CPS Data Streams Analytics based on Machine Learning for Cloud and Fog Computing: A Survey

Fei, X, Shah, N, Verba, N, Chao, K-M, Sanchez-Anguix, V, Lewandowski, J, James, A & Usman, Z

DOI 10.1016/j.future.2018.06.042
ISSN 0167-739X
Publisher: Elsevier

CPS Data Streams Analytics based on Machine Learning for Cloud and Fog Computing: A Survey

Xiang Fei[a], Nazaraf Shah[a], Nandor Verba[a], Kuo-Ming Chao[a], Victor Sanchez-Anguix[a], Jacek Lewandowski[a, d], Anne James[b], Zahid Usman[c]

aa5861@coventry.ac.uk, aa0699@coventry.ac.uk, verban@uni.coventry.ac.uk, csx240@coventry.ac.uk, ac0872@coventry.ac.uk, aa6910@coventry.ac.uk, anne.james-taylor@ntu.ac.uk, Zahid.Usman@Rolls-Royce.com

[a]*Faculty of Engineering, Environment and Computing,*

Coventry University, Coventry, United Kingdom

[b]*College of Science and Technology, School of Science & Technology*

Nottingham Trent University, Nottingham, United Kingdom

[c]*Rolls-Royce plc, United Kingdom*

[d]*Department of Genetics, Wroclaw University of Environmental and Life Sciences, Poland*

Abstract

Cloud and Fog computing has emerged as a promising paradigm for the Internet of things (IoT) and cyber-physical systems (CPS). One characteristic of CPS is the reciprocal feedback loops between physical processes and cyber elements (computation, software and networking), which implies that data stream analytics is one of the core components of CPS. The reasons for this are: (i) it extracts the insights and the knowledge from the data streams generated by various sensors and other monitoring components embedded in the physical systems; (ii) it supports informed decision making; (iii) it enables feedback from the physical processes to the cyber counterparts; (iv) it eventually facilitates the integration of cyber and physical systems. There have been many successful applications of data streams analytics, powered by machine learning techniques, to CPS systems. Thus, it is necessary to have a survey on the particularities of the application of machine learning techniques to the CPS domain. In particular, we explore how machine learning methods should be deployed and integrated in Cloud and Fog architectures for better fulfilment of the requirements of mission criticality and time criticality arising in CPS domains. To the best of our knowledge, this paper is the first to systematically study machine learning techniques for CPS data stream analytics from various perspectives, especially from a perspective that leads to the discussion and guidance of how the CPS machine learning methods should be deployed in a Cloud and Fog architecture.

Keywords: Cyber-physical systems (CPS), Machine learning, Cloud computing, Fog computing, Edge computing, Analytics

I. Introduction and Motivation

In this section we discuss definitions of cyber-physical systems (CPS) and highlight application areas.

1. Cyber-physical systems: Definitions and characteristics

Recent advances in computing, communication and sensing technologies have given rise to CPS, one of the most prominent ICT technologies that pervade various sectors of the physical world and also an integral part of everyday life [1][2][3][4]. The term cyber-physical systems (CPS) was coined in the US in 2006 [5], with the realisation of the increasing importance of the interactions between interconnected computing systems [6]. There have been various definitions of CPS, each of them throwing some light at some of the relevant factors.

- The National Science Foundation [7] defines CPS as "Cyber-physical systems (CPS) are engineered systems that are built from, and depend upon, the seamless integration of computational algorithms and physical components. Advances in CPS will enable capability, adaptability, scalability, resiliency, safety, security, and usability that will far exceed the simple embedded systems of today. CPS technology will transform the way people interact with engineered systems -- just as the Internet has transformed the way people interact with information. New smart CPS will drive innovation and competition in sectors such as agriculture, energy, transportation, building design and automation, healthcare, and manufacturing."
- Lee [1] defines CPS as "A cyber-physical system (CPS) is an orchestration of computers and physical systems. Embedded computers monitor and control physical processes, usually with feedback loops, where physical processes affect computations and vice versa."
- The National Institute of Standards and Technology [4] defines the subject of CPS as "Systems that integrate the cyber world with the physical world are often referred to as cyber-physical systems (CPS). The computational and physical components of such systems are tightly interconnected and coordinated to work effectively together, sometimes with humans in the loop"

Despite their differences in length, detail and the semantics of some terms, there are some common characteristics that can be extracted from these definitions. More specifically, we argue that CPS have the following inherent characteristics:

- Integration of cyber elements (computation, software and networking), engineered elements (physical processes) [1][7][8][9][10][11], and human factors [4]
- Reciprocal feedback loops between physical processes and computations, (simulation and decision making), sensing and actuation elements, and monitoring and control elements [4][1][8][9][12]
- Networked physical components and tightly coupled, interconnected processes that require cooperation and coordination. [2][4][13]

In addition to this, the National Institute of Standards and Technology also highlights the fact that CPS require the integration and cooperation of two technologies for successful deployment [4]. Firstly learning and predictive capabilities are necessary to provide the integration of physical and digital models and, more importantly, to provide the ability for the digital world to autonomously change its logic based on the state of the physical world (e.g., diagnostics and prognostics). Secondly, it is stated that CPS require open architectures and standards that provide for modularity and composability of systems, thus allowing complex and dynamic applications.

Particularly, CPS interconnect virtual and physical worlds. The physical system typically has a virtual twin which can be used for monitoring and control. The desired predictive capabilities in CPS require these systems to potentially collect and analyse data from the physical and digital world. In the end, the predictive capability informs decision makers to take appropriate actions or control to change the course of physical world.

Finally it should be highlighted that current applications of CPS include automotive systems, manufacturing, medical devices, military systems, assisted living, traffic control and safety, process control, power generation and distribution, energy conservation, HVAC (heating, ventilation and air conditioning), aircraft, instrumentation, water management systems, trains, physical security (access control and monitoring), asset management and distributed robotics (telepresence, telemedicine) [1][14][15].

2. Data Stream Analytics in CPS

Mining data streams, acquired from various sensors and other monitoring components embedded in the physical systems, plays an essential role in CPS. It extracts insights and knowledge, provides learning and predictive capabilities for decision support and autonomous behaviour, enables the feedback from the physical processes to the cyber counterparts, and eventually facilitates the integration of cyber and physical systems [16].

Silva et al. [17] provides a formal definition of a data stream as:

> A data stream *S* is a massive sequence of data objects X^1, X^2 …, X^N, i.e., $S = \{X^i\}_{i=1}^{N}$, which is potentially unbounded ($N \rightarrow \infty$). Each data object is described by an n-dimensional attribute vector $X^i = [x_j^i]_{j=1}^{n}$ belonging to an attribute space Ω that can be continuous, categorical, or mixed.

Data streams feature massive, potentially unbounded sequences of data objects that are continuously generated at rapid rates [17], which leads to the fundamental shift in the data analytics (information source) from traditional a priori information alone based or off-line batch approaches, to stream analytics. The key challenge in stream analytics is the extraction of valuable knowledge in real time from a massive, continuous and dynamic data stream in only a single scan [18]. The reader should additionally consider that the insights extracted from physical devices, such as sensors, feature perishable insights, i.e., they have to be provided quickly, as otherwise they lose value to feed the logic of the CPS software. In CPS, data streams are most beneficial at the time they are produced, as any change reported by the data (e.g. a sensor anomaly, a fault in the physical process being sensed, or a change of system state) should be detected as soon as possible and acted upon. Furthermore, as opposed to stream analytics for purely software systems, the insights being revealed by data streams in CPS are often tied to a safety-critical action that must be performed to ensure the health of the CPS itself [16].

Analysis of these ever-growing data streams becomes a challenging task with traditional analytical tools. Innovative and effective analytic techniques and technologies are required to operate, continuously and in real-time, on the data streams and other sources data [19]. Machine learning is a discipline that aims to enable computers to, without being explicitly programmed, automate data-driven model building and hidden insights discovery, i.e., to automate behaviour or the logic for the resolution of a particular problem, via iterative learning from example data or past experience [20][21][22]. In the past, there have existed many successful applications of machine learning, including systems that analyse past sales data to predict customer behaviour, optimize robot behaviour so that a task can be completed using

minimum resources, and extract knowledge from bioinformatics data[22]. In this particular survey, we will focus on stream analytics methods that are based on machine learning algorithms.

3. Cloud and Fog Computing

The interconnection of sensor and actuator systems with decision making and analytics have traditionally been performed by either local static controllers or uploaded to the Cloud for analysis. Supported by the paradigms of Internet of Things (IoT), Cloud computing experts propose the virtualization of devices to provide their data-based capabilities and their connection as a service for users within a Sensing and Actuation as a Service (SAaaS) [23] or as Things as a Service (TaaS) [24]. Another role that Cloud computing has played in supporting CPS is focused on the analysis of the data received from devices. The Cloud can provide a vast number of processing and storage resources which can be used to analyse large amounts of data [25] or streams [26]. These Cloud capabilities are concentrated in centralized and remote data centres, which has several drawbacks. The security aspect of storing, analysing and managing data in the Cloud is an increasing concern [27], while the remote nature of the Cloud also has reliability and latency issues [28].

The paradigm of Fog computing as proposed by [29] extends the Cloud to the edge of the network to better utilize resources available on gateways and connected devices. This extension allows data to be stored and processed locally to increase reliability and security, while decreasing the latencies between devices and the processing elements [30]. Fog computing systems are typically characterized by a large number of heterogeneous nodes, increased mobility and a strong presence of streaming and real-time applications [29]. The hosts or gateways used in Fog systems vary from PC based computing nodes [31], mobile devices [32] and resource constrained System on Chip (SoC) devices [33], routers, switches, set top boxes, proxy servers and base stations [34]. These hosts all have varying storage, processing and networking capabilities. While computing nodes have the most resources and are most reliable, they usually communicate with devices using Ethernet or Wi-Fi based networks. The mobile devices and SoC based devices have fewer resources but provide a wider range of wireless communication possibilities for polyglot gateways [35], that can be used to connect to a wider range of heterogeneous devices using low-power Machine to Machine (M2M) communication protocols. These distinguishing properties of the Fog are essential for providing elastic resources and services to end users at the edge of networks [30]. Fog computing is rapidly finding its way into CPS and IoT.

Adopting IoT paradigms into CPS can provide several types of services, such as weather monitoring, smart grid, sensor and actuator networks in manufacturing environments, smart building control and intelligent transport. These services produce a large amount of data that need to be processed for the extraction of knowledge and system control [36].

The platforms deployed in Fog computing vary based on hosts and application domains, but they can be categorized in a similar way as in Cloud computing. Infrastructure based platforms allow users to deploy Virtual Machines (VM's) [37] or lightweight virtualization images [38]. Platform based solutions as in [39] provide a platform for users for application-style system deployments. The third type of platform provides networking and analytics capabilities that the user can configure and use without the need to program and deploy their own applications.

From the hosts' perspective there are a number of differences between the Cloud and the Fog. The main difference is the resources. While the Cloud is considered to have virtually unlimited storage and processing capabilities, in the Fog such resources are much more restricted

so their optimal management is crucial. Inter-host communication in the Cloud is fast due to high speed networks. In the Fog, due to wireless communication and varying network types, delays can occur and can vary largely in size between hosts. When we consider device to host communication, the Cloud adds significant networking delays when accessing remote devices. From a platform perspective we can see that Cloud solutions offer full control of resources using VM's style solutions or other Platform as a Service (PaaS) options, while Fog solutions tend to share more interdependent and constrained resources between users. Cloud computing has a well-established business model as compared to the relatively new concept of Fog computing. However, this fact has been recognised by researchers and efforts can be seen in literature resolving billing, accounting, monitoring and pricing for a Fog business model [40].

CPS requires large computational capabilities to process, analyse, and simulate the collected data from sensors to make decisions and to instruct controllers, in a limited timeframe. The volume and velocity of sensor and visualization data in CPS requires a large amount of storage and software applications to process them. The division of the labour of latency tolerant and deep analytics tasks between Fog and Cloud depends upon processing power of the edge nodes and application domain. The edge nodes with limited computational power may only focus on performance of latency sensitive tasks. On the other hand, machine learning algorithms that require intensive computing resources should be executed in the Cloud. The Cloud service model with elastic and flexible architecture presents an appropriate solution to support the emerging CPS. However, the study on how data and applications should be distributed between edge devices and the cloud has derived little attention from the academic and industry research communities. This obviously includes the decision on where machine learning methods for stream analytics should be executed: the edge or the cloud. The existing machine learning methods with different processing properties have their own strengths and weaknesses, so several methods or their variants have been proposed to address diverse requirements from different applications. Some methods, for example, may cope better than others with incomplete data sets or large data sets, while some may require more computational power than others.

Given the emerging and promising Cloud and Fog computing architecture and the foreseeable integration of CPS, more specifically the machine learning based data analytics in CPS, it is necessary to investigate what machine learning techniques have been employed in the context of CPS. Further we should consider how they should be adapted and deployed in the Cloud-Fog-Edge architecture for better fulfilment of the requirements of the application, such as mission criticality and time criticality. This research aims to identify and analyse the properties of current well-known machine learning methods employed in the context of CPS and the characteristics of stream data in CPS to provide a comprehensive analysis on their relation. This will help determine how to map data and machine learning methods to the Cloud and Edge computing to meet CPS requirements. More specifically, we will focus on the analysis of the machine learning models employed in stream analytics from the perspective of time complexity. This measure will provide important indications to the appropriateness of Edge computing to host tasks, as it has limited computational powers, RAM and storage whereas the Cloud has more flexibilities, capacities and capabilities to deal with resource-intensive tasks on demand. The required qualities for the outputs and the types of results (e.g. precision and accuracy rates) have significant influence on the resources and response time of the selected methods, so the

correlation between them should be investigated.

To the best of our knowledge, this paper is the first to systematically study the machine learning based data stream analysis in CPS and its deployment in the emerging Cloud-Fog-Edge architecture.

The remainder of the paper is organized as follows. We present related work in section 2. In section 3, the machine learning methods are reviewed from the perspective of the functions they provide for typical CPS applications. Then, the time complexities of general machine learning techniques are discussed in section 4, along with how machine learning methods should be deployed for effective and efficient integration within Cloud and Fog computing architecture. We conclude the paper with some future research directions.

II. Related work

Traditional CPS may have limited computation and storage capabilities due to the tiny size of the devices embedded into the systems. Chaâri et al. [2] investigated the integration of CPS into Cloud computing, and presented an overview of research efforts in three areas: (1) remote brain, (2) big data manipulation, and (3) virtualization. More specifically, real-time processing, enabled by offloading computation and big data processing to the Cloud, was explored. Nevertheless, Chaâri et al. [2] did not include an exhaustive analysis of the emerging Fog and Edge computing technologies, and how these technologies should co-operate with CPS.

The authors in [18] and [17] presented a survey on data stream analytics from the perspective of clustering algorithms. Apart of summarizing the unique characteristics of data stream processing by comparison with traditional data processing, in [18], data stream clustering algorithms were categorized into five methods (i.e., hierarchical methods, partitioning methods, grid-based methods, density-based methods, and model-based methods). Similarly, [17] analysed 13 most relevant clustering algorithms employed in the context of data stream analytics. In addition to the categories listed in [17], the authors in [18] identified three commonly-studied window models in data streams, i.e., sliding windows, damped windows, and landmark windows. Differently to [17] and [18], we do not solely focus on clustering algorithms, but we also extend analytics to other types of machine learning algorithms.

In [22], the authors studied machine learning techniques employed in transportation systems, and identified various conventional machine learning methods such as regression (linear regression, polynomial regression and multivariate regression), decision tree, artificial neural networks (ANNs), support vector machines (SVMs) and clustering. Despite the useful insights provided by the work, the analysis is exclusively carried out in the light of a very particular type of CPS application; and further, no advanced machine learning methods, e.g. deep learning methods, was introduced.

The survey provided in [41] recognized the changes that were needed to move from a conventional technology-driven transport system into a more powerful multi-functional data-driven intelligent transportation system (D2ITS), i.e. a system that employed machine learning and other intelligent methods to optimize its performance to provide a more privacy-aware and people-centric intelligent system. The paper identified both the data sources that drove intelligent transport systems (ITS), (e.g. GPS, laser radar, seismic sensor, ultrasonic sensor, meteorological sensor, etc.), and the learning mechanisms for real-time traffic control and transportation system analysis. Examples of the learning mechanisms were online learning (e.g., state-space neural network, real-time Kalman filter, combination of online nearest neighbour and fuzzy inference, hidden Markov model), adaptive dynamic programming (ADP),

reinforcement learning (RL) and ITS-oriented Learning. The article offers a thorough and sound view on transport systems, but the insights are not extrapolated to other CPS domains and applications.

The authors in [42] presented an analysis on a number of existing data mining and predictive machine learning methods for big data analytics with the goal of optimising the dynamic electrical market and consumers' expectations in the smart grid. Similarly, authors in [43] review the benefits and gaps of the combination of artificial neural networks, genetic algorithms, support vector machines and fuzzy logic for the forecasting of power grid. Another similar review is carried out in [44] to analyse the big data methods used to manage the smart grid. The authors identified different predictive tasks that can be carried out in the smart grid domain such as power generation management, power forecasting, load forecasting, operation and control fault diagnosis, and so forth. The authors mapped to the corresponding statistical or machine learning methods with the required data inputs or sources.

III. Machine Learning Methods in CPS Applications

In this section we consider some typical CPS applications

1. Typical CPS Applications

Smart Grid:

Smart grid is a complex system ranging from micro grid to national or international networks involving different levels of facility, management and technology. A smart grid is considered as a cyber physical system as it monitors and manages the power generation, loading, and consumptions through a number of sensors. These sensors gather the stream data that is fed to analytic methods and control systems to balance and distribute power generation and consumption [45].

Due to complexity and dynamics of power market, and the volatile nature of renewable energy, it is important to have good forecasting and prediction on market trend and energy production to correctly estimate the amount of power to generate. In addition to this purpose, applications of analytics to the smart grid also include fault detection at infrastructure, device, system and application levels [10]. Machine learning is a promising tool to analyse the data stream and providing results to inform decisions and actions.

Intelligent Transportation Systems (ITS)

An intelligent transportation system (ITS) is an advanced application which aims to provide innovative services relating to transport and traffic management, and enable users to be better informed and make safer, more coordinated, and smarter use of transport networks. ITS brings significant improvement in transportation system performance, including reduced congestion and increased safety and traveller convenience [46][47][48].

ITS meets the core characteristics of CPS. Enabled by Information and Communication Technologies (ICT), elements within the transportation system, such as vehicles, roads, traffic lights and message signs, are becoming intelligent by embedding microchips and sensors in them. In return, this allows communications with other agents of the transportation network, and the application of advanced data analysis and recognition techniques- to the data acquired from embedded sensors. As a result, intelligent transportation systems empower actors in the transportation system to make better-informed decisions, e.g. whether it's choosing which route to take; when to travel; whether to mode-shift (take mass transit instead of driving); how to optimize traffic signals; where to build new roadways; or how to hold providers of transportation services accountable for results [41][48].

Smart Manufacturing/Industrial 4.0:

Manufacturing applications, such as object detection, force and torque sensor based assembly operations, require accuracy of object detection, pose estimation and assembly to within few micrometres. Moreover, this accuracy has to pass the test of time and repeatability (i.e., the results should be precise).

Manufacturing in general and automotive manufacturing in particular, requires operation involving handling, inspection or assembly to be completed in few seconds. For example, BMWs mini plant in Oxford has a car coming of production line every 68 seconds [49]. Applications, such as welding, require real time data processing, analysis and results. For example, to track the position of joining plates on real time basis and adjust the movement of weld guns on real time basis for precise and accurate welding at high speed [50].

2. Machine Learning in a Nutshell

Machine learning is the discipline that aims to make computers and software learn how to program itself and improve with experience/data, with the goal of solving particular problems [51]. Typically, a machine learning algorithm is a specific recipe that tells a computer/software how to improve itself from experience. A model is the result of training a machine learning algorithm with a set of data or experiences of a given problem, and it can be employed to solve future related problems.

Machine learning algorithms fall into one of the following categories: supervised learning, unsupervised learning, and reinforcement learning. Next, we briefly discuss each of these categories and describe some of the most relevant techniques for each category:

- In supervised learning, the aim is learning a mapping from an input to an expected output that is provided by a supervisor or oracle (i.e., labelled data) [20]. Depending on the type of output, we say that we either have a classification or a regression problem. In the first case, we aim to produce a discrete and finite number of possible outputs, while in the second case the range of possible outputs are infinite and numeric [20].
- In unsupervised learning, there is no such supervisor and only the input data is present. The aim of these algorithms is finding regularities in the input [20][22].
- Finally, reinforcement learning applies to the cases where the learner is a decision-making agent that takes actions in an environment and receives reward (or penalty) for its actions in trying to solve a problem. Thus, the learning process is guided by a series of feedback/reward cycles [22]. Here, the learning algorithm is not based on given examples of optimal outputs, in contrast to supervised learning, but instead it must discover them by a process of trial and error [52]

Next, we describe some of the most usual machine learning algorithms employed in the context of CPS data stream analytics.

Decision Trees and Random Forests:

A decision tree is a supervised machine learning algorithm that is organized in a tree-like hierarchical structure composed by decision nodes and leaves. Leaves represent expected outputs, and decision nodes branch the path to one of the expected outputs according to the value of a specific input attribute. Decision tree algorithms exist in the form of classification and regression algorithms [20]. One of the main advantages of decision trees is that the model is human readable and understandable.

A Random Forest is an ensemble of random trees constructed by means of bagging. By this process, a training dataset of N samples is divided into k different datasets of *N*' samples uniformly sampled with replacement from the original dataset, and consisting of a random selection of the input attributes. Then, each dataset is employed to train a different decision tree,

guided by the heuristic that the combination of the resulting models should be more robust to overfitting. Each tree provides an output that can be aggregated by a wide variety of rules [53][54].

Artificial Neural Networks (ANNs) and variants:

ANNs are machine learning algorithms that resemble the architecture of the nervous system, organized as interconnected networks of neurons organized in layers. These versatile algorithms are typically employed for supervised, unsupervised, and reinforcement learning. The inputs of the network (input layer) are transformed by weighted (non) linear combinations that generate values that can be further transformed in other layers of the network until they reach the output layer. Due to their ability to represent potentially complex relationship between the inputs and the expected output, ANNs, such as the multilayer perceptron (MLP), have gained popularity in machine learning and data analytics realm. The multilayer perceptron is a nonparametric estimator that can be used for both classification and regression.

Convolutional Neural Networks (CNNs) exploit translational invariance within their structures by extracting features through receptive fields and learning by weight sharing. CNNs usually include two parts. The first part is a feature extractor, which learns features from raw data automatically and is composed of multiple similar stages and layers. The second part is a trainable fully-connected MLP or other classifiers such as SVM, which performs classification based on the learned features from the previous part [55][56].

Recurrent Neural Networks (RNNs) are a family of neural networks that has gained popularity in the last few years [57], and they are of special relevance to stream analytics due to this characteristic. In addition to this, the surge of data and computing power present in the last decade have given rise to deep neural networks [58] that stack multiple non-linear layers of neurons to represent more complex relationships between inputs and outputs or more efficient representations of the inputs. For various closely related definitions of deep learning, please refer to [58].

Support Vector Machines (SVMs):

Support vector machines (SVMs) are supervised learning methods that classify data patterns by identifying a boundary or hyperplane with maximum margin between data points of each class/category [22][53]. The support vector machine is fundamentally a two-class classifier, although multiclass classifiers can be built up by combining multiple two-class SVMs. Despite the fact that they were initially devised for classification tasks, SVMs have been further extended to regression problems [59].

Bayesian networks and variants

Bayesian networks are probabilistic graphical models based on directed acyclic graphs where the nodes are random variables and the direct arcs indicate the direct influences, specified by the conditional probability, between two random variables [20][60].

Some popular machine learning algorithms such as Naïve Bayes, a popular supervised classifier, and Hidden Markov models (HMMs) can be considered as special cases of Bayesian networks. The second specializes at processing sequences of outputs by learning implicit states that generate outputs [20] [52]. This paradigm has been used for both supervised and unsupervised tasks.

Evolutionary computation:

Evolutionary Computing is the collective name for a range of problem-solving techniques based on the principles of biological evolution, such as natural selection and genetic inheritance. The fundamental metaphor of evolutionary computing relates this powerful natural evolution to a particular style of problem solving – that is a pseudo trial-and-error guided by the

value of a given fitness function that measures the goodness of the evolved individual/solution [61]. Evolutionary computing techniques mostly involve metaheuristic optimization algorithms, such as genetic algorithms and swarm intelligence. Genetic algorithms have been employed in supervised[62], unsupervised [63], and reinforcement learning problems[64].

Clustering:

Clustering is an unsupervised family of algorithms that involve processing data and partitioning the samples into subsets known as clusters. The aim of this process is to classify similar objects into the same cluster while keeping dissimilar objects in different clusters [18]. The separation criteria may include (among others) maximization of similarities inside clusters, minimization of similarities between different clusters, and minimization of the distance between cluster elements and cluster centres. One of the most popular clustering algorithms is called **k-means clustering** where k denotes the number of clusters.

Self-organizing map (SOM):

SOM is an automatic data-analysis method widely applied to clustering problems. SOM represents a distribution of input data items using a finite set of models. These models are automatically associated with the nodes of a regular grid in an orderly fashion such that more similar models become automatically associated with nodes that are adjacent in the grid, whereas less similar models are situated farther away from each other in the grid. This organization, a kind of similarity diagram of the models, makes it possible to obtain an insight into the topographic relationships of data, especially of high-dimensional data items [65].

Q-learning:

Q-learning is a kind of reinforcement learning technique that is a simple way for agents to learn how to act optimally in controlled Markovian domains. It amounts to an incremental method for dynamic programming which imposes limited computational demands. It works by successively improving its evaluations of the quality of particular actions at particular states [66].

3. Machine Learning Methods in CPS

Table 1: Overview of machine learning methods in the context of CPS

Machine Learning Method	Domain	Functional Category	Task	Reference
ANN	Smart Grid	Forecasting/Prediction/Regression	Electrical Power prediction, load forecasting	[67][68][69][70][71][43]
	Transport	Pattern Recognition/ Clustering	Behaviour/Event Recognition	[53]
		Forecasting/Prediction/Regression	traffic flow features	[72]
			road-side CO and NO2 concentrations estimation	[73]
			travel time prediction	[74][75][76]
		Classification	obstacle detection and recognition	[77]
			Image Processing	[78]
	Manufacturing	Forecasting/Prediction/Regression/optimization	Energy Consumption/ Process parameters optimisation	[79] [80]
Random Forest	Smart Grid	Forecasting/Prediction/Regression	demand side load forecasting/Price forecasting	[67][81]
		Anomaly/Fault Detection	Power record faults	[82]
	Transport	Pattern Recognition/Clustering	Behaviour/Event Recognition	[53]
	Manufacturing	Anomaly/Fault Detection	Tooling wear/ Errors detection	[83] [84] [85]
SVM	Smart Grid	Forecasting/Prediction/Regression	Price Prediction	[86][87]
			Electrical Power prediction,	[88][69][71][89]

		Anomaly/Fault Detection	Non-Technical Loss detection	[71][90][91]
			Blackout Warning	[88] [92]
			Power Line Attacks	[92]
	Transport	Classification	Unintentional vehicle lane departure prediction	[93]
			Obstacles classification	[94][77]
		Pattern Recognition/ Clustering	Behaviour/Event Recognition	[53][95]
		Anomaly/Fault Detection	Mechanism Failure	[96]
		Forecasting/Prediction/Regression	Travel time prediction	[97][76]
	Manufacturing	Forecasting/Prediction/Regression	Machine Maintenance	[98]
		Design / Configuration	Feature Design; Production Processing	[99][100]
		Anomaly/Fault Detection	Quality Control	[101][102]
	Smart Home	Pattern Recognition/ Clustering	Activity recognition	[103][104]
Decision tree	Smart Grid	Anomaly/Fault Detection	fault detection predict an energy demand	[105] [106]
		Forecasting/Prediction/Regression		[106]
	Transport	Forecasting/Prediction/Regression	To predict the traffic congestion level and pollution level; bus travel time	[107] [108]
				[108]
		Anomaly/Fault Detection	Cyber Attacks / detect stereotypical motor movements	[109]
	Manufacturing	Classification/Diagnosis	Quality Control/Fault diagnosis	[110][111]
Bayesian Network	Transport	Classification	Event and behaviour detect	[53]
	Smart Grid	Anomaly/Fault Detection	Non-technical losses and fault detection	[105]
	Manufacturing	Anomaly/Fault Detection	Fault detection in the production line	[112]
		Forecasting/Prediction/Regression	Tool wear prediction/Energy consumption prediction	[113][114]
Self-Organising Map	Transport	Clustering	Obstacle detection and recognition	[77]
Evolutionary Computing	Smart Grid	Optimisation/ Forecasting/Prediction	Short Term load forecasting	[115]
Swarm Computing	Smart Grid	Optimisation	economic load dispatch/feature Selection	[116][117]
	Manufacturing	Anomaly/Fault Detection/Process optimisation	Fault detection, classification and location for long transmission lines/Process optimization Automatic fault diagnosis of bearings	[118][119][120]
HMM	Smart Grid	Optimisation	Optimal decisions on smart home usage	[121]
	Manufacturing	Anomaly/Fault Detection	Automatic fault diagnosis of bearings	[119][122]
Reinforcement learning /Q-learning-based ADP algorithm	Smart Grid	Optimisation	Aided Optimal Customer Decisions for an Interactive Smart Grid	[121]
	Transport	Optimisation	the road latent cost	[123]
Deep Learning/ Autoencoder model/ convolutional neural network (CNN)/ Recurrent Neural Networks (RNNs)	Smart Grid	Forecasting/Prediction/classification/ Regression	Building Energy consumption	[124]
	Transport		Traffic flow prediction; processing roads images / commanding Steering; detecting train door anomaly and predicting breakdowns Anomaly-based detection of malicious activity	[125][126][127] [128] [96]
	Other		To classify various human activities; To detect congestive heart failure	[56]
	Other			

Table 1 shows an overview of machine learning methods where they have been used in the loose context of CPS. They have been used to carry out tasks in three different application domains: smart grid, transport and manufacturing.

ANN is one of the most popular methods having been used in the various application domains, as it is capable of performing long term forecasting by regressing the time series stream to predict trends. For example, in smart grid and manufacturing, ANN can efficiently predict energy consumption of consumers which is beneficial for demand side management. Only few researchers use ANN for real-time or short-term predication [86][70], as it requires considerable time to process and tune the parameters before it can be deployed. Most applications require large amount of input data and training time to produce meaningful models with certain degree of accuracy and confidence [67][43][72]. ANN can work alone and produce acceptable results, but it often works with other learning methods such as SVM, GA, Bayesian etc. to improve training efficiency or modelling accuracy [43]. In the table, the term ANN was broadly used, but it has a lot of variants with various activation functions and structures forming a hybrid model to meet the purposes such as forecasting, classification, clustering, and regression for various applications. [43] has carried out detailed analysis of these variations and hybrid approaches. Here, we classify applications into this category using ANN as the main body for their solutions.

SVM has been widely adopted to address the issues in product feature design, fault detection, forecasting, clustering and pattern recognition across the application domains such as manufacturing, smart grid, transportation as well as smart home due to its maturity and transparency. The method can take different sizes of input data to carry out classification and regression, so it has been used in applications that require a short response time such as described in [87][88]. It also used in conjunction with other machine learning methods such as ANN, and Bayesian by exploiting its characteristics to provide complimentary functions to address complex problems [70][98][99]. The authors in [99] used a trained SVM classifier from the classified design examples such as features and components, which are obtained from a hierarchical clustering, to recommend different additive manufacturing design features. In the case study, it only shows 21 design features from hundreds that were used to train and to build the model.

The faults in products or tools in manufacturing can lead to a big loss of time and a serious consequence if they are not detected and resolved earlier. Authors in [83]and [84] reported the use of the Random Forest to analyse the big data for tooling condition monitoring in milling production and silicon in semiconductor manufacturing. It also has been used in predicting the short term electricity price from the historical data [81] and detecting false electricity records from the sensors [82]. [53] reported the use of Random Forest to model a driver profile effectively. These applications required a reasonably large amount of historic data for the training to achieve accurate and time was not considered to be a crucial factor.

Decision Tree is a well-known method for classification, so it is predicable that the researchers have used it to detect the faults in power system and motor movement and for quality management in production. It also has been used to predict energy demand, bus travelling time, and to determine the correlation between traffic congestion and air pollution.

The accuracy of fault detection, quality prediction, classification and rare events forecasting are associated with probabilities, as all the input factors cannot be certain due to the dynamic environments and complex human

behaviour and interactions. Bayesian Network is a well-studied method to model complex probability networks as it has been used in different applications to explain the possible occurrences of outputs with input variables. It does not require a large amount of input data to form the network, if the probabilities of variables are known. The network can be large and complex, but its processing time is linear. [53][105][112] showed the consistent characteristics in these applications.

Table 1 also shows where the machine learning methods have been used across four application domains and the tasks that have been carried out to gain the benefits of analyzing and interpreting large volumes of data streams generated. The most common area for researchers and industry practitioners adopting the methods is to increase accuracy of predication and forecasting in their CPS applications. The authors in [43][67][68][69][70][71][81][106] reported adoption of machine learning to predict electrical power consumption, demand, supply and load in order to improve demand response management in Smart Grid. Machine learning is a well employed tool to predict traffic flow, air population emitted by cars, traffic congestion and travel time by transport [72][73] [74][75][76][107][108][96][125][126][127][128]. Machine learning also has been extensively applied in manufacturing by predicting energy consumption in production line, machine maintenance, and tool wearing [79] [80] [104] [113][114]. Diagnosis and fault detection is another function for which machine learning has been widely used in the manufacturing domain. Fault detection applications include root cause of power faults in production, tooling wearing and mechanic faults, cause of the fault in components or products, and quality control [53][101][102][112][119][122]. Smart Grid also has several machine learning applications for anomaly and fault detection such as non-technical loss detection, blackout warning, power line and cyber attacks, faults in demand management and power line faults [71][90][91][88][92][82][105][106] .

The utilization of machine learning for mechanical fault diagnosis and prevention of cyber attacks in transport systems can be explored further, as only two [96][109] reported the benefits of machine learning in this area. Machine learning is also a popular solution to configure plant/production, optimize electrical load/dispatch, reduce road latent cost, and forecast short term in electricity usage [77][99][100][115][121][123]. Machine learning has been exploited in other applications such as clustering road obstacles, classifying driving behaviours and traffic incidents and improving production quality [53][77][93][110][111].

From Table 1, it can be seen that functions of MLs have brought various benefits to different application areas and they have generated different levels of impacts in various areas, but the potential of machine learning is not fully realized yet, as the field is still evolving and the prevailing complexity may currently hinder take-up.

IV. Temporal Complexity Analysis

Machine learning algorithms are able to learn from selected samples to derive rules, trends, patterns or properties of a true population. The concept or hypothesis space, however, can be large and complex such that it cannot be learned or modelled in polynomial time. In these cases, learning to achieve highly accurate results by exhaustively exploring parameter values may not be possible practically but approximation may be achieved. As it is natural, the goal of all machine learning applications is to minimise the differences between the target concept and the output produced by the trained models. The representation, quality and quantity of the selected samples, which are input parameters to the learning algorithms, are important

attributes to increase the possibility of the successful learning. The probability of reaching successful learning by increasing accuracy of approximating to the target concept also depends on the complexity of learning and time. Learning is a trade-off between time and accuracy. In principle, the higher accuracy, the more time is required for training. Information and computation are two main dimensions to measure the complexity of learning algorithms. The sample complexity is concerned with the number, distribution and sufficiency of training samples. The computational complexity of a solution method is the size of computational resources required to derive the concepts from the training data. This can be further classified into time and space complexity. Space complexity denotes the memory required for the computational model being selected to solve the problem. The time complexity is measured by the number of computational executions required to reach or approximate to the target concept. In this paper, we intend to show asymptotic time complexity rather than the actual runtime of the algorithms which will be various depending on its operating computational environment including hardware and software.

Table 2 shows a list of machine learning methods used by the applications illustrated in Table 1 and their corresponding time complexities, represented by big O, and the factors contributing to the complexities. Since there are many different variants to each machine learning method, it is not feasible to list them exhaustively, but some examples to illustrate measurement of complexity are given. For example, varieties of Bayesian Network models derived from various approximate and exact inference algorithms to infer unobserved variables, can lead to different computational complexities. Several hybrid learning methods have been proposed to resolve or improve the insufficiency of one individual method. This complicates the measurement of the execution due to the interdependency, as one method may reduce the complexity for the other in the model, but the overall complexity calculation still needs to consider all the methods involved. More algorithms and their time complexity can be found in [129].

Table 2: Time complexity of some of the most common machine learning algorithms

Machine learning method	**Asymptotic Time complexity**	**Factors**
Decision Tree Learning[130]	$O(M \cdot N^2)$	M: size of the training samples N: number of attributes
Hidden Markov model Forward-backward pass [54]	$O(N^2 \cdot M)$	N: number of states M: number of observations
Multilayer Perceptrons [129]	$O(n \cdot M \cdot P \cdot N \cdot e)$	n: input variables M: number hidden neurons P: number outputs N: number of observations e: Number of epochs
Deep Learning (Convolutional Neural Networks) [131]	$O(D \cdot N \cdot L \cdot S^2 \cdot M^2 \cdot e)$	L: number of input variables N:number of filters (width) S: spatial size (length) of the filter

		M:size of the output. D:number of convolutional layers (depth) e: number of epochs
Support vector machine [132]	$O(N^3)$ or $O(N^2)$	N: vectors C: upper bound of samples N^2 when C is small; N^3 when C is big
Genetic algorithms [129]	O(P·logP·I·C)	C: number of genes/chromosome P: population size I: Number of iterations
Radom forest [54][133]	(K·N· log N)	N:number of samples K:input variables
Self-organizing Map [134]	O(N·C)	N: input vector size C: cycle size
Reinforcement learning [135]	$O(N^3)$	N:number of steps to reach the goal
Particle swarm optimization (PSO) [136]	O(P+Gen·P·D)	P: number of particles D: number of dimensions Gen: number of generations
Bayesian Network (exact learning models of bounded tree-width)[137]	$O(3^N \cdot N^{(w+1)})$	N:size of nodes W: width of tree.

For example, [121] used Q-learning algorithms to model the interaction with users in smart homes There were a maximum of 20 steps to interact with users before appropriate recommendation could be made. Its asymptotic time complexity is up to 20^3 and the authors have concluded that Q-learning algorithm outperformed greedy or random decision strategies [121] in their simulated cases. Figure 1 shows the complexity level in big O when the number of steps decreases in the simulation. The authors did not report the actual runtime, so the asymptotic complexity cannot be correlated with the experimental one.

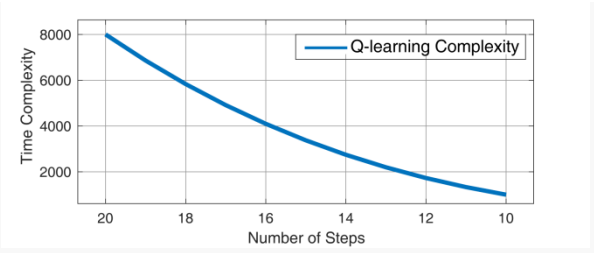

Figure 1: Complexity level and number of steps in Q-learning

[68] used three machine learning methods, SVM, LS-SVM and BPNN, for energy usage forecasting over 283 households with 500 point data (hours) for each. The total number of data points for training in the experiments was 141,500 (283*500). In their empirical study, the computational times of these methods are 335.39, 26.22, and 29.28 seconds respectively over a laptop to produce reasonably accurate results. The authors recommend running these approaches in Cloud and distributed computing to improve performance. SVM has better accuracy in reducing errors, but it took more time than others due to the overhead of using GA to find key parameters for SVM. The BPNN

has more errors than the other two and it requires more runtime than LS-SVM. The authors, however, did not include key parameter values such as generations and input points for GA and BPNN, so cannot derive their time complexity in relation to actual runtime. The time complexity of LS-SVM is $O(141500^2)$. Figure 2 shows the time complexity of LM-SVM by applying the data from [68] with simulation output and the actual runtimes in seconds.

It shows actual runtimes against the complexity level and the correlation between them without carrying out the actual experiments, the researchers can estimate its actual runtime by giving the number of samples when the underlying machine or environment has the same characteristics.

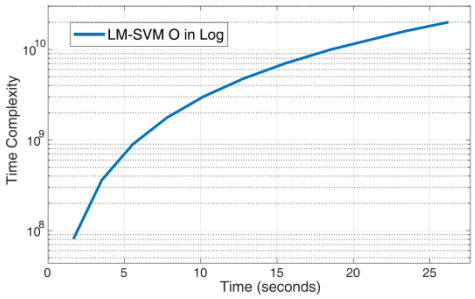

Figure 2: Time complexity of LM-SVM

The authors in [138] report the applications of the Particle Swarm Optimization (PSO) method to balance different loads by considering price to dispatch them. Their test case one includes 6 factors (dimensions), 6 generators (particles) and 100 generations to evolve, and its time complexity in theory is 3606 (6+6*100*6) before it has a satisfactory convergent result. In their test case two, it increases to 7 factors, 40 generators and 400 generations, so 40+40*400*7 (so the asymptotic time complexity is 112,040). In another test case it has 5 factors, 20 generators, and 400 generations (asymptotic time complexity is 40,020) and its actual computational runtime is 0.29282 second that is around 10 and 200 times slower than the other approaches [138] in the simulation. Figure 3 shows the relationship between complexity and actual runtime by extending the figures given in the paper. The line is the time complexity in log with respect to actual runtime. The researchers can refer this to approximate the actual runtime of an application with the same computational resources by giving key parameter values of the learning method. The approximation is not rigid, as we assume that the space complexity is changing linearly.

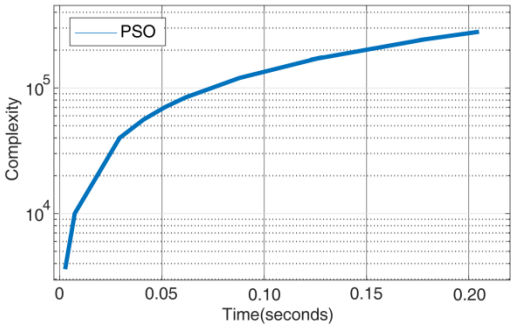

Figure 3: relationship between complexity and actual runtime of particle swarm optimization method

For deep NN learning methods such as CNN, the weights in the convolutional layers are trained and updated in a similar way as traditional ANNs/MLPs except that the number of filters and layers are orders of magnitude higher than those in traditional ANNs and MLPs. The authors in [131] report their experimental results on computational time complexity of a CNN model by varying different key parameters such as depth, filter size and number, width and pooling layer of the network to find their trade-offs between two parameters to investigate the overall performance in terms of time complexity and output accuracy. We share the same view with the authors [131] that introducing computational time and memory constraints can give better understanding the value of machine learning methods in realistic business applications.

The training of these deep NN models needs massive resources (e.g. to accommodate the training data) and time. Therefore they should be carried out on the Cloud. However, the operation time of these models is only proportional to the number of neurons no matter how large the training data is, the online analysis tasks can be deployed on the Edge/Fog.

As it has been observed in this analysis, few research outputs report the empirical time complexity of their approaches. Therefore, the estimation on the empirical time complexity of a training algorithm still has room for more extensive study. This information may be vital for decision making on-the-fly if a learning task can be deployed in the Edge devices.

V. On-line Learning Methods

If we take a look at Table 2 we will observe that the asymptotic complexity of the classic learning algorithms reported in the literature review normally takes into consideration many terms (e.g., number of samples, iterations, structure parameters, etc.). In theory, this could result in high order polynomial behaviour, which would deter the deployment of the learning phase in Edge devices. This is because firstly, over time more and more streaming data will be accumulated and it is impractical and often infeasible to accommodate large volumes of streaming data in the machine's main memory. Secondly, it is also infeasible to regularly reconstruct new models from the scratch with accumulated streaming data in real-time. Furthermore CPS data streams feature *perishable insights*, i.e., information that must be acted upon fast, as insights obtained from streaming data, such as from sensors, quickly lose their value if they were to be processed in 'batch mode' [18]. As a result, a new paradigm of learning, i.e. incremental and on-line learning algorithms should be adopted. Losing et al. [139] gives the definition of incremental learning for supervised learning as below (we change the notations/symbols for consistency reasons).

> An incremental learning algorithm generates, on a given stream of training data $S^1, S^2 \ldots, S^N$, a sequence of models H^1, H^2 …, H^N, where S^i is labeled training data $S^i = (X^i, Y^i) \in R^n \times \{1, \ldots, C\}$ and $H^i : R^n \{1, \ldots, C\}$ is a model function solely depending on H^{i-1} and the recent p examples $S^i, \ldots, S^{i-p}$, with p being strictly limited.

Losing et al. [139] further specify on-line learning algorithms as incremental learning algorithms which are additionally bounded in model complexity and run-time, capable of endless/lifelong learning on a device with restricted resources.

Incremental and online learning algorithms aim for minimal processing time and space; and thus fit in CPS data processing environments.

Losing et al. [139] evaluate eight popular incremental methods representing different algorithm classes such as Bayesian, linear, and instance-based models as well as tree-ensembles and neural networks. Experiments are carried out to evaluate these algorithms with respect to accuracy, convergence speed as well as model complexity, aiming at facilitating the choice of the best method for a given application. However, it primarily covers supervised incremental learning algorithms with stationary datasets, although robustness of the methods to different types of real concept drift are also investigated.

Gama et al. [140] considers dynamically changing and non-stationary environments where the data distribution can change over time yielding the phenomenon of concept drift, which applies to most of the real world CPS applications. Adaptive learning algorithms, defined as advanced incremental learning algorithms that are able to update predictive models online during their operation to react to concept drifts, are explored. A taxonomy for adaptive algorithms, presented in four modules as memory, change detection, learning, and loss estimation, is proposed; and the methods within each module are also listed. Gama et al. [140] focuses on online supervised learning.

Ade et al. [141] includes some unsupervised incremental learning approaches that learn from unlabelled data samples to adjust pre-learned concepts to environmental changes. Most of the

incremental clustering algorithms for pattern discovery rely on similarity measures between the data points. One approach is called Concept Follower (CF) that includes CF1 and CF2 [142]. CF1 and CF2 learn from unlabelled data samples to adjust pre-learned *concepts* to environmental changes. Initially, a supervised learner is used to learn and label a set of *concepts*. When a new sample is collected, CF1 calculates the distance of the sample to all *concepts* and the *concept* with the minimal distance to the sample is identified. If the distance is smaller than the predefined threshold, CF1 considers the *concept* a match and then slightly shifts, by a learning rate parameter, towards the classified sample to adjust to the concept drift; otherwise CF1 detects the abrupt change and repeats the initial supervised learning stage. Compared to CF1, CF2 supports problems areas with unbalanced sample ratio between *concepts*. This is done by CF2 adjusting all *concepts* in the proximity of the sample instead of, as does CF1, adjusting only the *concept* closest to the sample.

Next, we discuss on some of the most relevant online approaches to the machine learning algorithms identified in this article.

Artificial Neural Networks

Classically, ANN are trained using a training set and optimization methods such as gradient descent and backpropagation to minimize a cost function correlated to the error derived from the current state of the network.

The online version can adapt to the arrival of new data by pre-training the network with the available training set, and then adapting the pre-trained network by using stochastic gradient descent over the new series of available data. This type of setting would benefit from a combination of both Cloud technologies (i.e., for pre-training the network), and Edge computing (i.e. for adapting the network).

While the use of stochastic gradient descent allows adopting a batch algorithm like backpropagation in a non-batch setting, there are specialized learning algorithms, called on-line sequential learning methods, for training neural networks in an on-line setting in which data becomes available with time [143][144][145][146]. They can be efficiently deployed in an Edge device as they do not need to store past training samples. The online sequential learning methods tend to be ad-hoc for networks with specific activation functions, or with specific architectures (e.g., single hidden layer). Therefore, the complexity of problems represented by these networks may not be as vast as the one represented by classic neural networks or deep learning approaches.

Decision trees

Generating a classic decision tree requires that all of the training samples are considered [147]. This is hardly applicable in a stream analytics context, as training samples arrive constantly. Therefore, different learning mechanisms are required to properly train decision trees in a stream analytics context, in which the trees can evolve from a stream of data. Some approaches with a default tree structure provide a series of greedy steps to adapt to the new training samples. These include ID5R algorithm [148], an adaptation of the popular ID3 learning algorithm for stream data, and ITI [149]. These greedy changes were in some cases suboptimal and ended up in inappropriate adaptations to change.

The other approach to learning decision trees from streams is to maintain a set of statistics at nodes and only split a node when sufficient and statistically significant information is available to make the split. Hoeffding inequality [150][151][152] is the backbone to these approaches, which provide bounds for the number of observations that are necessary to obtain an estimated mean that does not differ from the mean of the underlying random variable. Some researchers have recently argued that the assumptions underlying the Hoeffding inequality are not appropriate when constructing

on-line trees. Some methods split the nodes of the decision tree based on other modelling paradigms such as McMiarmid's bound [153], or Gaussian processes [154].

Random forests

The general idea behind on-line random forests consists of providing both a method to carry out on-line bagging, and a method to carry out on-line learning of random trees. Abdulsalam et al. [155] take an approach that carries out on-line bagging by dividing the incoming samples of data into blocks with a certain size. Then, blocks of data randomly selected are employed for either training or testing a tree in the model. The training block is redirected to a chosen tree, and an on-line learning algorithm for trees is employed to update the current tree. Later on, the learning model is enhanced to adapt to the random arrival of labelled examples in the stream, with blocks of different sizes and frequency [156].

Another alternative to the on-line bagging process described above is employed by Saffari et al. [157]. In this case, each new sample is presented a number of times controlled by a Poisson distribution, to each random tree in the model. Then, the random trees gradually grow by creating random tests and thresholds at decision nodes and choosing the best one after a number of statistics have been gathered that guarantee that the test is the best from the ones randomly created at the decision node.

Other approaches opt for avoiding on-line bagging at the forest level, and the sub-sampling is carried out at the tree level [158]. When a new sample arrives to the random forest, this sample is presented to all of the trees. Then, the individual tree decides if the sample will be used to influence the structure of the tree, or used to estimate class membership probabilities in the leaf to which they are assigned.

Support vector machines

Classification in support vector machines is based on the idea of finding the maximum margin hyperplane that separates elements from different categories. By definition, one should have access to the entire training dataset in order to build such maximum margin hyperplanes. Otherwise, there would be no guarantee that estimated hyperplanes are optimal. This assumption limits the applicability of classic support vector learning algorithms to an online setting, and it forces scholars to devise new methods that are adapted to the online setting.

The incremental approach to support vector learning typically requires deciding if a new sample should become a support vector that modifies the current hyperplane. The algorithm also needs to determine if previously calculated support vectors are still as relevant after the observation of the new sample, and remove those that are no longer relevant. Otherwise, online approaches to support vector learning incur in the risk of growing linearly with the infinite number of samples [159]. To tackle this problem, there have been a number of proposals that aim to build a support vector model with adequate predictive performance while also minimizing the number of support vectors in the resulting model [159][160][161][162].

VI. Discussion

So far machine learning methods of various categories have been employed for data stream analysis purposes. Little literature has studied the integration of these methods to the Cloud and Fog computing architecture.

The very nature of CPS requires a computing paradigm that offers latency sensitive monitoring, intelligent control and data analytics for intelligent decision making. In contrast to the Cloud, the Fog performs latency-sensitive applications at the edge of network, however latency tolerant tasks are efficiently performed in the Cloud for deep analytics [163].

Cloud computing provides on demand and scalable storage and processing services that can scale up to requirements of IoT based CPS. However, for healthcare applications, manufacturing control applications, connected vehicle applications, emergency response, and other latency sensitive applications, the delay caused by transferring data to the Cloud and back to the application becomes unacceptable [164][165][166]. The latency sensitive applications rely on the Fog for their time critical functionality. The adoption of Fog computing not only greatly improves the response time of time sensitive applications but also brings some new challenges such as business model, security, privacy and scalability. It is perceived that in time critical services Fog computing is cost-effective compared to Cloud computing due to its lesser latency and in some cases due to spare capacity of locally available resources. The view is endorsed by study carried out in [165], which shows that with high number of latency sensitive applications Fog computing outperforms Cloud computing in terms of power consumption service latency and cost.

Since data stream analytics processes the data in one scan, due to perishable insights, some algorithms, are infeasible for streaming data as they require multiple scans of data [167]. In addition, for memory-based methods such as the Parzen probability density model and nearest-neighbour methods, the entire training set needs to be stored in order to make predictions for future data points. Also, a metric is required to be defined to measure the similarity of any two vectors in input space. These requirements are both memory consuming and generally slow at making predictions for test data points. Therefore they should not be employed for data stream analysis, even though the Fog computing is introduced.

ANN (MLP), DT and SVM are the most commonly used machine learning methods in surveyed CPS. In terms of accuracy, it is observed that the performance of these machine learning methods is task dependent. For example, [77] pointed out that the best classifier differs according to the weather conditions. The classifier based on MLP behaves better than SVM (and SOM) for sunny and foggy conditions, whereas for rainy conditions, the SVM-based model is the most appropriate. [168] concluded that in automatic Stereotypical Motor Movements (SMM) recognition, SVM appears to outperform DT on overall accuracy by ~6 percentage points (although at times DT did outperform SVM), regardless of feature set used. In terms of the operation (classification or regression) time, [109] discovered the noticeably lower detection latency provided by DT while [78] ascertained that SVM was not fast enough for real-time classification (classification time being around 2.2 seconds) compared to ANN with seven hidden nodes (classification time being around 100 milliseconds).

For those machine learning methods that need massive training data and take iterations to converge, such as ANN, HMM and reinforcement learning methods, it is recommended to deploy the training tasks onto the Cloud while deploying the on-line analysis tasks on the Edge/Fog.

For deep NN learning methods such as CNN, the weights in the convolutional layers are trained and updated in the similar way to traditional MLPs except that the number of weights and layers are orders of magnitude higher than MLPs. As the training of these deep NN models needs massive resources (e.g. to accommodate the training data) and time, they should be carried out on the Cloud. However, the operation latency of these models is only proportional to the number of neurons no matter how large the training data is, the online analysis tasks can be deployed on the Edge/Fog.

When machine learning methods are deployed on the Edge, trade-offs are needed among accuracy, operation time, and the parameters of these methods such as sliding window sizes,

number of iterations and prediction/forecast time lags [53][73].

Application dependent data pre-processing proved effective in improving the performance of the data analysis. For example, in [78], before employing an ANN classifier, a simple gradient detector and an intensity-bump detector with loose (low) threshold values are applied to quickly filter out non-lane markings. As the remaining samples are much smaller in number, the classification time was significantly reduced. Due to space limitations, this paper does not investigate the data pre-processing techniques for machine learning methods in CPS.

The distributed and parallel environment provided by Cloud and Fog computing may facilitate the execution of machine learning methods (such as Random Forest) to further reduce the classification time as the sets of sub-tasks (such as the decision trees involved in Random Forest) can be run in parallel.

The data stream properties also could affect the choice of the methods. For example, fuzzy logic is more capable of dealing with fuzzy information without requiring large volumes of samples. Existing deep learning methods will require substantial numbers of samples in the training process. In addition, ANN is likely to be more appropriate to deal with multiple variant data sets than reinforcement learning methods..

Conclusion and Future Research Directions

Data stream analytics is one of the core components in CPS and machine learning methods have proved to be effective techniques of data analytics. The rise of Cloud and Fog computing paradigm calls for the study of how the machine learning based CPS data stream analytics should be integrated to such a paradigm in order to better meet CPS requirements of mission criticality and time criticality. This paper investigated and summarized the existing machine learning methods for CPS data stream analytics from various perspectives, especially from the time complexity point of view. The investigation led to the discussion and guidance of how the CPS machine learning methods should be integrated to the Cloud and Fog architecture. In the future, more effective and efficient machine learning methods should be developed for analysing the ever growing data streams in CPS. of Distributed and parallel environments provided by the Cloud and Fog computing [169] may be utilised. Hierarchical and composable machine learning methods that are well suited to partitioned execution across the Cloud and the Edge are needed, as are transfer and continual learning techniques to deal with the non-stationarity of data streams. In the meanwhile, studies should be carried out on the development of Cloud and Edge systems that facilitate the CPS data stream analytics by accommodating the discrepancy and the heterogeneity between the capabilities of Edge devices and data centre servers and among the Edge devices themselves, providing uniformed APIs [170] and services [171][172].

Acknowledgement

This work is partially supported by EU H2020 programme (Project NOESIS under grant no 769980) and Innovate UK project (Project SABOT under grand no 103539).

Reference

[1] E.A. Lee, The past, present and future of cyber-physical systems: A focus on models, Sensors (Switzerland). 15 (2015) 4837–4869. doi:10.3390/s150304837.

[2] R. Chaâri, F. Ellouze, A. Koubâa, B. Qureshi, N. Pereira, H. Youssef, E. Tovar, Cyber-physical systems clouds: A survey, Comput. Networks. 108 (2016) 260–278. doi:10.1016/j.comnet.2016.08.017.

[3] A. Rayes, S. Salam, Internet of Things (IoT) Overview, Internet Things From Hype to Real. (2017) 1–34. doi:10.1016/J.FUTURE.2013.01.010.

[4] NIST, Strategic Vision and Business

Drivers for 21st Century Cyber-Physical Systems, 2013. https://www.nist.gov/sites/default/files/documents/el/Exec-Roundtable-SumReport-Final-1-30-13.pdf.

[5] E. a Lee, Cyber-Physical Systems - Are Computing Foundations Adequate?, October. 1 (2006) 1–9. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.8011&rep=rep1&type=pdf.

[6] L. Wang, M. Törngren, M. Onori, Current status and advancement of cyber-physical systems in manufacturing, J. Manuf. Syst. 37 (2015) 517–527. doi:10.1016/j.jmsy.2015.04.008.

[7] NSF, Cyber-Physical Systems (CPS), 2017. https://www.nsf.gov/pubs/2017/nsf17529/nsf17529.htm.

[8] J. Shi, J. Wan, H. Yan, H. Suo, A survey of Cyber-Physical Systems, 2011 Int. Conf. Wirel. Commun. Signal Process. (2011) 1–6. doi:10.1109/WCSP.2011.6096958.

[9] X. Guan, S. Member, B. Yang, C. Chen, W. Dai, Y. Wang, A comprehensive overview of cyber-physical systems: from perspective of feedback system, IEEE/CAA J. Autom. Sin. 3 (2016) 1–14. doi:10.1109/JAS.2016.7373757.

[10] S.K. Khaitan, J.D. McCalley, Cyber physical system approach for design of power grids: A survey, 2013 IEEE Power Energy Soc. Gen. Meet. (2013) 1–5. doi:10.1109/PESMG.2013.6672537.

[11] J. Lee, B. Bagheri, H.A. Kao, A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems, Manuf. Lett. 3 (2015) 18–23. doi:10.1016/j.mfglet.2014.12.001.

[12] S.A. Asadollah, R. Inam, H. Hansson, A survey on testing for cyber physical system, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 9447 (2015) 194–207. doi:10.1007/978-3-319-25945-1_12.

[13] A. Humayed, J. Lin, F. Li, B. Luo, Cyber-Physical Systems Security -- A Survey, 4662 (2017). doi:10.1109/JIOT.2017.2703172.

[14] R. Gravina, C. Ma, P. Pace, G. Aloi, W. Russo, W. Li, G. Fortino, Cloud-based Activity-aaService cyber–physical framework for human activity monitoring in mobility, Futur. Gener. Comput. Syst. 75 (2017) 158–171. doi:10.1016/j.future.2016.09.006.

[15] G. Fortino, G. Di Fatta, M. Pathan, A. V. Vasilakos, Cloud-assisted body area networks: state-of-the-art and future challenges, Wirel. Networks. 20 (2014) 1925–1938. doi:10.1007/s11276-014-0714-1.

[16] I. Akkaya, Data-Driven Cyber-Physical Systems via Real-Time Stream Analytics and Machine Learning, (2016) 136. https://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-159.pdf.

[17] J. a Silva, E.R. Faria, R.C. Barros, E.R. Hruschka, A.C.P.L.F. De Carvalho, J. Gama, Data Stream Clustering: A Survey, {ACM} Comput. Surv. 46 (2013) 13:1–13:31. doi:10.1145/2522968.2522981.

[18] M. Mousavi, A.A. Bakar, M. Vakilian, Data stream clustering algorithms: A review, Int. J. Adv. Soft Comput. Its Appl. 7 (2015) 1–15.

[19] M.D. Jayanthi, A Framework for Real-time Streaming Analytics using Machine Learning Approach, (2016) 85–92.

[20] E. Alpaydın, Introduction to Machine Learning Second Edition, 2010.

[21] SAS, Machine Learning: What it is & why it matters, (n.d.). https://www.sas.com/it_it/insights/analytics/machine-learning.html.

[22] P. Bhavsar, I. Safro, N. Bouaynaya, R. Polikar, D. Dera, Mchine Learning in Transportation Data Analysis, in: Data Anal. Intell. Transp. Syst., 2017: pp. 283–307.

[23] S. Distefano, G. Merlino, A. Puliafito, A utility paradigm for IoT: The sensing Cloud, Pervasive Mob. Comput. 20 (2015) 127–144. doi:10.1016/j.pmcj.2014.09.006.

[24] B. Christophe, M. Boussard, M. Lu, A. Pastor, V. Toubiana, The web of things vision: Things as a service and interaction

patterns, Bell Labs Tech. J. 16 (2011) 55–61. doi:10.1002/bltj.20485.

[25] Y. Zhang, M. Qiu, C.-W. Tsai, M.M. Hassan, A. Alamri, Health-CPS: Healthcare Cyber-Physical System Assisted by Cloud and Big Data, IEEE Syst. J. 11 (2017) 88–95. doi:10.1109/JSYST.2015.2460747.

[26] M.S. Hossain, M.M. Hassan, M. Al Qurishi, A. Alghamdi, Resource Allocation for Service Composition in Cloud-based Video Surveillance Platform, in: 2012 IEEE Int. Conf. Multimed. Expo Work., IEEE, 2012: pp. 408–412. doi:10.1109/ICMEW.2012.77.

[27] A. Botta, W. de Donato, V. Persico, A. Pescapé, Integration of Cloud computing and Internet of Things: A survey, Futur. Gener. Comput. Syst. 56 (2016) 684–700. doi:10.1016/j.future.2015.09.021.

[28] I. Stojmenovic, Fog computing: A cloud to the ground support for smart things and machine-to-machine networks, in: 2014 Australas. Telecommun. Networks Appl. Conf., IEEE, 2014: pp. 117–122. doi:10.1109/ATNAC.2014.7020884.

[29] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proc. First Ed. MCC Work. Mob. Cloud Comput. - MCC '12, ACM Press, New York, New York, USA, 2012: p. 13. doi:10.1145/2342509.2342513.

[30] A.V. Dastjerdi, R. Buyya, Fog Computing: Helping the Internet of Things Realize Its Potential, Computer (Long. Beach. Calif). 49 (2016) 112–116. doi:10.1109/MC.2016.245.

[31] M. Aazam, E.-N. Huh, Fog Computing and Smart Gateway Based Communication for Cloud of Things, in: 2014 Int. Conf. Futur. Internet Things Cloud, IEEE, 2014: pp. 464–470. doi:10.1109/FiCloud.2014.83.

[32] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, B. Koldehofe, Mobile fog: a programming model for large-scale applications on the internet of things, in: Proc. Second ACM SIGCOMM Work. Mob. Cloud Comput. - MCC '13, ACM Press, New York, New York, USA, 2013: p. 15. doi:10.1145/2491266.2491270.

[33] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, R.S. Tucker, Fog Computing May Help to Save Energy in Cloud Computing, IEEE J. Sel. Areas Commun. 34 (2016) 1728–1739. doi:10.1109/JSAC.2016.2545559.

[34] R. Mahmud, R. Kotagiri, R. Buyya, Fog Computing: A Taxonomy, Survey and Future Directions, in: 2018: pp. 103–130. doi:10.1007/978-981-10-5861-5_5.

[35] S.K. Datta, C. Bonnet, N. Nikaein, An IoT gateway centric architecture to provide novel M2M services, in: 2014 IEEE World Forum Internet Things, IEEE, 2014: pp. 514–519. doi:10.1109/WF-IoT.2014.6803221.

[36] S.F. Ochoa, G. Fortino, G. Di Fatta, Cyber-physical systems, internet of things and big data, Futur. Gener. Comput. Syst. 75 (2017) 82–84. doi:10.1016/j.future.2017.05.040.

[37] L.F. Bittencourt, M.M. Lopes, I. Petri, O.F. Rana, Towards Virtual Machine Migration in Fog Computing, in: 2015 10th Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput., IEEE, 2015: pp. 1–8. doi:10.1109/3PGCIC.2015.85.

[38] P. Bellavista, A. Zanni, Feasibility of Fog Computing Deployment based on Docker Containerization over RaspberryPi, in: Proc. 18th Int. Conf. Distrib. Comput. Netw. - ICDCN '17, ACM Press, New York, New York, USA, 2017: pp. 1–10. doi:10.1145/3007748.3007777.

[39] A. Al-Fuqaha, A. Khreishah, M. Guizani, A. Rayes, M. Mohammadi, Toward better horizontal integration among IoT services, IEEE Commun. Mag. 53 (2015) 72–79. doi:10.1109/MCOM.2015.7263375.

[40] S. Yi, C. Li, Q. Li, A Survey of Fog Computing: : Concepts, Applications and Issues, in: Proc. 2015 Work. Mob. Big Data - Mobidata '15, ACM Press, New York, New York, USA, 2015: pp. 37–42. doi:10.1145/2757384.2757397.

[41] J.. Zhang, F.-Y.. c Wang, K.. Wang, W.-H.. Lin, X.. Xu, C.. Chen, Data-driven intelligent transportation systems: A survey, IEEE Trans. Intell. Transp. Syst. 12 (2011) 1624–1639. doi:10.1109/TITS.2011.2158001.

[42] P.D. Diamantoulakis, V.M. Kapinas, G.K. Karagiannidis, Big Data Analytics for Dynamic Energy Management in Smart Grids, Big Data Res. 2 (2015) 94–101. doi:10.1016/j.bdr.2015.03.003.

[43] M.Q. Raza, A. Khosravi, A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings, Renew. Sustain. Energy Rev. 50 (2015) 1352–1372. doi:10.1016/j.rser.2015.04.065.

[44] K. Zhou, C. Fu, S. Yang, Big data driven smart energy management: From big data to big insights, Renew. Sustain. Energy Rev. 56 (2016) 215–225. doi:10.1016/j.rser.2015.11.050.

[45] X. Yu, Y. Xue, Smart Grids: A Cyber–Physical Systems Perspective, Proc. IEEE. 104 (2016) 1058–1070. doi:10.1109/JPROC.2015.2503119.

[46] EU, Framework for the Deployment of Intelligent Transport Systems in the Field of Road Transport and for Interfaces with Other Modes of Transport, 2010. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2010:207:0001:0013:EN:PDF.

[47] G. Dimitrakopoulos, P. Demestichas, Intelligent Transportation Systems: Systems Based on Cognitive Networking Principles and Management Functionality, IEEE Veh. Technol. Mag. (2009) 77–84.

[48] S. Ezell, Intelligent Transportation Systems, 2010. https://www.itif.org/files/2010-1-27-ITS_Leadership.pdf.

[49] MINI Plant Oxford, Assembly, (n.d.). http://miniplantoxford.co.uk/production/assembly.aspx.

[50] A. Kuss, T. Dietz, F. Spenrath, A. Verl, Automated Planning of Robotic MAG Welding Based on Adaptive Gap Model, Procedia CIRP. 62 (2017) 612–617. doi:10.1016/j.procir.2016.07.008.

[51] T.M. Mitchell, The discipline of machine learning, Carnegie Mellon University, School of Computer Science, Machine Learning Department, 2006.

[52] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, New York, USA, 2006.

[53] J. Ferreira, E. Carvalho, B. V. Ferreira, C. de Souza, Y. Suhara, A. Pentland, G. Pessin, Driver behavior profiling: An investigation with different smartphone sensors and machine learning, PLoS One. 12 (2017) e0174959. doi:10.1371/journal.pone.0174959.

[54] L. Breinman, Random forests, Mach. Learn. 45 (2001) 5–32. doi:10.1186/1478-7954-9-29.

[55] Z. Wang, T. Oates, Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks, Work. Twenty-Ninth AAAI Conf. Artif. Intell. (2015) 40–46. http://www.aaai.org/ocs/index.php/WS/AAAIW15/paper/view/10179.

[56] Y. Zheng, Q. Liu, E. Chen, Y. Ge, J.L. Zhao, Time Series Classification Using Multi-Channels Deep Convolutional Neural Networks, Web-Age Inf. Manag. SE - 33. 8485 (2014) 298–310. doi:10.1007/978-3-319-08010-9_33.

[57] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.

[58] L. Deng, D. Yu, Deep Learning: Methods and Applications, Found. Trends® Signal Process. 7 (2014) 197–387. doi:10.1561/2000000039.

[59] V. Drucker, H., Burges, C. J., Kaufman, L., Smola, A. J., & Vapnik, Support vector regression machines, in: Adv. Neural Inf. Process. Syst., 1997: pp. 155–161.

[60] G. Santafe, Advances on Supervised and Unsupervised Learning of Bayesian Network Models, University of the Basque Country, 2007. http://www.sc.ehu.es/ccwbayes/members/guzman/pdfs/guzmanTesis.pdf.

[61] A.E. Eiben, J.E. Smith, Introduction to Evolutionary Computing, Springer, 2003.

[62] H. Ishibuchi, T. Nakashima, T. Murata, A fuzzy classifier system that generates fuzzy if-then rules for pattern classification problems, in: Proc. 1995 IEEE Int. Conf. Evol. Comput., IEEE, 1995: pp. 759–764.

doi:10.1109/ICEC.1995.487481.

[63] U. Maulik, S. Bandyopadhyay, Genetic algorithm-based clustering technique, Pattern Recognit. 33 (2000) 1455–1465. doi:10.1016/S0031-3203(99)00137-5.

[64] S. Whiteson, Evolutionary Computation for Reinforcement Learning, in: 2012: pp. 325–355. doi:10.1007/978-3-642-27645-3_10.

[65] T. Kohonen, Essentials of the self-organizing map, Neural Networks. 37 (2013) 52–65. doi:10.1016/j.neunet.2012.09.018.

[66] C.J.C.H. Watkins, P. Dayan, Q-learning, Mach. Learn. 8 (1992) 279–292. doi:10.1007/BF00992698.

[67] G. Sideratos, A. Ikonomopoulos, N. Hatziargyriou, A Committee of Machine Learning Techniques for Load Forecasting in a Smart Grid Environment, Int. J. Energy Power. 4 (2015) 98. doi:10.14355/ijep.2015.04.016.

[68] L. Xiao, W. Shao, C. Wang, K. Zhang, H. Lu, Research and application of a hybrid model based on multi-objective optimization for electrical load forecasting, Appl. Energy. 180 (2016) 213–233. doi:10.1016/j.apenergy.2016.07.113.

[69] W. Yu, D. An, D. Griffith, Q. Yang, G. Xu, Towards statistical modeling and machine learning based energy usage forecasting in smart grid, ACM SIGAPP Appl. Comput. Rev. 15 (2015) 6–16. doi:10.1145/2753060.2753061.

[70] C. Guan, P.B. Luh, L.D. Michel, Y. Wang, P.B. Friedland, Very Short-Term Load Forecasting: Wavelet Neural Networks With Data Pre-Filtering, IEEE Trans. Power Syst. 28 (2013) 30–41. doi:10.1109/TPWRS.2012.2197639.

[71] A. Kusiak, Haiyang Zheng, Zhe Song, Short-Term Prediction of Wind Farm Power: A Data Mining Approach, IEEE Trans. Energy Convers. 24 (2009) 125–136. doi:10.1109/TEC.2008.2006552.

[72] K.Y. Chan, T.S. Dillon, J. Singh, E. Chang, Neural-Network-Based Models for Short-Term Traffic Flow Forecasting Using a Hybrid Exponential Smoothing and Levenberg–Marquardt Algorithm, IEEE Trans. Intell. Transp. Syst. 13 (2012) 644–654. doi:10.1109/TITS.2011.2174051.

[73] P. Zito, Haibo Chen, M.C. Bell, Predicting Real-Time Roadside CO and CO2 Concentrations Using Neural Networks, IEEE Trans. Intell. Transp. Syst. 9 (2008) 514–522. doi:10.1109/TITS.2008.928259.

[74] A. Khosravi, E. Mazloumi, S. Nahavandi, D. Creighton, J.W.C. Van Lint, Prediction intervals to account for uncertainties in travel time prediction, IEEE Trans. Intell. Transp. Syst. 12 (2011) 537–547. doi:10.1109/TITS.2011.2106209.

[75] J.W.C. van Lint, Online Learning Solutions for Freeway Travel Time Prediction, IEEE Trans. Intell. Transp. Syst. 9 (2008) 38–47. doi:10.1109/TITS.2008.915649.

[76] T. Yin, G. Zhong, J. Zhang, S. He, B. Ran, A prediction model of bus arrival time at stops with multi-routes, Transp. Res. Procedia. 25 (2017) 4623–4636. doi:10.1016/j.trpro.2017.05.381.

[77] F. Castaño, G. Beruvides, R. Haber, A. Artuñedo, Obstacle Recognition Based on Machine Learning for On-Chip LiDAR Sensors in a Cyber-Physical System, Sensors. 17 (2017) 2109. doi:10.3390/s17092109.

[78] Z. Kim, Robust Lane Detection and Tracking in Challenging Scenarios, IEEE Trans. Intell. Transp. Syst. 9 (2008) 16–26. doi:10.1109/TITS.2007.908582.

[79] S.-J. Shin, J. Woo, S. Rachuri, Predictive Analytics Model for Power Consumption in Manufacturing, Procedia CIRP. 15 (2014) 153–158. doi:10.1016/j.procir.2014.06.036.

[80] C. Shen, L. Wang, Q. Li, Optimization of injection molding process parameters using combination of artificial neural network and genetic algorithm method, J. Mater. Process. Technol. 183 (2007) 412–418. doi:10.1016/j.jmatprotec.2006.10.036.

[81] A. Lahouar, J. Ben Hadj Slama, Random forests model for one day ahead load forecasting, in: IREC2015 Sixth Int. Renew.

Energy Congr., IEEE, 2015: pp. 1–6. doi:10.1109/IREC.2015.7110975.

[82] Jian Zhou, Zhaoqiang Ge, Shang Gao, Yanli Xu, Fault record detection with random forests in data center of large power grid, in: 2016 IEEE PES Asia-Pacific Power Energy Eng. Conf., IEEE, 2016: pp. 1641–1645. doi:10.1109/APPEEC.2016.7779771.

[83] D. Wu, C. Jennings, J. Terpenny, S. Kumara, Cloud-based machine learning for predictive analytics: Tool wear prediction in milling, in: 2016 IEEE Int. Conf. Big Data (Big Data), IEEE, 2016: pp. 2062–2069. doi:10.1109/BigData.2016.7840831.

[84] D. Gkorou, T. Hoogenboom, A. Ypma, G. Tsirogiannis, M. Giollo, D. Sonntag, G. Vinken, R. van Haren, R.J. van Wijk, J. Nije, Towards Big Data Visualization for Monitoring and Diagnostics of High Volume Semiconductor Manufacturing, in: Proc. Comput. Front. Conf. ZZZ - CF'17, ACM Press, New York, New York, USA, 2017: pp. 338–342. doi:10.1145/3075564.3078883.

[85] L. Auret, C. Aldrich, Unsupervised Process Fault Detection with Random Forests, Ind. Eng. Chem. Res. 49 (2010) 9184–9194. doi:10.1021/ie901975c.

[86] F. Oldewurtel, A. Ulbig, A. Parisio, G. Andersson, M. Morari, Reducing peak electricity demand in building climate control using real-time pricing and model predictive control, in: 49th IEEE Conf. Decis. Control, IEEE, 2010: pp. 1927–1932. doi:10.1109/CDC.2010.5717458.

[87] H. Chao, Efficient pricing and investment in electricity markets with intermittent resources, Energy Policy. 39 (2011) 3945–3953. doi:10.1016/j.enpol.2011.01.010.

[88] S. Gupta, R. Kambli, S. Wagh, F. Kazi, Support-Vector-Machine-Based Proactive Cascade Prediction in Smart Grid Using Probabilistic Framework, IEEE Trans. Ind. Electron. 62 (2015) 2478–2486. doi:10.1109/TIE.2014.2361493.

[89] W.Y. Zhang, W.-C. Hong, Y. Dong, G. Tsai, J.-T. Sung, G. Fan, Application of SVR with chaotic GASA algorithm in cyclic electric load forecasting, Energy. 45 (2012) 850–858. doi:10.1016/j.energy.2012.07.006.

[90] J. Nagi, A.M. Mohammad, K.S. Yap, S.K. Tiong, S.K. Ahmed, Non-Technical Loss analysis for detection of electricity theft using support vector machines, in: 2008 IEEE 2nd Int. Power Energy Conf., IEEE, 2008: pp. 907–912. doi:10.1109/PECON.2008.4762604.

[91] A.H. Nizar, Z.Y. Dong, M. Jalaluddin, M.J. Raffles, Load Profiling Method in Detecting non-Technical Loss Activities in a Power Utility, in: 2006 IEEE Int. Power Energy Conf., IEEE, 2006: pp. 82–87. doi:10.1109/PECON.2006.346624.

[92] M. Esmalifalak, Nam Tuan Nguyen, Rong Zheng, Zhu Han, Detecting stealthy false data injection using machine learning in smart grid, in: 2013 IEEE Glob. Commun. Conf., IEEE, 2013: pp. 808–813. doi:10.1109/GLOCOM.2013.6831172.

[93] A.A. Albousefi, H. Ying, D. Filev, F. Syed, K.O. Prakah-Asante, F. Tseng, H.-H. Yang, A two-stage-training support vector machine approach to predicting unintentional vehicle lane departure, J. Intell. Transp. Syst. 21 (2017) 41–51. doi:10.1080/15472450.2016.1196141.

[94] A. Ponz, C.H. Rodríguez-Garavito, F. García, P. Lenz, C. Stiller, J.M. Armingol, Laser Scanner and Camera Fusion for Automatic Obstacle Classification in ADAS Application, in: 2015: pp. 237–249. doi:10.1007/978-3-319-27753-0_13.

[95] T. Liu, Y. Yang, G.-B. Huang, Y.K. Yeo, Z. Lin, Driver Distraction Detection Using Semi-Supervised Machine Learning, IEEE Trans. Intell. Transp. Syst. 17 (2016) 1108–1120. doi:10.1109/TITS.2015.2496157.

[96] R.P. Ribeiro, P. Pereira, J. Gama, Sequential anomalies: a study in the Railway Industry, Mach. Learn. 105 (2016) 127–153. doi:10.1007/s10994-016-5584-6.

[97] S. Moridpour, T. Anwar, M.T. Sadat, E. Mazloumi, A genetic algorithm-based support vector machine for bus travel time prediction, in: 2015 Int. Conf. Transp. Inf. Saf., IEEE, 2015: pp. 264–270. doi:10.1109/ICTIS.2015.7232119.

[98] G.A. Susto, A. Schirru, S. Pampuri, S. McLoone, A. Beghi, Machine Learning for Predictive Maintenance: A Multiple Classifier Approach, IEEE Trans. Ind. Informatics. 11 (2015) 812–820. doi:10.1109/TII.2014.2349359.

[99] X. Yao, S.K. Moon, G. Bi, A hybrid machine learning approach for additive manufacturing design feature recommendation, Rapid Prototyp. J. 23 (2017) 983–997. doi:10.1108/RPJ-03-2016-0041.

[100] A. Madureira, J.M. Santos, S. Gomes, B. Cunha, J.P. Pereira, I. Pereira, Manufacturing rush orders rescheduling: a supervised learning approach, in: 2014 Sixth World Congr. Nat. Biol. Inspired Comput. (NaBIC 2014), IEEE, 2014: pp. 299–304. doi:10.1109/NaBIC.2014.6921895.

[101] T. Wuest, C. Irgens, K.-D. Thoben, An approach to monitoring quality in manufacturing using supervised machine learning on product state data, J. Intell. Manuf. 25 (2014) 1167–1180. doi:10.1007/s10845-013-0761-y.

[102] H. Ma, Y. Wang, K. Wang, Automatic detection of false positive RFID readings using machine learning algorithms, Expert Syst. Appl. 91 (2018) 442–451. doi:10.1016/j.eswa.2017.09.021.

[103] D.J. Cook, A.S. Crandall, B.L. Thomas, N.C. Krishnan, CASAS: A Smart Home in a Box, Computer (Long. Beach. Calif). 46 (2013) 62–69. doi:10.1109/MC.2012.328.

[104] N.C. Krishnan, D.J. Cook, Activity recognition on streaming sensor data, Pervasive Mob. Comput. 10 (2014) 138–154. doi:10.1016/j.pmcj.2012.07.003.

[105] I. Monedero, F. Biscarri, C. León, J.I. Guerrero, J. Biscarri, R. Millán, Detection of frauds and other non-technical losses in a power utility using Pearson coefficient, Bayesian networks and decision trees, Int. J. Electr. Power Energy Syst. 34 (2012) 90–98. doi:10.1016/j.ijepes.2011.09.009.

[106] Y. Simmhan, S. Aman, A. Kumbhare, R. Liu, S. Stevens, Q. Zhou, V. Prasanna, Cloud-Based Software Platform for Big Data Analytics in Smart Grids, Comput. Sci. Eng. 15 (2013) 38–47. doi:10.1109/MCSE.2013.39.

[107] E. Osaba, E. Onieva, A. Moreno, P. Lopez-Garcia, A. Perallos, P.G. Bringas, Decentralised intelligent transport system with distributed intelligence based on classification techniques, IET Intell. Transp. Syst. 10 (2016) 674–682. doi:10.1049/iet-its.2016.0047.

[108] F.C.C. Garcia, A.E. Retamar, Towards building a bus travel time prediction model for Metro Manila, in: 2016 IEEE Reg. 10 Conf., IEEE, 2016: pp. 3805–3808. doi:10.1109/TENCON.2016.7848775.

[109] T.P. Vuong, Cyber-physical intrusion detection for robotic vehicles, University of Greenwich, 2017. http://gala.gre.ac.uk/17445/7/Tuan Vuong 2017.pdf.

[110] 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. doi:10.1016/j.procir.2013.05.033.

[111] V. Sugumaran, V. Muralidharan, K.I. Ramachandran, Feature selection using Decision Tree and classification through Proximal Support Vector Machine for fault diagnostics of roller bearing, Mech. Syst. Signal Process. 21 (2007) 930–942. doi:10.1016/j.ymssp.2006.05.004.

[112] Y. Liu, S. Jin, Application of Bayesian networks for diagnostics in the assembly process by considering small measurement data sets, Int. J. Adv. Manuf. Technol. 65 (2013) 1229–1237. doi:10.1007/s00170-012-4252-7.

[113] L. Hao, L. Bian, N. Gebraeel, J. Shi, Residual Life Prediction of Multistage Manufacturing Processes With Interaction Between Tool Wear and Product Quality Degradation, IEEE Trans. Autom. Sci. Eng. 14 (2017) 1211–1224. doi:10.1109/TASE.2015.2513208.

[114] S. Nannapaneni, S. Mahadevan, S. Rachuri, Performance evaluation of a manufacturing process under uncertainty using Bayesian networks, J. Clean. Prod.

113 (2016) 947–959. doi:10.1016/j.jclepro.2015.12.003.

[115] V.N. Coelho, I.M. Coelho, B.N. Coelho, A.J.R. Reis, R. Enayatifar, M.J.F. Souza, F.G. Guimarães, A self-adaptive evolutionary fuzzy model for load forecasting problems on smart grid environment, Appl. Energy. 169 (2016) 567–584. doi:10.1016/j.apenergy.2016.02.045.

[116] S. Chakraborty, T. Senjyu, A. Yona, A.Y. Saber, T. Funabashi, Solving economic load dispatch problem with valve-point effects using a hybrid quantum mechanics inspired particle swarm optimisation, IET Gener. Transm. Distrib. 5 (2011) 1042. doi:10.1049/iet-gtd.2011.0038.

[117] C.C.O. Ramos, A.N. Souza, G. Chiachia, A.X. Falcão, J.P. Papa, A novel algorithm for feature selection using Harmony Search and its application for non-technical losses detection, Comput. Electr. Eng. 37 (2011) 886–894. doi:10.1016/j.compeleceng.2011.09.013.

[118] P. Ray, D.P. Mishra, Support vector machine based fault classification and location of a long transmission line, Eng. Sci. Technol. an Int. J. 19 (2016) 1368–1380. doi:10.1016/j.jestch.2016.04.001.

[119] M. Yuwono, Y. Qin, J. Zhou, Y. Guo, B.G. Celler, S.W. Su, Automatic bearing fault diagnosis using particle swarm clustering and Hidden Markov Model, Eng. Appl. Artif. Intell. 47 (2016) 88–100. doi:10.1016/j.engappai.2015.03.007.

[120] T. Navalertporn, N. V. Afzulpurkar, Optimization of tile manufacturing process using particle swarm optimization, Swarm Evol. Comput. 1 (2011) 97–109. doi:10.1016/j.swevo.2011.05.003.

[121] D. Li, S.K. Jayaweera, Machine-Learning Aided Optimal Customer Decisions for an Interactive Smart Grid, IEEE Syst. J. 9 (2015) 1529–1540. doi:10.1109/JSYST.2014.2334637.

[122] A.H. Tai, W.-K. Ching, L.Y. Chan, Detection of machine failure: Hidden Markov Model approach, Comput. Ind. Eng. 57 (2009) 608–619. doi:10.1016/j.cie.2008.09.028.

[123] J. Zheng, L.M. Ni, Modeling heterogeneous routing decisions in trajectories for driving experience learning, in: Proc. 2014 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput. - UbiComp '14 Adjun., ACM Press, New York, New York, USA, 2014: pp. 951–961. doi:10.1145/2632048.2632089.

[124] E. Mocanu, P.H. Nguyen, M. Gibescu, W.L. Kling, Deep learning for estimating building energy consumption, Sustain. Energy, Grids Networks. 6 (2016) 91–99. doi:10.1016/j.segan.2016.02.005.

[125] Y. Lv, Y. Duan, W. Kang, Z. Li, F.-Y. Wang, Traffic Flow Prediction With Big Data: A Deep Learning Approach, IEEE Trans. Intell. Transp. Syst. (2014) 1–9. doi:10.1109/TITS.2014.2345663.

[126] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L.D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, K. Zieba, End to End Learning for Self-Driving Cars, (2016). http://arxiv.org/abs/1604.07316.

[127] Z. Chen, X. Huang, End-to-end learning for lane keeping of self-driving cars, in: 2017 IEEE Intell. Veh. Symp., IEEE, 2017: pp. 1856–1860. doi:10.1109/IVS.2017.7995975.

[128] A. Taylor, Anomaly-based detection of malicious activity in in-vehicle networks Anomaly-based detection of malicious activity in in-vehicle networks, University of Ottawa, 2017. https://www.google.co.uk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0ahUKEwi1-cnir6XYAhVNY1AKHYY-DPEQFgg2MAE&url=https%3A%2F%2Fruor.uottawa.ca%2Fbitstream%2F10393%2F36120%2F3%2FTaylor_Adrian_2017_thesis.pdf&usg=AOvVaw1mk_GeMwTMT0Yn6kUxiMK.

[129] P. Nicolas, Time Complexity: Graph and Machine Learning Algorithms, (2015). http://www.scalaformachinelearning.com/2015/11/time-complexity-in-machine-learning.html.

[130] J. Su, H. Zhang, A Fast Decision Tree Learning Algorithm, 21st Natl. Conf. Artif. Intell. - Vol. 1. 5 (2006) 500–505.

[131] K. He, J. Sun, Convolutional Neural Networks at Constrained Time Cost, in: IEEE Conf. Comput. Vis. Pattern Recognit., 2015: pp. 5353–5360. doi:10.1109/CVPR.2015.7299173.

[132] L. Bottou, C. Lin, Support Vector Machine Solvers, (2006).

[133] G. Louppen, Understanding Random Forest from theory to practice, University of Liège, 2014. https://arxiv.org/pdf/1407.7502.pdf.

[134] D.G. Roussinov, H. Chen, A Scalable Self-organizing Map Algorithm for Textual Classification: A Neural Network Approach to Thesaurus Generation, Commun. Cogn. Artif. Intell. Spring. 15 (1998) 81–112.

[135] S. Koenig, R.G. Simmons, Complexity Analysis of Real-Time Reinforcement Learning, Proc. AAAI Conf. Artif. Intell. (1993) 99–105.

[136] D. Simon, Biogeography-Based Optimization, IEEE Trans. Evol. Comput. 12 (2008) 702–713. doi:10.1109/TEVC.2008.919004.

[137] J.H. Korhonen, P. Parviainen, Exact Learning of Bounded Tree-width Bayesian Networks, in: Proc. 16th Int. Conf. AI Stat., 2013: pp. 370–378.

[138] A. Bhattacharya, P.K. Chattopadhyay, Biogeography-Based Optimization for Different Economic Load Dispatch Problems, IEEE Trans. Power Syst. 25 (2010) 1064–1077. doi:10.1109/TPWRS.2009.2034525.

[139] V. Losing, B. Hammer, H. Wersing, Incremental on-line learning: A review and comparison of state of the art algorithms, Neurocomputing. 275 (2018) 1261–1274. doi:10.1016/j.neucom.2017.06.084.

[140] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Comput. Surv. 46 (2014) 1–37. doi:10.1145/2523813.

[141] A. R.R, D. P. R, Methods for Incremental Learning : A Survey, Int. J. Data Min. Knowl. Manag. Process. 3 (2013) 119–125. doi:10.5121/ijdkp.2013.3408.

[142] D. Hadas, G. Yovel, N. Intrator, Using unsupervised incremental learning to cope with gradual concept drift, Conn. Sci. 23 (2011) 65–83. doi:10.1080/09540091.2011.575929.

[143] Nan-Ying Liang, Guang-Bin Huang, P. Saratchandran, N. Sundararajan, A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks, IEEE Trans. Neural Networks. 17 (2006) 1411–1423. doi:10.1109/TNN.2006.880583.

[144] S. Suresh, K. Dong, H.J. Kim, A sequential learning algorithm for self-adaptive resource allocation network classifier, Neurocomputing. 73 (2010) 3012–3019. doi:10.1016/j.neucom.2010.07.003.

[145] H.T. Huynh, Y. Won, Regularized online sequential learning algorithm for single-hidden layer feedforward neural networks, Pattern Recognit. Lett. 32 (2011) 1930–1935. doi:10.1016/j.patrec.2011.07.016.

[146] L. Guo, J. Hao, M. Liu, An incremental extreme learning machine for online sequential learning problems, Neurocomputing. 128 (2014) 50–58. doi:10.1016/j.neucom.2013.03.055.

[147] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1986) 81–106. doi:10.1007/BF00116251.

[148] P.E. Utgoff, Incremental induction of decision trees, Mach. Learn. 4 (1989) 161–186. doi:10.1023/A:1022699900025.

[149] D. Kalles, T. Morris, Efficient incremental induction of decision trees, Mach. Learn. 24 (1996) 231–242. doi:10.1007/BF00058613.

[150] P. Domingos, G. Hulten, Mining high-speed data streams, in: Proc. Sixth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD '00, ACM Press, New York, New York, USA, 2000: pp. 71–80. doi:10.1145/347090.347107.

[151] J. Gama, R. Rocha, P. Medas, Accurate decision trees for mining high-speed data streams, in: Proc. Ninth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD '03,

ACM Press, New York, New York, USA, 2003: p. 523. doi:10.1145/956750.956813.

[152] B. Pfahringer, G. Holmes, R. Kirkby, New Options for Hoeffding Trees, in: Proc. 20th Aust. Jt. Conf. Adv. Artif. Intell., Springer-Verlag, Berlin, Heidelberg, 2007: pp. 90–99. http://dl.acm.org/citation.cfm?id=1781238.1781251.

[153] L. Rutkowski, L. Pietruczuk, P. Duda, M. Jaworski, Decision Trees for Mining Data Streams Based on the McDiarmid's Bound, IEEE Trans. Knowl. Data Eng. 25 (2013) 1272–1279. doi:10.1109/TKDE.2012.66.

[154] L. Rutkowski, M. Jaworski, L. Pietruczuk, P. Duda, The CART decision tree for mining data streams, Inf. Sci. (Ny). 266 (2014) 1–15. doi:10.1016/j.ins.2013.12.060.

[155] H. Abdulsalam, D.B. Skillicorn, P. Martin, Streaming Random Forests, in: 11th Int. Database Eng. Appl. Symp. (IDEAS 2007), IEEE, 2007: pp. 225–232. doi:10.1109/IDEAS.2007.4318108.

[156] H. Abdulsalam, D.B. Skillicorn, P. Martin, Classification Using Streaming Random Forests, IEEE Trans. Knowl. Data Eng. 23 (2011) 22–36. doi:10.1109/TKDE.2010.36.

[157] A. Saffari, C. Leistner, J. Santner, M. Godec, H. Bischof, On-line Random Forests, in: 2009 IEEE 12th Int. Conf. Comput. Vis. Work. ICCV Work., IEEE, 2009: pp. 1393–1400. doi:10.1109/ICCVW.2009.5457447.

[158] M. Denil, D. Matheson, N. De Freitas, Consistency of Online Random Forests, in: Int. Conf. Mach. Learn., 2013: pp. 1256–1264.

[159] J. Kivinen, A.J. Smola, R.C. Williamson, Online Learning with Kernels, IEEE Trans. Signal Process. 52 (2004) 2165–2176. doi:10.1109/TSP.2004.830991.

[160] P. Laskov, C. Gehl, S. Kruger, K.-R. Muller, Incremental Support Vector Learning: Analysis, Implementation and Applications, J. Mach. Learn. Res. 7 (2006) 1909–1936.

[161] M. Karasuyama, I. Takeuchi, Multiple Incremental Decremental Learning of Support Vector Machines, IEEE Trans. Neural Networks. 21 (2010) 1048–1059. doi:10.1109/TNN.2010.2048039.

[162] B. Gu, V.S. Sheng, K.Y. Tay, W. Romano, S. Li, Incremental Support Vector Learning for Ordinal Regression, IEEE Trans. Neural Networks Learn. Syst. 26 (2015) 1403–1416. doi:10.1109/TNNLS.2014.2342533.

[163] B. Tang, Z. Chen, G. Hefferman, S. Pei, T. Wei, H. He, Q. Yang, Incorporating Intelligence in Fog Computing for Big Data Analysis in Smart Cities, IEEE Trans. Ind. Informatics. 13 (2017) 2140–2150. doi:10.1109/TII.2017.2679740.

[164] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, E. Riviere, Edge-centric Computing: Vision and Challenges, ACM SIGCOMM Comput. Commun. Rev. 45 (2015) 37–42. doi:10.1145/2831347.2831354.

[165] L. Gu, D. Zeng, S. Guo, A. Barnawi, Y. Xiang, Cost Efficient Resource Management in Fog Computing Supported Medical Cyber-Physical System, IEEE Trans. Emerg. Top. Comput. 5 (2017) 108–119. doi:10.1109/TETC.2015.2508382.

[166] B. Xu, L. Xu, H. Cai, L. Jiang, Y. Luo, Y. Gu, The design of an m-Health monitoring system based on a cloud computing platform, Enterp. Inf. Syst. 11 (2017) 17–36. doi:10.1080/17517575.2015.1053416.

[167] Q. Tu, J.F. Lu, B. Yuan, J.B. Tang, J.Y. Yang, Density-based hierarchical clustering for streaming data, Pattern Recognit. Lett. 33 (2012) 641–645. doi:10.1016/j.patrec.2011.11.022.

[168] M.S. Goodwin, M. Haghighi, Q. Tang, M. Akcakaya, D. Erdogmus, S. Intille, Moving towards a real-time system for automatically recognizing stereotypical motor movements in individuals on the autism spectrum using wireless accelerometry, in: Proc. 2014 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput. - UbiComp '14 Adjun., ACM Press, New York, New York, USA, 2014: pp. 861–872. doi:10.1145/2632048.2632096.

[169] N. Verba, K.-M. Chao, A. James, D. Goldsmith, X. Fei, S.-D. Stan, Platform as a

service gateway for the Fog of Things, Adv. Eng. Informatics. 33 (2017) 243–257. doi:10.1016/j.aei.2016.11.003.

[170] I. Stoica, D. Song, R.A. Popa, D. Patterson, M.W. Mahoney, R. Katz, A.D. Joseph, M. Jordan, J.M. Hellerstein, J. Gonzalez, K. Goldberg, A. Ghodsi, D. Culler, P. Abbeel, A Berkeley View of Systems Challenges for AI, 2017. http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-159.html.

[171] G. Baryannis, K. Kritikos, D. Plexousakis, A specification-based QoS-aware design framework for service-based applications, Serv. Oriented Comput. Appl. 11 (2017) 301–314. doi:10.1007/s11761-017-0210-4.

[172] W. Wang, K. Lee, D. Murray, A global generic architecture for the future Internet of Things, Serv. Oriented Comput. Appl. 11 (2017) 329–344. doi:10.1007/s11761-017-0213-1.