T. Kotsiopoulos, P. Sarigiannidis, D. Ioannidis, and D. Tzovaras, "Machine Learning and Deep Learning in smart manufacturing: The Smart Grid paradigm," Computer Science Review, vol. 40, p. 100341, May 2021, DOI: https://doi.org/10.1016/j.cosrev.2020.100341

Machine Learning and Deep Learning in Smart Manufacturing: The Smart Grid Paradigm

Thanasis Kotsiopoulos[a,b], Panagiotis Sarigiannidis[a], Dimosthenis Ioannidis[b], Dimitrios Tzovaras[b]

[a] *Dept. of Electrical and Computer Engineering, University of Western Macedonia, Karamanli & Ligeris Street, 50100 Kozani, Greece*
[b] *Information Technologies Institute, Centre for Research & Technology Hellas, 57001 Thermi, Greece*

Abstract

Industry 4.0 is the new industrial revolution. By connecting every machine and activity through network sensors to the Internet, a huge amount of data is generated. Machine Learning (ML) and Deep Learning (DL) are two subsets of Artificial Intelligence (AI), which are used to evaluate the generated data and produce valuable information about the manufacturing enterprise, while introducing in parallel the Industrial AI (IAI). In this paper, the principles of the Industry 4.0 are highlighted, by giving emphasis to the features, requirements, and challenges behind Industry 4.0. In addition, a new architecture for AIA is presented. Furthermore, the most important ML and DL algorithms used in Industry 4.0 are presented and compiled in detail. Each algorithm is discussed and evaluated in terms of its features, its applications, and its efficiency. Then, we focus on one of the most important Industry 4.0 fields, namely the smart grid, where ML and DL models are presented and analyzed in terms of efficiency and effectiveness in smart grid applications. Lastly, trends and challenges in the field of data analysis in the context of the new Industrial era are highlighted and discussed such as scalability, cybersecurity, and big data.

Keywords: Industry 4.0, Machine Learning, Deep Learning, Industrial AI, Smart Grid.

1. Notation

a activation function of hidden layers in Auto Encoders

ABOD Angle-Base Outlier Detection

AM Additive Models

AMQP Advanced Message Queuing Protocol

AE Auto Encoders

ANN Artificial Neural Networks

AI Artificial Intelligence

Ai...An conditional probability

ARIES smArt gRid Intrusion dEtection System

b regression coefficient

CNN Convolutional Neural Networks

CPS Cyber Physical Systems

C Definition for a class in Bayesian algorithm

DBN Deep Belief Network

DIDEROT Dnp3 Intrusion DetEction pReventiOn sysTem

DL Deep Learning

DODAG Destination Oriented Directed Acyclic Graph

e residual term

E(v,h,theta) Energy function of RBM model

ECNN Enhanced Convolutional Neural Network

ELM Extreme Learning Machine

FDI False Data Injection

Fp Activation function

fj(.) smooth terms

g(.) Smooth functions

*Corresponding author: Panagiotis Sarigiannidis
Email address: psarigiannidis@uowm.gr (Panagiotis Sarigiannidis)

GS Grid Search

H non linear transformation function

IBL Instance Based Learning

IAI Industrial Artificial Intelligence

IETF Internet Engineering Task Force

IF Isolation Forest

IoT Internet of Things

IIoT Industrial Internet of Things

k hidden layer representation of Auto Encoders

L Loss function

(l,h) loss function bounding its capacity of the learning machine

LLNs Low Power and Lossy Networks

LSTM Long Short Term Memory

LWL Locally Weighted Learning

m centroids of the clusters

MAE Mean Absolute Error

MAPE Mean Absolute Percentage Error

MBR Memory Based Reasoning

ML Machine Learning

MSE Mean Square Error

MQTT Message Queue Telemetry Transport

NB Bayesian Networks

O() notation for computational complexity

p* optimal average reward of SMART algorithm

PCA Principal Component Analysis

PEFL privacy-enhanced federated learning

PILCO Probabilistic Inference for Learning Control

phi activation function of CNN

R cost (error) function of Support Vector Machine

R(x,a) Action Values on SMART algorithm

RBM Restricted Boltzmann Machine

RC Reservoir Computing

RFE Recursive Feature Elimination

RL Reinforcement Learning

RMSE Root Mean Square Error

RNN Recurrent Neural Network

RUL Remaining Usage Life

SLT Statistical Learning Theory

SMART Semi-Markov average Reward Technique

SMOTE Synthetic Minority Over-Sampling TEchnique

SOM Self Organizing Maps

SOS Stochastic Outlier Selection

SPEAR Secure and Private Smart Grid

SVM Support Vector Machine

SVR Support Vector Regression

theta optimal control policy of PILCO

Thetak identically distributed random vectors

u Filter Vector of CNNs

V* optimal Value Function of SMART algorithm

W differantiable transformation function

w weight of the connections of a neural network

XGB XG-Boost

x input (predictors) of an algorithm

y output (target variable) of an algorithm

Z normalization factor of RBM

z Output values of Auto Encoders

2. Introduction

The large amount of data produced daily on the planet and the rise of recent exponentially growing technologies (e.g. IoT, Big Data, cloud computing) in combination with the need for faster and better production of products and services, have created a new trend in industry, the Industry 4.0. Industry 4.0 combines several technologies. CPS, IoT, cloud computing and Big Data Analytics are used to automate the production process, optimize products, reduce cost, reduce energy waste and provide useful information by analyzing the data collected from different aspects across the manufacturing enterprise, including manufacturing equipment, manufacturing process, labour activity, and environmental conditions. In general, Industry 4.0 optimizes the computerization of Industry 3.0. Once computers were introduced in Industry 3.0, due to the addition of a completely new technology, it was disruptive. Today, and as Industry 4.0 progresses in the future, machines are linked and collaborating with each other to make decisions without human involvement in the end (1),(2).

It can be easily spoken that Industry 4.0 is driven by four fundamental aspects. First, is the digitization of product and service offerings. The integration of new data collection and analysis methods, such as the expansion of existing products or the creation of new digitized products, helps companies to generate product usage data and, therefore, to refine products in order to best meet the needs of customers. Second, it is the digitization and integration of vertical and horizontal value chain. Industry 4.0 incorporates processes throughout the enterprise, such as processes in product development, production, distribution and service, while Industry 4.0 vertically covers internal operations from manufacturers to consumers and all key value chain partners. Third is the digital business models and customer access. Customer satisfaction is a multi-stage, neverending process that needs to be changed at the moment as the needs of consumers change all the time. Companies therefore expand their offerings by setting up disruptive digital business models to provide their customers with digital solutions that best suit their needs (2; 3).

By implementing the IoT technology in industry to obtain data from the manufacturing enterprise, a huge amount of data is generated. Nowadays, it is easier to handle and process this amount of data due to the growth of computational power and cloud computing. ML and DL make use of the data collected by sensors and actuators of the product line. In this way, the application of ML and DL help to reduce costs of the manual inspection personnel for defects on products and also help to reduce the cost in the total value of the production. By extracting knowledge from aggregated data, ML or DL techniques play a key role in identifying standards and patterns, producing valuable information about the state of the manufacturing equipment-manufacturing process and introducing the principles of AI in the industrial sector, forming this way the Industrial AI (4),(5).

Different levels of data analytics can be generated using the aforementioned techniques, such as predictive analytics, diagnostic analytics, prescriptive analytics and descriptive analytics. Predictive analytics use statistical models to predict the possible size of the production and/or the RUL of the machinery. Diagnostic analytics examine and report faults on the machinery and the product. Prescriptive analytics propose taking over actions to optimize the production and forecast the impact of these actions. Descriptive analytics aim to summarize and describe the conditions occurring in the manufacturing process and the manufacturing environment (6), (7). The combined usage of them, could lead to provide an automated solution to industries for maximizing the profit, identifying early possible defects in the structure of the product and predicting the cause of any defect might occur, by calculating the RUL of the machinery.

The impact of ML and DL technologies in the world is rising and promising. Applications of ML and DL can be initially found in condition monitoring of electric machines. Models for fault prediction on electric machines and rolling bearings are emerging and provide solid and accurate measures. Furthermore, applications can also be found in logistics and supply chains. As new information is presented, a connected supply chain will adapt and accommodate it. If a shipment is related to a weather delay, a connected system can proactively adjust to that reality and change priorities for manufacturing. Transportation is also another sector, where ML and DL models are applied. There are shipping yards that use autonomous cranes and trucks to streamline operations as shipping containers from ships are accepted. In addition, secure IIoT architectures are developed, to store and process scalable sensor data (big data) for health care applications. Smart Grids is another framework where models of ML and DL are applied. Due to the demand and the growth of Smart Grid applications and scenarios, this paper aims to collect and present in section vi the use cases and models developed for the Smart Grid case study.

The contribution of this paper can be summarized in three main pillars. First, we provide a new approach

for applying AI in industrial ecosystems, while we also describe and present in details, scenarios where IAI is met. Second, we present and analyze ML and DL algorithms and models utilized for the needs of IAI and provide guidelines to correctly choose a ML/DL algorithm or model under different cases. Third, we present several ML and DL algorithms and models utilized in smart grid applications.

The reminder of this paper is organized as follows. Section 3 presents the fundamental elements and the Ecosystem of the Industrial AI. In details, scenarios where Industrial AI is applied are highlighted, while we present a new architecture for applying AI in Industry. Industrial AI applications and the correlation with ML and DL technologies are also presented. In Section 4 and 5, ML and DL algorithms used in the industry are surveyed and presented. Section 6, addresses ML and DL models utilized for the needs of the Smart Grid framework. At the end, section 7 concludes this paper and discusses the challenges and trends on the fields of the Industry 4.0 and on the fields of Smart Grid.

3. Industrial Artificial Intelligence

AI was introduced to the public in 1956 and it is broadly defined as "the science of making computers do things that the human needs intelligence to do". Nowadays with the big development of cloud computing and computational power it is easy to apply AI or AI's subsets such as Machine Learning or Deep Learning, in any field like image recognition, automotive, industry (8).

In (9) - [40] are distinctively presented AI applications or potential AI applications in the industrial field. Automotive, manufacturing, financial services, healthcare and supply chains benefit from AI. The research focused on discovering the fundamental elements of IAI. First, we researched for papers containing the terms IAI, Industry 4.0, ML and DL. The analysis of these papers has risen questions about the applicable fields of the IAI. To this end, we investigated also for papers containing the terms automotive, manufacturing, financial services, healthcare and supply chain on the Industry 4.0. In a nutshell, the following paragraphs present the analysis of the investigated papers about the applicability of the IAI in the aforementioned sections.

The rapid rise of mobile bank fraud due to the increasing usage of mobile banking applications may cause profit loss, which can be translated into billion of dollars. Creating a ML model to efficiently detect anomalies in customers behavior (log-in location, money transfer) and producing descriptive analytics about anomaly detection to mobile bank applications and to e-banking applications, would be a huge step in attenuating the problem ((9), (10; 11; 12; 13; 14; 15; 16; 17)).

We should also take under consideration the Capax Global's Solution for financial institutes. Capax Global by applying Machine Learning algorithms, predict the amount of cash an ATM is expected to allocate, on any day of the week. This is important because knowing the amount of cash dispensing from the ATMs, financial institutes can store the needed amount of cash in each ATM without having to leave extra cash that can otherwise be used for lending to customers and generating a profit (9).

Healthcare is also a sector in which AI can be applicable. Data collection from patients (e.g. blood pressure, diabetes), hospitals, drug stores (e.g. availability in drugs) and doctors (e.g. files with historical medical and treatment data) can provide valuable information. Delivering them through IoT techniques to a ML algorithm, generates predictive analytics about patients, hospitals and drug stores.((18; 19; 20; 21; 22; 23; 24; 25; 26)).

Another application of AI is in the supply chain. AI brings contextual intelligence to the supply chain that can be used by them to minimize running costs to successfully manage inventory. In addition, businesses use AI to gain new insights into diverse fields, including warehouse operations, distribution and supply chain management ((27; 28; 29; 30; 31; 32; 33; 34; 35; 36; 37)).

Furthermore, as it is presented in (32), many people tend to use online shops for their needs. This has an enormous impact to the traditional shops, which struggle to keep their customers. A proposed solution, to this problem is to apply face recognition in every person entering the shop floors, so as to deliver specified information due to the customer's need. We believe that this application must be taken into further consideration because it raises questions about the security of the personal data, how and by whom the data are stored and processed.

Last but not least, AI provide new solution to mobile communications (38) and to cognitive computing (39). AI is also applicable in Industry, by estimating the RUL of the Machinery, generating predictive, diagnostic, prescriptive, and descriptive analytics about the manufacturing Enterprise ((40; 41; 42; 43)). More information about this topic are presented in Section III and Section IV respectively.

Although an interpretation of the basic components of industrial artificial intelligence and its architecture is given, a lot of research needs to be done.

As Lee et al. in (44) states, the key elements in Industrial AI can be characterized by the rule of "ABCDE". "A" stands for Analytics, which is the core of AI but it is only valuable if other elements are present. "B" stands for Big Data, which provide the data, the source of the information. "C" is about the Cloud infrastructure, providing a platform for Industrial AI. "D" is referred to Domain know-how, which is about understanding the architecture of the system and how it works. Also it is about dealing with any problems to solve and understanding the physical meanings of the information, how they are related with the machinery and how they vary from machine to machine. Last but not least, "E" is related with Evidence, which is the feedback given to the AI system in order to evaluate and improve itself.

Lee et al. also in (44) defines the Industrial AI architecture as a pyramid of six layers, presented in Fig. 1. The first layer includes the Industrial Sectors. The embedded AI Devices, the Resilient Factory, Smart Human and Health Performance, Productive Energy Systems, Worry-free Transportation and Industrial AI - based Education System. The second layer meets the needs of Industry 4.0. This layer contains all the attributes of every intelligent system, named as: self-aware, self- compare, self- predict, self-optimize, resilient. The third layer highlights the challenges an IAI System could come up with. These challenges includes Data Quality, Operational Regimes, Machine to Machine Variations, Expert Knowledge and Cyber Security. The fourth layer presents the enabling technologies for constructing Industrial AI systems: Data Technology, which refers to identifying the appropriate equipment and mechanism to acquire useful data , Analytics Technology, which refers to converting the data obtains from sensors into useful information, Platform Technology, which refers to the hardware architecture for manufacturing data storage, analysis and feedback and finally Operations Technology, which refers to a series of decisions made and actions taken based on the information extracted from data. The fifth layer includes the development tools, the Machine or Deep Learning methodology and the platforms used to develop the system. In conclusion, the sixth layer is referred to the impact an AI system could have.

Bearing in mind the proposed ecosystem described above and our research about the requirements for a system to belong in Industry 4.0, we propose a slightly different approach of the Industrial AI Ecosystem, as shown in Fig. 2.

The first layer from bottom up is about obtaining data from the environment. Collecting data from machinery in the manufacturing, data about the summation/duration/faults of the production, and data-knowledge about the procedure of the Logistics is crucial to form a solid and well performed Industrial AI system. Of course, in this level we cannot exclude the hardware components we use to obtain the data, such as sensors, actuators and Embedded AI devices.

The next layer establishes the communication connections between the industrial sectors and the proper acquisition of the data. In addition, it is responsible to store the data collected from the first layer, secure them with the best tools and processes and if needed, extract valuable features from a pre-processing procedure.

The third layer corresponds to the demands of Industry 4.0, which an industrial AI system must fulfill. A CPS should follow the proposed architecture described below, to tackle the needs of the new Industrial revolution. It consists of 5 levels: Configuration level, Cognition level, Cyber level, Conversion level and Connection level. Configuration level is about providing feedback from the user or from the action taken from the cognition level. Cognition level refers to collecting data from the machines and the machinery network and providing a comprehensive knowledge of the system. Cyber level introduces all the connected machines and operates as a central hub for data processing in the Industry 4.0. Conversion level is responsible for collecting the raw data values from the field and converting them in useful information.To conclude, Connection level refers to the correct data handling methodologies to accurate acquire the data (6).

Furthermore, the fourth layer introduces the foundations of data analytics. ML, DL, RL and Cognitive Models are the tools to produce valuable information about the manufacturing Enterprise. The available HPC platforms used to build the models (e.g. Microsoft Azure (45), Amazon Web Services (46), etc) play key role in this layer (47), as also the model building - processing toolkit, which refers to the available programming languages (e.g. python, R, C++) and libraries (Scikit learn (48), PyTorch (49), Keras (50), Theano (51), TensorFlow (52)).

The fifth and last layer includes the results of an Industrial AI system. It produces information about the conditions occurring in the manufacturing process and manufacturing environment. Propose taking over actions to optimize the manufacturing Enterprise and forecast the impact of these actions. Examines and reports faults on the machines and predicts the possible size of the production as also the RUL of the machinery.

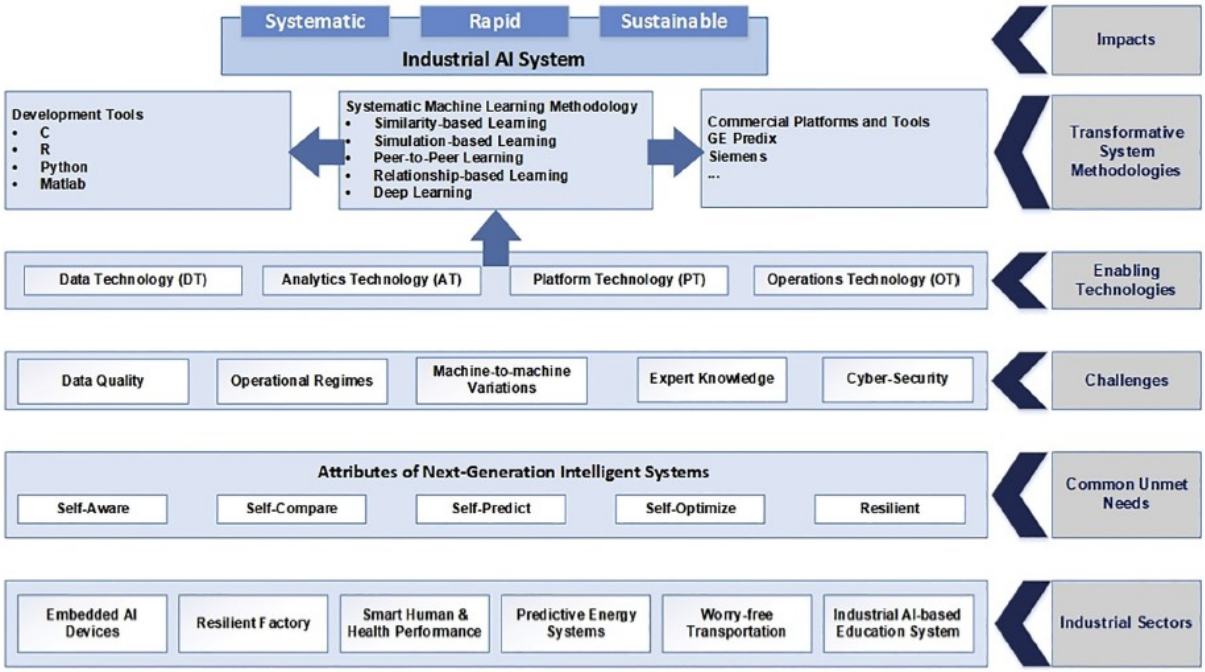

Figure 1: Industrial AI Architecture proposed by Lee et Al. in (44)

4. MACHINE LEARNING IN INDUSTRY 4.0

ML is the science of getting machines to take action without specific programming. Machine learning has brought us self-driving vehicles, functional speech recognition, successful web search and a much enhanced understanding of the human genome over the past decade. There are several ML Algorithms used in the Industrial sector. They could be categorized as supervised or unsupervised, depending on how they learn from the data. In the following paragraphs are presented the surveyed ML algorithms used in Industry. To begin with, Statistical Learning Theory (SLT) is a supervised ML framework used in Industry. SLT's major advantage is the variety of possible application in strategies and scenarios. By using SLT it is also possible to overcome the observer variability issue. It's main purpose is the best estimation of the output for previously unseen inputs. However, a large number of samples is needed to perform and the application of SLT in some cases might lead to over-fitting (53).

4.1. Bayesian Networks

An application of SLT is BNs or NBN. BNs describe the probability relationship between several variables. Similar to BN are NBNs, a simplest form of Bayesian Networks. Fig. 3 presents our proposal about the structure of NBNs. From a theoretic approach, NBNs could be described as follows. Given a class label C, the naive Bayesian classifier learns from the data (training data) the conditional probability Ai of each attribute. Classification is applied then, by utilizing the Bayes rule to compute the probability of C, given the particular instance of $A_1, ..., An$ and prediction is made about the highest class with the posterior probability. This computation is generally based on the assumption that all the Ai attributes are conditionally independent given the value of the class C. Here, independence stands for probabilistic independence, that is A independent of B given C whenever

$$P_r(A|B, C) = P_r(A|C) \tag{1}$$

for all possible values of A, B and C, whenever $Pr(C) > 0$ (54). Although BNs require limited storage, they are robust to missing values and they are easy to grasp output, the tolerance to interdependent and redundant attributes are very limited (55)-(56; 57; 58). Bayesian Networks where used in (54) to predict fraudulent samples received by the tea tasting unit of the Sri Lanka Tea Board.

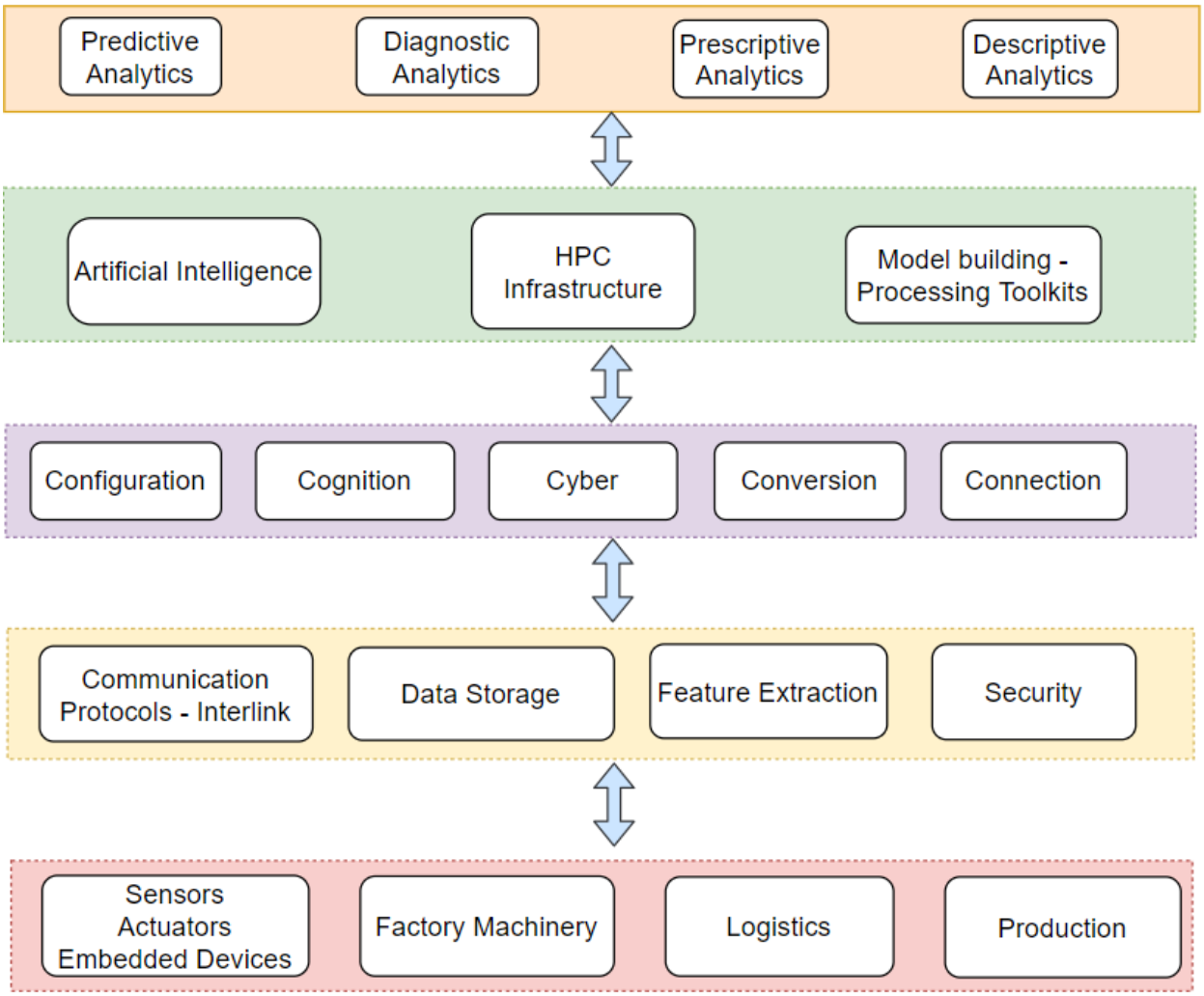

Figure 2: Our Proposal about Industrial AI Architecture

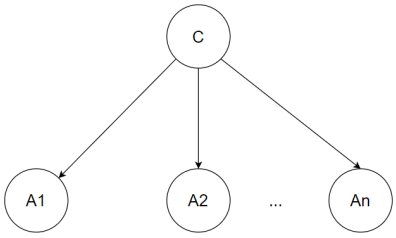

Figure 3: Structure of a Naive Bayesian Network

4.2. Support Vector Machine

SVM is an algorithm for two group classification, who could best apply the theoretical background of SLT. SVM achieve high performance,high accuracy and has the ability to handle high-dimensional multi-variate datasets (53; 59; 60; 61). In order to have a good generalization property, SVM keeps the value of training error equal to zero or equal to some acceptable level and it minimizes the confidence interval.The utilization of this approach drives to resolving the trade between under-fitting and over-fitting. The cost (error) function of the SVM is (62):

$$R = \sum_{i=1}^{l} L_e + W(l, h) \tag{2}$$

where Le is a loss function, which assert the closeness to data and $W(l, h)$ is a function bounding the capacity of the learning machine. In particular, Le is a typically 0-1 loss function but in regression problems is the Vup-

nik's ε-insensitivity loss (error) function described by the following equation:

$$Le = |y - f(x,w)|_e = 0, if|y - f(x,w)|e \leq e \quad (3)$$

or

$$Le|y - f(x,w)|_e = |y - f(x,w)| - e \quad (4)$$

where e is a radius of a tube where the regression function should lie, after successful training (63)-(64). Applications of SVM can be found in (65), monitoring machine condition and fault diagnosis. Also SVM was used in (66), to predict the physical quality of intermediate products in interlinked manufacturing processes and in (67), to calculate Steel Quenching Degree. Similar to SVM algorithm is the Support Vector Regression algorithm. The difference is that SVR is used for regression problems. It is found that SVR was used in (68) to estimate the manufacturing cost of jet engine components during the early design face.

4.3. Instance-Based Learning

Another supervised ML algorithm is Instance - Based Learning (IBL) or Memory - Based Reasoning (MBR). These algorithms are applied in regression and classification and can achieve a high classification accuracy and a solid performance. However, if a large training set needs to be processed, then there will be a huge time complexity. Also, they cannot set the weight vector in unknown or little known domains and they tend to over fitting with noisy data (53),(69; 70; 71). Table 1 describes the IBL algorithm.

Instance Based Algorithm
CD=0
For each x ∈ Training set do
1.For each y ∈ DC do
$Sim(y) = Similarity(x,y)$
2. $y_{max} = somey \in CDwithmaximalSim[y]$
3. if class(X) = class(y_{max})
then classification = correct
else classification = incorrect
4. CD=CD ∪ x

Table 1: The algorithm of the Instance Based Learning (69)

4.4. K-Nearest Neighbors

K-nearest neighbors algorithm belongs to the family of IBL algorithms and it was used in (72) to predict and estimate machine specification data, such as machine geometry and design, motor performance, range and cost. Furthermore, it was used in (67) to predict the physical quality of intermediate products in interlinked manufacturing processes and in (68), to estimate the manufacturing cost of jet engine components. Table 2 describes the K-Nearest neighbor algorithm. The Nearest Neighbor classification rule assigns an input y sample vector, which is of unknown classification. The classifier also require no pre-processing of the labeled sample set prior to their use ((73; 74; 75; 76)).

K-nearest Algorithm
Let $W = X1, X2, \Delta\Delta\Delta, Xn$ be a set of n labeled samples.
1. BEGIN
Input y, of unknown classification.
Set K, 1 <= K <= n
Initialize i = 1
2. DO UNTIL (K-nearest neighbors found)
Compute distance from y to X_i
3. IF (i! = K) THEN
Include X_i in the set of K-nearest neighbors
4. ELSE IF (X_i is closer to y
than any previous nearest
neighbor) THEN
Delete farthest in the set of K-nearest neighbors
Include X_i, in the set of K-nearest neighbors.
END IF
Increment i.

Table 2: The K-nearest algorithm (73)

4.5. Ensemble Methods

Ensemble Methods are a ML algorithm family, where a weighted committee of learners is used to solve a regression or a classification problem. The ensemble's base learners could be Neural Networks, nearest neighbor or trees. Base learners from the same algorithm family form the "Homogeneous ensemble" and base learners from different families form the "Heterogeneous ensemble". It is demonstrated that "Ensemble Methods" lead to a better generalization model than a single classifier (53).

4.6. Neural Networks

ANN "simulate the decentralized 'computation' of the central nervous system by parallel processing". Decentralization gives the ability to process information by dynamic response to external inputs.The building blog of an ANN could be expressed as (53)::

$$y_p = f_p(\sum_{i \in inputs} w_{p,i} \times x_{p,i}) \quad (5)$$

where, y_p is the target variable, $X_{p,i}$ are the predictors with associated weights $W_{p,i}$ and F_p is the activation function. Their advantages are the wide applicability, and the good handling of high-dimensional and multi-variate data. However, there might be in some cases an over-fitting of the training data, but it is acceptable in high variance algorithms. Other challenges include the intolerance in missing values, the time consuming training and the complexity of the models produced (53; 77; 78). ANN are applied in different aspects in manufacturing like semiconductor manufacturing or process control (72). Furthermore ANNs were used in (68) to estimate the manufacturing cost of jet engine components during the early design face, in (67) to predict and estimate machine specification data, such as machine geometry and design, motor performance, range and cost. Last but not least, ANNs where used in (65) to real-time monitor machine tools.

Other supervised ML algorithms used in industry are Multiple Linear Regression, Decision or Regression Tree, Gradient Boosted Trees, Additive Models, Logistic Regression, Random Forest, Bag of Words and Locally weighted training. Multiple Linear Regression, Addictive Models, Gradient Boosted Trees found to be used in (67) for the calculation of the Steel Quenching Degree. Decision or Regression Trees algorithm was used in ((67),(65),(79),(80)), while Random Forest algorithm was implemented in ((53),(81)) and in (82) to predict possible equipment stoppages (or faults) of an industrial equipment for anode production. Logistic Regression was used in (72), Locally weighted training in (83) and Bag of Words was used in (84) to identify melting pool points.

4.7. Multiple Regression-Logistic Regression

Multiple Regression describes how a single response variable Y depends linearly on a number of predictor variables X. A Multiple Regression model with n predictor variables $X_1, X_2..., X_n$ and a response Y can be stated as follow ((85; 86; 87; 88; 89)):

$$y = b_0 + b_1x_1 + b_2x_2 + ... + b_nx_n + e \tag{6}$$

where, e are the residual terms of the model and the distribution assumption we place on the residuals. This term (e) will allow later to do inference on the rest model parameters. The $b_0, b_1, b_2....b_n$ are the regression coefficients. In addition, Logistic Regression is a slightly different approach (generalized linear model) of Linear Regression. The equation that best describes its operation is (86) :

$$y = 1 \quad \text{if } b_0 + b_1x_1 + e > 0 \tag{7}$$

else

$$y = 0 \tag{8}$$

4.8. Decision Tree

Decision Tree algorithm utilize a decision tree to observe data about an instance and produce a brief conclusive report about the instance's target value. The difference with Regression Trees is that Regression Trees are used to predict continuous values, while Decision Tree is used to predict categorical values. In those type of algorithms, data stream in the form of ((79; 80; 90)):

$$(X, y) = (x_1, x_2, x_3, ...x_k, Y) \tag{9}$$

where $x_1, x_2, x_3..., x_k$ are the predictor variables and Y is the target variable ((91; 92; 93)).

4.9. Gradient Boosted

Gradient boosted is a ML algorithm for regression and classification cases. The purpose of Gradient Boosted algorithm is to iterate over the prediction of weaker models and then, sum their predictions.The equation which describes the above statement is the following ((94; 95; 96)):

$$F_{k+1}(X) = F_k(x) + h(x), 1 \leq x \leq M \tag{10}$$

where the $h(x)$ must be able to fit $y - F_{k(x)}$. The Gradient Boosted algorithm steps are shown in Table 3.

Gradient Boosted Algorithm
1. A constant value F_0 is the initial model
$F_0(X) = argmin_g \sum_{i=1}^{n} L(g_i, g)$
2.a. Generate m learner through iterations
$g_{ik} = -(\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)})_{F(x)=F_{k-1(x)}}, i = 1, 2...n$
b. By calculating g_{im} through:
$g_k = argmin_g \sum_{i=1}^{n} L(g_i, F_{k-1}(x_i) + gh_k(x_i))$
3. Renew the model
$F_k(x) = F_{k-1}(x) + g_kh_k(x)$

Table 3: The Gradient Boosted algorithm (65)

4.10. Additive Models

Additive Models (AM) are an extension of Multiple Linear Models with a basic difference. They map the target variable as a sum of non linear transformation of the input variable, not as a sum of linear terms. The equation which describes best the Additive Model is:

$$g(E(Y)) = b_0 + \sum_{j=1} f_j(X., X.) \quad (11)$$

where $g(.)$ is a smooth function and $f\,j(.)$ are complex linear functions (smooth terms). Furthermore, AM belong to the non parametric models because the smooth terms are highly non linear and they are not pertaining to any predetermined family of functions ((97; 98; 99)).

4.11. Locally Weighted Learning

Locally Weighted Learning (LWL) is a Lazy Learning method utilized for ML techniques. LWL implementation requires dealing with insufficient amount of data (mostly for training data), regularization of the estimates for measured introduction bias and methods for predicting the prediction's quality. LWL could be used for outlier detection and noise filter (100)-(101). Table 4 is presented the Locally Weighted Learning algorithm.

Locally Weighted Learning Algorithm
1. Given a query point x_p
p training points (x_i, y_i) in memory
2. Compute prediction
a) compute diagonal weight matrix W
where $w_d = exp(-\frac{1}{2}(x_i - x_q)^T D(x_i - x_q))$
b) build matrix X and vector y such that:
$X = (x_1, x_2, ..., x_p)^T$ where
$x_i = ((x_i - x_q)1)^T$
$y = (y_1, y_2, ..., y_p)^T$
c) compile locally linear model
$\beta = (X^T WX) - 1X^T WY$
d) the prediction for x_q is thus
$y_q = \beta_{n+1}$

Table 4: The LWR algorithm (83)

In the d and final step of the LWR algorithm, the β_{n+1} denotes the n+1 element of the regression vector, while the computational complexity of the algorithm is $O(pn^2)$.

4.12. Bag of Words

Bag of Words is a generic visualization ML algorithm. It corresponds to a histogram of the number of instances of image patterns in a given image. The advantages of this algorithm are computational efficiency, simplicity, invariance to affine transformations, as well as lighting, occlusion and intra-class variations (84; 102).

4.13. Random Forest

Random Forest is a collective method, which is working based on the nearby neighbor predictor. It utilizes the divide and conquer algorithm to enhance its performance. In particular, Breiman L. in (103) define Random Forest algorithm as "as a classifier consisting of a collection of tree-structured classifiers $h(x, \Theta_k), k = 1...$,where the Θ_k are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x". Random Forest algorithm fits well with limited data and has high accuracy. On the contrast, it cannot achieve good accuracy on high dimensional datasets ((104; 105; 106)).

4.14. K-means

The unsupervised ML algorithms found to be used in industry are the k-means algorithm and the Self Organizing Map. K-means algorithm aims to partition n observations in k clusters, in which each observation belongs to the cluster with the nearest mean, by minimizing the average Euclidean Distance between the point in each cluster. Afterwards, it calculates the mean vector of each cluster and reassigns each data point to the cluster with the closest mean. Table 5 presents the aforementioned described in brief algorithm, which is Lloyd's approach ((107; 108; 109)).

K-means Algorithm
For an initial set of k-means $m_1,_m 2....m_n$
proceed by alternating between two steps:
1. Assign instances to the cluster whose mean
has the least squared Euclidean Distance
$S_i^t = x_p : \|x_p - m_i^t\|^2 \leq \|x_p - m_j^t\|^2 \; \forall j, 1 \leq j \leq k$
2. Calculate the new mean cendroids of the clusters:
$m_i^{(t+1)} = \frac{1}{\|S_i^{(t)}\|} \sum_{x_j \in S_i^{(t)}} x_j$

Table 5: The k-means algorithm (110)

Kmeans are simple to implement and east to adapt to new data. However, it is noticed to cluster a lot of outliers ((111; 112; 113)).

4.15. Self Organizing Map

Self Organizing Map (SOM) is an ANN with the purpose to reduce dimensions and use competitive learning instead of error-correction learning than the other ANN. Concretely, SOM achieves clustering and dimensionality reduction by mapping inputs from a higher dimensions into a fixed number of points on a lower dimensional grid. SOMs are capable of clustering large datasets, but they are slow to training and are diffucult

to generalize (114; 115). K -means algorithm and Self Organizing Map were used in (67), to predict the physical quality of intermediate products in interlinked manufacturing processes.

4.16. SMART algorithm -PILCO algorithm

SMART and PILCO are the two Reinforcement Learning algorithms found to be used in manufacturing. SMART, which stands for Semi-Markov average Reward Technique, was used in (116; 117) to optimize a transfer line. This algorithm utilizes action values $R(x, a)$ to represent value function. To approximate the action values, one could run a simulation model of the manufacturing domain and use a feed-forward neural network. SMART algorithm can be derived through the Bellman's equation:

$$V^*(x) = max_a(r(x, a) - p^* y(x, a) + \sum_{z \in S} P_{xz}(a) V^*(z)) \quad (12)$$

where $p*$ is the optimal average reward and $V*$ is the optimal value function. Table 6 depicts the SMART algorithm.

SMART algorithm
1. Set decision epoch n=0 and initialize action values $R_n(x, a) = 0$. Choose the current state x arbitrarily. Set the total reqard c_n and total time t_n to 0.
2. While n¡MAX STEPS do a. With high probability p_n, choose an action a that maximizes $R_n(x, a)$, otherwise choose a random action.
b. Perform action α. Let the state at the next decision epoch be z, the transition time be r, and r_{imm} be the cumulative reward earned in this epoch as a result of taking action α in state x.
c.Update $R_n(x, a)$ using: $R_{n+1}(x, a) = (r_{imm} - \rho_t + max_b R_n(x, b)$
d. In case a nonrandom action was chosen in step 2(a) update total reward $c_n = c_n + r_{imm}$ update total time $t_n = t_n + r$ update average reward $\rho_n = \frac{c_n}{t_n}$
e. Set current state x to new state z, and n=n+1

Table 6: The SMART algorithm (116)

PILCO (Probabilistic Inference for Learning Control) uses a Gaussian process (GP) as a non-parametric approximation of the system and belongs to the RL Algorithms. PILCO was used in (118), to design a feedback control strategy for the swing-up of the double pendulum on a cart. PILCO's algorithm is explained in Table 7.

PILCO Algorithm
1. Select random controller parameter θ
2. Apply a random control sequence and collect data repeat
3. Learn system dynamics f by means of GP based on the existing data
4.Determine the optimal control policy $\theta = \pi(x_i, \theta*)$ by minimizing $J(\theta)$
5. Apply the control strategy $\pi(x_i, \theta*)$ and collect further data,which are addet to the existing until Control Task is fulfilled

Table 7: The PILCO's algorithm (118)

4.17. Discussion

Table 8 illustrates the ML algorithms found to be used in industry, categorized as supervised, unsupervised or reinforcement learning algorithms. Table 9 depicts the advantages and the disadvantages of the aforementioned algorithms. Before we provide our opinion about the utilization of ML algorithms for classification, clustering and regression purposes, we feel the need to define what we think as a small dataset and what as a large one. A small dataset is a set of data that can be processed for acceptable time at a regular computer. A large dataset is a set of data that need a Hyper Performance Computing in order to get processed. The authors are of the opinion that classification problems with a limited amount of data, can be handled by applying SVM or Logistic Regression. SVM and Logistic Regression are able to deliver an enhanced classification accuracy when the dataset is small. On the opposite side, RF performs well enough when we have to deal with a mid range dataset. In the case where the previously ML models cannot fit and address the high amount of data, ANNs can handle the problem really well, due to the ability of finding correlations between the independent variables without any prepossessing. When a clustering problem occurs, K-means algorithm is able to provide a good adoption to new samples. For mid range datasets, SOMs can assist the problem in a sufficient way. Regression problems can efficiently be solved with Multiple Linear Regression and Logistic Regression in a small range dataset problem. RF is also a good solution for regression problems in mid range dataset. For large datasets, Artificial Neural Networks provide a solid way to deal with these kind of data, as it is explained before. In a nutshell, we strongly suggest that all the available ML Algorithms could be utilized in the industrial sector,not only the ones mentioned above. However, they should be tested and then examined by

their performance(Accuracy, training loss, testing loss) over the problem's data-set.

Machine Learning Algorithms in Industry	
Supervised Learning	SVM/SVR Bayesian Networks/ Naive Bayesian Networks K-nearest Neighbors Artificial Neural Networks Multiple Linear Regression Decision Tree/ Regression Tree Addictive Models Logistic Regression Bag of Words Locally Weighted Training
Unsupervised Learning	K-means, Self Organizing Map
Reinforcement Learning	PILCO, SMART

Table 8: Categorization of Machine Learning Algorithms used in Industry by supervised Learning, unsupervised learning and reinforcement learning

Algorithm	Advantages	Disadvantages
SVM/SVR	high performance, high accuracy, good handling of high dimensional data (59; 60; 61)	lack of transparacy in high dimensional data, extensive memory requirements(59; 60; 61)
Bayesian Networks	limited storage requirements,robust to missing values (57; 58)	very limited tolerance towards independent attributes, computational expensive (57; 58)
K-Nearest Neighbors	Intuitive and simple, easy to implement for multiclass problems (69; 76)	computational expensive in large datasets, performance depends on dimensionality (69; 76)
Artificial Neural Networks	Good at handling large datasets, detect all possible interactions between prediction variables, implicitly detection complex non linear relationships between depended and independent variables (77; 78)	high hardware depedencies (GPU), Unexplained behavior of the network, the duration of the network is unknown (77; 78)
Multiple Linear Regression	ability to determine the relative influence of one or more predictor variables to the target value (78; 88; 89)	difficulties on handling incomplete data (78; 88; 89)
Decision Tree/ Regression Tree	scaling and normalization of data is not required, missing data do not affect the building of the algorithm, very intuitive and simple (91; 92; 93)	small change in data can cause instability of the algorithm, involves higher time to train the data (91; 92; 93)
Additive Models	model highly complex nonlinear relationships when the number of potential predictors is large (97; 98; 119)	high propensity for overfitting (97; 98; 119)
Logistic Regression	highly interpretable,easy to regularize, outputs well-calibrated predicted probabilities (89)	cannot solve non linear problems,not a powerfull algorithm can be easily outperformed by others (89)
Bag of Words	very simple to understand and implement, great success on prediction problems,offers a lot of flexibility in data customization (102)	vocabulary requires careful design, sparse representations hard to model (102)
Locally Weighted Training	non parametric prediction by local cost functions (100; 101)	computational expensive, memory requirements increase with bigger datasets (100; 101)
Random Forest	fits well with limited data, high accuracy (104; 105; 106)	cannot improve accuracy on high dimensional datasets (104; 105; 106)
K-means	Simple to implement, easy adoption to new examples (111; 112; 113)	scale to large datasets, clustering outliers,depend on initial values scaling with number of dimensions (111; 112; 113)
Self Organizing Maps	capable of clustering large datasets, data mapping easily interpreted (114; 115)	slow training, do not build a generative model for data (114; 115)
PILCO	Data efficient, does not rely on expert knowledge (118)	get stack in a local optimum because of zero gradients, does not take temporal correlation int account (118)
SMART	handle continuous state and action spaces,allow incorporation of domain knowledge in the parametrization (117)	convergence rate is often slow in discrete problems,difficult to use in off-policy settings (117)

Table 9: Advantages and Disadvantages of Machine Learning Algorithms used in Industry

5. DEEP LEARNING IN INDUSTRY 4.0

Deep Learning is a branch of ML that process data by multiple no -linear processing layers (120). Deep learning is a specific ML subfield: a new take on data learning representations, that emphasizes learning successive layers of increasingly meaningful representations. The "deep" in deep learning is not a reference to any kind of deeper understanding that the approach achieves, rather, it stands for this notion of effective layers of representations. Modern deep learning often involves tens or even hundreds of successive layers of representations — and from exposure to training data they are all learned automatically. Meanwhile, other approaches to ML tend to focus on learning just one or two layers of data representations, thus, they are sometimes referred to as shallow learning (121). There are various deep learning models used in Industry. In this section, first the authors describe their research on the deep learning models used in Industry. Second, the general concept for every deep learning model is discussed. Finally, a discussion about the deep learning models, their advantages and disadvantages is presented.

5.1. Auto Encoders

Auto Encoders (AEs) consist of two components: the encoder and the decoder. Both of them are designed to learn a new representation of data by trying to reformulate the input data. Encoder is used to perform data compression by mapping input into a hidden layer. Decoder is used to reconstruct the given input. When input data is highly nonlinear, more hidden layers are required to deal with this complexity. They are mostly used in dimensionality reduction,like a non linear Principal Component Analysis. From a more mathematical approach, the encoder takes a given input x and transforms it into a hidden representation k as follows (122)-(123):

$$k = \alpha(Wx + b) \tag{13}$$

where α is an activation function. Then the decoder maps the hidden representation into its actual value as:

$$z = \alpha(W'x + b') \tag{14}$$

Model parameters $t = [W, b, W', b']$ are optimized to minimize the error (reconstruction error) between x and $z = f_t(x)$ Bearing in mind the aforementioned equations, in Fig. 4 is presented an architecture of an Auto Encoder including input layer (Encoder) and output Layer (Decoder).

Auto Encoder can handle large dimensionality datasets and they work well with no prior knowledge

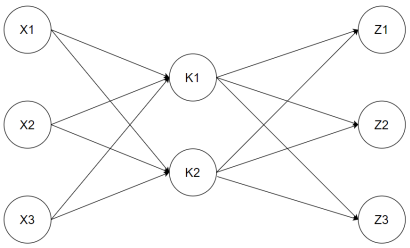

Figure 4: Structure of an Auto Encoder

of the data. However, training an Auto Encoder can be computational and hardware expensive (124; 125). Auto Encoder has several variants such as Stacked Auto Encoder, Sparse Auto Encoder, Denoising Auto Encoder and Contractive Auto Encoder (123). Auto Encoder and its variants found to be used in many industrial applications. It was used in (122) for fault detection in wind turbines, in (123) for fault diagnosis on rotor bearing systems. Stacked Auto Encoder and Denoising Auto Encoder were utilized in ((126; 127; 128; 129)), to monitor the RUL of the machines. It is highly controversial in which learning category do Auto Encoders belong. In ((126; 127; 128)) it is stated that Auto Encoders are unsupervised DL algorithms. Data Scientists in many forums are strongly suggesting that this type of DL algorithm is supervised learning. They highlight that the output of an Auto Encoder will be the input again so it can be seen as the model learning from the target variable (input). Our opinion states that AE is unsupervised learning algorithms. The definition of unsupervised learning is to learn from inputs without any target variables (outputs-labels). Supervised learning is from a given input, to select a function, which maps correctly the input to output and at the same time, input is different from the output.

5.2. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have an important role in smart manufacturing. They are a specialized version of Neural Networks, designed to process data in the form of multiple arrays. CNNs consist of convolutional layers, nonlinear layers and pooling layers. The convolutional layers handle raw input data and generate invariant local features. The non linear layers apply the activation function such as reflected linear function or gradient based backpropagation. The pooling layers extract the most important features by applying pooling operations such as max pooling and average pooling (121; 123). Assuming that the input data

is $x_1, x_2...., x_n$, the convolutional operation could be described as:

$$c_i = \phi((ux_{i:i+m-1}) + b) \tag{15}$$

where $x_{i:i+m-1}$ is a concatenation vector representation, b and ϕ declare bias term and non-linear activation function. u is a filter vector where: $u \in R^{md}$. A future map could be given as follows, if we slide the filtering window from the beginning through the ending time step:

$$c_j = c_1, c_2, ..., c_{i-m+1} \tag{16}$$

where index j represents the $j-th$ filter. CNNs are good at pattern completition and feature extraction but the speed of training depends on the available GPU and large datasets are needed for training (130; 131). Convolutional Neural Networks are used in (132) to find defective metallic parts and identify the cause of the defection. CNNs were also implemented in (133) to diagnose faults in rotor bearing systems and in (134), to diagnose the attachment of silicon die or other wire bondable components on printed circuits boards. In (135), CNNs were utilized for photo-voltaic installation, while in (136) CNNs were implemented to localize slab identification numbers. In ((137; 138; 139; 140; 141)) CNNs were utilized to monitor thre RUL of the machinery. CNNs can be either supervised or unsupervised and it depends highly on how the model is trained. Either for feature extraction or for classification purposes. In Fig. 5 is presented a structure of a convolutional neural network, based on the mathematical approach described above.

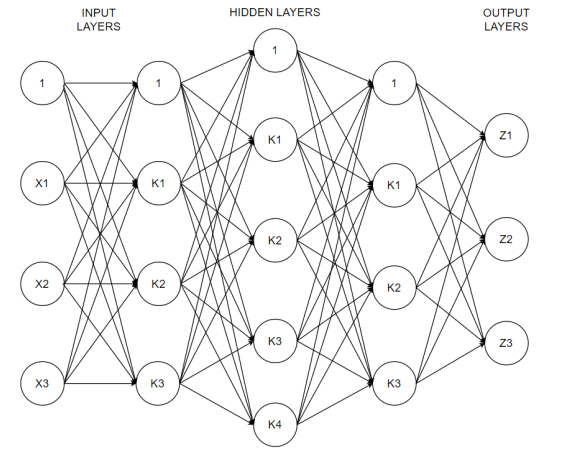

Figure 5: Structure of a Convolutional Neural Network

5.3. Restricted Boltzman Machine

Another DL model used in industry is Restricted Boltzmann Machine (RBM) and its variants. Restricted Boltzmann Machine is a two layer neural network. It consist of a visible (v) and a hidden layer (h). The visible layer is used to input data while the hidden layer is used to extract features. Given the parameters $\theta = [W, b, a]$ the energy function of the model is:

$$E(v, h, \theta) = -\sum_{i=1}^{i}\sum_{j=1}^{j} w_{ij}v_i h_j - \sum_{i=1}^{i} b_i v_i - \sum_{j=1}^{j} a_j h_j \tag{17}$$

where W_{ij} is the weight between the visible unit and the hidden unit. a_j and b_i denote the bias terms for hidden and visible unit respectively. The joint distribution is given by:

$$p(v, h, \theta) = \frac{exp(-E(v, h, \theta))}{Z} \tag{18}$$

where

$$Z = \sum_{h,v} exp(-E(v, h, \theta)) \tag{19}$$

is the normalization factor (122; 142). Afterwars, the conditional probabilities of the visible and hidden layer can be given as:

$$p(v_i = 1|v, \theta) = \delta(\sum_{j=1}^{J} w_{ij}h_j + b_i) \tag{20}$$

$$p(v_i = 1|v, \theta) = \delta(\sum_{i=1}^{I} w_{ij}h_i + a_j) \tag{21}$$

where δ is a logistic function. RBM's are trained through contrastive divergence method to maximize the joint probability. In Fig. 6 is shown a possible architecture of a RBM based on the mathematical approach discussed above.

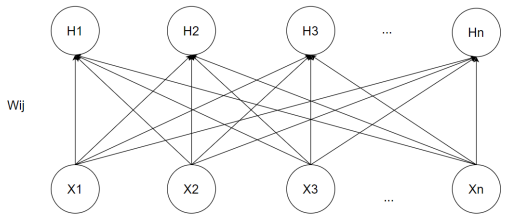

Figure 6: Architecture of a Restricted Boltzmann Machine

Deep Belief Network (DBF) is a variant of RBM and in order to train them a mix of greedy and contractive

wake-sleep algorithms are used. Its architecture consist of stacking multiple RBMs together (121),(143). RBM found to be used in (122) to detect faults in wind turbines and in (143; 144) to monitor the machinery health.RBMs can also be either supervised or unsupervised,depending on the model's training.

5.4. Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are feed-forward neural networks, if they are unfolded in time scale and they are both supervised and unsupervised DL models. RNNs consist of a structure of directed cycles among hidden units. The inputs of the hidden unit come from the output of the previous hidden unit at the past time plus the input unit at the current time. The equation which describes the above is:

$$h_t = \phi(Wx_t, Hh_{t-1}) + b \tag{22}$$

where H and W are non linear and differentiable transformation functions. ϕ denotes the nonlinear activation function and b is the bias vector(145; 146). in Fig. 7 presents a representation of the architecture of a Reccurent Neural Network. RNNs are well suited for natural speech recognition (146). Long Short-Term Memory (LSTM) is a variant of RNN which has the capability to learn long term dependencies and it is consisting of a memory cell which store continuously information during the procession stage (147). LSTM was used in (147) to predict the Remaining Usage Life of battery sets. RNNs were also used in (148) to monitor the machinery health. RNNs are better than other techniques at handling sequantial data, however, there are a lot of input parameters to tune in order to achieve good accuracy (149; 150).

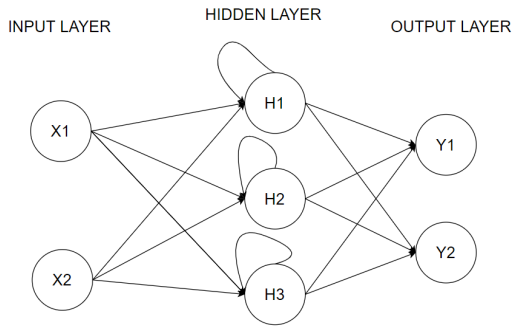

Figure 7: Architecture of a Recurrent Neural Network

5.5. Multilayer Perceptron

Multilayer Perceptron is a supervised feed forward neural network and consist of at least an input layer a hidden layer and an output layer. The hidden layer and the output layer use a nonlinear activation function. The total input x_j^{d+1} received by a neuron j in layer $d+1$ could be declared as:

$$x_j^{d+1} = \sum_i y_i^d w_{ij}^d - \alpha_j^{d+1} \tag{23}$$

where y_i is the state of the i_{th} neuron in the d_{th} layer,and W_{ij} is the weight of the i_{th} neuron in layer d to the j_{th} neuron in layer d+1. While θ is the threshold of the j_{th} neuron in the d+1 hidden layer. Also, the output of a neuron in any layer except the input layer could be given as (151; 152; 153):

$$y_j^d = \frac{1}{1 + e^{-x_j^d}} \tag{24}$$

Fig.8 illustrates a possible structure of a Multilayer Perceptron, with respect to the above mathematical approach. In order to train Multilayer Perceptron networks, a back propagation algorithm should be implemented. Multilayer Perceptron were used in (122) to detect faults in wind turbines and in (154) to predict six-digit NAICS codes.

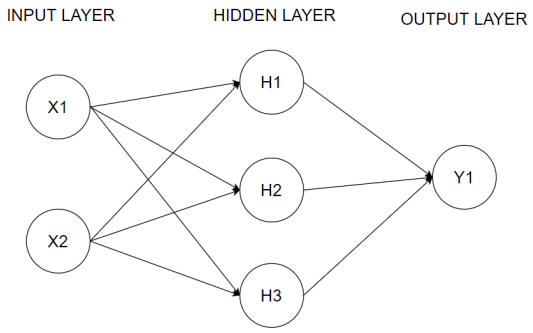

Figure 8: Structure of a Multilayer Perceptron

5.6. CAMP-BD model - YOLOv2 model

Other DL models used in Industry is the CAMP-BD model and the YOLOv2 model. The CAMP- BD model is a combination of a CNN and a RNN and it was used in (155) to predict distortion within laser-based addictive manufacturing tolerance limits by considering the local heat transfer for point-wise distortion prediction. CAMP-DB has two advantages. First it leverages large datasets captured in industrial environment by utlizing DL. Second, the model could be generalized to

solve other challenges of the process, like residual stress and porosity by providing the needed dataset. Furthermore, CAMP-BD is backpropagated on the loss and optimized using Adam optimizer. ReLU activation function is used for training except the for the final output layer,where it is used a linear function(155).

YOLOv2 is an object detection system targeted for real-time processing and consist of twenty four convolutional neural networks.In more details, YOLOv2 utilizes a single neural network to predict class probabilities and bounding boxes directly from full images. Input image is divided into SxS grids. Each cell of the grid predicts c confidence scores of the boxes and bouncing boxes. Also it predicts a C conditional class probabilities. YOLOv2 detection network has 24 convolutional layers followed by 2 fully connected layers (156). YOLO architectures are goot at real-time processing and can be trained end to end in order to improve accuracy. Nevertheless, YOLO struggles to generalize groups of small objects (157; 158; 159). YOLOv2 was used in (160) to recognize oil industry facilities.

5.7. Discussion

Table 10 summarizes the models described in Section V and categorizes them in supervised or unsupervised learning models. Table XI presents the advantages and disadvantages of DL models used in Industry. Convolutional Neural Networks coud be used to achieve high performance in image recognition as well as for feature extraction purposes. The benefit of using CNNs is their ability to develop a two-dimensional image's internal representation. This allows the model throughout the data to learn position and scale, which is critical when working with images (130; 131). Recurrent Neural Networks and especially LSTMs can efficiently handle time series data for prediction purposes. By working with sequences of words and paragraphs, generally called natural language processing, RNNs in general and LSTMs in particular have achieved an enhanced accuracy than other DL approaches. This includes both text sequences as well as spoken language sequences represented as a time series. They are also used as generative models requiring a sequence output, not only with text, but also for applications like generating handwriting (149; 150). AE are fitting well enough with high dimensional data and are a good choice for anomaly detection techniques. However, the AE model will work very well if you have correlated input data since the encoding operation relies on the correlated features to compress the data (124; 125). RBMs and especially DBNs are very efficient on pattern recognition and feature extraction (161; 162). In conclusion YOLOv2 can handle in an efficient way real time object detection problems (157; 158; 159), while CAMP-BD is a new DL approach and there is not sufficient literature work in order to form an opinion about its capabilities.

Deep Learning Models in Industry	
Supervised Learning	Convolutional Neural Network Recurrent Neural Network Restricted Boltzmann Machine Multiple Linear Perceptron YOLOv2
Unsupervised Learning	Auto Encoders Convolutional Neural Network Recurrent Neural Network Restricted Boltzmann Machine CAMP-BD

Table 10: Summary of Deep Learning Models used in Industry

Model	Advantages	Disadvantages
Convolutional Neural Network	high accuracy in image recognition,very good at feature extraction (130; 131)	large dataset needed for training purposes, high computational cost, speed of training depending on GPU (130; 131)
Recurrent Neural Networks	better than other techniques at handling sequantial data (149; 150)	more data are required for training than in other options, there are lots of input parameters to tune, high computational complexity, hardware expensive (149; 150)
Restricted Boltzmann Machines	good at pattern completion and feature extraction (161; 162)	complex to train (161; 162)
Auto Encoders	Intuitive, no prior knowledge of the data is needed, good handling of large dimensional datasets (124; 125)	computational and hardware expensive (124; 125)
CAMP-BD	leverage of large data captured in industry 4.0 environment, generalized to solve other LBAM proccess control challenges such as porosity and residual stress (155)	due to the beginning state of this algorithm drawbacks are not yet presented
YOLOv2	good at real-time proccessing, can be trained end to end to improve accuracy (157; 158; 159)	struggle to generalize groups of small objects (157; 158; 159)

Table 11: Advantages and Disadvantages of Deep Learning Algorithms used in Industry

6. Case study scenario: The Smart Grid

The previous sections analysed ML and DL algorithms and models utilized in Industry 4.0. This section covers the applicability of the previously discussed and presented algorithms and models in the Smart Grid framework, a rising field of the Industry 4.0.

6.1. Machine Learning Algorithms

6.1.1. Bayesian Networks

Several ML Algorithms are implemented for the Smart Grid framework. In (163), Bayesian classification was used to predict the energy produced by PV systems over a very short term period of 15 minutes ahead. For the training and the testing phase of the classifier, four months of real historical data of a 30kWp PV system were used. Bayesian classification was also selected in (164), for detection and classification of complex power quality disturbances. Specifically,a three-level multiply connected Bayesian-Network is utilized, consisting of Features Evidence Layer, Disturbances State Layer and Circumstance Evidence Layer, aiming to extract features from the sample signal. The computation of the posterior marginal probabilities of each event implements the classification. The training and testing of the classifier was conducted not only by the extracted features, but also with historical data. In addition, Babar et al. in (165) developed a secure demand-side management engine in order to detect intrusions in the SG use case. Concretely, the authors implemented a Naive Bayes classifier in order to to control intrusions and preserve the efficient utilization of energy based on priorities.

6.1.2. Support Vector Machines

Support Vector Machines are used in general for load forecasting in Smart Grid. Analytically, in (166), is presented an energy consumption forecasting methodology based on the SVM. The data used for the training and the testing were collected through the SCADA Office Intelligent Context Awareness Management (SOICAM) system. The system is implemented in real facilities in GECAD Research Center, with more than 30 researchers, located in the Institute of Engineering – Polytechnic of Porto (ISEP/IPP), Portugal. De Yong et al in (167) also used the wavelet transform and the SVM for classifying the power quality. The suggested algorithm is made up of a set of straightforward binary SVM classifiers. Each SVM node is trained individually to allow parallels. The training phase is performed using single events, but it enables the system to detect complex incidents due to the composition of the chosen SVM methodology. Real, complicated signals were used to test and train the model. Furthermore in (168), SVM was used to recognize the state of the operation for heat pumps. The method is tested on real-world power consumption time series with 1s resolution from three distinct structures tracked over 1 year. Training information for the SVM algorithm are distilled by outlier removal and subsequent K-means classification from the assessment information due to the lack of ground truth validation information. To conclude, Behera and Misra in (169) used SVM classifier for predicting and re-engineering the hourly energy demand in a residential building. The data for the training and testing phase were collected from iAWE dataset read by two meters (primary, secondary) from a residential building in Delhi during 2013. Additional sensor information installed in the smart building were recorded to play the function of explanatory variables. Electricity consumption was evaluated at three levels: electricity meter (using Schneider electric sensor EM6400), circuit panel and level of appliances. Power outages are also taken into account during data set recording. Efstathopoulos et al in (170) developed an anomaly-based IDS designed specifically for SG using real power plant operational data. Many ML approaches have been considered for detection of anomalies, including One Class-SVM, Isolation Forest, Angle-Base Outlier Detection (ABOD), Stochastic Outlier Selection (SOS), Principal Component Analysis (PCA), and deeply connected AE. The model is trained and tested on operational data obtained from the Lavrio Unit 5 power plant.

6.1.3. K-Nearest Neighbor

Weng et al in (171) made a system for state estimation based on K-Nearest Neighbor algorithm. They build data-driven state estimation method based on recent targeted sensor, data processing and electronics investment. The proposed architecture use physics and patterns of the power system to systematically clean historical data and perform supervised learning, using historical related measurements and their states to learn the relationship between current measurement and state. Kernel trick is used to generate linear mapping in a carefully selected higher-dimensional space to deal with non-linearity. They evaluate the power system data set and discover its clustering property due to the periodic pattern of power systems to accelerate the information-driven approach for online services. In (172) K Nearest Neighbor is used for forecasting low voltage demand. The inputs of this model are historic smart meter data and it can predict the next day's load without the consumer's explicit knowledge.

6.1.4. Artificial Neural Networks

Artificial Neural Networks are also utilized to assist the operation of Smart Grids. Macedo et al. in (173) built a demand side management system based on Artificial Neural Networks for the optimisation of power system management in real time. The system is trained through patterns extracted from digital meter data. Furthermore, Forderer et al in (174) utilized an ANN to represent the devices and act as models of surrogacy. The main benefit of this approach is its capacity for arbitrary versatility in energy and the resulting universal applicability in different usage patterns.

6.1.5. Multiple Linear Regression

Kim in (175) developed a Multiple Linear Regression model to predict electric loads. In details, the utilized model adopts a statistical approach which assumes that past load and weather data are predictable. It defines a reference load before the target time and collects past loads similar to the reference load to render the data vector and condition matrix observed. Lee and Benjapolakul in (176) present a Bagged Averaging of Multiple Linear Regression model, handling data from the phasor measurement unit. The proposed model handles and manages the missing values quickly and efficiently in synchronized frequency data calculation. This methodology is based on the ensemble learning to estimate missing values by bootstrapping and integrating several different linear regressions while data from the Texas synchrophasor network were used to train and test the model.

6.1.6. Decision Tree

Eissa et al in (177) present a new control procedure to reduce the risk of disconnecting charged feeders from electrical grids due to low frequency shedding (UFLS) relays malfunction. The suggested technique is based on a decision-tree algorithm for taking a precise decision to control thermostatically controllable loads (TCL). Terzi et al in (178) also presented a model based on Decision Tree algorithm. The proposed system utilizes detection methods for cyber attacks on Smart Grid scenario. The model is evaluated with the Power System Dataset which was covered by Industrial Control System (ICS) Cyber Attack Datasets. In addition, Achlerkar et al in (179) implemented a model based on Variational Mode Decomposition and Decision Tree for detection and classification of power quality disturbances in Grid-Connected Distributed Generation System. Attributes such as central frequency mode(MCFs), relative energy ratios (RMERs), zero crossings and instantaneous amplitude (IA) are derived using a decision tree algorithm to distinguish single and mixed PQ disturbances. A collection of simulated test signals, disturbance signals from real events as well as signals produced from the Real Time Digital Simulator (RTDS) platform are used to test the effectiveness of the proposed method in different operating scenarios and noise levels of the device.

6.1.7. Gradient Boosted

Gradient Boosted algorithm is also used in Smart Grid applications. Bessa et al. in (180), utilized a probabilistic Solar Power forecasting model based on Gradient Boosted algorithm. In particular, this work proposes a new six-hour forecasting algorithm based on the model of vector auto regression, which incorporates the time series information collected by the infrastructure of the Smart Grid. For residential solar photovoltaic (PV) and secondary substation rates, probabilistic forecasts are produced. The test case consists of 44 Smart Grid pilot units and 10 secondary substations in Évora, Portugal. Punmiya et al in (181) utilized a model for energy theft detection. This work presents a gradient boosting theft detector (GBTD) based on the three latest gradient boosting classifiers. 1) extreme gradient boosting; 2) categorical boosting; and 3) light gradient boosting method. The dataset used to evaluate the model is the Irish smart energy dataset, which contains half-hour recorded use of each customer(in kWh) for about 420 days, of which 361 and 59 days were used respectively in the training set and test set. Razavi, Rouzbeh et al also in (182) suggested a new model-agnostic, feature-engineering architecture for smart grid theft detection based on Gradient Boosted algorithm. The architecture uses a combination of Finite Mixture Model clustering for customer segmentation and a Genetic Programing algorithm to identify new predictable features.

6.1.8. Additive Model

Pompey et al. in (183) presented a broad simulation system emulating electrical load throughout the electrical network, based on General Additive Models. The platform supports customer portfolio shift simulation and consumer behavior, installing new distributed generation capability at any network level, and adaptive network reconfiguration. Taieb et al. in (184) built a model for probabilistic time series forecasting that allow the inclusion of a possibly large set of exogenous variables in Smart Grids. The approach was based on boosted additive models. The evaluation of the model was made by conducting extensive experiments on aggregated and dis-aggregated scales using electricity smart meter data. Thouvenot et al. in (185) present a

load forecasting model which is using Additive Models. The authors submit an automated explanatory factor selection method in an additive model and display how to correct short-term forecast errors.

6.1.9. Locally Weighted Training

Zhang et al in (186) proposed a system for instantaneous electromechanical dynamics monitoring in Smart transmission Grid. Big data can be acquired by measuring sensors mounted in the smart transmission system for tracking electromechanical dynamics. The data obtained from the time series carry information regarding the instantaneous system oscillation modes relationship with regard to operating conditions. To extract this information, they suggested a parallel online supervised learning algorithm k-nearest neighbors called "local weighted linear regression" (KNN-LWLR), a systematic combination of two well-known machine-learning algorithms: 1) KNN learning and 2) LWLR learning.

6.1.10. Random Forest

In (187) Lahouar and Slama are proposing a short-term load prediction model based on random forests that can predict the next 24 hours of load. They utilize random forest algorithm to construct the model following an online learning process. The inputs are optimized by expert feature selection using a set of if – then rules to include the country climate or market's own consumer preferences and generalize the forecasting capacity. The proposed solution is checked by the Tunisian Power Company's real historical collection. Furthermore, Lin et al. in (188) propose a model on fault prediction in the smart distribution network. They utilize a voted based Random Forest approach to increase the predictive accuracy of the faults. Through re-designing the voting algorithm, they modify the decision process through adding several SVM models for voting model training a basic NSGA algorithm is used to find the best voting model based on the trained models. In (189) Singh et al. developed four models-including a Random Forest model- to analyze energy consumption data and associated weather data at different periods of time and discuss the training strategies limitations.

6.1.11. Isolation Forest

Ahmed et al in (190) presented a machine learning-based scheme to identify data integrity assaults in SG communications networks using non-labeled data. Analytically, an Isolation Forest algorithm was introduced by the developers, which distinguishes data integrity assaults based on the premise that the attack in a built random forest has the shortest average path range. They also utilized a PCA algorithm to tackle with the dimensional reduction issue.

6.1.12. K-Means

Wakeel et al in (191) developed a load estimation algorithm based on k-means cluster analysis. To predict missing and potential measurements, the algorithm applies cluster centers – from previously clustered load profiles – and distance functions. Several case studies were conducted using aggregated smart meter daily and segmented load profiles. Segmented profiles span a time window equal to or below 24 hours. To enable better monitoring and control of distribution networks, the developed load estimation algorithm can be combined with state estimation or other network operational tools. Yu et al in (192) developed a model utilizing K Means algorithm for modeling and forecasting these individual household electricity loads. The new method is tested for the period from September 2011 to August 2013 on a data set of 5000 households in a joint project with Chattanooga's electric power commission. Starke et al in (193) utilized a hybrid, distributed and decentralized SDN architecture for resilient smart systems. They implemented a k-means algorithm in order to detect anomalies in the traffic of the network.

6.1.13. Self Organizing Maps

In a nutshell, Llanos et al in (194) presented a Load estimation for microgrid planning based on a self-organizing map methodology. In details, this approach presents a load estimation method for isolated communities not receiving or consuming power for a limited period of time each day. The framework proposed contains the components below. First the inputs are analyzed on the basis of surveys of residents living in any socio-economic level of housing and neighborhood. Second, family groups are classified using an SOM from which relevant information that distinguishes one family from another is obtained. Then, each cluster's load profiles are selected from a database. In addition, social aspects and relevant information on energy supply from communities with similar characteristics are used to generate the database required. The SOM for clustering community families with available energy measurements is used as an initial assumption for clustering community families with unknown energy measurements.

6.2. Deep Learning Algorithms

Several Deep Learning algorithms found to be utilized for the Smart Grid infrastructure. AE, CNNs, RNNs, RBMs and Multilayer Perceptron.

6.2.1. Auto Encoders

Tong et al in (195) propose a deep learning model that first refines features from historical electricity load information and related temperature parameters by stacked denoising auto encoders (SDAs). The deep learning model trains a support vector regression (SVR) model to predict the total day-to-day electricity charge. Then it is evaluated by comparing with plain SVR and artificial neural networks (ANNs) models. Lu, Shixiang, et al in (196) proposed a trend-based load characterising approach. Initially, the candlestick chart concept is used as an innovative load classification method. Additionally, trend indexes for electricity, such as stochastic oscillator and moving average convergence / divergence, were implemented as parameters for characterizing the load. Then, stacked auto-encoders are used to predict future loads based on historical pattern index data. The evaluation of the model was implemented by data obtained from a large user in Foshan, Guangdong province of China. Yang et al in (197) built a variational autoencoder for voltage stability evaluation in Smart Grid. Various data-driven indicators were proposed for comparison and evaluation on the basis of sparse stacked AE and adversarial AE. Their methods were tested in IEEE-14, IEEE-57 and IEEE-118 bus standard system, and compared mutually. Ahmed et al in (198) suggested a model to recreate sensor-collected power network measurement data by eliminating the impacts of the hidden information-integrity attack. The model is utlized by a denoising autoencoder, which learns from the data about robust nonlinear representations to root out the bias that a smart attacker has added to the sensor measurements.The scheme was evaluated utilizing standard IEEE 14-bus, 39-bus, 57-bus, and 118-bus systems.

6.2.2. Convolutional Neural Networks

Convolutional Neural Networks were used in (199) to build an electricity-theft detection method. The architecture consists of a wide CNN and a deep CNN. The deep CNN component can identify the non periodicity of electricity theft and the periodicity of normal electricity usage based on 2-D electricity consumption data. Meanwhile, the wide component can capture the global features of 1-D electricity consumption data. The evaluation of the model is made by a dataset released from the State Grid Corporation of China. Zahid et al in (200) built a model for Electricity Price and Load Forecasting. They used XGB, DT, RFE and RF for feature selection and feature extraction. Then, they used an Enhanced CNN and Enhanced SVR as classifiers. GS is used to improve the performance of the classifier parameters. Ultimately, the proposed models were compared with various stability analysis benchmark schemes. MSE, RMSE, MAE, and MAPE performance metrics are used to measure the performance of the proposed models. Kuo et al in (201) built a model for short-term load forecasting in Smart Grids. Analytically, in order to predict the amount of load needed in short term, they initialize a CNN. The neural network consists of 3 convolutional layers as well as 3 pooling layers. In order to evaluate the model, the USA District public consumption dataset and electric load dataset from 2016 provided by the Electric Reliability Council of Texas were used.

6.2.3. Recurrent Neural Networks

Kong et al in (202) proposed a LSTM, for short-term residential load forecasting. The proposed framework is tested on the Australia's project Smart Grid Smart City dataset, which contains smart meter data for about 10,000 different customers in New South Wales. Ouyang et al in (203), presented an approach of using LSTM networks to identify false data of smart terminals in the Smart Grid. In the master station gateways, they use LSTM to detect abnormal data from smart terminals. For the evaluation phase, the accuracy metric and equal error rate metric were used. In (204) Hassan et al. proposed an electricity theft detection system based on CNNs and LSTMs. A SMOTE for data preprocessing was also introduced in this work to measure the missing instances in the dataset based on the local values relative to the missing data point. In addition, the number of users of electricity theft was relatively low in this dataset, which might have made the model inefficient in identifying users of theft. Nasser and Mahmoud in (205) developed a photovoltaic power forecasting model with LSTM-RNN network. The proposed method is evaluated using hourly datasets of different cities (Aswan and Cairo) for a year. The model is able to predict an hour-ahead power of PV. Marino et al in (206) built LSTM networks for energy load forecasting. Their research discusses two versions of the LSTM:1) standard LSTM and 2) LSTM-based Sequence to Sequence (S2S) architecture. The methods are applied for a single residential consumer on a reference dataset of electricity consumption, called "Personal household electricity use." The dataset included measurements of power consumption that were obtained with one-minute resolution between December 2006 and November 2010. The dataset provided an effective cumulative power load for the entire house and three submetering for the house's three parts.

6.2.4. Restricted Boltzman Machines

He et al in (207) developed a system for real-time detection of false data injection attacks in Smart Grid, based on Deep Belief Networks.They applied deep Belief Network techniques to recognize the behavior patterns of FDI attacks using the historical measurement data and employ the revealed features to detect the FDI attacks in real-time. The performance of the proposed strategy is illustrated through by using IEEE 118-bus test system. The scalability of the system is also evaluated through the use of the IEEE 300-bus test system. Menon and Radhika in (208) used Deep Belief Networks to detect intrusions in smart home area network. In details, the build a Deep Belief Network to detect the normal and abnormal behaviors in the traffic pattern of Smart Grid data. Deep belief Network was deployed to classify the data traffic anomalies in the Smart Grid to prevent intrusion. SVM algorithm is then used for identification of invasion. He Yusen et al in (209) use deep learning techniques to identify the behavioral patterns of FDI attacks using historical measurement data and use the features discovered to detect FDI attacks in real time. The data were gathered from an urbanized area in Texas, USA. Forecasting hourly power load in four different seasons in a selected year is examined. Two forecasting scenarios, day-ahead and week-ahead forecasting were performed using the proposed methods and compared to classical neural networks, SVR, ELM and classical DBN.

6.2.5. Multilayer Perceptron

Hamedani et al in (210) presented a method for attack detection of Smart Grids with wind power generators using reservoir computing (RC), temporal encoding and multilayer perceptrons. Moon et al in (211) presented a hybrid short-term load forecasting scheme using Random Forest and Multilayer Perceptron. They collected six-year electrical load data from a university campus to develop this model and divided it into learning, validation, and test sets. They also categorized the information for the training set using a decision tree with input parameters including time, weekday, holiday, and academic year. In (212), Alimi et al proposed a hybrid Support Vector Machine and Multilayer Perceptron Neural Network (SVMNN) algorithm combining Support Vector Machine (SVM) and Multilayer Perceptron Neural Network (MPLNN) algorithms to predict and detect cyber intrusion attacks on power system networks. As case studies, a modified version of the IEEE Garver 6-bus test system and a 24-bus test system was used. The IEEE Garver 6-bus test system was used to define the attack scenarios, while the load flow analysis was carried out on real-time data from a modified Nigerian 24-bus network to produce the bus voltage dataset which considered multiple cyber attacks for the hybrid algorithm. In conclusion, Wahid et al in (213) used Multi-layer perceptron and Random Forest to identify residential buildings based on their consumption of energy. Hourly historical data, of two types of buildings, are predicted: high-power and low-power buildings. The prediction consists of three stages: recovery of data, extraction of features, and prediction. The hourly data collected on a daily basis is extracted from the server at the data retrieval stage. Statistical features are calculated from the recovered data in the extraction stage of the feature: mean, standard deviation, skewedness and kurtosis. In Table 12 are summarized the ML models and Deep Learning models used in the Smart Grid scenario.

6.3. Discussion

The accurate and fast detection of cyberattacks relies on the proper selection of the Machine Learning and Deep Learning model, which will detect the anomalous incident. Different aspects should be considered in order to select a ML/DL model. Aspects like the size of the dataset, the application of a DL/ML model on other cases and the generalization of the model on previously unseen data. More guidelines for the proper selection of the ML/DL model are given in subsection “Model Selection” of section “Challenges and Trends”. However, in order to tackle with the generalization problem and increase the accuracy of intrusion detection, in our previous works ((214; 215)) we created ML/DL schemes, which combine both supervised and unsupervised models. Concretely, DIDEROT (214) is a detection and intrusion system for the DNP3 protocol.It combines both supervised (Decision Tree, Support Vector Machine) and unsupervised ML/DL models (DIDEROT autoencoder, Local Outlier Factor) capable of discriminating whether a DNP3 network flow is related to a particular DNP3 cyberattack or anomaly.ARIES (215) is an anomaly-based intrusion detection system with supervised (Decision Tree, Support Vector Machine) models and a novel unsupervised Generative Adversarial Network. ARIES integrates three levels of surveillance dedicated to the identification of suspected cyberattacks and anomalies. In specific, a supervised multi-class classification is conducted by the first layer, understanding DoS, brute force attacks, port scanning attacks and bots. The second layer detects potential Modbus packet-related abnormalities, while the third layer tracks and recognizes operating data anomalies.

The on-time detection of anomalous incidents (cyber-attacks) is another emerging topic on the Smart Grid framework. The fast response and recovery of the Smart Grid components after a possible cyberattack must be the target of all the developing solutions. The solutions on this topic should also maximize the observability, minimize the latency and maximize the QoS of the Smart Grid. SPEAR and SDN-microSENSE are two European Commission projects with aim to handle and provide cybersecurity solutions on the Smart Grid. SPEAR project in brief, is aiming to develop an integrated platform of methods, processes, and tools for in time detection of cyber attacks and risk assessment. SDN-microSENSE project is aiming to provide and demonstrate a secure, resilient to cyber-attacks, privacy-enabled, and protected against data breaches solution for decentralised EPES.

Model	Usage	Model	Usage
Bayesian Networks	load prediction, intrusion detection (163; 164; 165)	**Support Vector Machine**	Energy consumption forecasting, power quality classification, power consumption classification, anomaly based IDS (166; 167; 168; 169; 170)
K-Nearest Neighbor	power state estimation,forecasting low voltage demand (171; 172)	**Artificial Neural Networks**	power management systems (173; 174)
Multiple Linear Regression	load prediction (175; 176)	**Decision Tree**	control procedures, cyber-attack detections, classification of power quality disturbances (177; 178; 179)
Gradient Boosted	solar power forecasting, energy theft detection (180; 181; 182)	**Additive Models**	developing simulation platforms for electric grids, load forecasting (183; 184; 185)
Locally Weighted Training	electromechanical dynamics monitoring (186)	**Random Forest**	short term load prediction, fault prediction, energy consumption analysis (187; 188; 189)
K-Means	load estimation-forecasting, anomaly detection (191; 192; 193)	**Self Organizing Maps**	load estimation (194)
Auto Encoders	feature extraction, load characterizing, voltage stability evaluation (195; 196; 197; 198)	**Convolutional Neural Networks**	load estimation, theft detection, price and load forecasting (199; 200; 201)
Recurrent Neural Networks	load forecasting, false data identification, theft detection, power forecasting (202; 203; 204; 205; 206)	**Restricted Boltzman Machines**	cyber attack detection, intrusion detection (207; 208; 209)
Multilayer Perceptron	cyber attack detection, load forecasting (210; 211; 212; 213)	**Isolation Forest**	intrusion detection (190)

Table 12: Smart Grid: Machine Learning models, Deep Learning models and their usage

7. Challenges and Trends

Despite the potential vision of Industrial AI, many challenges and trends need to be addressed to understand fully its capabilities. Scalability, Cyber Security, Fault Tolerance, Network latency from the one hand and data handling, Machine or Deep Learning model selection from the other have a great impact on the application of Industrial AI. Machine Learning and Deep Learning techniques found to play an important role in Industry 4.0. The ability to handle high dimensional and multi-variate data, in combination with the ability to reduce cycle time and scrap, improve resource utilization. The ability also, to discover formerly unknown knowledge, makes these techniques a crucial factor in Industry 4.0 (216),(217).

7.1. Data Quality and Processing

First of all, a basic need is the handling of missing or imbalanced data. In some cases, there might be missing data or bad quality of data, aspects that have a strong influence on the performance of ML and DL algorithms. Even if there are techniques that allow us to tackle with these problems, we should take into account the possibility that they could be not successful (53). Dealing with missing data is crucial. Replacing missing data is an important factor because the original dataset is influenced. The goal is to reduce the negative influence as much as possible, in order to achieve a good performance, which will lead in a successful application (53),(216).

7.2. Model Selection

To the authors opinion, the most important challenge is the proper selection of ML/DL algorithm. The increased attention of researchers on the field of ML/DL in manufacturing, developed a large number of algorithms. So, the question raised is which ML/DL technique should be used?

First, we have to look at the available data and their format, to choose the appropriate approach between a supervised, unsupervised or Reinforcement Learning algorithm (53).

In addition, the availability of each algorithm to handle specific data-sets has to be considered and investigated. ML algorithms should be used for datasets that can be processed for acceptable time at a regular computer. DL algorithms should be used for a set of data that need a Hype Performance Computing in order to get processed.

Last but not least, previous applications of the chosen algorithm has to be investigated to identify a suitable algorithm (53).

After gathering a set of suitable algorithms, each one should be applied to the problem's dataset and notice how they correspond. Next, the model's accuracy should be evaluated relying on different metrics - such as Root Mean Square Error, Adjusted R Square or R Square - and techniques.

Azmoodeh et al in (3) highlighted that the most crucial parameters in order to use a Deep Learning model is the generalization of the model and the selection of the correct training parameters (e.g batch size, learning rate, weight decay).

To ensure generalization, several approaches in statistical learning theory have been deployed, which include: Hypothesis-space complexity, stability and robustness. Hypothesis-space relates to the decoupling of the model function from its training data and the worst-case distance for functions in the space hypothesis. Stability refers to dealing with the dependence of the model on the dataset, by taking into consideration also the stability of the learning algorithm with regard to different datasets. Robustness refers to the elimination of some specifics of the model function 's reliance on the dataset, by taking into account the robustness of the learning algorithm for all possible datasets (3).

7.3. Fault Tolerance

Every machine is connected to sensors,embedded devices, as well as to other machines. Faults due to machine failure, software faults in the cloud, or malicious attacks will significantly affect the reliability and availability of the system. It is vital to design a fault-tolerant system and fault recovery mechanisms to adapt to unforeseen failures that may lead to service degradation or unavailability. Different approaches are developed to improve the system reliability. In (217) a hardware fault handling system was developed to to achieve a predictive maintenance and reduces the chance of system failure. In (218) an edge computing system is deployed to reduce uncertainty in the data transfer.

7.4. Network Latency

In industrial AI systems machines, sensors, actuators and devices should properly work together and achieve real time monitoring and data transmission. Thus it is very important to ensure reliable and low latency data transmission, to achieve reliable and efficient industrial systems. To reduce the latency in industrial AI systems one can apply fog computing or edge computing. Fog computing is suitable for industrial systems

that require low and predictable latencies and real-time performance(219)

Fog computing is a computational tool that provides storage, networking services, computing and networking services between end devices and Clouder Servers typically, but not exclusively located at the edge of network. Benefits of cloud computing could be considered the low latency, widely distributed deployments of interconnected devices, mobility, real time interactions, heterogeneity, interoperability and federation(220).

Edge Computing is referring to the processes and technologies allowing the computation to be performed at the edge of the network. The term "edge" is connected with the computing and network resources along the path between the physical layer of the Internet of Things Architecture and the layer which refers to the cloud computing. Shi W. et al in (218) state that edge computing is "interchangeable" with fog computing, but, edge computing relies more on the embedded devices, sensors and actuators while fog computing focus more in the architecture of an IoT system. Furthermore, at the "edge" interconnected things can not only request service and content from the cloud but also perform processes from the cloud. Such processes are data storage,computing offloading, distribute request as well as caching and delivery service from cloud to user. The benefits of edge computing could be summarized as energy reduction of the system (about 20% - 40%), reduced response time and reduced running time(218).

7.5. Scalability

The control systems are usually independently engineered and do not scale. Therefore,it is a challenge here to enable heterogeneous devices and systems to communicate and collaborate. It is highly required to reduce the manual effort in order to communicate and collaborate heterogeneous devices. To tackle with this challenge, different communication protocols are developed, such as Message Queue Telemetry Transport (MQTT) (221), Advanced Message Queuing Protocol (AMQP) (222), Data Distribution Service (223), Low Power Wide Area Networks (224) and Narrow Band-IoT (225).

MQTT is a communication protocol for machine to machine (M2M)/"Internet of Things. "It has been developed as an extremely lightweight messaging transportation to publish / subscribe. It is useful for remote location connections where a limited code footprint is required and/or a premium network bandwidth(221).

A common standard for transmitting business messages between applications or organizations is the Advanced Message Queuing Protocol (AMQP). It links networks, feeds workflows with the data they require, and transmits the guidance for achieving their goals efficiently (222).

The Data-Distribution Service for Real-Time Systems is the first open global middleware interface that directly addresses real-time and embedded publish-subscribe communications. DDS implements a digital Global Data Space where applications can share information simply by reading and writing addressed data objects using an application-defined name (Topic) and a password (223).

Low-power WAN (LPWAN) is a wide-area wireless network technology that interconnects low-bandwidth, battery-powered devices over long ranges with low bit rates. Designed for machine-to-machine (M2 M) and Internet of Things (IoT) networks, LPWANs run more effectively than conventional mobile networks at a lower cost.This can also support more connected devices over a larger area. LPWANs can handle packet sizes ranging from 10 to 1,000 bytes at uplink speeds up to 200 Kbps. Depending on the system, the long range of LPWAN ranges from 2 km to 1,000 km (224).

Narrowband Internet of Things (NB-IoT) is a Low Power Wide Area Network (LPWAN) radio technology standard developed by 3GPP. NB-IoT focuses on indoor coverage, low cost, long battery life, and high density of communication. NB-IoT uses an LTE standard subset which limits the frequency to a single 200kHz narrow band (225).

7.6. Cyber Security

Security is crucial to Industrial AI systems. Interconnected machines, sensors, actuators are driving towards the development of security systems to ensure authentication and data confidentiality. The ability to connect to the Internet and deliver data through it, leaves potential vulnerability for attackers to exploit and take control of the system. Therefore, the software needs to be protected from malicious attacks without interfering with the control process.

In order to tackle with this problem various security services should be considered. Authentication is one parameter, which addresses the capability to ensure the identity of any communicating object. Confidentiality is another parameter, which ensures that data is accessible only to the intended recipients. Next User's Privacy is also vital. It guarantees that any data related to the user, could not be obtained without its explicit approval, and will be used only for the intended purposes. Data Integrity is another aspect which should be considered and ensures that received data were not modified in an unauthorized way(226),(227).

7.6.1. The RPL protocol

In previous years many solutions were developed to secure the Industrial Internet. One of them is the RPL security protocol (228). It was created by the Internet Engineering Task Force(IETF) and is used to route messages in Low Power and Lossy Networks (LLNs). It operates by creating a Destination Oriented Directed Acyclic Graph (DODAG) that initiates an objective function. The existance of secure variations of the RPL packets (DIS,DIO,DAO,ACK) and the capability to apply three security modes is the base of the security in the RPL Protocol. Integrity, replay protection, delay protection, and optional confidentiality are provided by these variations. To conclude, the confidentiality, the integrity and the authenticity of the information in the RPL protocol is guaranteed by the significant security mechanism it provides.

7.6.2. The TLS protocol

Another solution is the TLS security protocol. The TLS protocol consist of individual protocols and it is formed by two layers.Record Protocol is included in the first layer while Alert Protocol, the Change Cipher Spec Protocol, the Heartbeat protocol and the Handshake protocol are included in the second layer (228).

The operation of the Record Protocol is initially to separate the application data into blocks of 214 bytes or less. Then, a symmetric encryption algorithm is used to encrypt them as so the message authentication code (mac). A mac is computed for the specified blocks. The final step is the addition of a specified header. After the aforementioned procedure, the protocol channels the above data in a Transmission Control Protocol packet and then, transmits them (228).

The Change Cipher Spec Protocol is based on a single byte which has the value 1 and shows the the pending state to the current state to update the cryptographic algorithm (228).

The Alert Protocol provides alert over all the operation of the TLS. While, the Handshake Protocol implements an authentication process for the server and the client and a negotiation process of the encryption algorithm,the mac and the cryptographic keys (228).

The Heartbeat protocol assures the sender that the receiver is on and listening. Then, it creates additional network activity to avoid the closure of these connections by a firewall. Although TLS protocol assure the principles of confidentiality, integrity and authenticity of communication, it is an expensive protocol (228).

7.6.3. The DTLS protocol

A variation of the TLS protocol is the DTLS protocol. DTLS operates over data, which can be lost, or received in wrong format. It supports additional mechanisms such as the extension of the TLS Record Protocol with two additional fields, an epoch and a sequence number the forbidden utilization of the stream cirphers and the improved operation of the Handshake Protocol with the addition of a stateless cookie (228).

7.6.4. The CoAP protocol

Another solution is the CoAP security protocol, a lightweight version of the HTTP protocol. It runs over the User Datagram Protocol utilizing 6LoWPAN and its architecture consist of the methods GET, POST, DELETE and PUT. CoAP consists of two layers the request/response layer and the message layer. Message layer controls the communication over the UDP protocol, while the request/response protocol is responsible for sending the right messages in the proper way (228).

Although the aforementioned protocols are efficient to provide secure communication link for the data, we should consider and provide solutions to protect systems from jamming and intuition attacks. In (229), the authors provide a method based on the Colonel Blotto, where a controller device can hinder jamming attacks against IoT devices. Also in (230), the authors implement a hierarchical game, which mitigates the jamming attacks by developing a probabilistic method. Chen et al. in (231) provide a deep reinforcement learning model which is devoted to saving power and optimizing the transmission performance,thus mitigating the jamming attacks.

7.7. Federated Learning

IAI has been introduced to address numerous industrial issues in Industry 4.0 by exploiting ML/DL-based technology. However, conventional centralized training may not be sufficient for critical industrial data-driven situations, such as healthcare and autopilot, for privacy purposes (232). Furthermore, in most industries, data exists in the form of isolated islands. To this end, Federated Learning is addressed, which is a new approach for training DL models. Instead of exchanging and disclosing the training dataset with the server, vast communities of interconnected computers, operating as local trainers, jointly refine the model parameters (e.g., neural network weights and biases) (233).

Yang et al in (234) define federated learning as a learning process in which the data owners collaboratively train a model M_{fed} and process any data that

owner does not expose to others. The authors also propose three different architectures for three subcategories of Federated Learning, namely Horizontal Federated Learning, Vertical Federated Learning and Federated Transfer Learning.

Horizontal Federated Learning is introduced in the scenarios that data sets share the same feature space but they aquire different samples of this feature space. Vertical Federated Learning is implemented in cases were two data sets share the same sample ID space but differ in feature space, while Federated Transfer Learning applies when the two data sets differ both in samples and in feature space (234). Mohri et al in (235) proposed a new Federated Learning approach, called as Agnostic Federated Learning. In Agnostic Federated Learning, the centralized model is configured for every possible distribution of the objective generated by a client distribution combination. The authors identify an agnostic and more risk-averse goal rather than optimizing the centralized model for a particular distribution, with the high risk of a mismatch with the target.

Several communication algorithms have been proposed for the model parameters' transfer from owners to a central server in a reliable and secure way. Concretely, Shokri et al in (236) suggested the first distributed learning system, where participants selectively share small part of the gradients to ensure the privacy of training data. Hao et al. in (232) proposed a communication algorithm called PEFL. In each secure aggregation, PEFL is non-interactive. The homomorphic ciphertext of private gradients is inserted into the term Augmented Learning with Error (A-LWE) to achieve safe aggregation protocol. In specific, the authors provide a clear example of using an optimized BGV homomorphic encryption device that reduces the key-switching operation and improves the key-switching activity. Savazzi et al. in (233) developed a communication algorithm, which updates both the model and the gradients of the model, by relying solely on local cooperation with neighbors, and local in-network (as opposed to centralized) processing.

Furthermore, Xu et al in (237) proposed VerifyNet, a privacy-preserving and verifiable federated learning framework. The authors developed a double-masking protocol to guarantee the confidentiality of users' local gradients during the training phase. Qu et al in (238) developed a congitive model based on federated learning for industrial purposes. In order to guarantee the security of the process, they enabled a blockchain framework to secure the engine from poisoning attacks.

8. Conclusions

In this paper, the new industrial revolution and the key role of the Artificial Intelligence are surveyed and discussed. Initially, the fundamental elements and the Ecosystem of the Industrial AI are analysed and a new application scheme of the Industrial AI is proposed. Furthermore, the ML and DL algorithms and models used in manufacturing are discussed and presented thoroughly. An analysis of the ML and DL models and algorithms on the Smart Grid, an important field of Industry 4.0, is also implemented in terms of its efficiency and its applications. In conclusion, the challenges and trends on the Industrial AI are also documented. The authors are of the opinion that Industry 4.0 has not fully incorporated Artificial intelligence into its operations and there is still much to be done. Cybersecurity is an area that needs special attention due to the interconnection of the manufacturing components to the internet. SPEAR and SDN-microSENSE projects are working to provide overall solutions in this field.

As future work, the authors aim to apply and examine the capabilities and the accuracy of the aforementioned models and algorithms in the use cases of the SPEAR and SDN-microSENSE project. In particular, the models and algorithms will be utilized for anomaly detection, RUL estimation and cost prediction.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 787011 (SPEAR).

References

[1] S. Otles, A. Sakalli, Industry 4.0: The smart factory of the future in beverage industry, in: Production and Management of Beverages, Elsevier, 2019, pp. 439–469. doi:10.1016/b978-0-12-815260-7.00015-8.

[2] C. Yang, W. Shen, X. Wang, The internet of things in manufacturing: Key issues and potential applications, IEEE Systems, Man, and Cybernetics Magazine 4 (1) (2018) 6–15. doi:10.1109/msmc.2017.2702391.
URL https://doi.org/10.1109/msmc.2017.2702391

[3] J. Sakhnini, H. Karimipour, A. Dehghantanha, R. M. Parizi, Ai and security of critical infrastructure, in: Handbook of Big Data Privacy, Springer, 2020, pp. 7–36.

[4] Y. Lu, Industry 4.0: A survey on technologies, applications and open research issues, Journal of Industrial Information Integration 6 (2017) 1–10. doi:10.1016/j.jii.2017.04.005.
URL https://doi.org/10.1016/j.jii.2017.04.005

[5] J.-Q. Li, F. R. Yu, G. Deng, C. Luo, Z. Ming, Q. Yan, Industrial internet: A survey on the enabling technologies, applications, and challenges, IEEE Communications Surveys & Tutorials 19 (3) (2017) 1504–1526. doi:10.1109/comst.2017.2691349.
URL https://doi.org/10.1109/comst.2017.2691349

[6] A. Kampker, H. Heimes, U. Bührer, C. Lienemann, S. Krotil, Enabling data analytics in large scale manufacturing, Procedia Manufacturing 24 (2018) 120–127. doi:10.1016/j.promfg.2018.06.017.
URL https://doi.org/10.1016/j.promfg.2018.06.017

[7] P. Lade, R. Ghosh, S. Srinivasan, Manufacturing analytics and industrial internet of things, IEEE Intelligent Systems 32 (3) (2017) 74–79. doi:10.1109/mis.2017.49.
URL https://doi.org/10.1109/mis.2017.49

[8] B. J. Copeland, D. Proudfoot, Artificial intelligence, in: Philosophy of Psychology and Cognitive Science, Elsevier, 2007, pp. 429–482. doi:10.1016/b978-044451540-7/50032-3.
URL https://doi.org/10.1016/b978-044451540-7/50032-3

[9] A. Masood, A. Hashmi, Cognitive Computing Recipes, Apress, 2019. doi:10.1007/978-1-4842-4106-6.
URL https://doi.org/10.1007/978-1-4842-4106-6

[10] M. D. Fethi, F. Pasiouras, Assessing bank efficiency and performance with operational research and artificial intelligence techniques: A survey, European journal of operational research 204 (2) (2010) 189–198.

[11] S. Palle, Artificial intelligence using dbs-qos in banking organizations, Journal of scientific research & engineering trends 5 (1) (2019) 2395–566X.

[12] M. Jakšič, M. Marinč, Relationship banking and information technology: The role of artificial intelligence and fintech, Risk Management 21 (1) (2019) 1–18.

[13] J. M. Chen, Models for predicting business bankruptcies and their application to banking and to financial regulation, Available at SSRN 3329147 (2019).

[14] C. Liu, H. Huang, S. Lu, Research on personal credit scoring model based on artificial intelligence, in: International Conference on Application of Intelligent Systems in Multi-modal Information Analytics, Springer, 2019, pp. 466–473.

[15] C. Perez, –digitalisation and artificial intelligence: the new face of the retail banking sector. evidence from france and spain, Virtuous circles between innovations, job quality and employment in Europe? Case study evidence from the manufacturing sector, private and public service sector (2018) 178.

[16] S. Ransbotham, P. Gerbert, M. Reeves, D. Kiron, M. Spira, Artificial intelligence in business gets real, MIT Sloan Management Review and Boston Consulting Group (2018).

[17] J. M. Huerta, A. Anand, Machine learning and artificial intelligence in consumer banking, Journal of Digital Banking 3 (1) (2018) 22–32.

[18] D. Dao, S. Trinh, H.-B. Ly, B. Pham, Prediction of compressive strength of geopolymer concrete using entirely steel slag aggregates: Novel hybrid artificial intelligence approaches, Applied Sciences 9 (6) (2019) 1113. doi:10.3390/app9061113.
URL https://doi.org/10.3390/app9061113

[19] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, Y. Wang, Artificial intelligence in healthcare: past, present and future, Stroke and vascular neurology 2 (4) (2017) 230–243.

[20] Y. Zang, F. Zhang, C.-a. Di, D. Zhu, Advances of flexible pressure sensors toward artificial intelligence and health care applications, Materials Horizons 2 (2) (2015) 140–156.

[21] H. C. Koh, G. Tan, et al., Data mining applications in healthcare, Journal of healthcare information management 19 (2) (2011) 65.

[22] P. Hamet, J. Tremblay, Artificial intelligence in medicine, Metabolism 69 (2017) S36–S40.

[23] J. He, S. L. Baxter, J. Xu, J. Xu, X. Zhou, K. Zhang, The practical implementation of artificial intelligence technologies in medicine, Nature medicine 25 (1) (2019) 30–36.

[24] E. J. Topol, High-performance medicine: the convergence of human and artificial intelligence, Nature medicine 25 (1) (2019) 44–56.

[25] S. Reddy, J. Fox, M. P. Purohit, Artificial intelligence-enabled healthcare delivery, Journal of the Royal Society of Medicine 112 (1) (2019) 22–28.

[26] M. Begli, F. Derakhshan, H. Karimipour, A layered intrusion detection system for critical infrastructure using machine learning, in: 2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE), IEEE, 2019, pp. 120–124.

[27] B. Tjahjono, C. Esplugues, E. Ares, G. Pelaez, What does industry 4.0 mean to supply chain?, Procedia Manufacturing 13 (2017) 1175–1182.

[28] F. R. Lima-Junior, L. C. R. Carpinetti, Quantitative models for supply chain performance evaluation: a literature review, Computers & Industrial Engineering 113 (2017) 333–346.

[29] S. Zhang, C. K. M. Lee, K. Wu, K. L. Choy, Multi-objective optimization for sustainable supply chain network design considering multiple distribution channels, Expert Systems with Applications 65 (2016) 87–99.

[30] M. T. C. Cardoso, C. de Sousa, Artificial intelligence: the new source of productivity in the fashion supply chain, Ph.D. thesis (2019).

[31] G. Baryannis, S. Dani, S. Validi, G. Antoniou, Decision support systems and artificial intelligence in supply chain risk management, in: Revisiting Supply Chain Risk, Springer, 2019, pp. 53–71.

[32] G. Baryannis, S. Validi, S. Dani, G. Antoniou, Supply chain risk management and artificial intelligence: state of the art and future research directions, International Journal of Production Research 57 (7) (2019) 2179–2202.

[33] A. Calatayud, J. Mangan, M. Christopher, The self-thinking supply chain, Supply Chain Management: An International Journal (2019).

[34] B. Hellingrath, S. Lechtenberg, Applications of artificial intelligence in supply chain management and logistics: Focusing onto recognition for supply chain execution, in: The Art of Structuring, Springer, 2019, pp. 283–296.

[35] A. Chawla, A. Singh, A. Lamba, N. Gangwani, U. Soni, Demand forecasting using artificial neural networks—a case study of american retail corporation, in: Applications of Artificial Intelligence Techniques in Engineering, Springer, 2019, pp. 79–89.

[36] I. Barclay, A. Preece, I. Taylor, Defining the collective intelligence supply chain, arXiv preprint arXiv:1809.09444 (2018).

[37] A. Salamanis, D. D. Kehagias, C. K. Filelis-Papadopoulos, D. Tzovaras, G. A. Gravvanis, Managing spatial graph dependencies in large volumes of traffic data for travel-time prediction, IEEE Transactions on Intelligent Transportation Systems 17 (6) (2015) 1678–1687.

[38] E. Dahlman, S. Parkvall, J. Peisa, H. Tullberg, H. Murai, M. Fujioka, Artificial intelligence in future evolution of mobile communication, in: 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), IEEE, 2019. doi:10.1109/icaiic.2019.8669012.
URL https://doi.org/10.1109/icaiic.2019.8669012

[39] R. S. Bapi, K. S. Rao, M. V. Prasad, First international conference on artificial intelligence and cognitive computing.

[40] S. Makridakis, Forecasting the Impact of Artificial Intelligence, Part 3 of 4: The Potential Effects of AI on Businesses, Manufacturing, and Commerce, Foresight: The International Journal of Applied Forecasting (49) (2018) 18–27.
URL https://ideas.repec.org/a/for/ijafaa/y2018i49p18-27.html

[41] T. Küfner, T. H.-J. Uhlemann, B. Ziegler, Lean data in manufacturing systems: Using artificial intelligence for decentralized data reduction and information extraction, Procedia

CIRP 72 (2018) 219–224. doi:10.1016/j.procir.2018.03.125.
URL https://doi.org/10.1016/j.procir.2018.03.125

[42] D. J. Crandall, Artificial intelligence and manufacturing, Manufacturing Policy Initiative [2019]: Smart Factories: Issues of Information Governance. School of Public and Environmental Affairs, Indiana University (2019) 10–17.

[43] T. Vafeiadis, S. Krinidis, C. Ziogou, D. Ioannidis, S. Voutetakis, D. Tzovaras, Robust malfunction diagnosis in process industry time series, in: 2016 IEEE 14th International Conference on Industrial Informatics (INDIN), IEEE, 2016, pp. 111–116.

[44] J. Lee, H. Davari, J. Singh, V. Pandhare, Industrial artificial intelligence for industry 4.0-based manufacturing systems, Manufacturing letters 18 (2018) 20–23.

[45] M. Copeland, J. Soh, A. Puca, M. Manning, D. Gollob, Microsoft Azure: Planning, Deploying, and Managing Your Data Center in the Cloud, 1st Edition, Apress, USA, 2015.

[46] F. P. Miller, A. F. Vandome, J. McBrewster, Amazon Web Services, Alpha Press, 2010.

[47] J. L. Berral-García, A quick view on current techniques and machine learning algorithms for big data analytics, in: 2016 18th international conference on transparent optical networks (ICTON), IEEE, 2016, pp. 1–4.

[48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[49] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035.

[50] F. Chollet, et al., Keras, https://github.com/fchollet/keras (2015).

[51] Theano Development Team, Theano: A Python framework for fast computation of mathematical expressions, arXiv e-prints abs/1605.02688 (May 2016).
URL http://arxiv.org/abs/1605.02688

[52] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015).
URL http://tensorflow.org/

[53] T. Wuest, D. Weimer, C. Irgens, K.-D. Thoben, Machine learning in manufacturing: advantages, challenges, and applications, Production & Manufacturing Research 4 (1) (2016) 23–45.

[54] I. H. Wijesinghe, S. D. Viswakula, Machine learning for preauction sample selection, in: 2018 National Information Technology Conference (NITC), IEEE, 2018, pp. 1–8.

[55] T. D. Nielsen, F. V. Jensen, Bayesian networks and decision graphs, Springer Science & Business Media, 2009.

[56] F. V. Jensen, et al., An introduction to Bayesian networks, Vol. 210, UCL press London, 1996.

[57] K. M. Al-Aidaroos, A. A. Bakar, Z. Othman, Naive bayes variants in classification learning, in: 2010 International Conference on Information Retrieval & Knowledge Management (CAMP), IEEE, 2010, pp. 276–281.

[58] O. Ardhapure, G. Patil, D. Udani, K. Jetha, Comparative study of classification algorithm for text based categorization, IJRET: International Journal of Research in Engineering and Technology 5 (2016) 217–220.

[59] D. Anguita, A. Ghio, N. Greco, L. Oneto, S. Ridella, Model selection for support vector machines: Advantages and disadvantages of the machine learning theory, in: The 2010 international joint conference on neural networks (IJCNN), IEEE, 2010, pp. 1–8.

[60] L. Auria, R. A. Moro, Support vector machines (svm) as a technique for solvency analysis (2008).

[61] E. J. Bredensteiner, K. P. Bennett, Multicategory classification by support vector machines, in: Computational Optimization, Springer, 1999, pp. 53–79.

[62] L. Wang, Z. Zhang, C. X. R. C. Design, Theory and applications, Support Vector Machines, Springer-Verlag, Berlin Heidelber (2005).

[63] J. A. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural processing letters 9 (3) (1999) 293–300.

[64] B. Schölkopf, A. J. Smola, F. Bach, et al., Learning with kernels: support vector machines, regularization, optimization, and beyond, MIT press, 2002.

[65] C.-R. Liu, L.-H. Duan, P.-W. Chen, C.-C. Yang, Monitoring machine tool based on external physical characteristics of the machine tool using machine learning algorithm, in: 2018 First International Conference on Artificial Intelligence for Industries (AI4I), IEEE, 2018, pp. 5–8.

[66] W. An, Y. Sun, D. Wang, Study on support vector machine in calculating steel quenching degree, in: 2006 6th World Congress on Intelligent Control and Automation, Vol. 2, IEEE, 2006, pp. 7780–7783.

[67] D. Lieber, M. Stolpe, B. Konrad, J. Deuse, K. Morik, Quality prediction in interlinked manufacturing processes based on supervised & unsupervised machine learning, Procedia Cirp 7 (2013) 193–198.

[68] J.-L. Loyer, E. Henriques, M. Fontul, S. Wiseall, Comparison of machine learning methods applied to the estimation of manufacturing cost of jet engine components, International Journal of Production Economics 178 (2016) 109–119.

[69] D. W. Aha, D. Kibler, M. K. Albert, Instance-based learning algorithms, Machine learning 6 (1) (1991) 37–66.

[70] H. Brighton, C. Mellish, Advances in instance selection for instance-based learning algorithms, Data mining and knowledge discovery 6 (2) (2002) 153–172.

[71] J. Zhang, Selecting typical instances in instance-based learning, in: Machine Learning Proceedings 1992, Elsevier, 1992, pp. 470–479.

[72] L. Romeo, M. Paolanti, G. Bocchini, J. Loncarski, E. Frontoni, An innovative design support system for industry 4.0 based on machine learning approaches, in: 2018 5th International Symposium on Environment-Friendly Energies and Applications (EFEA), IEEE, 2018, pp. 1–6.

[73] J. M. Keller, M. R. Gray, J. A. Givens, A fuzzy k-nearest neighbor algorithm, IEEE transactions on systems, man, and cybernetics (4) (1985) 580–585.

[74] K. Fukunaga, P. M. Narendra, A branch and bound algorithm for computing k-nearest neighbors, IEEE transactions on computers 100 (7) (1975) 750–753.

[75] I. Triguero, D. García-Gil, J. Maillo, J. Luengo, S. García,

F. Herrera, Transforming big data into smart data: An insight on the use of the k-nearest neighbors algorithm to obtain quality data, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 9 (2) (2019) e1289.

[76] S. B. Imandoust, M. Bolandraftar, Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background, International Journal of Engineering Research and Applications 3 (5) (2013) 605–610.

[77] M. M. Mijwel, Artificial neural networks advantages and disadvantages, Retrieved from LinkedIn: https://www. linkedin. com/pulse/artificial-neuralnet works-advantages-disadvantages-maad-m-mijwel (2018).

[78] J. V. Tu, Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes, Journal of clinical epidemiology 49 (11) (1996) 1225–1231.

[79] M. A. Friedl, C. E. Brodley, Decision tree classification of land cover from remotely sensed data, Remote sensing of environment 61 (3) (1997) 399–409.

[80] C. W. Olanow, R. L. Watts, W. C. Koller, An algorithm (decision tree) for the management of parkinson's disease (2001):: Treatment guidelines, Neurology 56 (suppl 5) (2001) S1–S88.

[81] D. Wu, C. Jennings, J. Terpenny, R. X. Gao, S. Kumara, A comparative study on machine learning algorithms for smart manufacturing: tool wear prediction using random forests, Journal of Manufacturing Science and Engineering 139 (7) (2017).

[82] N. Kolokas, T. Vafeiadis, D. Ioannidis, D. Tzovaras, Forecasting faults of industrial equipment using machine learning classifiers, in: 2018 Innovations in Intelligent Systems and Applications (INISTA), IEEE, 2018, pp. 1–6.

[83] S. Schaal, C. G. Atkeson, S. Vijayakumar, Real-time robot learning with locally weighted statistical learning, in: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), Vol. 1, IEEE, 2000, pp. 288–293.

[84] L. Scime, J. Beuth, Using machine learning to identify in-situ melt pool signatures indicative of flaw formation in a laser powder bed fusion additive manufacturing process, Additive Manufacturing 25 (2019) 151–165. doi:10.1016/j.addma.2018.11.010.
URL https://doi.org/10.1016/j.addma.2018.11.010

[85] R. H. Myers, R. H. Myers, Classical and modern regression with applications, Vol. 2, Duxbury press Belmont, CA, 1990.

[86] K. J. Preacher, P. J. Curran, D. J. Bauer, Computational tools for probing interactions in multiple linear regression, multilevel modeling, and latent curve analysis, Journal of educational and behavioral statistics 31 (4) (2006) 437–448.

[87] D. F. Andrews, A robust method for multiple linear regression, Technometrics 16 (4) (1974) 523–531.

[88] Y.-M. Wang, T. M. Elhag, A comparison of neural network, evidential reasoning and multiple regression analysis in modelling bridge risks, Expert Systems with Applications 32 (2) (2007) 336–348. doi:10.1016/j.eswa.2005.11.029.
URL https://doi.org/10.1016/j.eswa.2005.11.029

[89] T. Ayer, J. Chhatwal, O. Alagoz, C. E. Kahn Jr, R. W. Woods, E. S. Burnside, Comparison of logistic regression and artificial neural network models in breast cancer risk estimation, Radiographics 30 (1) (2010) 13–22.

[90] S. R. Safavian, D. Landgrebe, A survey of decision tree classifier methodology, IEEE transactions on systems, man, and cybernetics 21 (3) (1991) 660–674.

[91] D. Westreich, J. Lessler, M. J. Funk, Propensity score estimation: neural networks, support vector machines, decision trees (cart), and meta-classifiers as alternatives to logistic regression, Journal of clinical epidemiology 63 (8) (2010) 826–833.

[92] P. H. Swain, H. Hauska, The decision tree classifier: Design and potential, IEEE Transactions on Geoscience Electronics 15 (3) (1977) 142–147.

[93] V. Podgorelec, P. Kokol, B. Stiglic, I. Rozman, Decision trees: an overview and their use in medicine, Journal of medical systems 26 (5) (2002) 445–463.

[94] L. Bottou, Large-scale machine learning with stochastic gradient descent, in: Proceedings of COMPSTAT'2010, Springer, 2010, pp. 177–186.

[95] S. Ruder, An overview of gradient descent optimization algorithms, arXiv preprint arXiv:1609.04747 (2016).

[96] L. Mason, J. Baxter, P. L. Bartlett, M. R. Frean, Boosting algorithms as gradient descent, in: Advances in neural information processing systems, 2000, pp. 512–518.

[97] T. J. Hastie, R. J. Tibshirani, Generalized additive models, Vol. 43, CRC press, 1990.

[98] A. Buja, T. Hastie, R. Tibshirani, Linear smoothers and additive models, The Annals of Statistics (1989) 453–510.

[99] A. Guisan, T. C. Edwards Jr, T. Hastie, Generalized linear and generalized additive models in studies of species distributions: setting the scene, Ecological modelling 157 (2-3) (2002) 89–100.

[100] C. G. Atkeson, A. W. Moore, S. Schaal, Locally weighted learning for control, in: Lazy learning, Springer, 1997, pp. 75–113.

[101] S. Vijayakumar, S. Schaal, Locally weighted projection regression: An o (n) algorithm for incremental real time learning in high dimensional space, in: Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Vol. 1, 2000, pp. 288–293.

[102] H. K. Kim, H. Kim, S. Cho, Bag-of-concepts: Comprehending document representation through clustering words in distributed representation, Neurocomputing 266 (2017) 336–352.

[103] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.

[104] A. Statnikov, L. Wang, C. F. Aliferis, A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification, BMC bioinformatics 9 (1) (2008) 319.

[105] T. D. Buskirk, Surveying the forests and sampling the trees: an overview of classification and regression trees and random forests with applications in survey research, Survey Practice 11 (1) (2018) 1–13.

[106] Y. Ao, H. Li, L. Zhu, S. Ali, Z. Yang, The linear random forest algorithm and its advantages in machine learning assisted logging regression modeling, Journal of Petroleum Science and Engineering 174 (2019) 776–789.

[107] J. A. Hartigan, M. A. Wong, Algorithm as 136: A k-means clustering algorithm, Journal of the royal statistical society. series c (applied statistics) 28 (1) (1979) 100–108.

[108] K. Krishna, M. N. Murty, Genetic k-means algorithm, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 29 (3) (1999) 433–439.

[109] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, A. Y. Wu, An efficient k-means clustering algorithm: Analysis and implementation, IEEE transactions on pattern analysis and machine intelligence 24 (7) (2002) 881–892.

[110] J. M. Pena, J. A. Lozano, P. Larranaga, An empirical comparison of four initialization methods for the k-means algorithm, Pattern recognition letters 20 (10) (1999) 1027–1040.

[111] M. Santini, Advantages & disadvantages of k-means and hierarchical clustering (unsupervised learning), URL: http://santini. se/teaching/ml/2016/Lect_10/10c_Unsupervise

dMethods. pdf (Accesed 17.04. 2019) (2016).
[112] N. Litvinenko, O. Mamyrbayev, A. Shayakhmetova, M. Turdalyuly, Clusterization by the k-means method when k is unknown, in: ITM Web of Conferences, Vol. 24, EDP Sciences, 2019, p. 01013.
[113] C. Yuan, H. Yang, Research on k-value selection method of k-means clustering algorithm, J—Multidisciplinary Scientific Journal 2 (2) (2019) 226–235.
[114] X. Qu, L. Yang, K. Guo, L. Ma, M. Sun, M. Ke, M. Li, A survey on the development of self-organizing maps for unsupervised intrusion detection, Mobile Networks and Applications (2019) 1–22.
[115] O. E. Dragomir, F. Dragomir, M. Radulescu, Matlab application of kohonen self-organizing map to classify consumers' load profiles., in: ITQM, 2014, pp. 474–479.
[116] S. Mahadevan, G. Theocharous, Optimizing production manufacturing using reinforcement learning., in: FLAIRS Conference, Vol. 372, 1998, p. 377.
[117] T. Das, A. Gosavi, S. Mahadevan, N. Marchalleck, Solving semi-markov decision problems using average reward reinforcement learning, Management Science 45 (11 1998). doi:10.1287/mnsc.45.4.560.
[118] M. Hesse, J. Timmermann, E. Hüllermeier, A. Trächtler, A reinforcement learning strategy for the swing-up of the double pendulum on a cart, Procedia Manufacturing 24 (2018) 15–20.
[119] F. Dominici, A. McDermott, S. L. Zeger, J. M. Samet, On the use of generalized additive models in time-series studies of air pollution and health, American journal of epidemiology 156 (3) (2002) 193–203.
[120] C. Ning, F. You, Optimization under uncertainty in the era of big data and deep learning: When machine learning meets mathematical programming, Computers & Chemical Engineering 125 (2019) 434–448.
[121] J. Wang, Y. Ma, L. Zhang, R. X. Gao, D. Wu, Deep learning for smart manufacturing: Methods and applications, Journal of Manufacturing Systems 48 (2018) 144–156.
[122] G. Helbing, M. Ritter, Deep learning for fault detection in wind turbines, Renewable and Sustainable Energy Reviews 98 (2018) 189–198.
[123] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, R. X. Gao, Deep learning and its applications to machine health monitoring, Mechanical Systems and Signal Processing 115 (2019) 213–237.
[124] L. Mescheder, S. Nowozin, A. Geiger, Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks, arXiv preprint arXiv:1701.04722 (2017).
[125] X. Zhu, T. Fujii, Modulation classification for cognitive radios using stacked denoising autoencoders, International Journal of Satellite Communications and Networking 35 (5) (2017) 517–531.
[126] F. Jia, Y. Lei, J. Lin, X. Zhou, N. Lu, Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data, Mechanical Systems and Signal Processing 72 (2016) 303–315.
[127] J. Guo, X. Xie, R. Bie, L. Sun, Structural health monitoring by using a sparse coding-based deep learning algorithm with wireless sensor networks, Personal and ubiquitous computing 18 (8) (2014) 1977–1987.
[128] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, X. Chen, A sparse auto-encoder-based deep neural network approach for induction motor faults classification, Measurement 89 (2016) 171–178.
[129] L. Wang, X. Zhao, J. Pei, G. Tang, Transformer fault diagnosis using continuous sparse autoencoder, SpringerPlus 5 (1) (2016) 1–13.
[130] A. Kamilaris, F. X. Prenafeta-Boldú, A review of the use of convolutional neural networks in agriculture, The Journal of Agricultural Science 156 (3) (2018) 312–322.
[131] A. Mikołajczyk, M. Grochowski, Data augmentation for improving deep learning in image classification problem, in: 2018 international interdisciplinary PhD workshop (IIPhDW), IEEE, 2018, pp. 117–122.
[132] T. Kotsiopoulos, L. Leontaris, N. Dimitriou, D. Ioannidis, F. Oliveira, J. Sacramento, S. Amanatiadis, G. Karagiannis, K. Votis, D. Tzovaras, P. Sarigiannidis, Deep multi-sensorial data analysis for production monitoring in hard metal industry, The International Journal of Advanced Manufacturing Technology (Oct. 2020). doi:10.1007/s00170-020-06173-1.
URL https://doi.org/10.1007/s00170-020-06173-1
[133] S. Ma, F. Chu, Ensemble deep learning-based fault diagnosis of rotor bearing systems, Computers in Industry 105 (2019) 143–152.
[134] N. Dimitriou, L. Leontaris, T. Vafeiadis, D. Ioannidis, T. Wotherspoon, G. Tinker, D. Tzovaras, Fault diagnosis in microelectronics attachment via deep learning analysis of 3-d laser scans, IEEE Transactions on Industrial Electronics 67 (7) (2020) 5748–5757. doi:10.1109/tie.2019.2931220.
URL https://doi.org/10.1109/tie.2019.2931220
[135] B. R. Sutherland, Locating photovoltaic installations with deep learning, Joule 2 (12) (2018) 2512–2513.
[136] S. J. Lee, J. Ban, H. Choi, S. W. Kim, Localization of slab identification numbers using deep learning, in: 2016 16th International Conference on Control, Automation and Systems (ICCAS), IEEE, 2016, pp. 1174–1176.
[137] O. Janssens, V. Slavkovikj, B. Vervisch, K. Stockman, M. Loccufier, S. Verstockt, R. Van de Walle, S. Van Hoecke, Convolutional neural network based fault detection for rotating machinery, Journal of Sound and Vibration 377 (2016) 331–345.
[138] C. Lu, Z. Wang, B. Zhou, Intelligent fault diagnosis of rolling bearing using hierarchical convolutional network based health state classification, Advanced Engineering Informatics 32 (2017) 139–151.
[139] X. Guo, L. Chen, C. Shen, Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis, Measurement 93 (2016) 490–502.
[140] D. Weimer, B. Scholz-Reiter, M. Shpitalni, Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection, CIRP Annals 65 (1) (2016) 417–420.
[141] R. Ren, T. Hung, K. C. Tan, A generic deep-learning-based approach for automated surface inspection, IEEE transactions on cybernetics 48 (3) (2017) 929–940.
[142] I. Sutskever, G. E. Hinton, G. W. Taylor, The recurrent temporal restricted boltzmann machine, in: Advances in neural information processing systems, 2009, pp. 1601–1608.
[143] H. Shao, H. Jiang, X. Zhang, M. Niu, Rolling bearing fault diagnosis using an optimization deep belief network, Measurement Science and Technology 26 (11) (2015) 115002.
[144] P. Tamilselvan, P. Wang, Failure diagnosis using deep belief learning based health state classification, Reliability Engineering & System Safety 115 (2013) 124–135.
[145] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE transactions on neural networks 5 (2) (1994) 157–166.
[146] T. Mikolov, S. Kombrink, L. Burget, J. Černockỳ, S. Khudanpur, Extensions of recurrent neural network language model, in: 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, 2011, pp. 5528–5531.
[147] P. Tamilselvan, P. Wang, Failure diagnosis using deep belief learning based health state classification, Reliability Engineer-

ing & System Safety 115 (2013) 124–135.
[148] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen, J. Wang, Machine health monitoring using local feature-based gated recurrent unit networks, IEEE Transactions on Industrial Electronics 65 (2) (2017) 1539–1548.
[149] B. A. Pearlmutter, Gradient calculations for dynamic recurrent neural networks: A survey, IEEE Transactions on Neural networks 6 (5) (1995) 1212–1228.
[150] W. Chan, I. Lane, Deep recurrent neural networks for acoustic modelling, arXiv preprint arXiv:1504.01482 (2015).
[151] S. K. Pal, S. Mitra, Multilayer perceptron, fuzzy sets, classifiaction (1992).
[152] M. W. Gardner, S. Dorling, Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences, Atmospheric environment 32 (14-15) (1998) 2627–2636.
[153] D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxley, B. W. Suter, The multilayer perceptron as an approximation to a bayes optimal discriminant function, IEEE Transactions on Neural Networks 1 (4) (1990) 296–298.
[154] S. Wood, R. Muthyala, Y. Jin, Y. Qin, N. Rukadikar, A. Rai, H. Gao, Automated industry classification with deep learning, in: 2017 IEEE International Conference on Big Data (Big Data), IEEE, 2017, pp. 122–129.
[155] J. Francis, L. Bian, Deep learning for distortion prediction in laser-based additive manufacturing using big data, Manufacturing Letters 20 (2019) 10–14.
[156] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.
[157] D. Haifeng, H. Siqi, Natural scene text detection based on yolo v2 network model, in: Journal of Physics: Conference Series, Vol. 1634, IOP Publishing, 2020, p. 012013.
[158] W. Zhihuan, C. Xiangning, G. Yongming, L. Yuntao, Rapid target detection in high resolution remote sensing images using yolo model, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 42 (2018) 3.
[159] R. Cheng, A survey: Comparison between convolutional neural network and yolo in image identification, in: Journal of Physics: Conference Series, Vol. 1453, 2020, p. 012139.
[160] N. Zhang, Y. Liu, L. Zou, H. Zhao, W. Dong, H. Zhou, H. Zhou, M. Huang, Automatic recognition of oil industry facilities based on deep learning, in: IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2018, pp. 2519–2522.
[161] T. Li, B. Wang, Y. Jiang, Y. Zhang, Y. Yan, Restricted boltzmann machine-based approaches for link prediction in dynamic networks, IEEE Access 6 (2018) 29940–29951.
[162] M. H. Rafiei, W. H. Khushefati, R. Demirboga, H. Adeli, Supervised deep restricted boltzmann machine for estimation of concrete., ACI Materials Journal 114 (2) (2017).
[163]
[164] Y. Luo, K. Li, Y. Li, D. Cai, C. Zhao, Q. Meng, Three-layer bayesian network for classification of complex power quality disturbances, IEEE Transactions on Industrial Informatics 14 (9) (2018) 3997–4006. doi:10.1109/tii.2017.2785321.
URL https://doi.org/10.1109/tii.2017.2785321
[165] M. Babar, M. U. Tariq, M. A. Jan, Secure and resilient demand side management engine using machine learning for iot-enabled smart grid, Sustainable Cities and Society 62 (2020) 102370.
[166] E. Vinagre, T. Pinto, S. Ramos, Z. Vale, J. M. Corchado, Electrical energy consumption forecast using support vector machines, in: 2016 27th International Workshop on Database and Expert Systems Applications (DEXA), IEEE, 2016. doi:10.1109/dexa.2016.046.
URL https://doi.org/10.1109/dexa.2016.046
[167] D. D. Yong, S. Bhowmik, F. Magnago, An effective power quality classifier using wavelet transform and support vector machines, Expert Systems with Applications 42 (15-16) (2015) 6075–6081. doi:10.1016/j.eswa.2015.04.002.
URL https://doi.org/10.1016/j.eswa.2015.04.002
[168] P. Schuetz, R. Durrer, D. Gwerder, M. Geidl, J. Worlitschek, Poster abstract: state of operation recognition for heat pumps from smart grid monitoring data, Computer Science - Research and Development 33 (1-2) (2017) 259–261. doi:10.1007/s00450-017-0372-5.
URL https://doi.org/10.1007/s00450-017-0372-5
[169] S. Behera, R. Misra, SmartPeak, in: Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference on ZZZ - AICCC 18, ACM Press, 2018. doi:10.1145/3299819.3299833.
URL https://doi.org/10.1145/3299819.3299833
[170] G. Efstathopoulos, P. R. Grammatikis, P. Sarigiannidis, V. Argyriou, A. Sarigiannidis, K. Stamatakis, M. K. Angelopoulos, S. K. Athanasopoulos, Operational data based intrusion detection system for smart grid, in: 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), IEEE, 2019. doi:10.1109/camad.2019.8858503.
URL https://doi.org/10.1109/camad.2019.8858503
[171] Y. Weng, R. Negi, C. Faloutsos, M. D. Ilic, Robust data-driven state estimation for smart grid, IEEE Transactions on Smart Grid 8 (4) (2017) 1956–1967. doi:10.1109/tsg.2015.2512925.
URL https://doi.org/10.1109/tsg.2015.2512925
[172] O. Valgaev, F. Kupzog, H. Schmeck, Low-voltage power demand forecasting using k-nearest neighbors approach, in: 2016 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia), IEEE, 2016. doi:10.1109/isgt-asia.2016.7796525.
URL https://doi.org/10.1109/isgt-asia.2016.7796525
[173] M. Macedo, J. Galo, L. de Almeida, A. de C. Lima, Demand side management using artificial neural networks in a smart grid environment, Renewable and Sustainable Energy Reviews 41 (2015) 128–133. doi:10.1016/j.rser.2014.08.035.
URL https://doi.org/10.1016/j.rser.2014.08.035
[174] K. Förderer, M. Ahrens, K. Bao, I. Mauser, H. Schmeck, Towards the modeling of flexibility using artificial neural networks in energy management and smart grids, in: Proceedings of the Ninth International Conference on Future Energy Systems, ACM, 2018. doi:10.1145/3208903.3208915.
URL https://doi.org/10.1145/3208903.3208915
[175] J. Kim, S. Cho, K. Ko, R. R. Rao, Short-term electric load prediction using multiple linear regression method, in: 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), IEEE, 2018. doi:10.1109/smartgridcomm.2018.8587489.
URL https://doi.org/10.1109/smartgridcomm.2018.8587489
[176] N. T. Le, W. Benjapolakul, A data imputation model in phasor measurement units based on bagged averaging of multiple linear regression, IEEE Access 6 (2018) 39324–39333. doi:10.1109/access.2018.2856768.
URL https://doi.org/10.1109/access.2018.2856768
[177] M. Eissa, A. Ali, K. Abdel-Latif, A. Al-Kady, A frequency control technique based on decision tree concept by managing thermostatically controllable loads at smart grids, International Journal of Electrical Power & Energy Systems 108 (2019) 40–51.
[178] D. S. Terzi, B. Arslan, S. Sagiroglu, Smart grid security evaluation with a big data use case, in: 2018 IEEE 12th International

Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG 2018), IEEE, 2018, pp. 1–6.
[179] P. D. Achlerkar, S. R. Samantaray, M. S. Manikandan, Variational mode decomposition and decision tree based detection and classification of power quality disturbances in grid-connected distributed generation system, IEEE Transactions on Smart Grid 9 (4) (2016) 3122–3132.
[180] R. J. Bessa, A. Trindade, C. S. Silva, V. Miranda, Probabilistic solar power forecasting in smart grids using distributed information, International Journal of Electrical Power & Energy Systems 72 (2015) 16–23.
[181] R. Punmiya, S. Choe, Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing, IEEE Transactions on Smart Grid 10 (2) (2019) 2326–2329.
[182] R. Razavi, A. Gharipour, M. Fleury, I. J. Akpan, A practical feature-engineering framework for electricity theft detection in smart grids, Applied energy 238 (2019) 481–494.
[183] P. Pompey, A. Bondu, Y. Goude, M. Sinn, Massive-scale simulation of electrical load in smart grids using generalized additive models, in: Modeling and Stochastic Learning for Forecasting in High Dimensions, Springer, 2015, pp. 193–212.
[184] S. B. Taieb, R. Huser, R. J. Hyndman, M. G. Genton, et al., Probabilistic time series forecasting with boosted additive models: an application to smart meter data, Department of Economics and business statistics, Monash University (2015).
[185] V. Thouvenot, A. Pichavant, Y. Goude, A. Antoniadis, J.-M. Poggi, Electricity forecasting using multi-stage estimators of nonlinear additive models, IEEE Transactions on Power Systems 31 (5) (2015) 3665–3673.
[186] J. Zhang, C. Chung, Z. Wang, X. Zheng, Instantaneous electromechanical dynamics monitoring in smart transmission grid, IEEE Transactions on Industrial Informatics 12 (2) (2015) 844–852.
[187] A. Lahouar, J. B. H. Slama, Day-ahead load forecast using random forest and expert input selection, Energy Conversion and Management 103 (2015) 1040–1051.
[188] R. Lin, Z. Pei, Z. Ye, B. Wu, G. Yang, A voted based random forests algorithm for smart grid distribution network faults prediction, Enterprise Information Systems 14 (4) (2020) 496–514.
[189] S. Singh, A. Yassine, R. Benlamri, Towards hybrid energy consumption prediction in smart grids with machine learning, in: 2018 4th International Conference on Big Data Innovations and Applications (Innovate-Data), IEEE, 2018, pp. 44–50.
[190] S. Ahmed, Y. Lee, S.-H. Hyun, I. Koo, Unsupervised machine learning-based detection of covert data integrity assault in smart grid networks utilizing isolation forest, IEEE Transactions on Information Forensics and Security 14 (10) (2019) 2765–2777.
[191] A. Al-Wakeel, J. Wu, N. Jenkins, K-means based load estimation of domestic smart meter measurements, Applied energy 194 (2017) 333–342.
[192] C.-N. Yu, P. Mirowski, T. K. Ho, A sparse coding approach to household electricity demand forecasting in smart grids, IEEE Transactions on Smart Grid 8 (2) (2016) 738–748.
[193] A. Starke, J. McNair, R. Trevizan, A. Bretas, J. Peeples, A. Zare, Toward resilient smart grid communications using distributed sdn with ml-based anomaly detection, in: International Conference on Wired/Wireless Internet Communication, Springer, 2018, pp. 83–94.
[194] J. Llanos, R. Morales, A. Núñez, D. Sáez, M. Lacalle, L. G. Marín, R. Hernández, F. Lanas, Load estimation for microgrid planning based on a self-organizing map methodology, Applied Soft Computing 53 (2017) 323–335.
[195] C. Tong, J. Li, C. Lang, F. Kong, J. Niu, J. J. Rodrigues, An efficient deep model for day-ahead electricity load forecasting with stacked denoising auto-encoders, Journal of parallel and distributed computing 117 (2018) 267–273.
[196] S. Lu, G. Lin, H. Que, L. Chen, H. Liu, C. Ye, D. Yi, Electric load data characterising and forecasting based on trend index and auto-encoders, The Journal of Engineering 2018 (17) (2018) 1915–1921.
[197] H. Yang, R. C. Qiu, X. Shi, X. He, Deep learning architecture for voltage stability evaluation in smart grid based on variational autoencoders, arXiv preprint arXiv:1808.05762 (2018).
[198] S. Ahmed, Y. Lee, S.-H. Hyun, I. Koo, Mitigating the impacts of covert cyber attacks in smart grids via reconstruction of measurement data utilizing deep denoising autoencoders, Energies 12 (16) (2019) 3091.
[199] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, Y. Zhou, Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids, IEEE Transactions on Industrial Informatics 14 (4) (2017) 1606–1615.
[200] M. Zahid, F. Ahmed, N. Javaid, R. A. Abbasi, H. S. Zainab Kazmi, A. Javaid, M. Bilal, M. Akbar, M. Ilahi, Electricity price and load forecasting using enhanced convolutional neural network and enhanced support vector regression in smart grids, Electronics 8 (2) (2019) 122.
[201] P.-H. Kuo, C.-J. Huang, A high precision artificial neural networks model for short-term energy load forecasting, Energies 11 (1) (2018) 213.
[202] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, Y. Zhang, Short-term residential load forecasting based on lstm recurrent neural network, IEEE Transactions on Smart Grid 10 (1) (2017) 841–851.
[203] X. Ouyang, Z. Ma, Using lstm networks to identify false data of smart terminals in the smart grid, in: 2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS), IEEE, 2017, pp. 765–768.
[204] M. Hasan, R. N. Toma, A.-A. Nahid, M. Islam, J.-M. Kim, et al., Electricity theft detection in smart grid systems: a cnn-lstm based approach, Energies 12 (17) (2019) 3310.
[205] M. Abdel-Nasser, K. Mahmoud, Accurate photovoltaic power forecasting models using deep lstm-rnn, Neural Computing and Applications 31 (7) (2019) 2727–2740.
[206] D. L. Marino, K. Amarasinghe, M. Manic, Building energy load forecasting using deep neural networks, in: IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2016, pp. 7046–7051.
[207] Y. He, G. J. Mendis, J. Wei, Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism, IEEE Transactions on Smart Grid 8 (5) (2017) 2505–2516.
[208] D. M. Menon, N. Radhika, A secure deep belief network architecture for intrusion detection in smart grid home area network, IIOAB Journal 7 (2016) 479–483.
[209] Y. He, J. Deng, H. Li, Short-term power load forecasting with deep belief network and copula models, in: 2017 9th International conference on intelligent human-machine systems and cybernetics (IHMSC), Vol. 1, IEEE, 2017, pp. 191–194.
[210] K. Hamedani, L. Liu, R. Atat, J. Wu, Y. Yi, Reservoir computing meets smart grids: Attack detection using delayed feedback networks, IEEE Transactions on Industrial Informatics 14 (2) (2017) 734–743.
[211] J. Moon, Y. Kim, M. Son, E. Hwang, Hybrid short-term load forecasting scheme using random forest and multilayer perceptron, Energies 11 (12) (2018) 3283.
[212] O. A. Alimi, K. Ouahada, A. M. Abu-Mahfouz, Real time security assessment of the power system using a hybrid support

vector machine and multilayer perceptron neural network algorithms, Sustainability 11 (13) (2019) 3586.
[213] F. Wahid, R. Ghazali, A. S. Shah, M. Fayaz, Prediction of energy consumption in the buildings using multi-layer perceptron and random forest, IJAST 101 (2017) 13–22.
[214] P. Radoglou-Grammatikis, P. Sarigiannidis, G. Efstathopoulos, P.-A. Karypidis, A. Sarigiannidis, Diderot: an intrusion detection and prevention system for dnp3-based scada systems, in: Proceedings of the 15th International Conference on Availability, Reliability and Security, 2020, pp. 1–8.
[215] P. R. Grammatikis, P. Sarigiannidis, G. Efstathopoulos, E. Panaousis, ARIES: A novel multivariate intrusion detection system for smart grid, Sensors 20 (18) (2020) 5305. doi:10.3390/s20185305.
URL https://doi.org/10.3390/s20185305
[216] Z. Ge, Z. Song, S. X. Ding, B. Huang, Data mining and analytics in the process industry: The role of machine learning, Ieee Access 5 (2017) 20590–20616.
[217] K. Medjaher, D. A. Tobon-Mejia, N. Zerhouni, Remaining useful life estimation of critical components with application to bearings, IEEE Transactions on Reliability 61 (2) (2012) 292–302.
[218] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, IEEE internet of things journal 3 (5) (2016) 637–646.
[219] H. Pang, K.-L. Tan, Authenticating query results in edge computing, in: Proceedings. 20th International Conference on Data Engineering, IEEE, 2004, pp. 560–571.
[220] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the first edition of the MCC workshop on Mobile cloud computing, 2012, pp. 13–16.
[221] U. Hunkeler, H. L. Truong, A. Stanford-Clark, Mqtt-s—a publish/subscribe protocol for wireless sensor networks, in: 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08), IEEE, 2008, pp. 791–798.
[222] S. Vinoski, Advanced message queuing protocol, IEEE Internet Computing 10 (6) (2006) 87–89.
[223] G. Pardo-Castellote, Omg data-distribution service: Architectural overview, in: 23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings., IEEE, 2003, pp. 200–206.
[224] K. E. Nolan, W. Guibene, M. Y. Kelly, An evaluation of low power wide area network technologies for the internet of things, in: 2016 international wireless communications and mobile computing conference (IWCMC), IEEE, 2016, pp. 439–444.
[225] R. Ratasuk, B. Vejlgaard, N. Mangalvedhe, A. Ghosh, Nb-iot system for m2m communication, in: 2016 IEEE wireless communications and networking conference, IEEE, 2016, pp. 1–5.
[226] C. Bekara, Security issues and challenges for the iot-based smart grid., in: FNC/MobiSPC, 2014, pp. 532–537.
[227] P. I. R. Grammatikis, P. G. Sarigiannidis, I. D. Moscholios, Securing the internet of things: Challenges, threats and solutions, Internet of Things 5 (2019) 41–70.
[228] A. Triantafyllou, P. Sarigiannidis, T. D. Lagkas, Network protocols, schemes, and mechanisms for internet of things (iot): Features, open challenges, and trends, Wireless communications and mobile computing 2018 (2018).
[229] N. Namvar, W. Saad, N. Bahadori, B. Kelley, Jamming in the internet of things: A game-theoretic perspective, in: 2016 IEEE Global Communications Conference (GLOBECOM), IEEE, 2016, pp. 1–6.
[230] X. Tang, P. Ren, Z. Han, Jamming mitigation via hierarchical security game for iot communications, IEEE Access 6 (2018) 5766–5779.
[231] Y. Chen, Y. Li, D. Xu, L. Xiao, Dqn-based power control for iot transmission against jamming, in: 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), IEEE, 2018, pp. 1–5.
[232] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, S. Liu, Efficient and privacy-enhanced federated learning for industrial artificial intelligence, IEEE Transactions on Industrial Informatics 16 (10) (2020) 6532–6542. doi:10.1109/tii.2019.2945367.
URL https://doi.org/10.1109/tii.2019.2945367
[233] S. Savazzi, M. Nicoli, V. Rampa, Federated learning with cooperating devices: A consensus approach for massive iot networks, IEEE Internet of Things Journal 7 (5) (2020) 4641–4654.
[234] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, ACM Transactions on Intelligent Systems and Technology (TIST) 10 (2) (2019) 1–19.
[235] M. Mohri, G. Sivek, A. T. Suresh, Agnostic federated learning, arXiv preprint arXiv:1902.00146 (2019).
[236] R. Shokri, V. Shmatikov, Privacy-preserving deep learning, in: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, 2015, pp. 1310–1321.
[237] G. Xu, H. Li, S. Liu, K. Yang, X. Lin, VerifyNet: Secure and verifiable federated learning, IEEE Transactions on Information Forensics and Security 15 (2020) 911–926. doi:10.1109/tifs.2019.2929409.
URL https://doi.org/10.1109/tifs.2019.2929409
[238] Y. Qu, S. R. Pokhrel, S. Garg, L. Gao, Y. Xiang, A blockchained federated learning framework for cognitive computing in industry 4.0 networks, IEEE Transactions on Industrial Informatics (2020) 1–1doi:10.1109/tii.2020.3007817.
URL https://doi.org/10.1109/tii.2020.3007817