Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 176 (2020) 1586–1595

Procedia Computer Science

www.elsevier.com/locate/procedia

24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Evaluation of Machine Learning for Sensorless Detection and Classification of Faults in Electromechanical Drive Systems

Tobias Grüner [1,a], Falco Böllhoff[a], Robert Meisetschläger[a], Alexander Vydrenko[a], Martyna Bator[b], Alexander Dicks[b], Andreas Theissler [2,a]

[a]Aalen University of Applied Sciences, Beethovenstraße 1, 73430 Aalen, Germany
[b]inIT - Institute Industrial IT, Campusallee 6, 32657 Lemgo, Germany

Abstract

Obtaining new information and creating value from present measurements without introducing additional sensors is cost-efficient and mitigates data that is collected and stored by information systems but not used. In electromechanical drive systems, defect states of synchronous motors can be detected based on measurements of the motor current. While there is a tendency to (exclusively) apply Deep Learning models to such problems, we argue that, for appropriate problem settings, alternatives should also be evaluated and at least be used as benchmarks. This paper addresses the question of whether non-Deep Learning methods are competitive to Deep Learning ones for sensorless detection and classification of faults in electromechanical drive systems. For this multi-class classification problem, a systematic evaluation of selected traditional, ensemble, and Deep Learning classifiers is conducted for a data set with one normal state and ten fault states of an electromechanical drive system. In addition to working on the raw input data, the impact of Recursive Feature Elimination is compared to dimensionality reduction with Principal Component Analysis. Accuracy, computational complexity, and engineering effort of the different Machine Learning pipelines are compared. A key finding is that the appropriate combination of feature elimination and Machine Learning model yields high accuracies while allowing to massively reduce the number of features, hence making the detection of fault states less computationally expensive.

Keywords: sensorless fault classification; machine learning; deep learning; feature elimination; electromechanical drive systems

1. Introduction

Increasing amounts of data are being captured with the integrated digitisation of manufacturing and production processes. Methods of Artificial Intelligence or respectively Machine Learning (ML) develop huge optimisation potentials, such as increasing the efficiency, flexibility, and individuality of the production process. Faster identification of faults through condition monitoring, anomaly detection, as well as preventive maintenance or improvement of

[1] https://orcid.org/0000-0002-0913-0472
[2] https://orcid.org/0000-0003-0746-0424

1877-0509
10.1016/j.procs.2020.09.170

product quality through data-driven process modelling, enable an increase in the autonomy of production plants. This provides predictive maintenance of production plants and improves the monitoring of industrial processes. The integration of sensor functions into the drive system is an integral part of an autonomous, intelligent subsystem [12]. The autonomous self-diagnostic capabilities of the individual components, the entire drive system, and the process can be increased by locally available ML algorithms. Within this context, this paper addresses sensorless detection and classification of faults in electromechanical drive systems based on current measurements. The evaluation in this paper is performed with a data set, which was previously employed in several researches [4, 28], where the problem setting is to classify motor currents as either normal or one of ten defects.

Recently Deep Learning [20] led to a boost in the accuracy of ML applications. These high accuracies are achieved by models found in a high-dimensional parameter space utilising enormous computing power. For specific types of input data, Deep Learning has been shown to be clearly superior, leveraging the specific properties of the data, e.g. long short-term memories [15, 27] for sequential data and Convolutional Neural Networks [19, 26] for images. This may tempt one to exclusively rely on these types of ML models. We argue that for problem settings with data represented in attribute vectors – as is the case in the underlying problem of fault detection and classification in electromechanical drive systems – other methods should also be analysed comparatively. While Deep Learning often outperforms traditional methods, the computational complexity, and the engineering effort to find good models are often significantly higher. For example, while a soft-margin Support Vector Machine (SVM) with a Radial Basis Function (RBF)-kernel [1] has two hyperparameters (C and γ) and a unique solution per parameter set, deep Artificial Neural Networks (ANNs) have a high number of hyperparameters and a tremendous number of adjustable weights with random parts involved. Typical hyperparameters are the number of layers, nodes per layer, type of activation function, drop-out rate [30], and the optimiser (e.g. Adam [18]). Consequently, a wide set of hyperparameters is tested [6].

For the underlying multi-class classification problem, a systematic evaluation of selected traditional, ensemble and Deep Learning classifiers is conducted in connection with different preprocessing and dimensionality reduction steps. The traditional methods K-nearest Neighbor (KNN) and SVM (see e.g. [14]), the tree-based ensemble methods random forests [7] and extreme gradient boosting machines (XGBoost) [9], and two ANNs [20] are trained to classify the measurements of electromechanical drive systems. These classifiers are selected in order to represent different families of algorithms like distance-based (KNN), kernel-based (SVM), and promising ensemble methods. For the traditional and ensemble methods, the effect of using the raw data in contrast to a reduced feature space is evaluated utilising Principal Component Analysis (PCA) and Recursive Feature Elimination (RFE) [13]. This results in different ML pipelines, which are evaluated on the accuracy, computational complexity, and engineering effort.

The paper is motivated by the observation that in research, industry and academia, there is a tendency to apply Deep Learning models to problems, where less computationally expensive and more interpretable models possibly lend themselves. We argue that – if the problem setting is appropriate – one should also evaluate alternatives and at least utilise them as benchmarks. This is formulated into the following research question guiding this paper: *Are non-Deep Learning approaches competitive to Deep Learning for sensorless detection and classification of faults in electromechanical drive systems?* Wherefore the following contributions are made:

1. Creation of six different ML models for fault detection and classification alongside their evaluation in terms of generalisation, engineering effort, and computational complexity.
2. Dimensionality reduction techniques RFE and PCA are evaluated for the given data set in terms of accuracy and training time, showing huge potential in reducing the feature set.
3. Systematic evaluation of different ML pipelines, showing a procedure applicable to similar problems.

2. Problem Setting: Fault Detection and Classification in Electromechanical Drive Systems

The data set used in this work contains measured phase-related motor currents for fault detection in electromechanical drive systems. The data is obtained in the research project *AutASS* [2]. There, methods are explored to utilise drives as sensors and to draw conclusions about the operated engines in terms of self-diagnosis by analysing the drive behaviour. The aim is to integrate the diagnostic functions into the drives utilising sensor and information methods without using additional sensors. Various errors and damages in the components of the drive train are simulated and

evaluated. Reasons for their occurrence are e.g. wear, incorrect assembly, or overload. Only the respective phase-related motor currents are evaluated. The basis for data acquisition is a demonstrator that can be operated under different operating conditions, i.e. different bearing loads, torque loads, and speeds. Various intact and defective components can be installed, and the corresponding motor currents can be recorded. A test bench with a 425 W permanent magnet synchronous motor, axles, bearings, and load transducers is used to simulate different defect drive states and produce the required data. A test module is used to generate different types of faults and damage. For the individual damage conditions, reference measurements with different operating parameters form the data basis for the classification. More information about the measurement procedure and a detailed description of the demonstrator can be found in [21]. For the data set the fault-free state and 10 different fault states are examined, where each state is measured three times at a total of 12 different operating conditions, i.e. at various speeds, load torques, and load forces.

To enable the classification of different operating states, suitable features are calculated from the time signals of the synchronous motor. The decomposition of the current signals by means of Empirical Mode Decomposition (EMD), part of the Hilbert Huang Transformation [16], which is often used for time series analysis, forms the basis for feature generation. The original signal is decomposed into so-called intrinsic modal functions (IMF), and residuals, each of them contains a frequency contained in the signal. Each IMF and each residual is further divided into intervals related to the mechanical properties of the drive, from which, in turn, the statistical features empirical mean, standard deviation, skewness, and excess are determined. The interval length depends on the motor speed, i.e. the higher the speed, the more samples an interval has. It is shown that already the first three IMF of the two current phases and their residuals are sufficient as a basis for the generation of features. For a detailed description of the EMD process and the feature extraction procedure, please refer to [4]. The generation of the features of the data set is shown schematically in Fig. 1.

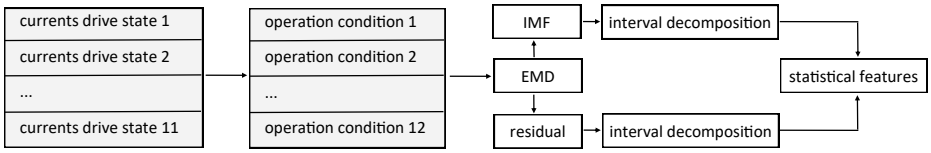

Fig. 1: Schematic representation of the generation of the database.

The data set consists of 48 features, which are labeled as one of 11 drive states. The problem setting is to classify the feature vectors, deduced from the motor currents, as shown in Fig. 1, as one of the 11 drive states.

Nomenclature

A	Full data set with 11 classes, 58,509 feature vectors, and 48 dimensions per feature vector ($n = 48$).
M	Training (or modeling) set with raw features ($n = 48$).
T	Test set with raw features ($n = 48$).
$M/T_{RFE_{25}}$	Training/test set with optimal feature set ($n = 25$) according to RFE.
M/T_{RFE_9}	Training/test set with an nearly optimal feature set ($n = 9$) according to RFE.
$M/T_{PCA_{11}}$	Training/test set with 11 components inferred using PCA to capture 0.99 variance.
M/T_{PCA_5}	Training/test set with 5 components inferred using PCA to capture 0.95 variance.
ANN-3	Artificial Neural Network with 3 hidden layers trained for the given problem setting.
ANN-20	Artificial Neural Network with 20 hidden layers trained for the given problem setting.

3. Related Work

Data-driven fault detection can be addressed with ML methods from the field of anomaly detection. A common approach is to use one-class classification [32], where the models are trained on normal data and in the inference step, new data is classified as either normal or as a potential fault. Alternatively, two-class or multi-class classifiers can be trained on the normal state and the faults, which – if representative measurements of the fault states exist – typically yields better accuracies. In the case that an

unrepresentative set of faults is available during training, an extensible one-class classifier like SVDDneg [32] can be used or an ensemble of one-class and two-class classifiers as proposed in [33]. For the problem addressed in this paper, a data set with a representative set of fault states is available, i.e. the problem in this paper is addressed with multi-class classification.

The data set [3] used in this paper was previously addressed e.g. in [4, 5, 24], yet with different approaches and goals. [24] and [5] evaluate a Modified-Fuzzy-Pattern-Classifier in combination with Linear Discriminant Analysis (LDA) and Proper Orthogonal Decomposition to detect known classes. [24] additionally presents an approach that can detect known classes using EMD and LDA. Both also demonstrate the detection of previously unknown motor faults using a fuzzy classification approach. In contrast, this work focuses on the classification of known classes, wherefore, an exploration of unknown motor states is not part of the scope. Moreover, in this work, several ML models are evaluated regarding their generalisation, computational complexity, and engineering effort. In [4] different preprocessing techniques (LDA, Prediction Analysis for Microarrays (PAM), and RFE) are compared on a subset of the classes ($n = 4$) using an SVM classifier, whereby the LDA and the RFE techniques yield similar results. In this work, we systematically evaluate the impact of RFE and PCA on the classification performance using all classes. The impact on four classifiers is compared, whereby the RFE technique performs considerably better.

There are various options for condition monitoring on drive systems such as vibration monitoring, noise monitoring, temperature monitoring, or torque monitoring [10, 17, 31]. However, these methods require additional sensor systems to observe the conditions. Another opportunity is offered by the motor current signature analysis (MCSA) [35]. The current signal is transformed into the spectrum by a spectrum analyser or a specialised MCSA instrument. Subsequently, these spectra are further analysed with e.g. the Wavelet Transform (WT) or Fast Fourier Transform (FFT) [23]. On a related problem, the fault detection for three-phase induction motors with binary classifiers, [11] shows that SVMs may perform better than a seven layered or a fifty layered Deep Learning model. Also, the integration of feature reduction techniques is suggested for future work, as their work only focuses on raw data. The present paper extends the work of [11], as in addition to an SVM, different types of classifiers are evaluated. Moreover, we distinguish between ten fault states and one normal state instead of a binary classification. Also, we evaluate feature reduction in the context of Deep Learning vs. non-Deep Learning techniques.

The competitiveness of traditional ML methods in comparison to Deep Learning methods is shown empirically by [8] and [22] on multiple datasets. Their results show that non-Deep Learning methods can perform better than Deep Learning approaches, but the best solution is always problem specific. In the present paper, traditional, ensemble, and Deep Learning methods are compared in terms of performance (generalisation), computational complexity, and engineering effort, concluding high competitiveness of traditional and ensemble methods for the fault detection and classification in electromechanical drive systems.

4. Approach: Machine Learning for Sensorless Fault Detection and Classification

In this section, the investigated ML pipelines are introduced. They include of different methods of data scaling and dimensionality reduction and the traditional, ensemble, and Deep Learning models. The following ML methods are utilised for the classification of the normal and fault states in the underlying data set of measurements of the motor current:

- Traditional methods: KNN, SVM (see e.g. [1])
- Ensemble methods: Random forests [7], extreme gradient boosting machines (XGBoost) [9]
- Deep Learning [20]: Two fully-connected feed-forward ANNs with three (ANN-3) and 20 (ANN-20) hidden layers.

For the implementation, the python libraries scikit-learn[1] (KNN, SVM, Random forests), XGBoost[2] and Tensorflow[3] (ANNs) are used. For the traditional and ensemble methods, the effect of training on the raw data in contrast to a lower-dimensional feature space deduced with PCA and RFE is evaluated. For the two Deep Learning models, no feature engineering is conducted. This results in different ML pipelines, as shown in Fig. 2.

4.1. Data Preparation

Three data preparation steps are performed: The data set is split into a train and a test set. Thereafter, the best feature scaling for each classifier is determined (see Sec. 4.1.1). Additionally, for the non-Deep Learning models, the dimensionality reduction techniques RFE and PCA are applied on M (see Sec. 4.1.2), resulting in the data sets given in Fig. 3.

[1] https://scikit-learn.org
[2] https://xgboost.readthedocs.io/en/latest
[3] https://www.tensorflow.org

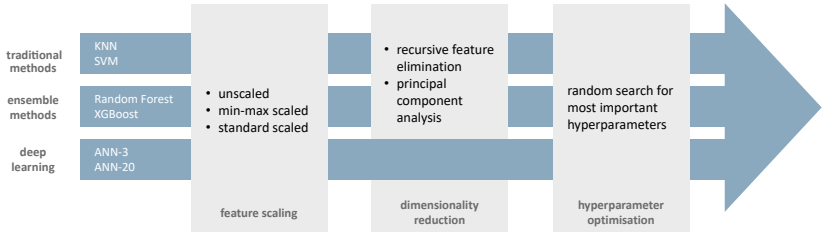

Fig. 2: ML pipelines used for the different ML models.

The full data set A is separated into a training (or modeling) set M and a test set T in order to compare the performance of the different classifiers. The feature extraction transformed the time series to statistical features: hence a random split into train and test set is conducted such that the train set contains $\frac{2}{3}$ and the test set $\frac{1}{3}$ of the data. The class distributions are not skewed, in addition it was made sure that there is no class imbalance in the train and test set. There is no evidence that requires weight-specific fault types. Consequently the accuracy metric is utilised to measure the performance of the classifiers [29].

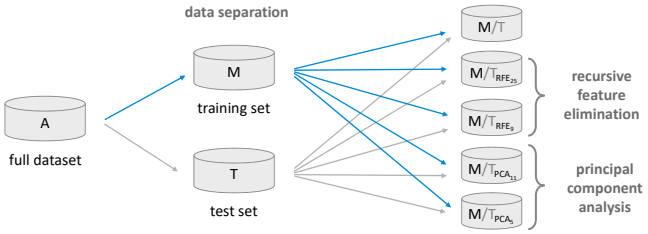

Fig. 3: Overview of all data sets created in the preprocessing phase through the preprocessing steps Data Separation and Dimensionality Reduction (RFE and PCA). (The Feature scaling step is not shown for clarity of illustration.)

4.1.1. Feature Scaling

The 48 features vary in their value ranges. As some of the classifiers rely on homogeneous feature ranges, the features are scaled for the classifiers to work properly. In order to determine the best feature scaling, each classifier's validation accuracy using its default hyperparameters[4] is determined from the training set M, the normalised M and the standardised M. As shown in Fig. 4, scaling has the highest impact on the performance of the traditional methods. In particular, the KNN classifier is highly dependent on scaled features due to the distance-based neighbour determination. As both ensemble methods rely on tree-based techniques,that consider only one feature at a time, the scaling has no impact on the performance of the Random Forest and the XGBoost classifier. For both classifiers, the unscaled data set is used. The effect of different feature scalings is low for both ANNs.

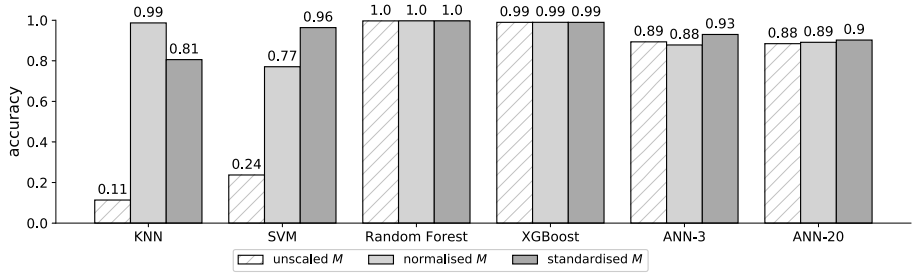

Fig. 4: Accuracy for all of classifiers on the raw, normalised, and standardised validation set of M (5-fold cross-validation).

Their performance differs only slightly between the unscaled and the normalised or standardised data. Nevertheless, the ANNs as well as the SVM, perform best on the standardised data. Only the KNN classifier performs best on the normalised data set. All

[4] For the *non-optimised* hyperparameters the default configuration of the *scikit-learn* (`https://scikit-learn.org/`) Python library is used.

classifiers, even though their hyperparameters are not optimised, achieve high accuracy on at least one of the data sets (≥ 0.9). Both ensemble methods and the KNN classifier achieve a very high validation accuracy (≥ 0.99) on the training set.

4.1.2. Dimensionality Reduction: RFE and PCA

The dimensionality reduction methods RFE [13] and PCA [25] are evaluated for the traditional and ensemble methods. By using the RFE technique forward-sequentially, the most important features are determined. The XGBoost classifier is used to rank the features by their importances. The five-fold validated accuracy on the train set M for each RFE iteration is displayed in Fig. 5 (left). While the accuracy increases fast for the first selected features at the beginning, two numbers of features stand out from the RFE results:

1. The best accuracy is achieved when using 25 of the 48 features. This data set is referred to as $M_{RFE_{25}}$.
2. The accuracy is only slightly worse when using 9 instead of the 25 optimal features. The corresponding data set is referred to as M/T_{RFE_9}.

In order to determine whether new features perform better than just eliminating existing features, the PCA technique is applied to obtain the dimensions with the highest explained variance. The variance captured by the principal components of the PCA-reduced data representation can be seen in Fig. 5 (right). Two PCA data sets are built using the declared variance thresholds 0.95 and 0.99. The resulting data sets consist of five ($n = 5$, M_{PCA_5}) and eleven ($n = 11$, $M_{PCA_{11}}$) principal components.

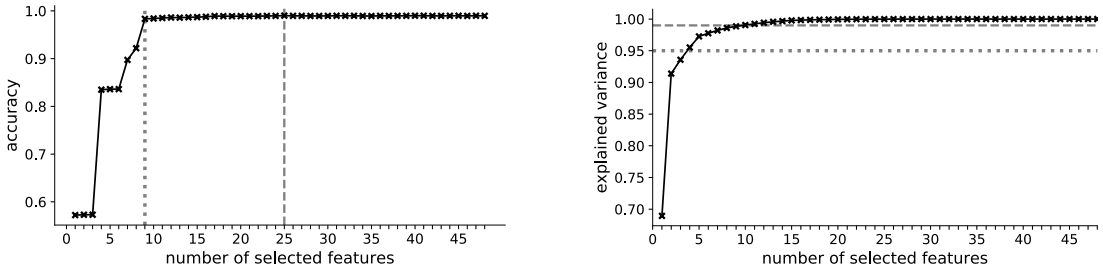

Fig. 5: Application of dimensionality reduction techniques to the training set M. Left: Five-fold validated accuracy of the XGBoost classifier on forward-sequentially selected features using RFE. Right: Percentage of variance for different number of components created using PCA.

4.2. Hyperparameter Optimisation for Machine Learning Models

For the traditional, the ensemble, and the Deep Learning models, the hyperparameters are optimised on the (scaled) variant of the data set, as discussed in Sec. 4.1.1.

4.2.1. Traditional and Ensemble Methods

For the traditional and ensemble methods, different data sets and hyperparameter ranges are evaluated. The optimised hyperparameters, their search ranges, and their respective best data set are shown in Table 1. As a baseline classifier, the KNN is used, which determines a data point's k nearest neighbours and assigns the majority class [14]. The best accuracy with KNN was achieved on the $M_{RFE_{25}}$ data set with k=1 neighbours. Support vector machines have been widely employed for fault detection (e.g. [33, 34]) and are hence evaluated. An SVM describes the decision with support vectors [14]. The RBF kernel is used, in order to allow for non-linear decision functions. For the SVM, the regularisation parameter C and the kernel parameter γ, which determines the sensitiveness per sample, are optimised [14]. The SVM performs best on the $M_{RFE_{25}}$ data set with the hyperparameters C=100 and γ=0.1.

With XGBoost[9], an advanced model combining the benefits of tree methods with ensembles is employed, which uses optimised gradient-boosted decision trees to classify samples. We consider the hyperparameters *max_depth*, which represents the depth of a tree, and γ, which sets a threshold up to which a partition of a tree's leaf is performed. XGBoost performs best on the full data set M with the hyperparameters *max_depth*=7 and γ=4. The random forest classifier is evaluated due to its simplicity and efficiency when applied to high-dimensional data. A random forest is a tree-based method, that relies on multiple trees classifying a sample based on a majority vote [7]. The randomness is introduced in the training where for each tree, a random subset of the features is considered [7]. The hyperparameter *n_estimators* controls the number of trained trees, the *criterion* parameter defines how the quality of a partition is measured, and the *min_samples_split* parameter determines the minimum number of samples required for a

Table 1: Hyperparameter optimisation for traditional and ensemble methods.

classifier	best data set	optimised hyperparameters	hyperparameter ranges
KNN	$M_{RFE_{25}}$	$n\_neighbors = 1$	$n\_neighbors \in [1,10]$
SVM (RBF)	$M_{RFE_{25}}$	$C = 100$ $\gamma = 0.1$	$C \in [0.01, 1000]$ $\gamma \in [0.01, 1000]$
XGBoost	M	$max\_depth = 7$ $\gamma = 4$	$max\_depth \in [1,10]$ $\gamma \in [1,5]$
Random Forest	M	$n\_estimators = 200$ $criterion = entropy$ $min\_samples\_split = 6$	$n\_estimators \in [10, 500]$ $criterion \in \{gini, entropy\}$ $min\_samples\_split \in [2,6]$

split. The random forest classifier performs best on the full data set with 200 trees (*n_estimators=200*) using the information gain criterion (*criterion=entropy*) and six minimum samples to perform a partition (*min_samples_split*=6).

4.2.2. Deep Learning Methods

ANNs rely on a weighted connection between neurons [14]. The neurons are organised in layers and have activation functions determining a neuron's output based on its inputs [14]. By utilising back-propagation, the weights of the connections are updated in order to optimise the network [14]. For the ANN-3 classifier, different numbers of neurons per layer are considered, while for the ANN-20 classifier, each of the 20 hidden layers has the same number of neurons. For the ANNs the following five hyperparameters are considered: the optimisation algorithm (*optimiser*), the way of randomly initialising the weights (*init*), the number of training repetitions on the data set (*epochs*), the activation function for each neuron (*activation*), and the number of neurons per layer. Hyperparameter tuning within the ranges given in Table 2 yields the following parameters: Both ANNs perform best using the Adam optimiser. The weights of the ANN-3, are initialised with a normal distribution, while the weights of the ANN-20 are initialised using the *glorot_uniform*. For the ANN-3 the *tanh* activation function performs best while for the ANN-20 the rectifier activation performs best. Furthermore, the best number of epochs varies: The best ANN-3 is trained for 100 epochs, the best ANN-20 is trained for 150 *epochs* (see Table 2).

Table 2: Hyperparameter optimisation for Deep Learning models.

parameter	searched range	ANN-3	ANN-20
optimiser	{*Adam*, *RMSprop*, *SGD*}	*Adam*	*Adam*
init	{*glorot uniform*, *normal*, *uniform*}	*normal*	*glorot uniform*
epochs	[50,300]	100	150
activation	{*tanh*, *ReLu*, *sigmoid*}	*tanh*	*ReLu*
neurons	[25, 120]	90, 120, 90	30 (per layer)

5. Evaluation

5.1. Impact of Preprocessing Techniques

The impact of the preprocessing techniques on the traditional and ensemble classifiers is evaluated for the training time and the accuracy (see Fig. 6). The training time is measured for each data set on 100 exemplary training iterations of the random forest classifier. The classifier has the longest training time on the training set M, followed by the RFE data sets. On the PCA data sets, the shortest training time is required for the training iterations. For both the PCA and the RFE data sets, the corresponding data set with fewer features requires less training time. The classifier takes more time on the M_{RFE_9} data set than on the $M_{PCA_{11}}$ data set, even though it has less number of features. In order to measure the impact of the different preprocessing techniques on the classifiers' performance, we compare the mean accuracy of all classifiers on each of the test data sets. For all data sets, the classifiers achieve high accuracy (above > 0.98). The $T_{RFE_{25}}$ data set clearly stands out from the others for two reasons: (1) All classifiers achieve higher accuracy on this data set (mean accuracy is 0.05 higher than for other classifiers) and (2) the deviation between all classifiers is the lowest for this data set. Both data sets, created using the RFE technique, achieve higher accuracy compared to the PCA data sets. The raw test set T achieves a slightly worse mean accuracy than the RFE data sets but shows a higher deviation between the classifiers.

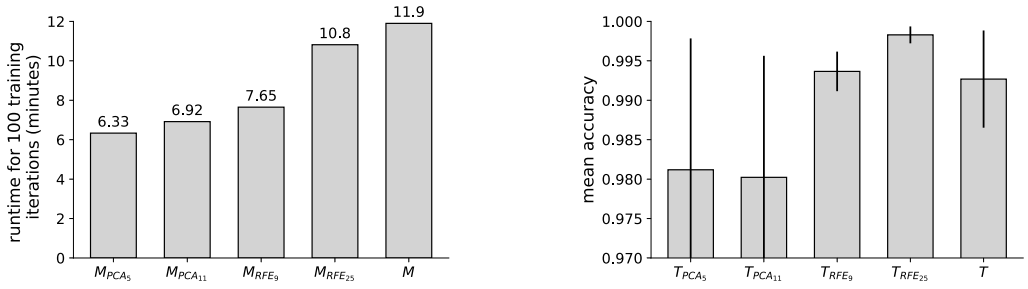

Fig. 6: Comparison of training times and accuracies between the different representations of the data set. Left: Training time for 100 iterations of the random forest classifier on the original training set M and the training sets with reduced dimensionality. Right: Mean accuracies and deviation of all traditional and ensemble method classifiers on the test data set (T) aggregated per data set.

5.2. Comparison of Machine Learning Approaches

In addition to the typically reported *generalisation* capability, we compare the utilised ML approaches by the criteria *computational complexity* and *engineering effort*. The criteria are reported qualitatively (*high*, *moderate*, *low*), allowing for an overview given in Table 3. KNN stands out because of its good performance in all criteria. Additionally, the ensemble methods XGBoost and Random Forest perform well due to their moderate computational complexity and engineering effort.

Generalisation. To indicate which classifiers generalise best, the validation accuracy on the train set M is compared to the accuracy on the test set (see Fig. 7). For each classifier the respective best data set is used (see Sec. 4.2). KNN, XGBoost, and Random Forest achieve the highest accuracy on the test set (≥ 0.998). For KNN, XGBoost, Random Forest, and ANN-20 the accuracy on the test set varies only slightly from the train set accuracy, while the SVM and ANN-3 generalise slightly worse. The complex ANN (ANN-20) achieves the lowest accuracy compared to the other classifiers, whereby the less complex network (ANN-3) performs similar to the other methods. This does in no way indicate that ANNs perform worse in general, but rather that no satisfactory hyperparameters or weights were found during training. The test set accuracies are categorised into *high* (> 0.9975), *moderate* (0.9975...0.995), and *low* (< 0.995) and are reported in Table 3.

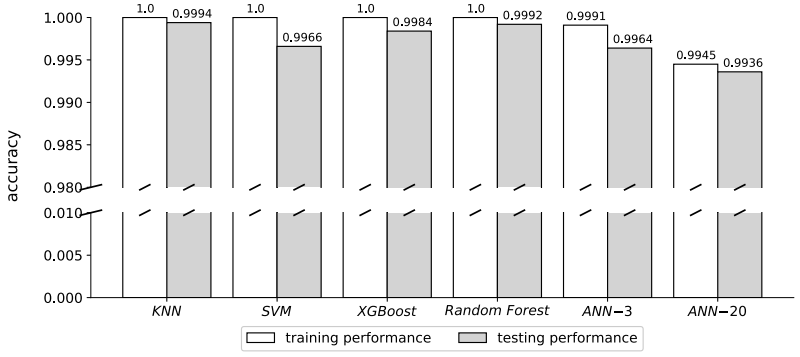

Fig. 7: Overview of all classifiers' validation accuracies on train set and accuracies on test set.

Computational complexity. The computational complexity of the models is essentially determined by the dimensionality of the hyperparameter space and the amount of required training data. As shown in Table 1, for KNN (rated as *low*) a single hyperparameter is tuned. For SVM and XGBoost two hyperparameters and for Random Forests, three hyperparameters are tuned. The aforementioned classifiers are hence rated as *moderate*. ANNs require to tune a high number of hyperparameters (see Table 2) using a large training set. Their computational complexity is hence rated as *high*.

[4] The classifiers are trained on a *D16s v3* Azure Windows VM with 16 virtual CPUs and 64 GB RAM.

Engineering effort. In practical applications, in addition to a potentially high accuracy, the engineering effort is of high relevance. Under this term we summarise the number of hyperparameters, the amount of required training data, required preprocessing, and the development effort to achieve a first viable solution. Since these are hard to quantify, we assess the engineering effort in a qualitative manner.

For the KNN classifier, the best scaling method and one parameter have to be determined, and the effort is consequently rated as *low*. An SVM with RBF-kernel has two parameters, and the results are sensitive to a good choice of these. The utilised ensemble methods achieve high accuracy in their default configuration without scaling. In addition, their accuracy can be improved with a small number of hyperparameters. Hence, engineering efforts of SVM, XGBoost, and Random Forest are rated as *moderate*. Despite the fact that most of the traditional and ensemble methods work out-of-the-box, a dimensionality reduction is helpful to tune their performance. The Deep Learning models have a higher number of hyperparameters and require a large training set and long training times to tune the weights. Therefore their engineering effort is rated as *high*.

Table 3: Qualitative overview of the evaluation criteria generalisation, computational complexity, and engineering effort for all classifiers.

approach	classifier	generalisation	computational complexity	engineering effort
traditional methods	KNN	high	low	low
	SVM	moderate	moderate	moderate
ensemble methods	XGBoost	high	moderate	moderate
	Random Forest	high	moderate	moderate
Deep Learning	ANN-3	moderate	high	high
	ANN-20	low	high	high

6. Conclusion and Outlook

In this paper, we evaluate different ML models, together with preprocessing steps for the detection and classification of faults in electromechanical drive systems. In order to determine the best-suited classifier for this problem, representative classifiers for three types of approaches (traditional methods, ensemble methods, and Deep Learning) are considered. For the traditional and ensemble methods, the effect of training on raw data in contrast to a lower-dimensional feature space is evaluated. We show that the traditional and ensemble methods achieve equal or better results for the problem at hand with less engineering effort compared to the Deep Learning approach.

Given the limited range of classifiers and their respective parameters, further research focusing either on other classifiers or more extensive parameter optimisation should be done to verify our results. The latter could specifically improve the performance of the ANNs. We show that the classifiers' performance for the traditional and ensemble methods can be improved by applying RFE to the given data set, whereby applying PCA leads to worse performance. Those findings could be complemented by a comparison of other dimension reduction methods like LDA or t-SNE. The use of other approaches, regarding classifiers or dimensionality reduction, might lead to different results, meaning that PCA might perform better than RFE when used with certain classifiers.

While the utilisation of Deep Learning might seem attractive due to the absence of manual feature engineering, we demonstrated that for this problem settings, the usage of traditional and ensemble methods could, despite the additional feature engineering, result in less effort and even a better classifier performance. This finding may be a result of the underlying data set, which seems to be rather easy to separate as all classifiers show high accuracies. This might indicate that the data set does not allow the ANNs to play to their strengths. In addition to the ensemble methods – usually more robust than individual classifiers – the straightforward and traditional approach, KNN appears to be the right candidate for the given underlying problem. On the downside, KNN has a higher computational effort during the inference step on large data sets.

Addressing the initial research question of whether non-Deep Learning methods are competitive for fault classification, the results of the experiment indicate that this is the case for the given problem. While it was neither the aim nor the scope of the work to make general statements, this suggests to at least consider, e.g., advanced tree-based ensemble methods, in addition to Deep Learning for similar problems. More data sets should be considered in future work in order to investigate whether general criteria can be identified, which indicate which problem characteristics traditional or ensemble approaches are competitive to Deep Learning.

References

[1] Abe, S., 2010. Support Vector Machines for Pattern Classification (Advances in Pattern Recognition). 2 ed., Springer-Verlag London Ltd.

[2] AutASS, 2011. Autonomous Drive Technology by Sensor Fusion for Intelligent, Simulation-based Production Facility Monitoring and Control: BMWi-funded Research Project, Grant Number: 01MA09006A. URL: https://www.th-owl.de/init/en/forschung/projekte/b/filteroff/129/single.html.

[3] Bator, M., 2015. UCI machine learning repository. URL: https://archive.ics.uci.edu/ml/datasets/Dataset+for+Sensorless+Drive+Diagnosis.

[4] Bator, M., Dicks, A., Mönks, U., Lohweg, V., 2012. Feature extraction and reduction applied to sensorless drive diagnosis, in: Hoffmann, F. (Ed.), Proc. / 22. Workshop Computational Intelligence. Hannover and Karlsruhe. Schriftenreihe des Instituts für Angewandte Informatik - Automatisierungstechnik, Universität Karlsruhe (TH), pp. 163–177.

[5] Bayer, C., Bator, M., Enge-Rosenblatt, O., Mönks, U., Dicks, A., Lohweg, V., 2013. Sensorless drive diagnosis using automated feature extraction, significance ranking and reduction. 18th IEEE Int'l Conf. on Emerging Tech. and Factory Automation (ETFA), Cagliari, Italy.

[6] Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization. Journal of Machine Learning Research 13, 281–305.

[7] Breiman, L., 2001. Random forests. Machine Learning 45, 5–32. doi:10.1023/a:1010933404324.

[8] Caruana, R., Niculescu-Mizil, A., 2006. An empirical comparison of supervised learning algorithms, in: Proceedings of the 23rd int'l conf. on Machine learning - ICML 06, ACM Press. doi:10.1145/1143844.1143865.

[9] Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA. p. 785–794. doi:10.1145/2939672.2939785.

[10] Choi, S., Haque, M.S., Tarek, M.T.B., Mulpuri, V., Duan, Y., Das, S., Garg, V., Ionel, D.M., Masrur, M.A., Mirafzal, B., Toliyat, H.A., 2018. Fault diagnosis techniques for permanent magnet ac machine and drives—a review of current state of the art. IEEE Transactions on Transportation Electrification 4, 444–463.

[11] Coelho, D., Barreto, G., Medeiros, C., Santos, J., 2014. Performance comparison of classifiers in the detection of short circuit incipient fault in a three-phase induction motor. doi:10.1109/CIES.2014.7011829.

[12] Eiermann, K.I., Vornholt, C., Blasco, J., 2015. Industry 4.0, urban development and German international development cooperation. Acatech Position Paper, Herbert Utz Verlag GmbH, München.

[13] Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. Machine learning 46, 389–422. doi:10.1023/a:1012487302797.

[14] Han, J., Kamber, M., Pei, J., 2012. Data Mining - Concepts and Techniques. 3 ed., Morgan Kaufmann Publishers.

[15] Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780. doi:10.1162/neco.1997.9.8.1735.

[16] Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N.C., Tung, C.C., Liu, H.H., 1998. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 454, 903–995. doi:10.1098/rspa.1998.0193.

[17] Isermann, R., 2011. Fault-Diagnosis Applications: Model-Based Condition Monitoring: Actuators, Drives, Machinery, Plants, Sensors, and Fault-tolerant Systems. Springer, Heidelberg.

[18] Kingma, D., Ba, J., 2014. ADAM: A method for stochastic optimization. International Conference on Learning Representations .

[19] LeCun, Y., Bengio, Y., 1998. The handbook of brain theory and neural networks, MIT Press, Cambridge, MA, USA. chapter Convolutional Networks for Images, Speech, and Time Series, pp. 255–258.

[20] LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–44. doi:10.1038/nature14539.

[21] Lessmeier, C., Enge-Rosenblatt, O., Bayer, C., Zimmer, D., 2014. Data Acquisition and Signal Analysis from Measured Motor Currents for Defect Detection in Electromechanical Drive Systems. doi:10.13140/2.1.3499.3289.

[22] Lim, T.S., Loh, W.Y., Shih, Y.S., 2000. Machine Learning 40, 203–228. doi:10.1023/a:1007608224229.

[23] Miljković, D., 2015. Brief review of motor current signature analysis. CrSNDT Journal 5, 14–26.

[24] Paschke, F., Bayer, C., Bator, M., Mönks, U., Dicks, A., Enge-Rosenblatt, O., Lohweg, V., 2013. Sensorlose Zustandsüberwachung an Synchronmotoren.

[25] Pearson, K., 1901. Liii. on lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2, 559–572.

[26] Rawat, W., Wang, Z., 2017. Deep convolutional neural networks for image classification: A comprehensive review. Neural Computation 29, 1–98. doi:10.1162/NECO_a_00990.

[27] Sak, H., Senior, A.W., Beaufays, F., 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling, in: Interspeech.

[28] Scardapane, S., Comminiello, D., Hussain, A., Uncini, A., 2017. Group sparse regularization for deep neural networks. Neurocomputing 241, 81–89. doi:10.1016/j.neucom.2017.02.029.

[29] Sokolova, M., Japkowicz, N., Szpakowicz, S., 2006. Beyond accuracy, f-score and roc: A family of discriminant measures for performance evaluation, in: Sattar, A., Kang, B.h. (Eds.), AI 2006. Springer, Berlin. volume 4304 of *Lecture notes in computer science Lecture notes in artificial intelligence*, pp. 1015–1021.

[30] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1929–1958.

[31] Tavner, P.J., 2008. Review of condition monitoring of rotating electrical machines. IET Electric Power Applications 2, 215–247.

[32] Tax, D., Duin, R., 2004. Support vector data description. Machine Learning 54, 45–66.

[33] Theissler, A., 2017a. Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection. Knowledge-Based Systems 123, 163–173. doi:10.1016/j.knosys.2017.02.023.

[34] Theissler, A., 2017b. Multi-class novelty detection in diagnostic trouble codes from repair shops, in: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), pp. 1043–1049. doi:10.1109/INDIN.2017.8104917.

[35] Thomson, William T., C.I., 2017. Current Signature Analysis for Condition Monitoring of Cage Induction Motors: Industrial Application and Case Histories. IEEE Series on Power Engineering, Wiley-IEEE Press, USA.