

This article has been accepted for publication in IEEE Transactions on Industry Applications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIA.2025.3546202

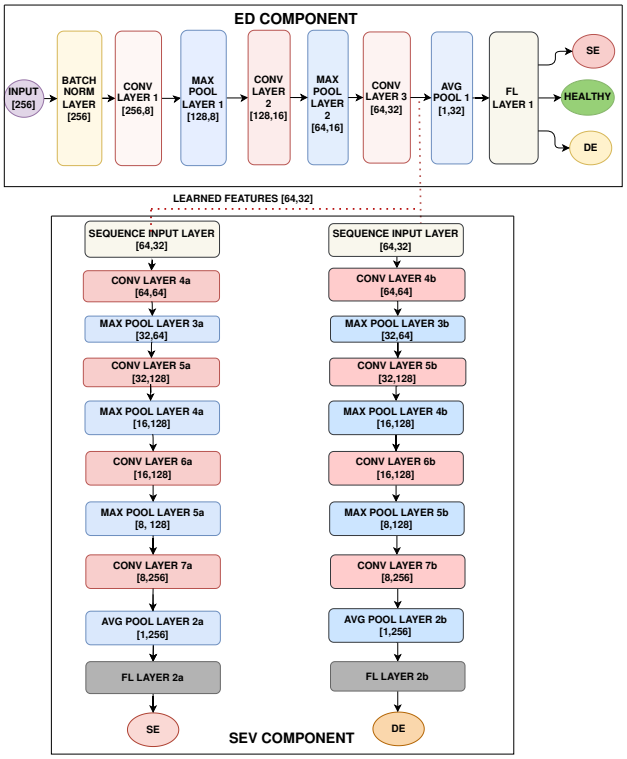

Fig. 5. Proposed HCNN for Eccentricity Classification and Severity

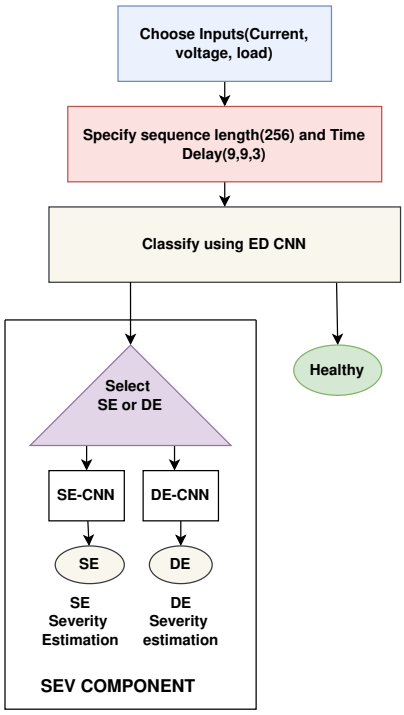

Fig. 6. HCNN Workflow

classification of the SPSM as *HL*, *SE* or *DE*. The convolutional layers have filters of size 8. The strides in the max pooling layers are set to 2, while that of the convolutional layers is set to 1. Subsequent convolutional layers have progressively increasing filter numbers (8, 16 and 32) to capture more complex features from the inputs. After each convolutional layer, an Exponential Linear Unit (ELU) layer is used as our nonlinear transformation. The ELU layer, as shown in (1), performs a threshold operation on each element of the input, where any value less than zero is exponentially reduced and kept negative.

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ \exp(x) - 1 & \text{if } x < 0 \end{cases} \tag{1}$$

The normalization layer follows each ELU layer and is employed to standardize a mini-batch of data across all channels for each observation separately. After the third convolutional layer, the normalization layer is followed by a global average pooling layer. It is used to reduce each feature map to a single value by averaging, which helps prevent any potential overfitting and reduces the number of parameters in the model. Finally, a fully connected layer followed by a softmax layer is used to compute the probability associated with each class *HL*, *SE* and *DE*. The *ED* component is trained using ADAM optimizer with a batch size of 200, an initial learning rate of 0.001, an L2 regularization of 0.0001 and a maximum of 1000 epochs. The regularized cross-entropy, $\mathcal{L}_{ED}$, between network predictions and target values was used as our loss function, and it is analytically computed as in (2).

$$\mathcal{L}_{ED} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{K} y_{i,k} log(\hat{y}_{i,k}) + \lambda \sum_{j=1}^{p} w_j^2, \tag{2}$$

where N is the total number of sequences in a batch, n is the sequence index ranging from 1 to N, K is the number of classes (3 in our case - *HL*, *SE*, and *DE*), $y_{i,k}$ is a binary indicator (0 or 1) if class label k is the correct classification for sequence i, $\hat{y}_{i,k}$ is the predicted probability that the sequence i belongs to class k, λ is the regularization parameter, w_j are the weights of the model, and p is the total number of weights.

In all the scenarios, we kept track of the loss function (2) for both the training and the validation data, and confirmed that the model has indeed converged within the allocated 1000 epochs.

B. Severity Estimation (SEV Component)

The *SEV* component comprises two CNNs, each responsible for estimating the severity of *SE* and *DE* depending on the predicate class obtained from the *ED* component. The learned features in the third convolutional layer of the *ED* component are passed on to the two CNNs in the *SEV* component. The *activations* function in MATLAB is used to extract the intermediate feature maps from the third convolutional layer of the *ED* component. Fig 7 shows a t-distributed Stochastic

Neighbor Embedding (t-SNE) of I_s to visualize the feature map of both *SE* and *DE* just before the average pooling layer of the *ED* component for scenario (I, L).

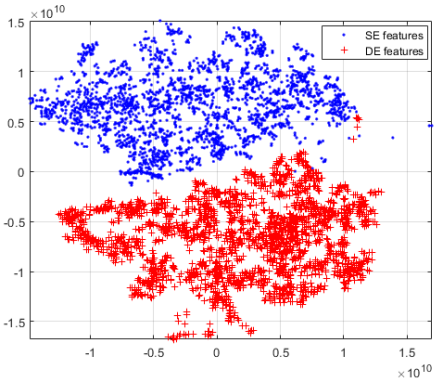

Fig. 7. t-SNE feature map for Scenario (I,L)

In the *SEV* component, each of the CNNs has four convolutional layers, three max pooling layers, one average pooling layer, one fully connected layer, and one regression layer, which is responsible for severity estimation. The number of filters in the convolutional layers are 64, 128, 128, 256, and 512. The filter size, stride, initial learning rate, batch size, and L2 regularization used for both CNNs remain the same as those used for the CNN in the *ED* component. Both *SEV* CNNs were trained for at least 3000 epochs, and the loss function is the MSE error given in (3).

The root mean square propagation (RMSProp) was used to optimize the loss function algorithm.

$$\mathcal{L}_{SEV} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{p} w_j^2, \tag{3}$$

where N is the total number of sequences in a batch, n is the sample index ranging from 1 to N, K is the number of classes, y_i is the actual value of the severity associated with sequence i, $\hat{y}_i$ is the predicted severity for sequence i, λ is the regularization parameter, w_j are the weights, and p is the total number of weights in the model.

In all the scenarios, we kept track of the loss function (3) for both the training and the validation data, and confirmed that the model has indeed converged within the allocated epochs. In fact, the TD based-model required 5000 epochs to converge while the others only 3000 epochs. Table II shows the hyperparameters used to train the *ED* and *SEV* components of the HCNN, respectively.

TABLE II
HYPERPARAMETERS USED FOR HCNN

Hyperparameter	*ED* **Component**	*SEV* **Component**
Batch size	200	200
No of Epochs	1000	3000
L2 Regularization	0.0001	0.00001
Learning rate	0.001	0.0001
Training algorithm	Adam	RMSProp
Dropout	-	0.1

IV. Results and Discussion

A. Eccentricity Detection and Classification

Table III summarizes the classification results of the *ED* component for the different scenarios. All the metrics in the table can be computed from the confusion matrix. For that, let us assume that the confusion matrix has the following form:

	Predicted *SE*	Predicted *DE*	Predicted *HL*
Actual *SE*	TP_1	FN_{12}	FN_{13}
Actual *DE*	FP_{21}	TP_2	FN_{23}
Actual *HL*	FP_{31}	FP_{32}	TP_3

where:

- TP_i: True Positives for class i
- FP_i: False Positives for class i (sum of instances incorrectly predicted as class i):

$$FP_i = \sum_{j\neq i} FP_{ji}$$

- FN_{ij}: False Negatives for class i when the actual class is j
- TN_i: True Negatives for class i (sum of true predictions for all classes except class i) :

$$TN_i = \sum_{j\neq i} TP_j + \sum_{j\neq i} FN_{ji}$$

The accuracy is a global metric is:

$$\text{Accuracy} = \frac{TP_1 + TP_2 + TP_3}{TP_1 + TP_2 + TP_3 + FP_1 + FP_2 + FP_3}$$

For the rest of the metrics, we first define the metrics associated with each fault class i as:

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i}$$

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_{ij}}$$

$$\text{Specificity}_i = \frac{TN_i}{TN_i + FP_i}$$

$$\text{F1Score}_i = \frac{2\times\ \text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}$$

We then take the averages across the classes to obtain the expressions for the final, *macro*, metrics used to gauge the performance of the models:

$$\text{Precision} = \frac{1}{3}\sum_{i=1}^{3}\text{Precision}_i$$

$$\text{Recall} = \frac{1}{3}\sum_{i=1}^{3}\text{Recall}_i$$

$$\text{Specificity} = \frac{1}{3}\sum_{i=1}^{3}\text{Specificity}_i$$

$$\text{F1Score} = \frac{1}{3}\sum_{i=1}^{3}\text{F1Score}_i$$

TABLE III
ED PERFORMANCE FOR THE DIFFERENT SCENARIOS IN PERCENTAGE

Scenario	Accuracy	Recall	Precision	Specificity	F1Score
(I, L)	99.91	99.91	99.75	99.95	99.87
(I, L, TD)	100	100	100	100	100
(I)	99.97	99.92	99.99	99.99	99.95
(V)	99.95	99.92	99.98	99.98	99.98
(V, TD)	100	100	100	100	100

From the classification performance table, we can easily see that *ED* component was able to achieve excellent performance in all scenarios. These results are in line with what we reported before regarding CNN and are better than those of ANN [46]. In fact, the overall accuracy of the different ANN models ranged from 94% to 99%, while CNN achieved beyond 99%. The novelty of HCNN, however, shines when we want to quantify *DE* and *SE*, as discussed next.

B. Eccentricity Severity Estimation

As opposed to the ANN and CNN models presented in [46], which can only classify SE and DE types of eccentricity faults, our HCNN model can estimate and quantify their severity. Depending on the eccentricity fault predicted by the *ED* component, the proper *SEV* component is called to estimate its associated severity. The metric used to quantify the performance of the *SEV* output is RMSE, which is defined in (15):

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=i}^{N}(O_i - \hat{O}_i)^2}, \tag{4}$$

where N is the number of sequences in the test dataset, O_i is the actual severity of the eccentricity under consideration, and $\hat{O}_i$ is the estimated severity from the associated *SEV* component.

TABLE IV
PERFORMANCE OF *SEV* COMPONENT FOR *DE* AND *SE* SEVERITY ESTIMATION

Scenario	*SE* RMSE	*DE* RMSE
(I, L)	0.0575	0.0218
(I, L, TD)	0.0406	0.0352
(I)	0.0321	0.0418
(V)	0.0587	0.0610
(V, TD)	0.0046	0.0188

As an illustration, the confusion chart for (I, L) and (V, TD) are shown side to side in Fig 8. Their associated *SE* and *DE* scatter plots and box plots for their severity are depicted in Fig 9 and Fig 10.

Fig 9 shows the scatter plots for the testing of the HCNN's SEV component of (I, L) and (V, TD) for SE and DE, respectively. These scatter plots visually compare the predicted severity values to the actual severity labels. The closer the blue dots are to the red bars (representing the actual severity levels), the more accurate the predictions are. Each blue dot represents a predicted severity value for a sample that was identified as being *SE* or *DE* by the *ED* component and passed to the associated *SEV* CNN.

Fig 10 shows the box plots for the severity estimates of (I, L) and (V, TD) for SE and DE, respectively. Additionally, Table V and Table VI present the predicted severity for the median, 25th percentile (Q1) and 75th percentile (Q3) of the box plot for both (I, L) and (V, TD).

TABLE V
SUMMARY OF PREDICTED VALUES OF THE BOX PLOT FOR (I, L)

Severity Label	Q1	Median	Q3
20SE	18.68	19.16	19.61
40SE	39.56	40.31	41.13
60SE	59.72	61.40	62.35
20DE	18.73	18.46	18.98
40DE	40.11	40.69	40.95
60DE	61.40	60.23	61.86

TABLE VI
SUMMARY OF PREDICTED VALUES OF THE BOX PLOT FOR (V, TD)

Severity Label	Q1	Median	Q3
20SE	19.96	19.95	20.01
40SE	39.96	39.94	39.99
60SE	59.68	59.99	60.02
20DE	19.82	19.99	20.01
40DE	39.86	39.94	40.07
60DE	59.98	59.98	60.01

The box plot effectively visualizes the variance between predictions and actual values. The size of each box (the interquartile range, or IQR) represents the middle 50% of the predicted values, indicating the variability in the predictions. The small, almost flat IQR boxes suggest minimal variability, reflecting the model's high precision in predictions. Also, the tight clustering of the majority of the data points around the true values suggests that the model is almost accurate, effectively capturing the true severity levels with minimal prediction error. The ranges for correctly predicted labels are defined as $15\% - 25\%$ for 20SE or 20DE, $35\% - 45\%$ for 40SE or 40DE, and $55\% - 65\%$ for 60SE or 60DE. This means that if the predicted severity falls within these ranges around the actual severity level, the label is set to 1; otherwise, it is set to 0. These ranges were chosen based on practical considerations in eccentricity fault estimation, where identifying the presence and approximate severity of a fault is more critical than obtaining exact values. This approach ensures the model detects significant faults, triggering maintenance when necessary, without being overly sensitive to minor deviations. Narrowing these ranges would improve precision but could reduce recall, leading to fewer correct predictions due to stricter criteria.

The RMSE results presented in Table IV show that using voltage as input, along with training data subdivided using the time delay strategy, yielded better outcomes (the lowest RMSE value for *SE* and *DE*) than the other scenarios, especially for estimating the severity of *SE*. When using current and load, we obtain slightly more accurate estimates for *DE* severity, while other scenarios provided better estimates for *SE*. Due to the inherent randomness in the training and optimization of deep models, we repeated the runs several times, consistently reaching the same conclusions.

This article has been accepted for publication in IEEE Transactions on Industry Applications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIA.2025.3546202

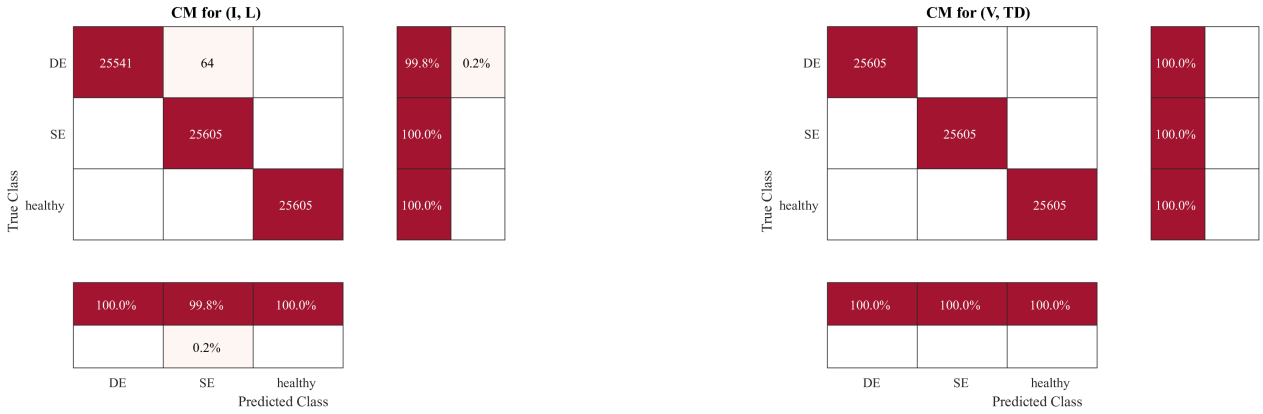

Fig. 8. Confusion charts for (I,L) scenario, left column, and for (V,TD) scenario, right column.

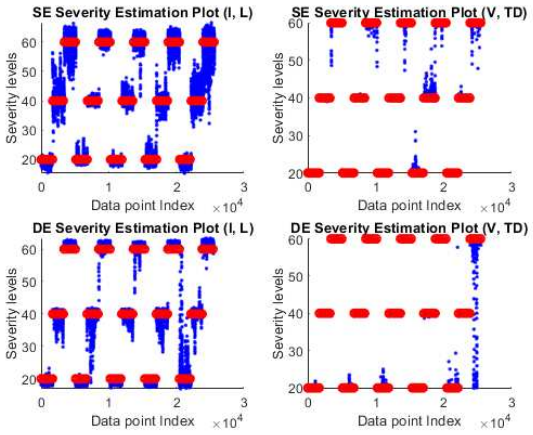

Fig. 9. Severity estimates of (I, L) and (V, TD) for *SE* and *DE*.

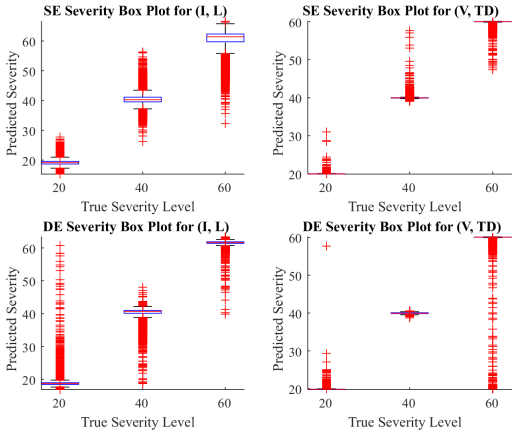

Fig. 10. Box plots for the severity estimates of (I, L) and (V, TD) for *SE* and *DE*.

Using time delay when training HCNN, especially when voltage is used as input, is essential for better performance of the *SEV* component. Incorporating time-delayed sequences not only allows the model to capture the dynamic behavior of voltage signals more effectively but also augments the training dataset, enabling us to train deep architectures effectively. This data augmentation technique can also help address class imbalance by enhancing the representation of less frequent fault conditions. Specifically, our time delay configuration of $(9, 9, 3)$ was chosen to represent better the healthy class, which was underrepresented in the original dataset. By considering many time-delayed sequences, the model is able to capture important dynamics across them and better identify patterns associated with eccentricity faults. This approach enhances sensitivity to subtle changes in the machine's performance, enabling a more robust quantification of eccentricity severity while avoiding overfitting.

It is important to highlight that the (V, TD) scenario yields superior results, and a comprehensive argument—deferred to future work—is needed to demonstrate fully why voltage outperforms current and load when quantifying eccentricity faults. A possible reason can be that the line voltage interacts with the machine's internal impedance Z, which changes due to faults like eccentricity. The total impedance of a machine is given as

$$Z = \sqrt{R^2 + (X_L)^2} \tag{5}$$

where X_L is the inductive reactance and depends on the inductance L, which is affected by the air gap variations caused by eccentricity. The Inductive reactance is:

$$X_L = 2\pi f L \tag{6}$$

As eccentricity alters the air gap, it changes the inductance L, leading to variations in impedance. These impedance changes affect how the machine absorbs and reacts to the line voltage. The relationship between voltage, current, and inductance is described as

$$V = L\frac{dI}{dt} + I\frac{dL}{dt} \tag{7}$$

The resulting impedance variations influence the motor's behavior, and these subtle fluctuations are encoded in the voltage signal over time.

Furthermore, during the tuning of the number of layers in the classification stage, we observed an intriguing phenomenon: increasing the depth of the *ED* component by adding more convolutional layers resulted in a decline in the performance of the *SEV* component on both the training and validation data (test data was excluded during this hyperparameter tuning phase). To explore this further, we analyzed the features generated after each convolutional layer and assessed their impact on the *SEV* component in the (I, L, TD) scenario. This was achieved by changing the layers in which the features were extracted and fed into the *SEV* component of the HCNN.

From Table VII *Deep HCNN Layer 17* indicates that the *ED* component's CNN architecture consists of 17 layers, with features extracted from the 17th layer for the *SEV* component. Similarly, *Deep HCNN Layer 13* refers to the same 17-layer *ED* component but with features extracted from the 13th layer. In the case of *Deep HCNN Layer 9*, the *ED* component still has 17 layers, but the features are extracted from the 9th layer. In contrast, for *HCNN Layer 13*, the *ED* component comprises 13 layers, with features being extracted from the 13th layer for use in the *SEV* component. For *HCNN Layer 9*, the ED component still consists of 13 layers, but features are drawn from the 9th layer. Similarly, *HCNN Layer 5* denotes the same 13-layer ED structure, but features are sourced from the 5th layer.

TABLE VII
RMSE VALUES FOR (I, L, TD) USING DIFFERENT HCNN LAYERS

HCNN Layer	*SE* RMSE	*DE* RMSE
HCNN Layer 13	0.0245	0.0200
HCNN Layer 9	0.0340	0.0385
HCNN Layer 5	0.0388	0.0290
Deep HCNN Layer 17	0.0786	0.0767
Deep HCNN Layer 13	0.0230	0.0286
Deep HCNN Layer 9	0.0096	0.0132

The findings are presented in Table VII. The results obtained show that, with a deeper *ED* component, the *SEV* component derives greater benefits from the early features extracted from *ED*. This phenomenon may be attributed to the fact that as the *ED* architecture deepens, the features in the later layers become increasingly specialized for the classification task, potentially losing essential features for the quantification task. While the final layer may effectively classify faults, it may struggle with quantifying their severity, as important patterns necessary for accurate quantification might have been forgotten within the deeper architecture.

V. HCNN GENERALIZATION AND APPLICABILITY

To further gauge the applicability of our HCNN models in other circumstances and check how well they generalize, we tested the model on an induction motor and on the mixed eccentricity *ME* faults (an eccentricity fault condition in which both *SE* and *DE* occurring simultaneously) in an SPSM.

A. HCNN for Induction Motors

An induction motor (IM) was used to evaluate the applicability of our HCNN model to other electric machines. Details of the IM used are provided in Appendix B. We introduced 20% and 40% *SE* and *DE* into the induction machine at the same loading conditions of the SPSM. Steady state line current and line-to-line voltage data were extracted from the DAQ for training and validating HCNN models. Similar to SPSM setup, 13 seconds of data was collected for training and validation, and 2 seconds of data was allocated exclusively for testing.

Tables VIII and IX show the results obtained from running (I, L, TD) and (V, TD) HCNN models on the test data. As we can see, the model performed excellently. Similar to SPSM, it was able to achieve 100% accuracy and quantify eccentricity faults reliably. The results show that using voltage with the TD strategy as input produces better estimates of severity than those from the current and load.

TABLE VIII
ED PERFORMANCE FOR CURRENT AND VOLTAGE INPUT FOR IM IN PERCENTAGE

Scenario	Accuracy	Recall	Precision	Specificity	F1Score
(I, L, TD)	100	100	100	100	100
(V, TD)	100	100	100	100	100

TABLE IX
RMSE VALUES FOR (I, L, TD) AND (V, TD) SCENARIOS FOR IM

Scenario	*SE* RMSE	*DE* RMSE
(I, L, TD)	0.0067	0.0235
(V, TD)	0.0055	0.0014

B. HCNN for Mixed Eccentricity

Mixed eccentricity (*ME*) fault condition was incorporated into the SPSM by using eccentric sleeves on both the shaft and the end plates. 20% and 40% of SE and DE were combined to give different *ME* conditions. The combination used were 20%SE 20%DE (**20SE20DE**), 20%SE 40%DE (**20SE40DE**), and 40%SE 20%DE (**40SE20DE**). These fault conditions were collected from the SPSM at the same loading conditions shown in Table I.

Although we did not specifically train our HCNN model to identify *ME* faults, we assessed its performance using *ME* datasets. We chose two models from the scenarios (I, L, TD) and (V, TD), which were trained on the *SE* and *DE* datasets rather than the *ME* dataset. To address the mixed nature of *ME*, the *ED* component was employed to detect healthy and non-healthy sequences by merging SE and DE. Any sequence identified as non-healthy was then analyzed using both SE and DE to estimate the severity of each. The binary classification for healthy and non-healthy sequences achieved 100% accuracy on the test data. The severity estimates are presented in Table X.

It can be observed that, although the (V, TD) HCNN model was neither trained on ME data nor specifically designed to quantify ME, it successfully generated strong estimates for DE severity. However, the model struggled with accurately

TABLE X
RMSE VALUES FOR (I, L, TD) AND (V, TD) SCENARIOS FOR *ME* DATA

Scenario	*SE* RMSE	*DE* RMSE
(I, L, TD)	0.2141	0.1334
(V, TD)	0.3008	0.0498

estimating SE severity, indicating a need for additional training on ME data to capture better the nuances associated with ME faults.

VI. CONCLUSION

This research builds on our previous work by extending the scope from eccentricity fault classification to severity estimation in Salient Pole Synchronous Machines (SPSMs). In earlier studies, we conducted a comparative analysis between Artificial Neural Networks (ANNs) and Convolutional Neural Networks (CNNs) using stator and field currents to classify eccentricity faults into Static Eccentricity (SE) and Dynamic Eccentricity (DE) under varying load conditions. Despite employing time-domain and frequency-domain features, along with complex preprocessing techniques to train the ANN, the network underperformed compared to CNN. The CNN architecture, with its ability to automatically extract features from preprocessed stator current inputs, demonstrated superior classification accuracy. Building on this foundation, the present work advances the concept by introducing an end-to-end deep hierarchical CNN (HCNN) that eliminates the need for preprocessing while effectively classifying and quantifying the severity of eccentricity faults. The HCNN architecture integrates feature extraction, classification, and severity estimation into a single framework, improving robustness and accuracy. Data from our experiments validated that the HCNN model can reliably classify both SE and DE faults and estimate their severity irrespective of the machine's loading conditions.

From our analysis, it can be concluded that using voltage inputs with time delays (V, TD) instead of current produced the best results, highlighting the advantages of voltage sensitivity to faults and insensitivity to load variations. In addition, further investigation into the HCNN architecture design revealed that extracting features from earlier layers in deeper models yielded better severity estimation performance. These findings provide valuable insights for the optimal design of future HCNN architectures aimed at fault diagnosis. Additionally, the model demonstrated adaptability across different electric machines, such as induction motors, and showed potential for handling mixed eccentricity faults. Equipped with this model, we believe that most plants should be able to address eccentricity faults efficiently and effectively, and even rely our model to do real-time monitoring of faults. They can also have it as part of their scheduled maintenance routine for their electrical machinery.

As a further work, we plan to extend the study to properly handle mixed eccentricity, and apply the framework to other operating conditions, such as the power factor of an SPSM.

APPENDIX A
SPECIFICATIONS OF THE SPSM USED IN OUR EXPERIMENTAL SETUP

Rated power = 2 kW
Stator voltage = 208 V
Number of phases = 3
Number of poles = 4
Speed = 1800 rpm
Frequency = 60 Hz
Type of stator winding = double layer, lap
Number of turns per phase = 144
Number of stator slots = 36
Number of rotor bars = 20 (5 bars per pole)
Stack length = 76 mm
Stator inner diameter = 148 mm
Rotor outer diameter = 146.8 mm
Stator resistance per phase = $0.6\ \Omega$
Stator leakage inductance per phase = 0.0079 H
Rotor bar resistance = $5.827\ \mu\Omega$
Rotor bar leakage inductance = $0.034\ \mu$H
End ring resistance = $0.4531\ \mu\Omega$
Eng ring leakage inductance = 17.5 nH
Field winding resistance = $81\ \Omega$
Field winding inductance = 6 H
Nominal air gap along d-axis = 0.6 mm
Nominal air gap along q-axis = 40.27 mm
Effective air gap along d-axis = 1.7769 mm
Effective air gap along q-axis = 59.1058 mm

APPENDIX B
SPECIFICATIONS OF THE INDUCTION MOTOR USED IN OUR EXPERIMENTAL SETUP

Rated power = 2 kW
Stator voltage = 208 V
Number of phases = 3
Number of poles = 4
Speed = 1770 rpm
Frequency = 60 Hz
Type of stator winding = double layer, lap
Number of turns per phase = 312
Number of stator slots = 36
Number of rotor bars = 45
Rotor bar resistance = $5.827\ \mu\Omega$
Rotor bar leakage inductance = $0.034\ \mu$H
End ring resistance =$0.4531\ \mu\Omega$
Eng ring leakage inductance = 0.175 nH
Mean radius = 73.75 mm
Airgap length = 0.5 mm
Carters coefficient = 1.95

ACKNOWLEDGMENT

The authors would like to extend their heartfelt gratitude to the associate editor and the reviewers for their thoughtful and constructive feedback on our previous version of the paper. Their insights and suggestions have significantly enhanced

the quality of our work. The authors would also like to acknowledge the University of Victoria and the Natural Sciences and Engineering Research Council of Canada (NSERC) for providing financial support, and the Digital Research Alliance of Canada (alliancecan.ca) for computational resources.

REFERENCES

[1] C. Bruzzese and G. Joksimovic, "Harmonic signatures of static eccentricities in the stator voltages and in the rotor current of no-load salient-pole synchronous generators," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 1606–1624, 2011.

[2] P. Pang, W. Liu, F. Chengwu, Y. Hang, L. Qinlong, R. Xiaochen, and J. Hailian, "Influence of rotor eccentricity types on the operating performance for a 100 kw hts maglev flywheel system," *IEEE Transactions on Applied Superconductivity*, vol. 31, no. 4, pp. 1–6, 2021.

[3] H. Toliyat and N. Al-Nuaim, "Simulation and detection of dynamic air-gap eccentricity in salient-pole synchronous machines," *IEEE Transactions on Industry Applications*, vol. 35, no. 1, pp. 86–93, 1999.

[4] P. Tavner, B. Gaydon, and D. Ward, "Monitoring generators and large motors," *IEE Proceedings B (Electric Power Applications)*, vol. 133, pp. 169–180(11), May 1986. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/ip-b.1986.0024

[5] S. Keller, M. T. Xuan, J. Simond, and A. Schwery, "Large low-speed hydro-generators – unbalanced magnetic pulls and additional damper losses in eccentricity conditions," *IET Electric Power Applications*, vol. 1, pp. 657–664(7), September 2007. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-epa_20060504

[6] H. C. Dirani, A. Merkhouf, A.-M. Giroux, and K. Al-Haddad, "Study of the impact of eccentricity in large synchronous generator with finite elements," in *2014 International Conference on Electrical Machines (ICEM)*, 2014, pp. 277–282.

[7] D. G. Dorrell and A. A. Salah, "Detection of rotor eccentricity in wound rotor induction machines using pole-specific search coils," *IEEE Transactions on Magnetics*, vol. 51, pp. 1–4, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:1085510

[8] J.-H. Im, J.-K. Kang, and J. Hur, "Static and dynamic eccentricity faults diagnosis in pm synchronous motor using planar search coil," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 9, pp. 9291–9300, 2023.

[9] U. Galfarsoro, A. McCloskey, X. Hernández, G. Almandoz, S. Zarate, and X. Arrasate, "Eccentricity detection procedure in electric motors by force transducer and search coils in a novel experimental test bench," in *2019 IEEE 12th International Symposium on Diagnostics for Electrical Machines, Power Electronics, and Drives (SDEMPED)*, 2019, pp. 220–226.

[10] Y.-L. He, Z.-J. Zhang, W.-Q. Tao, X.-L. Wang, D. Gerada, C. Gerada, and P. Gao, "A new external search coil based method to detect detailed static air-gap eccentricity position in nonsalient pole synchronous generators," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 8, pp. 7535–7544, 2021.

[11] I. Sadeghi, H. Ehya, and J. Faiz, "Eccentricity fault indices in large induction motors an overview," *2017 8th Power Electronics, Drive Systems & Technologies Conference (PEDSTC)*, pp. 329–334, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:6296317

[12] E. T. Esfahani, S. Wang, and V. Sundararajan, "Multisensor wireless system for eccentricity and bearing fault detection in induction motors," *IEEE/ASME Transactions on Mechatronics*, vol. 19, pp. 818–826, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:18750690

[13] R. Supangat, J. C. Grieger, N. Ertugrul, W. L. Soong, D. Gray, and C. Hansen, "Investigation of static eccentricity fault frequencies using multiple sensors in induction motors and effects of loading," *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, pp. 958–963, 2006. [Online]. Available: https://api.semanticscholar.org/CorpusID:22845499

[14] U. Dongare, B. Umre, M. Ballal, and V. Dongare, "Design of optimal mlp-neural network-based induction motor fault classifier," in *2022 IEEE International Conference on Power Electronics, Smart Grid, and Renewable Energy (PESGRE)*, 2022, pp. 1–6.

[15] D. G. Dorrell and C. Cossar, "A vibration-based condition monitoring system for switched reluctance machine rotor eccentricity detection," *IEEE Transactions on Magnetics*, vol. 44, pp. 2204–2214, 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:23805314

[16] O. O. Ogidi, P. S. Barendse, and M. A. Khan, "Detection of static eccentricity faults in afpm machine with asymmetric windings using vibration analysis," *2014 International Conference on Electrical Machines (ICEM)*, pp. 1549–1554, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:12701627

[17] O. Ogidi, P. S. Barendse, and M. A. Khan, "Detection of static eccentricities in axial-flux permanent-magnet machines with concentrated windings using vibration analysis," *IEEE Transactions on Industry Applications*, vol. 51, pp. 4425–4434, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:17767273

[18] V. Hegde and G. S. Maruthi, "Experimental investigation on detection of air gap eccentricity in induction motors by current and vibration signature analysis using non-invasive sensors," *Energy Procedia*, vol. 14, pp. 1047–1052, 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:108504807

[19] S. Nandi and H. Toliyat, "Fault diagnosis of electrical machines-a review," in *IEEE International Electric Machines and Drives Conference. IEMDC'99. Proceedings (Cat. No.99EX272)*, 1999, pp. 219–221.

[20] T. C. Ilamparithi and S. Nandi, "Detection of eccentricity faults in three-phase reluctance synchronous motor," *IEEE Transactions on Industry Applications*, vol. 48, no. 4, pp. 1307–1317, 2012.

[21] W. Dehina, M. Boumehraz, F. Kratz, and J. Fantini, "Diagnosis and comparison between stator current analysis and vibration analysis of static eccentricity faults in the induction motor," in *2019 4th International Conference on Power Electronics and their Applications (ICPEA)*, 2019, pp. 1–4.

[22] N. Bessous, S. E. Zouzou, S. Sbaa, and W. Bentrah, "A comparative study between the mcsa, dwt and the vibration analysis methods to diagnose the dynamic eccentricity fault in induction motors," in *2017 6th International Conference on Systems and Control (ICSC)*, 2017, pp. 414–421.

[23] R. Bousseksou, N. Bessous, L. Zarour, S. Sbaa, R. Pusca, R. Raphaël, M. M. Rezaoui, I. Merzouk, and A. Borni, "Detailed analysis of rotor eccentricity fault signatures in induction motors using dwt-fft technique," in *2022 19th International Multi-Conference on Systems, Signals & Devices (SSD)*, 2022, pp. 74–79.

[24] J. Shin, Y. Park, and S. B. Lee, "Flux-based detection and classification of induction motor eccentricity, rotor cage, and load defects," *IEEE Transactions on Industry Applications*, vol. 57, no. 3, pp. 2471–2480, 2021.

[25] J. Yu and V. Makis, "Wavelet analysis with time-synchronous averaging of planetary gearbox vibration data for fault detection and diagnostics," in *2011 IEEE International Conference on Computer Science and Automation Engineering*, vol. 1, 2011, pp. 417–421.

[26] M. Kiani, W.-J. Lee, R. Kenarangui, and B. Fahimi, "Detection of rotor faults in synchronous generators," in *2007 IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives*, 2007, pp. 266–271.

[27] W. Zhang, D. Yang, and H. Wang, "Data-driven methods for predictive maintenance of industrial equipment: A survey," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2213–2227, 2019.

[28] K. Alameh, N. Cité, G. Hoblos, and G. Barakat, "Feature extraction for vibration-based fault detection in permanent magnet synchronous motors," in *2015 Third International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE)*, 2015, pp. 163–168.

[29] E. Irgat, A. Ünsal, and H. T. Canseven, "Detection of eccentricity faults of induction motors based on decision trees," in *2021 13th International Conference on Electrical and Electronics Engineering (ELECO)*, 2021, pp. 435–439.

[30] R. K. Dussa and P. Kumar N, "Implementation of machine learning to analyze static eccentricity fault in spmsm using fem," in *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*, 2023, pp. 1–6.

[31] L. Yusuf and T. C. Ilamparithi, "Dynamic eccentricity fault detection in synchronous machines using principal component analysis," in *2023 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2023, pp. 348–353.

[32] D. Matić, F. Kulić, M. Pineda-Sanchez, and J. Pons-Llinares, "Artificial neural networks eccentricity fault detection of induction motor," in *2010 Fifth International Multi-conference on Computing in the Global Information Technology*, 2010, pp. 1–4.

[33] H. O. Alwan, N. M. Farhan, and Q. S.-A. Sabbagh, "Detection of static air-gap eccentricity in three phase induction motor by using artificial neural network (ann)," *International Journal of Engineering Research and Applications*, vol. 07, pp. 15–23, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:43464155

[34] X. Huang, T. G. Habetler, and R. G. Harley, "Detection of rotor eccentricity faults in a closed-loop drive-connected induction motor using an artificial neural network," *IEEE Transactions on Power Electronics*, vol. 22, pp. 1552–1559, 2007. [Online]. Available: https://api.semanticscholar.org/CorpusID:24011504

[35] I. M. Hussein and Z. Al-Hamouz, "Neural network based detection technique for eccentricity fault in lspms motors," in *2018 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2018, pp. 1–5.

[36] S. Wei, J. Li, Z. Zhao, and D. Yuan, "Artificial monitoring of eccentric synchronous reluctance motors using neural networks," *Computers, Materials & Continua*, no. 1, pp. 1036–1050, Feb. 2022. [Online]. Available: https://www.techscience.com/cmc/v72n1/46882/pdf

[37] C. Park, H. Kim, J. Lee, G. Ahn, M. Youn, and B. Youn, "A feature inherited hierarchical convolutional neural network (fi-hcnn) for motor fault severity estimation using stator current signals," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 22, pp. 1253–1266, 2021.

[38] H. Oh, J. H. Jung, B. C. Jeon, and B. D. Youn, "Scalable and unsupervised feature engineering using vibration-imaging and deep learning for rotor system diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3539–3549, 2018.

[39] S. Kiranyaz, A. Gastli, L. Ben-Brahim, N. Al-Emadi, and M. Gabbouj, "Real-time fault detection and identification for mmc using 1-d convolutional neural networks," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8760–8771, 2019.

[40] Y. Gao, L. Gao, X. Li, and S. Cao, "A hierarchical training-convolutional neural network for imbalanced fault diagnosis in complex equipment," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 8138–8145, 2022.

[41] B. Ganguly, R. K. Ray, A. Chatterjee, and S. Paul, "A deep learning aided intelligent framework for condition monitoring of electrical machinery," in *2023 IEEE Devices for Integrated Circuit (DevIC)*, 2023, pp. 82–86.

[42] J. S. L. Senanayaka, H. Van Khang, and K. G. Robbersmyr, "Toward self-supervised feature learning for online diagnosis of multiple faults in electric powertrains," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 3772–3781, 2021.

[43] R. Liu, G. Meng, B. Yang, C. Sun, and X. Chen, "Dislocated time series convolutional neural architecture: An intelligent fault diagnosis approach for electric machine," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1310–1320, 2017.

[44] M. Pavlovski, M. Alqudah, T. Dokic, A. A. Hai, M. Kezunovic, and Z. Obradovic, "Hierarchical convolutional neural networks for event classification on pmu measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.

[45] L. Wen, X. Li, and L. Gao, "A new two-level hierarchical diagnosis network based on convolutional neural network," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 2, pp. 330–338, 2020.

[46] A. Shejwalkar, L. Yusuf, and T. C. Ilamparithi, "Comparative analysis of machine learning algorithms for eccentricity fault classification in salient pole synchronous machine," in *2024 IEEE Texas Power and Energy Conference (TPEC)*, 2024, pp. 1–6.

[47] *HD-CNN: Hierarchical Deep Convolutional Neural Networks for Large Scale Visual Recognition*, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:206770495

Latifa Yusuf received her B.Eng and M.Eng degrees from the Federal University of Technology Minna, Nigeria. She is currently pursuing her Ph.D. at the University of Victoria in British Columbia, Canada. Her research focuses on utilizing machine learning algorithms for fault diagnosis in electrical machines.

Ashwin Shejwalkar received his B.Eng degree from Birla Institute of Technology and Science Pilani, India. He is currently pursuing his M.S. in Electrical and Computer Engineering at the University of Wisconsin-Madison, USA. His current research focuses primarily on modular multilevel converters for HVDC grid applications.

Belaid Moa received the Ph.D. degree in computer science from the University of Victoria, Victoria, BC, Canada, DEA diploma in Networks and Networks from Institute National Polytechnique de Toulouse, Toulouse, France, the M.Eng. degree in electronics and signal processing from École Nationale Supérieure, d'électrotechnique, d'informatique, d'hydraulique et des Télécommunications, Toulouse, France, and the B.Sc. degree in electrical engineering from École Hassania des Travaux Publics, Casablanca, Morocco. He is currently an adjunct faculty member in Electrical and Computer Engineering at the University of Victoria and an Advanced Research Computing Specialist with Digital Research Alliance of Canada/University of Victoria Systems, Victoria, BC, Canada.

T. Ilamparithi received his B.E. degree from Anna University, India, in 2005, and his M.Tech. degree from the Indian Institute of Technology, Delhi (IIT), India, in 2007. He obtained his Ph.D. degree from the University of Victoria in 2013. Between 2013 and 2017, He worked as a Modeling and Simulation Specialist at OPAL-RT Technologies Inc., a Montreal-based real-time simulator manufacturer. Since May 2017, he has been an Assistant Teaching Professor at the University of Victoria. His research interests include fault diagnosis of electrical machines and real-time simulations