Fast Predictive Maintenance in Industrial Internet of Things (IIoT) with Deep Learning (DL): A Review

Thomas Rieger[1], Stefanie Regier[2] , Ingo Stengel[2], Nathan Clarke[1]

[1] School of Computing and Mathematics, Plymouth University, United Kingdom
[2] Karlsruhe University of Applied Sciences, Germany
thomas.rieger@plymouth.ac.uk

Abstract: Applying Deep Learning in the field of Industrial Internet of Things is a very active research field. The prediction of failures of machines and equipment in industrial environments before their possible occurrence is also a very popular topic, significantly because of its cost saving potential. Predictive Maintenance (PdM) applications can benefit from DL, especially because of the fact that high complex, non-linear and unlabeled (or partially labeled) data is the normal case. Especially with PdM applications being used in connected smart factories, low latency predictions are essential. Because of this real-time processing becomes more important.

The aim of this paper is to provide a narrative review of the most current research covering trends and projects regarding the application of DL methods in IoT environments. Especially papers discussing the area of predictions and real-time processing with DL models are selected because of their potential use for PdM applications. The reviewed papers were selected by the authors based on a qualitative rather than a quantitative level.

Keywords: Predictive Maintenance, Industrial Internet of Things, IIoT, Deep Learning, Real-time, Data Streams

1 Introduction

This paper provides an analysis of selected literature applying DL techniques and Artificial Neural Networks (ANN) in the field of industrial IoT (IIoT) to produce fast predictions as required, among others, in maintenance applications. PdM attempts to predict failures before their possible occurrence to avoid unscheduled outages of machines and plants. The aim is to avoid breakdowns by their timely prediction and maximizing the service life at the same time. The predictions are based on data comprising accumulated knowledge and current conditions.

IIoT environments produce massive amounts of data. The necessity to perform data analytics on such massive data brings the characterizing features of Big Data into play, like the "5V's" volume, variety, velocity, variability, and veracity [1]. The high volume and the high complexity of data put massive demands on existing data processing techniques. Additionally, evolving data streams and real-time demands intensify the demands even more [2]. Sensors typically generate continuous streams of data. The term of data streams refers to data continuously generated typically at a high rate [3]. In fully automated industrial environments, obtaining information in real-time and react immediately becomes indispensable. In IIoT environments Machine to Machine (M2M) communication has high significance [4]. Intelligent sensors and devices not only sending data but communicating with their environment, anticipate immediate responses. In such IIoT Environments the characteristic of taking a snapshot of the entire data set and performing calculations with unpredictable response time contrasts with the demand for real-time communication and the presence of continuous flowing data streams [5]. To cope with such demands self-adaptive algorithms continuously learning and improving their models are essential. In addition, such algorithms should provide high performance and real-time behaviour. This is not only true when they are running on powerful cloud systems but also on fog and edge systems or IoT devices [6].

The methodological approach of this paper is a narrative review. The reviewed papers were selected by the authors based on a qualitative rather than a quantitative level. Papers covering the most current research for the topic fast predictions in IIoT with DL were given priority. There are many papers covering the topic of DL in (I)IoT. To the best of our knowledge, there is no paper in literature covering the specific topic of PdM in connection with DL an (I)IoT.

This review provides a classification of different DL approaches mentioned for use in industry und IoT. It also covers the topics of real-time processing and data streams in regard to the mentioned DL approaches. Techniques intended to improve the real-time and stream processing ability of different approaches mentioned in the reviewed papers are evaluated and classified. Special focus is set on the ability of the mentioned approaches to provide predictions. The paper concludes with a summary and outlook on future developments.

2 Deep Learning Approaches in Industrial Internet of Things

This section starts with a short introduction into DL and ANN. A classification of different DL methods mentioned for the use in industry und IoT will then be provided. The classification will be done by the theoretic approaches, application areas and strength and weaknesses in regard to the demands of PdM in IIoT environments. The reviewed papers covering the topics of DL methods in Cyber Physical Systems (CPS), IoT, Industry 4.0 (I4.0), as well as the topics of real-time and data stream processing.

DL can be defined as a subcategory of Machine Learning (ML) whereas ML is a segment in the field of Artificial Intelligence (AI). DL itself is often defined as a class of optimized ANNs comprising numerous layers (hidden layers). The high number of layers and neurons allow the abstraction of more complex problems and support further characteristics like the ability to unsupervised learning or automatic feature extraction [7]. Examples are Deep Neural Networks (DNN), Deep Belief Networks (DBN) or Recurrent Neural Networks (RNN).

The basic idea behind an ANN is to imitate the biological neural network in mammalian brains. Components of an ANN are neurons (in ANNs often called nodes) and connections between those nodes. The nodes are organized in layers producing non-linear output data based on the input data. The connections between the nodes transfer the output of one node to the input of another node. Weights assigned to each connection determine the relevance of the transferred signal. As in biological neural networks the output signal of a neuron (node) is ruled by a threshold function. To set up an ANN all weights have to be set to an initial value (often just simple estimates). By training the network those weights are adjusted in a holistic way following a defined learning rate to achieve a valid and balanced network. This is also often referred to as “connections developing over time with training". ANNs are known for more than 50 years and various ways have been developed since [21], [22], [23].

In [6] the following DL models are listed for IoT application: Auto-encoder (AE), RNN, Restricted Boltzmann Machine (RBN), DBN, Long Short-term Memory (LSTM), Convolutional Neural Network (CNN), Variational Auto-encoder (VAE), Generative Adversarial Network (GAN) and Ladder Net. The DL models are categorized in [6] into the three main groups of generative approaches (AE, RBM, DBN, VAE), discriminative approaches (RNN, LSTM, CNN) and hybrid (GAN, Ladder Net) as a combination of the two approaches mentioned before. This categorisation mainly refer to the underlying learning method whereas generative approaches basically follow the principle of unsupervised learning and discriminative approaches follow the principle supervised learning. Beside the definition of the required number of layers (complexity) the underlying learning method is a decisive factor for the selection of a DL approach. The categorization in generative and discriminative approaches chosen by [6] can be fundamentally found in many other works. In [6] different DL models are also categorized by their suitability in IoT applications. The relevant characteristics mentioned in [6] are the ability to work with (partially) unlabelled data (feature extraction, feature discovery), the magnitude of needed training dataset, dimensionality reduction abilities, the ability to deal with noisy data and time series data and their general performance classification. For the reduction of high dimensional data and to cope with unlabelled data [6] recommends the combination of RNN with DBN and AE. If the system is meant to make predictions like in PdM systems, DBN and AEs are often used as an upfront layer providing classified data to a subsequent RNN [6].

In case of spatial-temporal data like mobility data, RNNs are recommended because they show good results when data is developing in a sequential way. But if data also comprises long term dependencies, RNNs are not a good choice because RNNs does not memorize previous states and results [8]. An approach to handle sequential data streams from human mobility and transportation transition models containing long term dependencies (behaviours) is described in [8]. The described solution is a combination of RNN with LSTM in the form of a specialized RNN architecture. Besides the ability to handle long term dependencies the LMST also adds labelling and predictive functionality to that combination. The combination of RNN with LSTM to cope with data streams or time-series data comprising long-term dependencies (like certain behaviours or wear and tear of machineries) can be found in many other works [8], [9], [11], [18].

The paper "IoT Data Analytics Using Deep Learning" [9] describes how to select the right ANN to archive predictions from data streams and time-series data. To retrieve trends and predictions and also validate those trends and predictions in parallel by anomaly detection, a combination of LSTM with Naive Bayes models is proposed. The LSTM produces the predictions on data streams whereas the Naive Bayes model is responsible for anomaly detection performed on the results of the LSTM.

This paper also reflects on the fact that Simple Feedforward ANN (FNN) like Single-layer Perceptron (SLP) and Multi-layer Perceptron (MLP) using standard backpropagation (BP) for training are often not a good choice because they does not perform well in complex situations and on data streams with long-term dependencies. This is especially true when data streams comprise time series data and the aim of the model is to predict future events or trends. Data streams and time-series data usually have dependencies over time. Such dependencies are typical for IoT data and provide relevant insights. In simple ANNs data moves straight through the layers with the assumption that input data is independent from output data. Because of this, there is no way to remember previous input and output states (previous results). This is bad if previous data is linked to current data. Using RNN instead can archive better results in data streams and time-series data. Because the connections between nodes in a RNN are in the form of sequences or loops, it is possible to remember previous states. To avoid gradient explosions normally only a view states are remembered. Therefore only short-term dependencies are recognized. Because of this [9] recommends the application of LSTM in complex IoT environments to recognize long-term dependencies in the data. LSTM are a variant of RNN introducing memory units. Those memory units are able to remember important previous states and forget the unimportant ones [9].

To predict the behaviour of energy systems in the manner of smart grids [10] remark that more intelligent systems are necessary to produce accurate predictions on the future energy consumptions. In the paper “Deep learning for estimating building energy consumption” [10] it is stated that ANN-based prediction methods are a promising approach because of their ability to handle massive and highly non-linear time series data coming from different heterogeneous data sources (e.g. SmartMeter) and containing a lot of uncertainty (unlabelled data). In the paper [10] they benchmarked two different approaches of the RBN, namely Conditional Restricted Boltzmann Machine (CRBM) and Factored Conditional Restricted Boltzmann Machine (FCRBM), on a synthetic benchmark dataset. Based on this experiment the authors come to the conclusion that FCRBN outperforms in comparison to RNN, Support Vector Machine (SVM), as well as CBRM because of its added factored conditional history layer. A RBM is a stochastic ANN consisting of two layers, a visible layer and a hidden layer. In simple terms, the visible layer of a RBM contains a node for each possible value in the input data whereas the hidden layer defines categories of values. Because in a RBM each visible layer node is connected to any hidden layer node a RBN is good in feature classification, feature extraction and complexity reduction (by identifying the most important features). For DL RBMs can be stacked. In [10] RBM is extended by a conditional history layer (CRBM) enabling the RBN to detect long-term dependencies in time-series data. Additionally the output of one stacked CRMB layer is factored (FCRBM) to reduce the number of possible compositions.

Another paper in the field of energy management also emphasizes the very powerful forecasting abilities of DL. In [11] the application of AE and LSTM is described for predicting the power generation of solar systems. The accuracy reached by a combination of AE and LMST (Auto-LSTM) is compared to other neural networks (namely MLP) as well as to a physical model. The benchmark data is taken from 21 real solar power plants and the benchmark is taken from an experimental setup described in [11]. The following measurements are taken as benchmarks: average root-mean-square deviation (RMSD), average mean absolute error (MAE), average absolute deviation (Abs. Dev.), average BIAS and average correlation. The measured results show that all ANN- and DL-based models show far better results than the physical model. Among all ANN- and DL-based models Auto-LSTM is the best choice in this specific scenario and specific data set. The capability to extraction features in unlabelled data is mentioned as a decisive factor in making predictions.

The paper “An enhancement deep feature fusion method for rotating machinery fault diagnosis “ [12] points out the strength of AEs in feature extraction and feature learning. The paper describes how to further improve the feature learning ability with reduced influence of background noises by stacking Deep AE (noise reduction) and contractive AE (enhanced feature recognition), called deep feature fusion method.

3 Fast Predictions using DL

In many IoT applications real-time processing is essential. For example in a PdM system high latency could lead to unintentional reactive maintenance because of insufficient lead time to plan the maintenance tasks [5]. How fast real-time processing needs to be, strongly depends on the application case. According [13] in micro manufacturing systems, where vast volumes of micro parts are manufactured with high speed, the term real-time means microseconds. [13] shows that with systems for fault detection and PdM the rejection rate of the manufactured micro parts decrease by increasing processing speed [13]. In other scenarios, the term of real-time can mean seconds, minutes or hours. For example in PdM Applications for offshore wind turbines the frequency with which the data is available is mostly minutes and hours [14].

The paper "Metro Density Prediction with Recurrent Neural Network on Streaming CDR Data" [15] describes the implementation of a real-time public transportation crowd prediction system using a weight-sharing recurrent neural network in combination with parallel streaming analytical programming. Fast response time to emergent situations (e.g. entrance records in metro stations combined with telecommunication data) demand real-time analysis. The use of a powerful neural network model with strong learning capability offers a wide range of new insights but contrast with the need for fast response time. The way to meet this goal is described in [15] with three steps: a) adopting a RNN model to improve its ability to work on data streams, b) implement strategies for parallelization of RNNs and c) the use of parallel streaming analytical algorithms over a cloud-based stream processing platform. In the project described in [15] each metro station is modelled by an independent RNN. Shared layers are introduced to share weights from stations which are in similar "situations" (e.g. a downtown station during rush hour) across several models dynamically. Weight-sharing also enables co-training in parallel [15].

The application of RNNs and their many variations for fast data analytics is also recommended in [6]. Especially on typical sensor data like serial data, time-series data and data streams, RNNs can provide better performance than other models. Such sensor data is dominating in most PdM applications [1].

In order to be able to develop and permanently adapt models on massive data comprising the behaviour of people and their spatial and temporal attributes together with transportation capacities, real-time processing and real-time learning capabilities are essential. The paper [8] describes a multi-task deep LSTM learning architecture. The basic idea of this concept is not to use a joint feature vector but various LSTM tasks separated by their domain (e.g. respectively a separate task for mobility and transportation mode prediction). This architecture performs parallel learning whereas the results are aggregated depending on the intended insights [8].

Assistance systems in cars like traffic sign recognition must deliver accurate results with low latency. The paper [16] describes how to apply DNN in this field. The model of the system is continuously updated (online learning) and fed only with completely unlabelled data (raw images). A CNN with 9 layers is used for image recognition. To improve the performance of system max-pooling layers are combined with convolutional layers in an alternating way. The convolutional layers perform convolution on 2D input pixel maps. The max-pooling layer works like a pre-processor between two convolutional layers transforming the output of a preceding convolutional layer to the input of a subsequent convolutional layer by eliminating overlapping regions in the pixel maps. This eliminates redundant processing in the complex and time consuming convolutional layers. The approach described in [16] is referred to as Multi-Column DNN (MCDNN).

The paper [17] describes a real-time oriented solution for traffic sign detection and recognition. The primary focus is on the need for parallel processing because of the need to detect diverse traffic sign at the same time. In this approach also CNN is used for image processing in combination with AdaBoost to improve performance and parallel GPU processing.

Because of its memory cells LSTM models are good if data comprises long-term dependencies. If the data structure allows the separation of single entities with their specific behaviour as well as the formation of groups of entities, it could be then possible to process each entity and every group with its own neural network. This opens up parallel processing possibilities of the single neural networks. Normally each single and parallel processed neural network provides its result to an aggregation layer aggregating all outputs to an overall result. The paper “A Hierarchical Deep Temporal Model for Group Activity Recognition” [18] describes how to recognize situations in a volleyball match. One LSTM model per player predicts the behaviour of this player, remembering his previous behaviour in the match (long-term dependencies). Each single situation of the match is then modelled as a group of the players. The LSTMs are hierarchically ordered where the LSTM models of all involved players are subordinated to a scene. The scenes and the players behaviour is extracted based on images using CNN [18].

The paper [7] mentions that because of the demands for real-time processing, the organization of layers and connections have changed. Fully connected networks where each node of a layer is connected to all nodes of the subsequent layer can handle complex problems but also demand a lot computing power. Dropout all connections not really influencing the result is a strategy to reduce the complexity of a DL network, and therefore its computing demand, without affecting accuracy in a relevant manner. Besides dropout [7] also mention max pooling layers, batch normalization and transfer learning as additional strategies for performance optimization.

Despite all the mentioned papers discussing performance enhancements and real-time abilities of DL models, [19] considers that highest accuracy still stands over all in mostly all current DL projects. The paper “An Analysis of Deep Neural Network Models for practical Applications” [19] argues that numerous DL approaches described in literature are simply not suitable for practical use. This is for example because of their long processing time or excessive power consumption. In his paper he demands to spend more attention to performance issues because they are key factors in practical DL applications. The paper compares 14 different specific DL projects like AlexNet or GoogLeNet by comparing their accuracy, memory footprint, parameters, operations count, inference time, and power consumption. The paper shows that a small increase in accuracy lead to an enormous increase in computational power and computation time. It is recommended to define a maximum energy consumption for each DL project and adjust the accuracy according to it [19].

4 Conclusions

In this paper we provided a narrative review of selected literature applying DL techniques in the field of IIoT to produce fast predictions of maintenance issues. The papers have shown that the use of DL in IoT and PdM is a vital topic in industry. Many different applications are in use in practice and are constantly being developed and improved.

Frequently reported are combinations of different DL models to combine different advantages and strengths in one application. Also, the need for real-time processing of complex data and data streams has been demonstrated in certain application scenarios. This include in particular applications for predictions such as PdM. In order to increase the real-time capability, concepts of parallel DL networks using a final aggregation layer, or intermediate layers for the reduction of complexity are frequently used. Although many activities can be observed in the area of real-time processing of DL models, there are also critical voices criticizing the absolute focus on accuracy and calling for a greater focus on performance and lighter applications suitable for practical use. Almost all reports agree that a lot of research is still needed in this area.

Table 1 Summary of reviews papers with the DL-Methods mentioned

Ref.	DL-Methods	Characteristics	Typical applications
[6] Mohammadi, et al., 2018	AE, CNN, DBN, GAN, LSTM, RBM, RNN, VAE, Ladder Net	Feature extraction and dimensionality reduction of IoT Data with AE, DBN CNN for image recognition but needs large training set GAN, VAE and Ladder Net suitable for noisy data, used as classification layer for RNN to enable unsupervised learning LSTM provide good performance for data with long term dependencies RBM for feature extraction, dimensionality reduction and classification problems RNN especial for time-series data	Fault detection and predictions IoT environments Real-time and stream processing with different kinds of RNNs

[8] Song, et al., 2016	LSTM, RNN	LMST for data containing long term dependencies; time-series and IoT data Streams; LSTM adds labelling and predictive functionality in combination with RNN RNN good when sequential data and data streams	IoT, Transport, Mobility
[9] Xie, et al., 2017	LMST, RNN	LMST and RNN suitable for time-series and IoT data Streams	Predictions because of long-term dependencies in data RNN for short-term IoT applications like condition monitoring
[10] Mocanu, et al., 2016	RBM, CRBM, FCRBM	RBM for feature extraction, dimensionality reduction, classification CRBM extends RBM with long-term predictions by adding a conditional history layer FCRBM improves performance by reducing the number of possible compositions of each output layer in a stacked (C)BRM	Predictive IoT applications e.g. for smart cities or smart energy grids
[11] Gensler, et al., 2016	DBN, Auto-LSTM	DBN perform good for predictions on time-series data Auto-LSTM for predictions on time-series data, combination of AE and LSTM	Predictive IoT applications like power generation forecasts
[12] Shao, et al., 2017	AE	Good for feature extraction, unsupervised learning, noise reduction and compression (relevant feature detection), often used as pre-processing layer for complexity reduction, short-term dependencies only, not good for predictions	IoT applications like fault diagnosis
[15] Liang, et al., 2016	RNN	Adopted RNNs used for data streams and weight-sharing, as well as co-training in parallel	Applications running parallel RNNs with shared layers Cloud-based stream processing
[16] Ciresan, et al., 2012	CNN	Image recognition in real-time in combination with max-pooling layers, good for short-term dependencies, not good for predictions	Real-time and parallel processing IoT applications like traffic sign recognition
[17] Lim, et al., 2017	CNN	Image recognition in real-time in combination with max-pooling layers, good for short-term dependencies, not good for predictions	Real-time and parallel processing IoT applications like traffic sign recognition
[18] Ibrahim, et al., 2016	CNN, LSTM	CNN for Image recognition LSTM for predictions considering long-term dependencies; hierarchical LMST model for individuals and group behaviours	Recognition of individuals and groups e.g. to determine current behaviour or dynamics

Table 1 gives an overview of the reviewed papers with the DL-Methods mentioned. For each paper the characteristics (or strength and weaknesses) as well as the recommended application areas (like predictions) of the DL-Methods mentioned in the corresponding paper are summarized. Table 1 makes no statement regarding the validity of results in a quantitative way. The categorisation of the different DL models is only made in a qualitative way. This is because among all reviewed papers only in [19] concrete measured values are defined. All other papers solely provide qualitative statements. How to measure and evaluate the validity and quality of results of different DL methods is an open question [20]. So far, few approaches for measuring, evaluating and benchmarking have been developed. Moreover, those approaches are usually not verifiable as generally valid. For instance, in the case of classifications the use of accuracy estimation techniques, such as the "holdout method" or "n-fold cross-validation", can be used to evaluate performance, predictive ability and model accuracy [20]. As such, mentioned techniques divide a training set via varying approaches into data areas for learning and validation. For most models no measuring, evaluating and benchmarking concept has yet been defined. In general, the evaluation is done here by expert opinions [20]. The paper [20] points out that there is a demand for improved measuring and benchmark methods. Proven measurement methods to generate representative benchmarks are needed in order to be able to assess DL models.

The papers [1] to [5], [7], [13], [14] and [19] to [23] are not part of Table 1 because they are used as reference regarding basic statements and explanations made in this paper. These papers were not on the topic of DL methods and techniques.

References

1. Pusala, Murali, et al. 2016. Massive Data Analysis: Tasks, Tools, Applications and Challenges. Big Data Analytics. s.l. : Springer Verlag, 2016.
2. Zhang, Liangwei, et al. 2017. Sliding Window-Based Fault Detection From High-Dimensional Data Streams. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. 2017, Bd. 47, 2.
3. Krawczyk, Bartosz und Wozniak, Michal. 2015. Data stream classification and big data analytics. Neurocomputing. 2015, 150.
4. Ait-Alla, Abderrahim, et al. 2015. Real-time fault detection for advanced maintenance of sustainable technical systems. Procedia CIRP. 2015, 41.
5. Bauer, Dennis, Stock, Daniel und Bauernhansl, Thomas. 2017. Movement towards service-orientation and app-orientation in manufacturing IT. 10th CIRP Conference on Intelligent Computation in Manufacturing Engineering - CIRP ICME '16. 2017.
6. Mohammadi, Mehdi, et al. 2018. Deep Learning for IoT Big Data and Streaming Analytics: A Survey, IEEE COMMUNICATIONS SURVEYS & TUTORIALS, arXiv:1712.04301v2
7. Lee, L. N., et al. 2015. Risk Perceptions for Wearable Devices. Cornell University Library. [Online] 2015. http://arxiv.org/pdf/1504.05694.pdf.
8. Song, Xuan, et al. 2016, DeepTransport: Prediction and Simulation of Human Mobility and Transportation Mode at a Citywide Level, Center for Spatial Information Science, The University of Tokyo, Japan

9. Xie, Xiaofeng, et al. 2017, IoT Data Analytics Using Deep Learning, Key Laboratory for Embedded and Networking Computing of Hunan Province, Hunan University.
10. Mocanu, Elena, et al. 2016, Deep learning for estimating building energy consumption, Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands
11. Gensler, André, et al. 2016, Deep Learning for Solar Power Forecasting - An Approach Using Autoencoder and LSTM Neural Networks, 2016 IEEE International Conference on Systems, Man, and Cybernetics • SMC 2016 | October 9-12, 2016 • Budapest, Hungary
12. Shao, Haidong, et al. 2017, An enhancement deep feature fusion method for rotating machinery fault diagnosis, School of Aeronautics, Northwestern Polytechnical University, 710072 Xi'an, China
13. Rippel, Daniel, Lütjen, Michael und Freitag, Michael. 2015. SIMULATION OF MAINTENANCE ACTIVIES FOR MICRO-MANUFACTURING SYSTEMS BY USE OF PREDICTIVE QUALITY CONTROL CHARTS. 2015.
14. Freitag, Michael, et al. 2015. A Concept for the Dynamic Adjustment of Maintenance Intervals by Analysing Hereogenoeous Data. Applied Mechanics and Materials. 794, 2015.
15. Liang, Victor C., et al. 2016, Mercury: Metro Density Prediction with Recurrent Neural Network on Streaming CDR Data, ICDE 2016 Conference 978-1-5090-2020-1/16
16. Ciresan, Dan, et al. 2012, Multi-Column Deep Neural Network for Traffic Sign Classification, IDSIA - USI - SUPSI | Galleria 2, Manno - Lugano 6928, Switzerland
17. Lim, Kwangyong, et al. 2017, Real-time traffic sign recognition based on a general purpose GPU and deep-learning, Department of Computer Science, Yonsei University, 50 Yonsei-ro Seodaemun-gu, Seoul, Republic of Korea
18. Ibrahim, Mostafa S., et al. 2016, A Hierarchical Deep Temporal Model for Group Activity Recognition, School of Computing Science, Simon Fraser University, Burnaby, Canada
19. Canziani, Alfredo, et al. 2016, AN ANALYSIS OF DEEP NEURAL NETWORK MODELS FOR PRACTICAL APPLICATIONS, Weldon School of Biomedical Engineering Purdue University, Faculty of Mathematics, Informatics and Mechanics University of Warsaw, arXiv:1605.07678v4
20. Krawczyk, Bartosz und Wozniak, Michal. 2015. Data stream classification and big data analytics. Neurocomputing. 2015, 150.
21. Bhatia, Nidhi, et al. (2015), Deep Learning Techniques and its Various Algorithms and Techniques, International Journal of Engineering Innovation & Research, Volume 4, Issue 5, ISSN: 2277 – 5668
22. Chatfield, ken, et al. (2014), Return of the Devil in the Details: Delving Deep into Convolutional Nets, Visual Geometry Group, Department of Engineering Science, University of Oxford, arXiv:1405.3531v4
23. DEVCOONS Website, http://www.devcoons.com/literature-review-of-deep-machine-learning-for-feature-extraction/

This paper was submitted to the Collaborative European Research Conference (CERC 2019) https://www.cerc-conference.eu/