Modeling Electrical Motor Dynamics Using Encoder-Decoder with Recurrent Skip Connection

Sagar Verma,[1,2] Nicolas Henwood,[2] Marc Castella,[3] Francois Malrait,[2] Jean-Christophe Pesquet[1]

[1]Université Paris-Saclay, CentraleSupélec, Inria, Centre de Vision Numérique
[2]Schneider Toshiba Inverter Europe
[3]Samovar, CNRS, Télécom SudParis, Institut Polytechnique de Paris
{sagar.verma, jean-christophe.pesquet}@centralesupelec.fr, marc.castella@telecom-sudparis.eu,
{sagar.verma, nicolas.henwood, francois.malrait}@se.com

Abstract

Electrical motors are the most important source of mechanical energy in the industrial world. Their modeling traditionally relies on a physics-based approach, which aims at taking their complex internal dynamics into account. In this paper, we explore the feasibility of modeling the dynamics of an electrical motor by following a data-driven approach, which uses only its inputs and outputs and does not make any assumption on its internal behaviour. We propose a novel encoder-decoder architecture which benefits from recurrent skip connections. We also propose a novel loss function that takes into account the complexity of electrical motor quantities and helps in avoiding model bias. We show that the proposed architecture can achieve a good learning performance on our high-frequency high-variance datasets. Two datasets are considered: the first one is generated using a simulator based on the physics of an induction motor and the second one is recorded from an industrial electrical motor. We benchmark our solution using variants of traditional neural networks like feedforward, convolutional, and recurrent networks. We evaluate various design choices of our architecture and compare it to the baselines. We show the domain adaptation capability of our model to learn dynamics just from simulated data by testing it on the raw sensor data. We finally show the effect of signal complexity on the proposed method ability to model temporal dynamics.

1 Introduction

Electrical motors are so much a part of everyday life that we seldom give them a second thought. For example, when we switch on an electrical vehicle, we confidently expect it to run rapidly up to the correct speed, provide acceleration, stop when brakes are applied, and casually predict faults to avoid future mishaps. Electrical motors have very complex dynamics and it is essential to have a controller that can provide robust control based on these dynamics. Electrical motor controllers also provide protection and supervision of the electro-mechanical system (Campbell 1987; Sisking 1978). For these services, it is imperative to know the dynamical physical model of electrical motors. Accurate dynamics is derived from the first principles of physics. These dynamical models are dependent on different electrical motor physical quantities like currents, voltages, speed, fluxes, inductances, and resistances, which are measured directly or indirectly using sensors or estimators. Accurately measuring some of these quantities is hard due to the presence of noise. Operating conditions also affect some of these quantities, one example being thermal evolution of resistances with time. Therefore mathematical models cannot be fully trusted in the design of controllers. A large number of simulations and human expert knowledge is required to develop robust controllers. The focus of this work is to model relationships between different electrical quantities of an electrical motor. We focus on deriving currents and electromagnetic torque from voltages and speed recorded from electrical motors using sensors.

Internet-of-Things (IoT) has made it possible to monitor different electro-mechanical devices in real-time and also provides sensor data that can be used to learn the dynamics of the system under consideration. End-to-end learning of temporal dynamics from time-series data has been made easier due to methods like Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long-Short Term Memory (LSTM) structures. By providing a large amount of multi-dimensional data, it has been shown that RNN and LSTM approaches can model complex nonlinear feature interactions which are crucial to model complex nonlinear dynamics.

In line with the current work of (Miller and Hardt 2019), we found that one dimensional CNNs provide better results. We modified CNNs into an encoder-decoder architecture and incorporated recurrent skip connections between corresponding layers of the encoder-decoder architecture. In order to reduce the number of parameters, we further modified the recurrent architecture by diagonalizing its weights. We compared our proposed model with different benchmarks using a novel metric which takes into account the complexity of the predicted signals. We also proposed a novel loss function that takes into account the complexity of the signal while training the networks. We showed how using the proposed loss function for training leads to better generalization.

This paper makes the following contributions:

- This is one of the first works addressing the problem of learning nonlinear dynamics of electrical motors from recorded time-series data.

- We propose a new Encoder-Decoder architecture to learn time-series relationship between different electrical quantities.
- We validate our methodology on two datasets; a large dataset of simulated electrical motor operations and a small dataset of sensor data recorded from the real-world operations of electrical motors.
- We propose a novel loss function that uses fast variations present in the electrical motor signals to avoid model bias.
- We analyse the capability of the proposed method by using a new analysis technique and we demonstrate the transfer learning capability of our approach.

The paper is structured as follows: Section 2 provides a brief background on electrical motors and recent advances in physics and time-series modeling. Section 3 describes the data and more specifically how it is collected and preprocessed. Section 4 describes the benchmark and the proposed methods. Sections 5 and 6 contains experimental details and the obtained results. In the last section, conclusions are drawn and some possible extensions of this work are mentioned.

2 Background

The state space model of an induction motor is presented in (Jadot et al. 2009). Modeling of electrical motors based on analytical mechanics and energy consumption is presented in (Jebai et al. 2014). Existing methods for designing a controller for an induction motor can be done in two ways; when perfect knowledge of the parameters is available (Espinosa-Perez and Ortega 1994; Espinosa-Perez and Ortega 1995; Nicklasson et al. 1997) and when there is an uncertainty associated with the parameters estimation (Chan and Wang 1990; Stephan, Bodson, and Chiasson 1992; Marino, Peresada, and Tomei 1999; 2000). Electrical quantities like resistances and inductances are roughly estimated in most of the applications. These quantities also vary with change in temperature of the electrical motor environment. The control law is dependent on these quantities and measuring them requires high precision sensors and numerous expensive experiments. Due to this, acquiring perfect knowledge of the parameters is very impractical and has very limited applicability in industrial settings. Designing controllers in the presence of parametric uncertainty is done by using adaptation schemes. Two methods of adaptation are time-scale separation and time-varying adaptation (Anderson 1977; Zhang 2002; Jadot et al. 2009). (Silva, Bazzi, and Gupta 2013) presents a neural network classifier for fault diagnosis in electrical motor operations. They do not use dynamics modeling and only rely on supervised labels (Murphey et al. 2006), learn motor dynamics from simulated data and perform fault detection in simulated motor operations.

The first use of neural networks to model physical phenomena was presented in (Levin 1990). This paper presents a multi-layered neural networks for nonlinear prediction and system modeling from time-series data. Recently, deep neural networks have been used in learning physical dynamics from data in range of applications e.g., calorimetry (Carminati et al. 2017), drone landing (Shi et al. 2019), and nonlinear dynamics identification (Lusch, Kutz, and Brunton 2018). Karpatne et al. presents a physics-guided neural network (PGNN) that leverages the output of physics-based model simulations along with observational features to generate predictions using a neural network architecture (Karpatne et al. 2017). Furthermore, they present a novel framework for using physics-based loss functions in the learning objective of neural networks, in order to ensure that the model predictions not only show lower errors on the training set but are also scientifically consistent with the known physics on the unlabeled set.

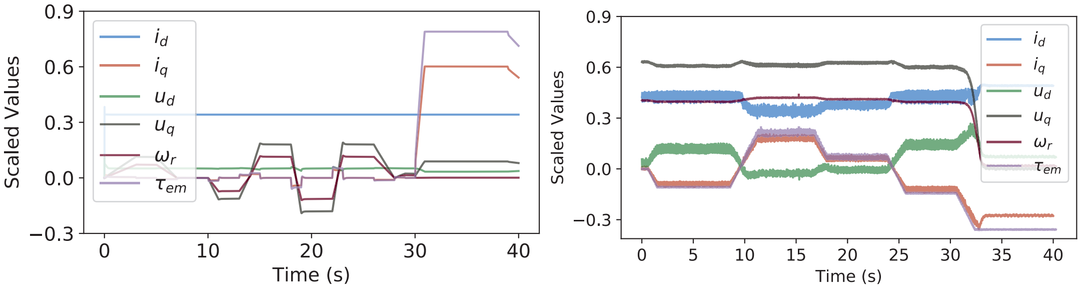
RNN and LSTMs have been shown to be very good at learning hidden temporal dynamics from data in various applications such as wind speed forecasting (Ghaderi, Sanandaji, and Ghaderi 2017), estimating missing measurements in time series (Yoon, Zame, and Schaar 2017), and consumer event forecasting (Laptev et al. 2017). Convolutional architectures have recently been shown to be competitive on many sequence modelling tasks when compared to the de-facto standard of recurrent neural networks (RNNs), while providing computational and modeling advantages due to inherent parallelism. In (Bai, Kolter, and Koltun 2018), authors provide an empirical comparison between convolutional and recurrent network in modeling time-series. Aksan et al. presents a stochastic variant of temporal convolutional network which performs better than stochastic RNNs (Aksan and Hilliges 2019). Miller et al. have shown that in some cases, feed-forward networks are better in modeling temporal patterns than sequential networks (Miller and Hardt 2019).

Li et al. showed that parameters in recurrent neural networks can be decreased by making neurons independent of each other (Li et al. 2018). In time-series prediction, different events often have different importance. This can be achieved using asymmetric loss function which weights distinct parts of signal differently as shown in (Christoffersen and Diebold 1997).

3 Available datasets

It seems there is no large electrical motor operations dataset available in the research community to train deep neural networks. We thus introduce two different datasets for our experiments; one dataset consists of simulations performed by using the control law proposed in (Jadot et al. 2009) and the second dataset is recorded from an industrial electrical motor. Data is collected at a sampling rate of 250Hz. We generate 100 hours of simulation data which cover a wide range of operating conditions.The dataset consists of the following electrical quantities; currents i_d and i_q, voltages u_d and u_q, rotor speed ω_r, stator pulsation ω_s, and torque τ_{em}. The indices d and q denote three phase quantities represented in a two phase orthogonal rotating reference frame. The real electrical motor data are collected from a 4-kilowatt induction motor. Data from 10 various operating conditions are collected. Rotor speed and torque load are different in each of the runs. In total, we collected 1207 seconds of raw sensor data.

In our experiments, we split the data into four parts; training and validation parts consist of 70% and 30% of the sim-



(a) Simulated sample.

(b) Real world sample.

Figure 1: Electrical motor quantities showing first 40 seconds of an electrical motor operation.

Model	**Architecture**		**Input**	**Output**
	Shallow	**Deep**		
Feed-Forward	4 Linear	5 Linear	Flattened vector	Middle Value
RNN	1 Recurrent → 2 Linear	2 Recurrent → 2 Linear	Channelized	Same length as input
LSTM	1 LSTM → 2 Linear	2 LSTM → 2 Linear	Channelized	Same length as input
CNN	3 Conv → 2 Linear	4 Conv → 2 Linear	Channelized	Middle Value

Table 1: Architectural details of the benchmark models.

ulation data, respectively. We use 20% of the raw sensor data to fine tune the model trained on the training set of the simulated data and the rest for testing. We do not use stator pulsation ω_s in our experiments since it has a trajectory similar to ω_r. To train our network, we normalize our data between $(-1, 1)$ as different electrical quantities have different ranges. Normalized signals from the first 40 seconds of a simulated sample and a raw sensor sample are shown in figure 1. It can be seen that raw sensor data has short term variations due to inherent noise present on the sensors.

4 Modeling

Benchmark Methods

To the best of our knowledge, this is the first work in modeling nonlinear dynamics of electrical motor and we would also like to emphasize that the proposed datasets are challenging. To illustrate this fact, we provide several benchmark methods that are derivatives of standard neural networks. Broadly, feed-forward network, convolutional neural network, recurrent neural network and Long-Short Term Memory (LSTM) structures are evaluated. Table 1 shows all the benchmark networks. For each type of network, we try two variations, shallow and deep, to evaluate the effect of the network depth on their learning capability.

Feedforward Neural Networks We use feedfoward neural networks (FNNs) to show that our proposed problem and dataset are quite difficult and FNNs have limited learning capabilities. Row 1 in table 1 provides the configuration details of the two experimented networks.

Sequential Neural Networks Sequential neural networks have been used widely to learn from sequential data. RNNs and LSTMs are two of the most commonly used sequential neural networks. Configuration details are shown in rows 2 and 3 in table 1.

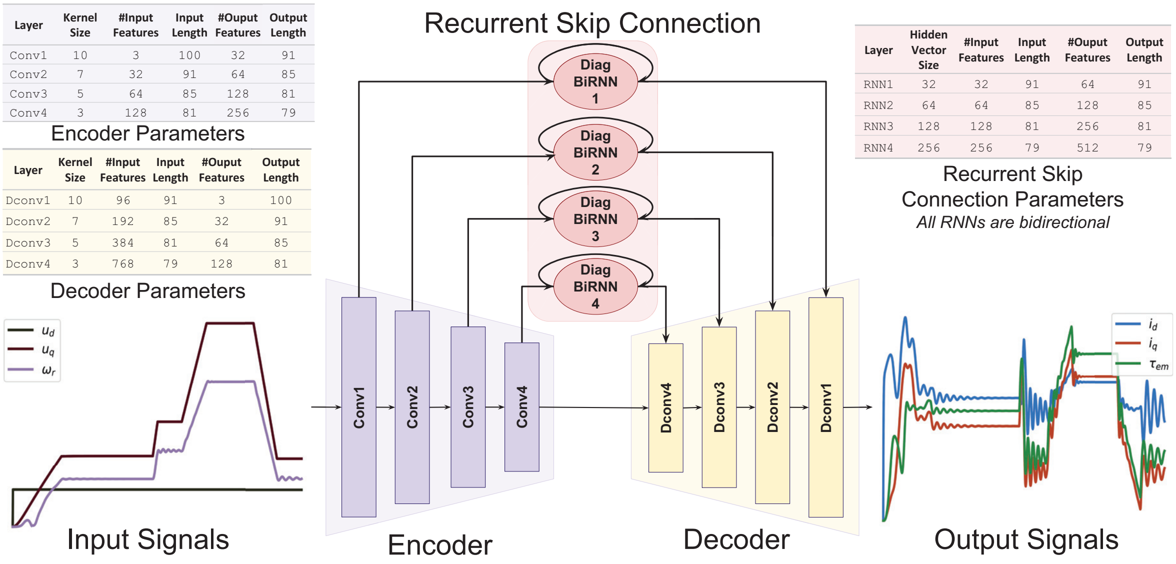
Convolutional Neural Networks FNNs have very limited learning capabilities when the input data is complex like sequential or multidimensional. Recently, convolutional neural networks (CNNs) have been shown to provide competitive performances on sequential data. The configuration of benchmark for CNNs is shown in row 4 of table 1.

Proposed Method

Traditionally, sequential networks have been used to model temporal dynamics. In our experiments, we found that RNNs and LSTMs do not provide as good learning capability as one dimensional CNNs. Since our task is to perform multivariate prediction over the same length as the input, we use an architecture where all layers are made of convolutions. We then carefully introduce several intuitive modifications to the encoder-decoder architecture which leads to a better and parameter efficient model.

Encoder-Decoder Network To capture temporal dynamics from complete input and output window we introduce an encoder-decoder network. It consists of encoding and decoding blocks with convolutional and deconvolutional layers, respectively. The convolutional and deconvolutional blocks are followed by ReLU activations. We do not use pooling as in our experiments we found that they deteriorate the results.

Encoder-Decoder Network with Skip Connection It has been shown that adding skip connection to encoder-decoder helps in transferring high level features directly from one encoding layer to its corresponding decoding layer (Mao, Shen, and Yang 2016). We also introduce skip connections between encoding and decoding layers.



Layer	Kernel Size	#Input Features	Input Length	#Ouput Features	Output Length
Conv1	10	3	100	32	91
Conv2	7	32	91	64	85
Conv3	5	64	85	128	81
Conv4	3	128	81	256	79

Encoder Parameters

Layer	Kernel Size	#Input Features	Input Length	#Ouput Features	Output Length
Dconv1	10	96	91	3	100
Dconv2	7	192	85	32	91
Dconv3	5	384	81	64	85
Dconv4	3	768	79	128	81

Decoder Parameters

Layer	Hidden Vector Size	#Input Features	Input Length	#Ouput Features	Output Length
RNN1	32	32	91	64	91
RNN2	64	64	85	128	85
RNN3	128	128	81	256	81
RNN4	256	256	79	512	79

Recurrent Skip Connection Parameters

All RNNs are bidirectional

Figure 2: Proposed architecture.

Encoder-Decoder Network with Recurrent Skip Connection Convolution operations are windowed over the kernel size, this means that convolution cannot learn temporal relationships which are out of the kernel sized windows. Adding recurrent layer over convolutional features can overcome this issue. This also helps in learning temporal patterns in the latent space. We add recurrent layers after every encoding layers. The output of the recurrent layer is then sent to the corresponding decoding layers.

Encoder-Decoder Network with Bidirectional Recurrent Skip Connection Bidirectional RNNs help in learning temporal patterns in both direction. For our use case, as we want to predict for each time step of the input window. Therefore, we also use bidirectional RNNs.

Encoder-Decoder Network with Bidirectional Diagonalized Recurrent Skip Connection Vanilla RNNs have a high number of parameters due to matrix multiplications between weights and features. Diagonalizing weights in the recurrent unit decreases the number of parameters.

The hidden state update equation of an RNN is given by:

$$h_t = \tanh(Wx_t + Uh_{t-1} + b) \tag{1}$$

where $x_t \in \mathbb{R}^M$ and $h_t \in \mathbb{R}^N$ are the input and hidden state at time t, respectively. $W \in \mathbb{R}^{N\times M}$, $U \in \mathbb{R}^{N\times N}$, and $b \in \mathbb{R}^N$ are the weights for the input and the hidden vector, and the bias of the neurons. We propose to impose diagonal structures for the weight matrices W and U by setting $N = M$. Let the diagonal vector of entries be denoted by vectors w and u, respectively. Practically, this amounts to replacing the matrix multiplication operations with Hadamard products $\odot$ of the involved vectors. The diagonalized recurrent network is described as:

$$h_t = \tanh(w \odot x_t + u \odot h_{t-1} + b) \tag{2}$$

where $w \in \mathbb{R}^M$, $u \in \mathbb{R}^M$, and $b \in \mathbb{R}^M$ are input weights.

Total Variation Weighted Mean Square Loss

In real world usage of electrical motors, large variations in the signals occurs less often than small variations. We observe this effect in our dataset, which causes model bias toward small variations when trained with mean square loss. This is not a desirable behavior if the learned model is used in controllers. To avoid this problem, we propose a novel asymmetric loss function that takes into account the signal variations:

$$\mathcal{L}_{\text{TV-WeightMSE}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T-1}|y_t^i - y_{t+1}^i|\frac{1}{T}\sum_{t=1}^{T}(y_t^i - \hat{y_t^i})^2 \tag{3}$$

where y_t^i and $\hat{y_t^i}$ are the values of output and predicted sample i at time-step t, respectively. N is the number of training samples where each sample is of duration T.

5 Experiments

For all our experiments we use an Ubuntu 18.04 OS with V100 GPU. PyTorch is employed to implement the benchmark and proposed architectures. Simulation data are collected from a Simulink model which is heavily used in the motor control industry. To show the requirements of the proposed method and why benchmark methods fail, we perform extensive experiments. We vary our architecture by trying different input lengths, number of layers, and RNN/LSTM hidden vector lengths. We try the following input lengths {5,10,15,20,25,50,100,200} and find out that an input length greater than 100 is better at capturing the motor operation dynamics. Depending on the architecture, different input and output structures are required. Feed-forward networks take a flattened vector and predict a single output which is the middle value of the output signal. RNNs and LSTMs take channelized input and predict output of the same length.

Model	Window Size	Parameters	MAE	SMAPE	R^2
Shallow Feed-Forward	25	751617	77.76	9.79%	-0.59
Deep Feed-Forward	20	1118209	78.91	8.53%	-0.39
Shallow RNN	100	9889	77.97	8.5%	-0.3
Deep RNN	150	12001	78.26	7.76%	-0.35
Shallow LSTM	50	13441	79.39	6.41%	-0.26
Deep LSTM	100	21889	79.58	6.29%	-0.11
Shallow CNN	100	518721	79.51	6.22%	-0.13
Deep CNN	100	650049	79.69	6.13%	-0.14
Shallow	100	309185	80.63	5.02%	0.08
Deep	100	1096385	81.21	4.57%	0.29
Skip	100	364801	28.96	3.71%	0.42
RNN-Skip	100	638145	28.18	3.42%	0.43
BiRNN-Skip	100	967105	27.96	3.31%	0.41
DiagBiRNN-Skip	100	618465	**26.88**	**1.09%**	**0.95**

Table 2: Experimental details and results for the benchmark models and the proposed model variants obtained on the simulated validation set. First 8 rows show results for the benchmark models and last 6 rows show results for the variants of the proposed model. Average of the three output quantities; currents i_d and i_q and electromagnetic torque τ_{em} is shown.

CNNs take channelized input and predict middle value of the output signal.

In encoder-decoder variations where an RNN is used, the hidden vector size is the same as the number of features in the input vector. In the encoder-decoder network the input and output lengths are the same. We train all our models using the proposed TV-weighted mean square loss function. To find the best architecture, we use the validation set of the simulated data. Then we fine-tune the best model on the training set of the raw data and test it on the raw data test set. We also train the best performing model using mean square loss to compare it with the proposed loss function.

To evaluate the capability of the proposed method, we use different metrics allowing us to compare the performance at global and local scope of the input signal. To analyse the learning capability at global scope, we report mean absolute error (MAE), symmetric mean absolute percentage error (SMAPE), and coefficient of determination R^2 (Cameron and Windmeijer 1997).

$$\text{MAE}(y, \hat{y}) = \frac{1}{T} \sum_{t=1}^{T} |y_t - \hat{y}_t| \tag{4}$$

$$\text{SMAPE}(y, \hat{y}) = \frac{100}{T} \sum_{t=1}^{T} \frac{|\hat{y}_t - y_t|}{|\hat{y}_t| + |y_t|} \tag{5}$$

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{t=1}^{T} (\hat{y}_t - \bar{y})^2}{\sum_{t=1}^{T} (y_t - \bar{y})^2} \tag{6}$$

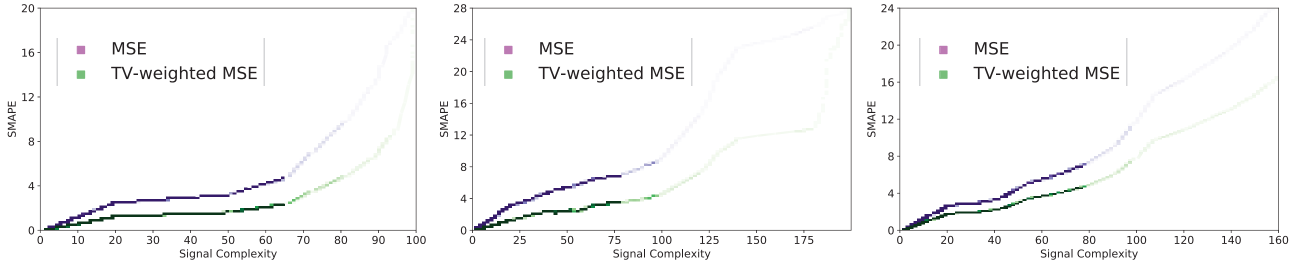
where y_t is the ground truth, $\hat{y}_t$ is the predicted output of the model at time t, and T is the total experiment duration. $\bar{y}$ denotes the mean of ground truth y.

MAE, SMAPE, and R^2 values do not provide enough information about the signal parts where the model is performing poorly or very well. Thus, we compute the signal complexity (SC) on sliding windows over the ground truth signal and plot it versus the corresponding window SMAPE value computed between the ground truth and the predicted signal. The signal complexity is given by $\text{SC}_y = \sum_{t=1}^{W-1} |y_t - y_{t-1}|$ for a small window length W. All metrics are reported on the original range of the respective quantities after re-scaling.

6 Results

We provide results for the benchmark models and the variations of the proposed architecture. The first eight rows of table 2 show the results obtained by different benchmark models. Window size column shows the input length on which the best result was obtained. Hidden vector size for both RNN and LSTM is 32. The number of parameters is also reported for all the models. For each of them we report MAE, SMAPE, and R^2 values. The average of the three output quantities are provided; current i_d, current i_q, and electromagnetic torque τ_{em}. All results were obtained on the validation set of the simulated data. Among benchmark models, we observe that MAE values are very close for all the models. But when we compare SMAPE and R^2 values, deep CNN and deep LSTM come out to be the best. In our experiments, we observe that the models perform better when the input length is 100 or more. For all the models, the performance gap between shallow and deep variants is small. This means that the network depth provides little advantage in learning nonlinear dynamics of electrical motors.

Based on the results obtained from the benchmark methods, we fix the input size to 100 for all our proposed model variants. The last 6 rows of table 2 show the results of the proposed model variants trained and validated on the simulated data. First and second rows show the results of the shallow and deep variant of the encoder-decoder architecture. We see that MAE is still comparable to the benchmark models but SMAPE and R^2 value improves. Third row shows the result of the model where skip connections have been added between encoder-decoder. MAE gets better in this case.



(a) Current i_d (b) Current i_q (c) Torque τ_{em}

Figure 3: Comparison between the proposed TV-weighted MSE loss and MSE loss used to train the proposed network. TV-weighted MSE is able to learn about signal windows which have a lot of variations.

Quantity	**MSE Loss**			**TV-weighted MSE Loss**		
	MAE	**SMAPE**	R^2	**MAE**	**SMAPE**	R^2
i_d **(A)**	28.1351	0.9702%	0.6572	27.9123	0.4619%	0.9224
i_q **(A)**	26.8912	2.3925%	0.9507	26.5234	1.9014%	0.9631
τ_{em} **(Nm)**	26.2321	1.5839%	0.9247	26.1978	0.9193%	0.9652

Table 3: Performance of proposed encoder-decoder with diagonalized recurrent skip connection model when trained using MSE loss and the proposed TV-weighted MSE loss. Results were obtained on the simulated data validation set.

Quantity	**Simulated Model**			**Fine-tuned Model**		
	MAE	**SMAPE**	R^2	**MAE**	**SMAPE**	R^2
i_d **(A)**	39.8392	4.1029%	0.3829	35.3167	2.6429%	0.5637
i_q **(A)**	47.3818	6.3729%	0.4113	42.9472	5.2841%	0.4936
τ_{em} **(Nm)**	38.5628	3.8128%	0.4997	32.3819	2.3891%	0.6017

Table 4: Results obtained for each of the output quantity on the raw test set. 'Simulated Model' column shows the results of the model trained on the simulated data and 'Fine-tuned Model' column shows the results of the model fine-tuned on the raw sensor data.

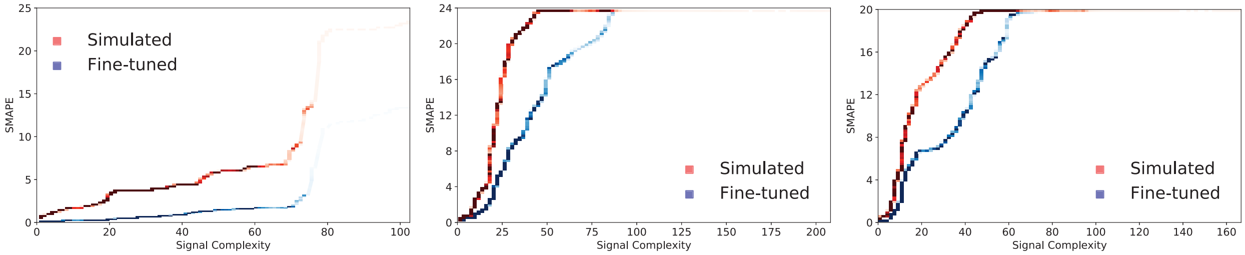
Fourth and fifth rows correspond to recurrent skip connections with unidirectional and bidirectional recurrence, respectively. Having recurrence in skip connections improves MAE and SMAPE values but comes at the cost of an increased number of parameters. It can be seen that bidirectionality has a positive effect on MAE and SMAPE. Last row shows the best version of our encoder-decoder model, where we replace RNNs in skip connections with diagonalized RNNs. This model outperforms all the methods and has fewer parameters when compared to other RNN variants.

Table 3 shows the results obtained by the proposed model when MSE loss and TV-weighted MSE loss were used in training. All three metrics for all three quantities improve when the proposed TV-weighted MSE loss is used in training the DiagBiRNN-Skip Encoder-Decoder network. Figure 3 shows how the SMAPE values increase when signal complexity increases, and compares the results with both loss functions. The SMAPE vs SC plots are 2D histograms where color intensity of each box represent the number of samples that are in that bin. We see that signal parts with higher signal complexity occur less often. Our model trained with MSE loss is able to predict more accurately parts of signal with small signal complexity. We observe that the model trained using TV-WeightMSE loss overcomes this issue.

Table 4 shows the results of simulated model and model fine-tuned on the raw data training set when tested on the raw data test set. It can be seen that the proposed model is able to learn the temporal dynamics of each of the quantities very well just from the simulated data. When the model is fine-tuned on the sensor data, it seems to be able to learn about the noise associated with the sensors and yields better results. Figure 4 shows the SMAPE vs signal complexity graph for the three output quantities obtained from the simulated and fine-tuned models. We observe that current i_q and torque τ_{em} have some signal parts which are more complex than current i_d. Figure 5 shows the results for one of the raw samples from the test set. It can be seen that the model trained on the simulated data has some offset in its prediction whereas the model fine-tuned on the sensor data is much closer to the ground truth, even if it is still not perfect.

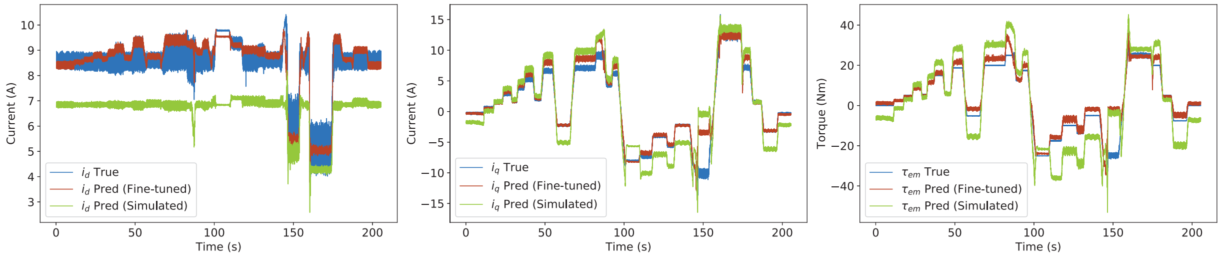
7 Conclusion

A novel problem has been investigated: the learning of electrical motor dynamics from time-series sensor data. We also have presented a novel encoder-decoder architecture that



(a) Current i_d (b) Current i_q (c) Torque τ_{em}

Figure 4: Comparison of simulated and fine-tuned model using SMAPE vs Signal Complexity graph.



(a) Current i_d (b) Current i_q (c) Torque τ_{em}

Figure 5: Predicted result of one of the experiments from test set.

uses diagonalized recurrent skip connections to learn the complex temporal dynamics. To learn the model, a novel loss function has been introduced that avoids prediction bias. We have used transfer learning to fine tune a model trained on large simulated data on a small raw sensor dataset. Our experiments have shown the promising performance of the proposed method on a noisy sensor dataset collected in an industrial context. We have also carried out a detailed analysis at the global and the local scope of the prediction performed on the test data. Our results show the feasibility of AI solutions in modeling electrical motor dynamics, thus opening a new avenue of research in this area.

References

Aksan, E., and Hilliges, O. 2019. STCN: Stochastic temporal convolutional networks. In *International Conference on Learning Representations*.

Anderson, B. 1977. Exponential stability of linear equations arising in adaptive identification. *IEEE Transactions on Automatic Control* 22(1):83–88.

Bai, S.; Kolter, J. Z.; and Koltun, V. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR* abs/1803.01271.

Cameron, A. C., and Windmeijer, F. A. 1997. An R-squared measure of goodness of fit for some common nonlinear regression models. *Journal of Econometrics* 77(2):329 – 342.

Campbell, S. J. 1987. *Solid-State AC Motor Controls*.

Carminati, F.; Pierini, G. K. M.; Farbin, A.; Hooberman, B.; Wei, W.; Zhang, M.; Pacela, V.; Vallecorsafac, S.; Spiropulu, M.; and Vlimant, J. 2017. Calorimetry with deep learning: Particle classification, energy regression, and simulation for high-energy physics. In *Deep Learning for Physical Sciences Workshop at NIPS (2017)*.

Chan, C. C., and Wang, H. 1990. An effective method for rotor resistance identification for high-performance induction motor vector control. *IEEE Transactions on Industrial Electronics* 37(6):477–482.

Christoffersen, P. F., and Diebold, F. X. 1997. Optimal prediction under asymmetric loss. *Econometric Theory* 13(6):808–817.

Espinosa-Perez, G., and Ortega, R. 1994. State observers are unnecessary for induction motor control. *Systems & Control Letters* 23(5):315 – 323.

Espinosa-Perez, G., and Ortega, R. 1995. An output feedback globally stable controller for induction motors. *IEEE Transactions on Automatic Control* 40(1):138–143.

Ghaderi, A.; Sanandaji, B.; and Ghaderi, F. 2017. Deep forecast: Deep learning-based spatio-temporal forecasting. In *Time Series Workshop in International Conference on Machine Learning*.

Jadot, F.; Malrait, F.; Moreno-Valenzuela, J.; and Sepulchre, R. 2009. Adaptive regulation of vector-controlled induction motors. *IEEE Transactions on Control Systems Technology* 17(3):646–657.

Jebai, A. K.; Combes, P.; Malrait, F.; Martin, P.; and Rouchon, P. 2014. Energy-based modeling of electric motors. In *53rd IEEE Conference on Decision and Control*, 6009–6016.

Karpatne, A.; Watkins, W.; Read, J.; and Kumar, V. 2017. How can physics inform deep learning methods in scientific problems?: Recent progress and future prospects. In *Deep Learning for Physical Sciences Workshop at NIPS (2017)*.

Laptev, N.; Yosinski, J.; Li, L.; and Smyl, S. 2017. Time-series extreme event forecasting with neural networks at uber. In *Time Series Workshop in International Conference on Machine Learning*.

Levin, E. 1990. Modeling time varying systems using hidden control neural architecture. In *International Conference on Neural Information Processing Systems*, 147–154.

Li, S.; Li, W.; Cook, C.; Zhu, C.; and Gao, Y. 2018. Independently recurrent neural network (IndRNN): Building a longer and deeper RNN. In *Computer Vision and Pattern Recognition*, 5457–5466.

Lusch, B.; Kutz, J. N.; and Brunton, S. L. 2018. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications* 9(1):4950.

Mao, X.; Shen, C.; and Yang, Y.-B. 2016. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in Neural Information Processing Systems 29*. 2802–2810.

Marino, R.; Peresada, S.; and Tomei, P. 1999. Global adaptive output feedback control of induction motors with uncertain rotor resistance. *IEEE Transactions on Automatic Control* 44(5):967–983.

Marino, R.; Peresada, S.; and Tomei, P. 2000. On-line stator and rotor resistance estimation for induction motors. *IEEE Transactions on Control Systems Technology* 8(3):570–579.

Miller, J., and Hardt, M. 2019. Stable recurrent models. In *International Conference on Learning Representations*.

Murphey, Y. L.; Masrur, M. A.; Chen, Z.; and Zhang, B. 2006. Model-based fault diagnosis in electric drives using machine learning. *IEEE/ASME Transactions on Mechatronics* 11(3):290–303.

Nicklasson, P. J.; Ortega, R.; Espinosa-Perez, G.; and Jacobi, C. G. J. 1997. Passivity-based control of a class of Blondel-Park transformable electric machines. *IEEE Transactions on Automatic Control* 42(5):629–647.

Shi, G.; Shi, X.; O'Connell, M.; Yu, R.; Azizzadenesheli, K.; Anandkumar, A.; Yue, Y.; and Chung, S. 2019. Neural lander: Stable drone landing control using learned dynamics. In *International Conference on Robotics and Automation*, 9784–9790.

Silva, A. A.; Bazzi, A. M.; and Gupta, S. 2013. Fault diagnosis in electric drives using machine learning approaches. In *2013 International Electric Machines Drives Conference*, 722–726.

Sisking, C. S. 1978. *Electrical Control Systems in Industry*.

Stephan, J.; Bodson, M.; and Chiasson, J. 1992. Real-time estimation of the parameters and fluxes of induction motors. In *IEEE Industry Applications Society Annual Meeting*, 578–585 vol.1.

Yoon, J.; Zame, W.; and Schaar, M. 2017. Multi-directional recurrent neural networks: A novel method for estimating missing data. In *Time Series Workshop in International Conference on Machine Learning*.

Zhang, Q. 2002. Adaptive observer for multiple-input-multiple-output (MIMO) linear time-varying systems. *IEEE Transactions on Automatic Control* 47(3):525–529.