Machine Learning and Complex Event Processing

A Review of Real-time Data Analytics for the Industrial Internet of Things

Jonas Wanner[*,a], Christopher Wissuchek[a], Christian Janiesch[ab]

[a] Faculty of Business Management and Economics, University of Würzburg, Würzburg, Germany
[b] Faculty of Business and Economics, Technische Universität Dresden, Dresden, Germany

Abstract. *In the Industrial Internet of Things, cyber-physical systems bridge the gap between the physical and digital world by connecting advanced manufacturing systems with digital services in so-called smart factories. This interplay generates a large amount of data. By analyzing the data, manufacturers can reap many benefits and optimize their operations. Here, the value of information is at its highest with low latency to its emergence and its value decreases over time. Complex Event Processing (CEP) is a technology, which enables real-time analysis of complex events (i.e., combined data values from different sources). In this way, CEP assists in the identification and localization of anomalous process sequences in smart factories. However, CEP comes with limitations that reduce its effectiveness. Setting up CEP requires in-depth domain knowledge and is primarily declarative as well as reactive by nature. Combining CEP with machine learning is a possible extension to circumvent these technological limitations. However, there is no up-to-date overview on the integration of both paradigms in research and no review of their transferability for application in smart factories. In this article, we provide (1) a synthesis of research on the integration of CEP and machine learning identifying supervised learning as the predominant approach, and (2) a transfer of potentials for the use in smart factories. Here, reactive and proactive policies are used in equal frequency.*

Keywords. Machine Learning • Complex Event Processing • Real-time Data Analytics • Industrial Internet of Things • Literature Review

Communicated by Peter Fettke. Received 2019-07-18. Accepted on 2019-12-23.

1 Introduction

The amount of generated data in the modern industry is increasing rapidly (Yin and Kaynak 2015). Traditionally data analytics was performed chiefly on historical data. However, the focus of contemporary research is shifting towards real-time analytics, which fosters situation-dependent and dynamic decision making (Bruns and Dunkel 2015). An approach to achieve this is complex event processing (CEP), a technological paradigm for processing continuous data streams based on events rather than batched log files (Luckham 2011).

* Corresponding author.
E-mail. jonas.wanner@uni-wuerzburg.de
This research and development project is funded by the Bayerische Staatsministerium für Wirtschaft, Landesentwicklung und Energie (StMWi) within the framework concept "Informations- und Kommunikationstechnik" (grant no. IUK536/003) and managed by the project management agency VDI+VDE Innovation + Technik GmbH.
Note: This paper was processed as a Fast Track submission extending Wanner, J., Wissuchek, C., Janiesch, C.: "Machine Learning und Complex Event Processing: Effiziente Echtzeitauswertung am Beispiel Smart Factory", presented at the 14. Internationale Tagung Wirtschaftsinformatik (WI). Siegen, 2019, pp. 47-61.

Real-time capability is a central design principle of the Industrial Internet of Things (IIoT). In smart factories, CEP can enable the real-time analysis of sensor and process data, which can lead to cost reduction and improved operational performance (Kagermann et al. 2013). Olsson and Janiesch (2015) suggest that the shorter the action distance, that is the lower the latency of the data, the higher the value of information can be for decision making. Despite the practicability of CEP in manufacturing environments, fundamental limitations exist: CEP processes and analyzes events based on manually specified rules (Akbar et al. 2015a; Turchin et al. 2009). A rule-based realization inhibits proactivity (Turchin et al. 2009). It requires costly and complex identification and implementation tasks, which have to be performed manually. In addition, a large amount of in-depth domain knowledge is necessary for the effective use in smart factories (Mutschler and Philippsen 2012). Optimization potential cannot be realized if machine events and, thus, machine states are not properly recognized.

As a result, researchers from the domain of CEP have recently turned to machine learning (ML) to change and improve the declarative nature of CEP (Akbar et al. 2015b; Artikis et al. 2010; Mehdiyev et al. 2015; Mousheimish et al. 2016; Widder et al. 2009). ML is an approach in which computer systems automatically learn from experience based on algorithms (Samuel 1959).

In summary, we observe a need to reduce the current limitations of CEP. ML techniques seem to be a suitable candidate to possibly even obliterate them altogether. In our work, we focus on the applicability of integrating CEP with ML for smart manufacturing in the IIoT. We derive the following two research questions (RQ):

RQ1 Which ML approaches and techniques are used integrated with CEP and how can they be synthesized?

RQ2 What potential can be derived from combining CEP and ML for smart factories in the IIoT?

For answering the research questions, we structure our work as follows: Sect. 2 provides an overview of the theoretical principles of the IIoT and smart factories, CEP, and ML. In Sect. 3, we introduce our research methodology. Sect. 4 addresses research question RQ1 based on a literature-based taxonomy for the identification of potential applications. We transfer our findings to investigate challenges and improvement potentials for real-time data analytics in smart factories in Sect. 5, answering the research question RQ2. In Sect. 6, we highlight identified gaps and propose items for a research agenda. Finally, we conclude with a brief summary and outlook.

2 Theoretical Background

Smart Factories and the Industrial Internet of Things. The IIoT is a fundamental pillar of the fourth industrial revolution. It describes industrial objects, whose capacity go beyond the collection of local data. They optimize based on the collected data and use it to interact with further objects or workers by communicating over a network (Janiesch et al. 2019). Hence, the primary production factor of the IIoT is information and communication technology (Bauernhansl 2017), which are necessary for developing cyber-physical systems. They include objects, buildings, tools, or manufacturing machines that can communicate through embedded systems, detect their surroundings using connected sensors, and interact with their environment using actuators. Altogether, this makes a factory intelligent and constitutes a so-called *smart factory*. By interconnecting cyber-physical systems, new optimization and automation potential manifest, which are enabled by the real-time analysis of events (Bauernhansl 2017).

Complex Event Processing. CEP is a technological paradigm for the real-time analysis of continuous event data streams (Luckham 2011). According to Luckham (2011), an event can be anything that happens or is expected to happen. Etzion et al. (2011) extend this and refer to an

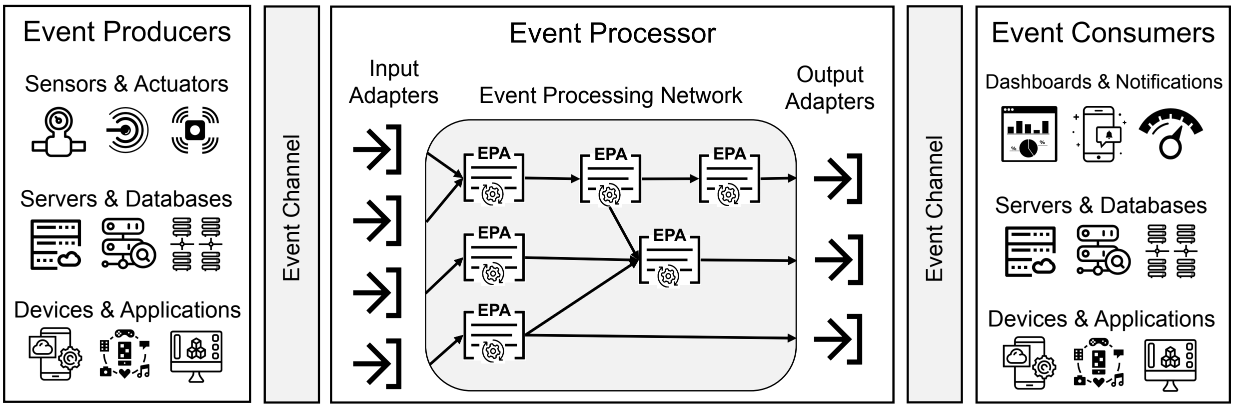

Figure 1: Concept of the Event Driven Architecture

event as an occurrence in a specific system or domain. Examples of events can be technical events such as sensor data in a production process, or business events such as financial transactions in a business process (Bruns and Dunkel 2015).

An event becomes complex if atomic events are related to each other. These relationships are based, for example, on time, causality, and aggregation (Luckham 2002), and enable abstraction across multiple levels. Low-level events comprise the lowest level of abstraction. That is the events that are transmitted directly from the event source (Luckham 2011). Low-level events can be abstracted into complex events. This abstraction is made possible by CEP through operations that are performed on (complex) events such as creating, reading, filtering, transforming, relating, and deleting events (Etzion et al. 2011).

The concept of the event-driven architecture (EDA) provides the foundation of the tripartition of the CEP reference architecture. EDA is an architectural style in which the components are event-driven and communicate unidirectional through events (Luckham 2011). The concept of the EDA is visualized in Fig. 1. It illustrates the central components of CEP and their interaction. An EDA comprises three roles with underlying components: the event producer, the event processor, and the event consumer (Bruns and Dunkel 2015).

Event producers are entities that generate low-level events and transfer them to the event processor. Examples are RFID sensors, software systems, and Web services (Bruns and Dunkel 2015). These events are forwarded through event channels, often connected through a publish-subscribe broker such as Apache Kafka (Apache Software Foundation 2019).

The event processor, that is the CEP system in a narrower sense, is the middle layer. It analyses events by step-wise processing and, thus, generates complex events to notify anyone who listens. According to Etzion et al. (2011), a CEP engine typically contains input adapters, the event processing network (EPN), and output adapters. Input adapters are responsible for converting the events of the event producers to the internal format of the CEP system. The events are then transferred to the EPN via an input event stream (Etzion et al. 2011). An EPN is a network of event processing agents (EPA) connected by information exchange channels. An EPA is an autonomous entity that process events based on manually predefined rules to detect pattern constellations of interest (Janiesch and Diebold 2016; Luckham 2011). They assume certain roles, such as filtering, transformation, and pattern recognition. Filtering describes the elimination of events that are not relevant depending on the given context or use case. Transformation describes the derivation of new events, which may become complex events. Exemplary operations are the translation, enrichment, projection, split, composition, and aggregation of events (Etzion et al. 2011). Another approach is pattern recognition. An event pattern can be regarded as a predefined template, which specifies a correlation of multiple events. After processing, (complex) events are passed to an output event stream, which passes them on to the event consumers via an output adapter for formatting (Etzion et al. 2011).

Event consumers perform the actual visible event handling in response to the event provided by CEP. In the IIoT, this entails enterprise applications or smart machinery. Examples of reactions are calling a Web service (e.g., to automate an action), updating an analytics dashboard (e.g., with live key performance indicators), publishing an event message (e.g., app notification) or triggering human process handling (Janiesch et al. 2012).

Machine Learning Approaches. ML is the science of mathematical models and algorithms that machines employ to solve tasks without relying on explicit instructions. In ML, we differentiate three fundamental learning approaches: supervised learning, unsupervised learning, and reinforcement learning.

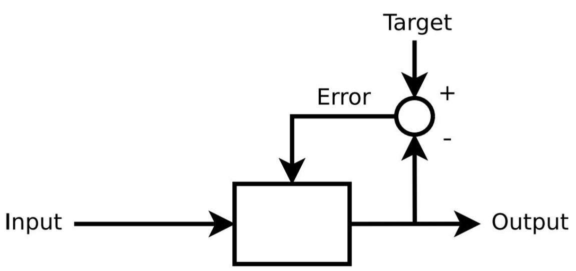

Figure 2: Supervised Learning (Tokic 2013)

Supervised learning describes a learning paradigm where inputs are assigned deliberately to predefined outputs (cf. Fig. 2). By transferring the input-output pairs for training purposes, the machine can automatically assign future cases (Marsland 2011; Shalev-Shwartz and Ben-David 2014; S. Wang et al. 2012). Supervised learning can be further sub-categorized into classification and regression approaches. In addition, techniques such as probabilistic methods, inductive logic programming, statistical estimators, and deep learning are used to train the artificial processing agents.

Classification comprises the transfer of input vectors and the decision of the assignment to classes (output). This is calculated on the basis of previously learned examples in the form of a training set (Marsland 2011). An example is a classification algorithm of detecting anomalies in financial transactions. At the outset, a training set is assigned to the algorithm with financial transactions (input) and their result of "OK" or "not OK" (output). Thereby, the algorithm derives a function that subsequently classifies other financial transactions independently (Shalev-Shwartz and Ben-David 2014; S. Wang et al. 2012). Prevalent algorithms for classification problems are based on decision trees, Bayesian networks, *k*-nearest neighbors, naive Bayes classifiers, support-vector machines (SVM), and artificial neural networks (Kotsiantis et al. 2007; Phyu 2009).

Regression analyses are statistical analysis procedures to establish relationships between variables, trend lines, and to make predictions using future output variables (Watt et al. 2016). Contrary to classification, where the output is a categorical or Boolean value, regression analysis returns a numerical output. Thus, it uses a representative line, curve, or hyper-level in relation to input-output data points (training data). At the beginning, the underlying function is unknown and must be found by training the selected algorithm using the available training data (Alpaydin 2004). Primary regression approaches are linear and non-linear techniques (Alpaydin 2004). Often, the concept is combined with other algorithmic approaches, such as support-vector regression (Drucker et al. 1997).

Probabilistic methods (or models) encompass ML techniques that use probability theory to express forms of uncertainty and noise associated with data. Among other things, this allows algorithms to infer unknown quantities, predictions, and learn from data (Ghahramani 2015). Examples for probabilistic techniques incorporated in

ML are graphical models (e.g., Bayesian networks or Markov networks), latent variable models (e.g., mixture models or linear factor analysis), and Markov-based models (e.g., Markov chains or hidden Markov models) (Bacciu et al. 2015).

Statistical estimators also referred to as optimal estimators, are computational algorithms employed in ML, which process measurements to derive the state of a system and its dynamics based on minimum error estimation using knowledge about the system itself. The analysis incorporates the fact that systems are subject to noise. By estimating an appropriate measurement error based on conditional information, noise is filtered out (Turchin et al. 2009). According to Gelb (1974), there are three types of estimation problems: when the time of an estimate corresponds to the last measurement point the problem is termed as *filtering*; when the time of interest falls within the range of available measurements, the problem is called *smoothing*; and when the time of interest is after the last available measurement, the problem is referred to as *prediction*. The most common and predominant estimation techniques are based on Kalman's work (e.g., discrete Kalman filter, continuous Kalman filter) (Kalman 1960).

Inductive logic programming is a sub-field of supervised learning based on logic programming, which uses logic as a uniform representation language for examples, background knowledge, and hypotheses (Amin 2003; Muggleton 1991). In addition, inductive logic programming also encompasses approaches that address learning from structured relational data. One example is statistical relational learning, which focuses on extending inductive logic programming to model uncertainty (De Raedt and Kersting 2008; Skarlatidis et al. 2015).

The techniques discussed above are predominantly task-specific (e.g., classify data inputs). They often require data that is easy to process mathematically. Hence, the reduction of complexity is essential and can be achieved through feature (or representation) learning methods (Argyriou et al. 2008).

Deep learning is a further, comprehensive class of ML algorithms based on artificial neural networks. It combines feature learning, but also task-specific techniques such as classification or regression, by using multiple layers to extract higher-level features from raw input progressively (Deng and Yu 2014). Consequently, deep learning is used typically on data that is not represented mathematically. This is a common case when information from the *real world* (e.g., images, speech, sensor data) is transferred to the *digital world* (Srivastava and Salakhutdinov 2012).

Figure 3: Unsupervised Machine Learning (Tokic 2013)

In *unsupervised learning*, the input-output pairs are not known (cf. Fig. 3). The machine receives only inputs. Thereby, the aim is to detect previously undiscovered patterns (Ghahramani 2003; Kubat 2017; Shalev-Shwartz and Ben-David 2014). Clustering and dimensionality reduction are the predominant unsupervised techniques.

In clustering, a data set is passed to the algorithm as input. The goal is the grouping of similar data, where the output consists of at least two clusters (Kubat 2017). Thus, distance and similarity between objects are the fundamental elements for constructing clustering algorithms. For quantitative data properties, distance is the preferred way to establish relationships between data. Qualitative analyses usually use similarity functions. Thus, clustering algorithms can be categorized based on their clustering model (Xu and Tian 2015). Traditional models include, for example, centroid-based models (e.g., *k*-means, *k*-medoids) and hierarchy-based algorithms (e.g., BIRCH, CURE, HAC). Modern approaches include more complex models, which include kernels, swarm intelligence, quantum theory, and spectral graph theory.

Dimensionality reduction is a learning process in which data is scaled from a higher dimensional level (input) to a lower level (output). The aim is often to interpret a data set and to structure it in a meaningful way (Shalev-Shwartz and Ben-David 2014). Common algorithms are based on mathematical methods, for example principal component analysis (Howley et al. 2005), linear discriminant analysis (Pang et al. 2005), or non-negative matrix factorization (Lee and Seung 2001).

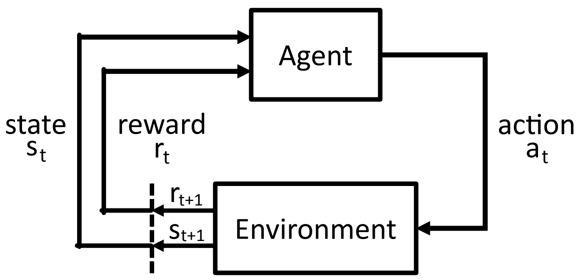

Figure 4: Reinforcement Learning (Ludvig et al. 2011)

Reinforcement learning is a paradigm, which automates learning and decision-making processes (cf. Fig. 4). In general, its logic follows a Markov decision process. A learning agent acts based on a predefined set of actions in a defined digital environment and is trained based on trial and error in combination with rewards. The aim is to maximize the cumulative rewards, whereby the learning agent learns the task-optimal behavior concerning conditions and actions (Alpaydin 2004; Sutton and Barto 2018; Tokic 2013; S. Wang et al. 2012). An example is training an artificial intelligence bot for chess. The bot adapts to the opponent's strategies and the current game situation and decides his next move accordingly. Combined with defined rewards about the changed state, it learns to improve its performance over time. Like a professional chess player, the agent refines its intuition with which he evaluates board configurations (Sutton and Barto 2018).

Combined learning integrates different approaches of ML (cf. Fig. 5). Thus, it joins techniques from supervised learning, unsupervised learning, or reinforcement learning to improve the overall result. In IT security, the detection of malware is a common example. The variety of malware is immense and continues to proliferate. Using ML a new framework has been developed to improve the accuracy of malware detection significantly. For this purpose, classification (supervised learning) was used to detect known malware classes. This was combined with the adaptability of an unsupervised learning technique to detect new classes (Comar et al. 2013).

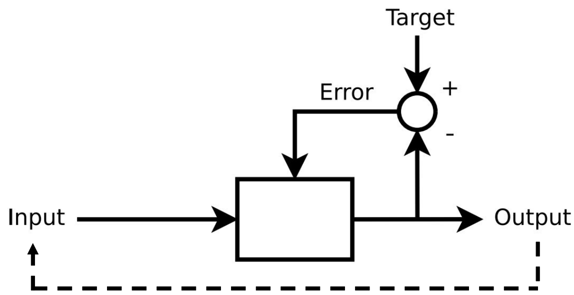

Figure 5: Combined Learning

In summary, ML approaches can be employed to overcome the barriers and limitations of manual specified processing instructions. They enable automated rule formation and optimization for known as well as yet undiscovered anomalies. For example, in CEP threshold-based filtering could be replaced by ML-based classification techniques.

3 Research Method

We employed the framework for structured literature research following vom Brocke et al. (2009). The method comprises five phases: The first phase, (1) the definition of the review scope, includes aspects defining the purpose or goal of a literature review. Subsequently, the (2) topic of interest has to be conceptualized, which encompasses the definition of key terms and issues of the research area under review. The (3) literature search covers the process of identifying relevant literature for surveying the research area under review. After collecting sufficient literature, it has to be (4) analyzed and synthesized. The goal is to arrange, discuss, and synthesize the literature to lastly (5) establish an agenda of future research. We address

Characteristics	Categories			
Focus	Research outcomes	Research methods	Theories	Applications
Goal	Integration	Criticism		Central issues
Perspective	Neutral representation		Espousal of position	
Coverage	Exhaustive	Exhaustive selective	Representational	Central
Organization	Historical	Conceptual		Methodological
Audience	Specialized scholars	General scholars	Pracitioners	General public

Figure 6: Definition of the Literature Search Scope based on Cooper (1988)

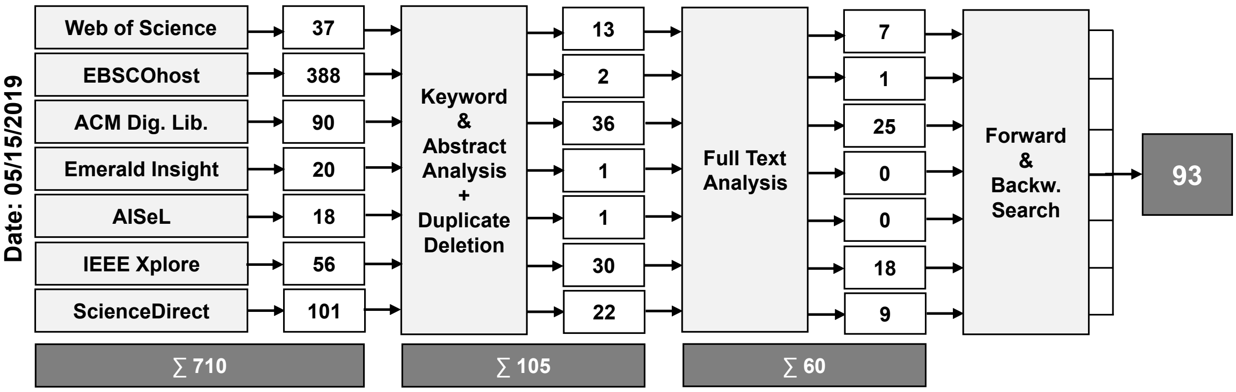

Figure 7: Structure and Result of the Literature Search based on vom Brocke et al. (2009)

the first three phases below. The synthesis is carried out in the subsequent sections (cf. Sect. 4 and 5) and the research agenda within our discussion (cf. Sect. 6).

Definition of Review Scope. Our review scope is based on the Cooper's taxonomy of literature reviews (Cooper 1988) as illustrated in Fig. 6. Our focus is on *research results* and *applications* with the aim to *integrate* the research areas of ML and CEP in the IIoT for smart factories. Thereby, we chose a *neutral representation* with a *representative coverage*. We do not claim our review to be exhaustive as we have not reviewed literature from neighboring fields relevant to the IIoT (such as robotics or industrial engineering) and our initial focus was narrower (central coverage). The results themselves are organized *conceptually* to answer our research questions. We define *specialized researchers* from the fields of ML or CEP as well as *general researchers* from the Information Systems (IS) domain as the target groups.

Conceptualization of Topic. We conceptualized the research topics of interest in our theoretical background (cf. Sect. 2) by defining key terms of ML, that is the three fundamental learning paradigms as well as techniques frequently applied in research. In addition, we presented the foundations of CEP by describing the event-driven architecture with its three main components, the event producer, the event processor, and the event consumer. Furthermore, we conclude that a relevant search result must apply ML techniques in CEP and be transferable to applications in manufacturing processes in smart factories in the IIoT. We consider an approach to be transferable if the application was not limited by the authors or is clearly unrelated.

Literature Search. According to vom Brocke et al. (2009) four sub-steps are necessary for the literature search: journal search, database search, keyword search, and forward and backward search.

To gather high quality, pivotal search results, we restricted the *journal search* initially to A+, A, and B journals of the Business Information Systems ranking of VHB-Jourqual 3 (Verband der Hochschullehrer für Betriebswirtschaft e.V. 2019). Due to the low number of relevant search results, we removed this restriction and opted for representative coverage.

For the *database selection* (see Fig. 7), we selected seven databases from the research areas of computer science and business management. The intention is to ensure that the search takes into account both the technological aspects of ML and CEP through IT-related databases as well as their business application and sub-disciplines relevant to the IIoT.

In the *keyword search*, we selected terminology from CEP combined with the generic term "Machine Learning". For a comprehensive overview of all search strings and filter options in the specific scientific databases, please see Tab. 1.

Initially, we identified 710 potentially relevant contributions. After analyzing title, abstract, keywords, and removing duplicates, we retained 105 relevant publications. The full-text analysis reduced this number to 60. A large number of search results were discovered in IT-related rather than IS-related databases (*IEEE Xplore* and *ACM Digital Library*). The latter publishes contributions from the highly relevant Distributed Event-Based Systems (DEBS) conference. *Backward search* was carried out based on the bibliography of relevant articles, for *forward search* we used Google Scholar citation data. Finally, the phase led to another 33 results which in sum led to a total of 93 contributions. We collected data from all years up to and including 2018 that was indexed by May 2019.

4 Literature Analysis and Synthesis

This section addresses research question RQ1: *Which ML approaches and techniques are used integrated with CEP and how can they be synthesized?* In the following, we present our review by synthesizing the results from the literature search. For an overview of the results, see Fig. 8.

Approach for Synthesis. As recommended by vom Brocke et al. (2009), we followed Webster and Watson (2002) and employed a concept-matrix-based approach for literature analysis and synthesis, which is based on the work of Salipante et al. (1982). Upon completion of the literature search, we subdivided topic-related concepts into units of analysis uncovered in the relevant text corpus. Here, we cross-validated our findings with existing literature reviews in the field of ML (Baumann et al. 2018; Buczak and Guven 2015; Qiu et al. 2016; Wuest et al. 2016), and initially identified the first dimension comprising the learning paradigms, as defined in Sect. 2. After grouping the contributions according to their learning paradigm, we introduced another sub-dimension to describe the specific technique (e.g., classification, clustering) employed in each contribution. Subsequently, we deemed it necessary to identify the type of algorithm used, which comprises the third sub-dimension of our concept matrix. Further explanations were kept in text form, to maintain comprehensibility and conciseness of the concept matrix. Specifically, we have also collected the data type that was used as input. We distinguish between sensor data, process data, network data, and simulation data among other minor data types occurring in this analysis. Finally, to address the application potential in smart factories, we added another dimension to illustrate the reactivity and proactivity of ML applications in CEP, which is discussed in Sect. 5.

Evaluation of Literature. The synthesis uncovers that the learning paradigm *supervised learning* dominates in CEP (n=56). Most of the contributions use *classification* (n=27) algorithms followed by *inductive logic programming* (n=7), *probabilistic method* (n=6), *regression* (n=5) and *supervised clustering* (n=4). In recent years, approaches from *deep learning* (n=3+1) have also been adopted. In addition, there are experiments

Table 1: Search Term per Database

Database	Search Term	#
Web of Science	TS=(("Complex Event Processing" OR "Event-driven" OR "Event Processing" OR "Event-driven Architecture" OR "Event Stream*" OR "Real time Event*" OR "*Real time Event" OR "Complex Event") AND "Machine Learning")	37
EBSCOhost Business Source Premier & Complete	("Complex Event Processing" OR "Event-driven" OR "Event Processing" OR "Event-driven Architecture" OR "Event Stream*" OR "Real time Event*" OR "*Real time Event" OR "Complex Event") AND "Machine Learning"	388
ACM Digital Library	("Complex Event Processing" OR "Event-driven" OR "Event Processing" OR "Event-driven Architecture" OR "Event Stream*" OR "Real time Event*" OR "*Real time Event" OR "Complex Event") AND "Machine Learning"	90
Emerald Insight	("Complex Event Processing" OR "Event-driven" OR "Event Processing" OR "Event-driven Architecture" OR "Event Stream*" OR "Real time Event*" OR "*Real time Event" OR "Complex Event") AND "Machine Learning"	20
AISeL	("Complex Event Processing" OR "Event-driven" OR "Event Processing" OR "Event-driven Architecture" OR "Event Stream*" OR "Real time Event*" OR "*Real time Event" OR "Complex Event") AND "Machine Learning"	18
IEEE Xplore	(("Complex Event Processing" OR "Event-driven" OR "Event Processing" OR "Event-driven Architecture" OR "Event Stream*" OR "Real time Event*" OR "*Real time Event") AND "Machine Learning")	56
ScienceDirect	tak("Complex Event Processing" OR "Event-driven" OR "Event Processing" OR "Event-driven Architecture" OR "Event Stream" OR "Real time Event" OR "Real time Event" OR "Complex Event") AND "Machine Learning"	101

combining techniques from supervised learning (n=1+1) and using *statistical estimators* (n=1).

The learning paradigm *unsupervised learning*, on the other hand, is used less (n=8). *Clustering* is the dominant approach (n=6). We also found one contribution using *dimensionality reduction* and another using *clustering; unsupervised classification*.

More prevalent is the combination of machine learning techniques across approaches (n=18), which we denote as *combined learning*. Most authors (n=12) use a combination of *clustering* (unsupervised learning) and *classification* (supervised learning). Further approaches combining two approaches from different approaches of ML are *clustering* extended by *deep learning* (n=1), *clustering* extended by a *probabilistic method* (n=1), and using a *regression* layer (supervised learning) for a *deep reinforcement learning* model (reinforcement learning) (n=1). Further, we identified combined learning approaches employing three kinds of techniques (for each n=1). These are a mix of *clustering*, *topic mining*, and *probabilistic method*; *clustering*, *topic mining*, and *probabilistic method*; *clustering*, *classification*, and *collaboration filtering*; and *dimensionality reduction*, *clustering*, and *classification*.

Reinforcement learning for CEP has not yet been considered as an approach on its own. It has been used in combination once and, thus, appears to have the potential to be subject for further research.

Further approaches, which could not be classified in our employed classification scheme, are included as *others* (n=11). Examples can be attributed to the domain of data mining (Bhandari 2012) or own developments such as Margara et al. (2013) with their Windows Learner. Since all of these *other* approaches represent unique developments, we found it necessary to exclude them from the classification scheme to reduce category proliferation. See Tab. 5 in the Appendix for an overview of the approaches we categorized as *others*.

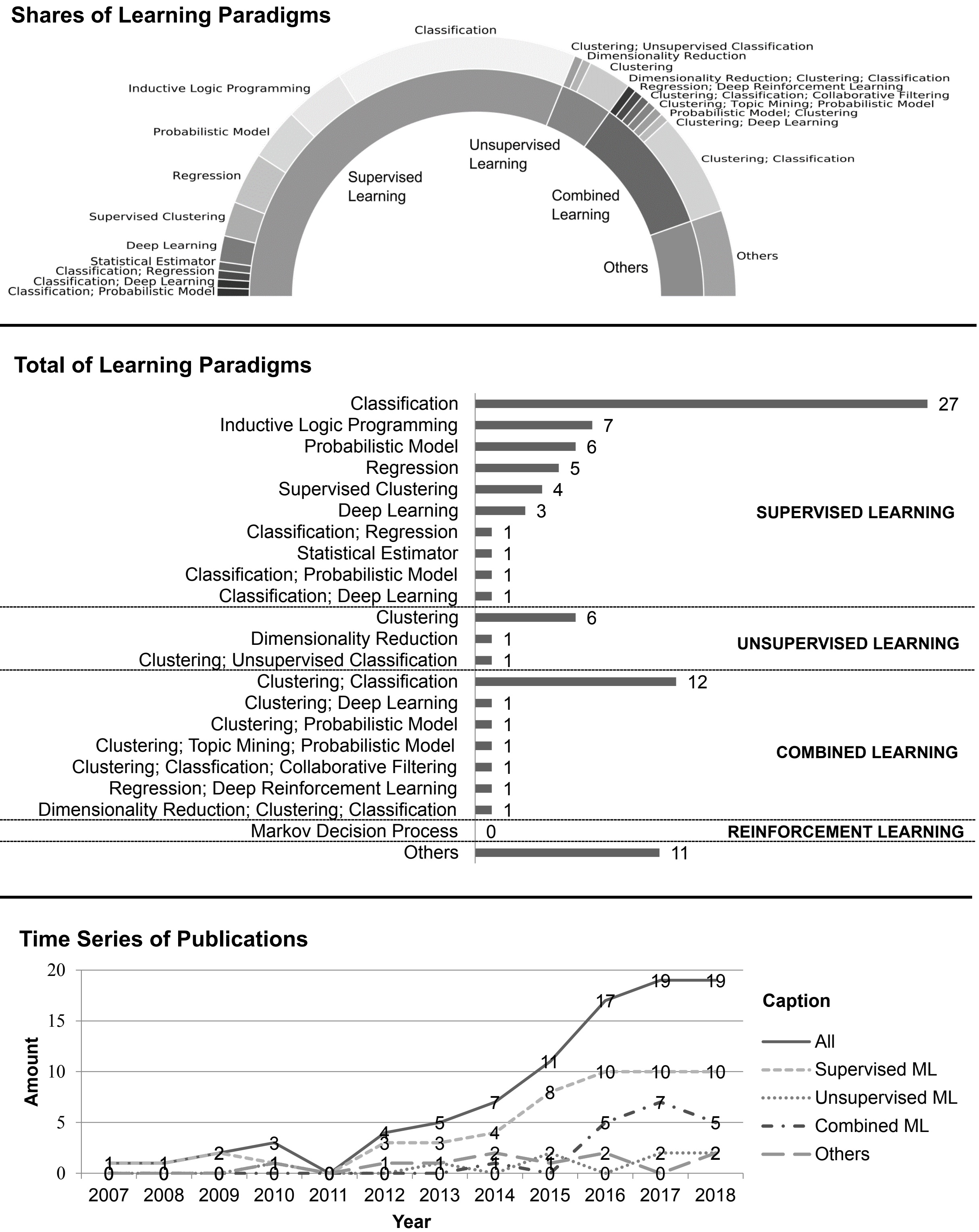

Figure 8: Overview of Literature Search Results

We also identified early approaches harnessing what is call *incremental* or *online learning* to improve the analysis by applying new knowledge not only in batches, but in a continuous fashion (Agarwal et al. 2008; Katzouris et al. 2016; Petersen et al. 2016). Online learning focuses on the automatic generation of the previously manually specified declarative CEP queries to achieve improved event recognition and processing. In contrast, traditional ML analyzes all available data in batches irregularly or periodically.

Development over time. The research area is comparably young with the earliest contribution published in 2007 (Widder et al. 2007). Over the following years, there has been an increase in research interest with peaks in the past two years of 2017 (n=19) and 2018 (n=19). While we included references from 2019 (n=4) in the categories, we excluded them from the development over time graph (cf. Fig. 8). This tendency illustrates the timeliness and relevance of the topic as well as the various possible applications of ML in CEP.

At the level of the learning paradigms, *supervised learning* was consistently of the highest research interest. Initially, the focus was on classification approaches. After 2013, we identified more diverse research efforts. Other approaches, such as regression-based techniques, are increasingly being considered. Nevertheless, classification algorithms remain dominant throughout the years. *Unsupervised learning*, on the other hand, (so far) is underrepresented in research on CEP. The first publication is from 2010. It follows a dimensionality reduction-based approach. From 2013, clustering-based approaches are published sporadically, with a maximum in 2015 and 2017 (with n=2 each). The *combination* of both ML paradigms shows a stronger interest. After only one publication in 2014 and 2015 each, there are five contributions in 2016, seven in 2017, and again five publications in 2018.

Conclusion. In CEP *supervised learning* is used most frequently, which can be explained by the reactive nature in relation to historical event data. In contrast, *unsupervised learning* is used typically in *combined learning* approaches. The combination enables the exploitation of the benefits of both paradigms. It is a fairly young field of application and requires more research to mature. Lastly, there is no substantial use and assessment of *reinforcement learning* approaches in CEP.

Summarizing, the historical analysis reveals that the research efforts in integrating CEP and ML are growing and evolving. They do not only focus on applying a limited and distinct set of techniques. Rather, there is a trend towards combining multiple techniques to generate a tangible added value for the real-time analysis of events.

5 Application Potential in Smart Factories of the Industrial Internet of Things

5.1 Harnessing ML for CEP in Smart Factories

From a business perspective, there are four targets of interest in production plants: improvement in cost, quality, time, and flexibility (Shepherd and Günter 2010). *Time* is a source of competitive advantage and can be considered as a key measure of performance in manufacturing. The objective of *cost* defines the reduction of monetary expenses. Furthermore, it comprises the best possible allocation or combination of manufacturing resources. *Quality* encompasses conformance of products to predefined specifications, based on the production process. *Flexibility* describes the ability to adapt efficiently to new circumstances and requirements of daily manufacturing operations, both in terms of product variety and production volume (Neely et al. 1995).

These targets are directly transferable to smart factories (Kagermann et al. 2013), where cyber-physical systems and related infrastructure enable potential improvements in all four areas of interest (Kagermann et al. 2013).

Based on the analysis and synthesis of the identified literature, we identify both reactive and proactive application potential for smart factories to achieve these targets. In the following, we discuss the distribution of the approaches in the different learning approaches. Subsequently, we

Table 2: Integration of CEP and ML in the Context of Smart Factories

Paradigm	ML technique(s)	Type of Algorithm	Reactive measures (∑=48) ∑	Reactive References*	Proactive measures (∑=47) ∑	Proactive References*
Supervised ML	Classification	Support-vector Machine	4	[1-4]	1	[5]
		Shapelet-based Algorithms	2	[6, 7]	3	[8-10]
		Decision Trees	3	[1, 11, 12]	5	[13-17]
		Artificial Neural Networks	4	[18-21]	2	[15, 22]
		Markov Model	1	[24]	1	[25]
		Discriminant Analysis	3	[18-20]	0	
		Nearest neighbor search	1	[1]	1	[26]
		Naive Bayes Classifier	1	[1]	1	[14]
		Own Development	5	[23, 27-30]	0	
	Probabilistic Method	Bayesian Networks	0		6	[14, 31-35]
		Markov Model	1	[36]	0	
		Conditional Density Estimation	0		1	[37]
	Inductive Logic Programming	Own Development	7	[38-44]	0	
	Regression	Support-vector Regression	0		3	[5, 45, 46]
		Symbolic Regression	0		1	[46]
		Own Development	0		3	[47-49]
	Supervised Clustering	Centroid-based Clustering	0		1	[50]
		Own Development	1	[51]	1	[52]
	Statistical Estimator	Kalman Filter	1	[53]	0	
	Deep Learning	Deep Neural Network	1	[54]	2	[17, 55]
		Convolutional Neural Network	1	[56]	0	
Unsupervised ML	Clustering	Density-based Clustering	2	[57, 58]	0	
		Hierarchical Clustering	1	[59]	0	
		Centroid-based Clustering	4	[60-63]	0	
	Dimensionality Reduction	Principal Component Analysis	1	[64]	0	
	Unsupervised Classification	Own Development	1	[63]	0	
Combined ML	Clustering; *Classification*	Centroid-based Clustering; *Markov Model*	0		8	[65-72]
		Centroid-based Clustering; *Gaussian Mixture Model*	1	[73]	0	
		Density-based Clustering; *Markov Model*	0		1	[74]
		Hierarchical Clustering; Centroid-based Clustering; Support-vector Machine; *Decision Trees*; *Naive Bayes Classifier*	0		1	[75]
		Unspecified Clustering Algorithm; *Naive Bayes Classifier*	1	[76]	0	
	Clustering; *Probabilistic Method*; Topic Mining	Density-based Clustering; *Markov Model*; Bag-of-words	0		1	[77]
	Clustering; *Classification*; Collaborative Filtering	Centroid-based Clustering; *Naive Bayes Classifier*; Matrix Factorization	0		1	[78]
	Clustering; *Probabilistic Method*	Density-based Clustering; *Bayesian Inference*	0		1	[79]
	Regression; *Deep Reinforcement Learning*	Logistic Regression; *Deep Belief Network*; *Markov Decision Process*	0		1	[80]
	Clustering; *Dimensionality Reduction*; Classification	Centroid-based Clustering; *Space-filling Curve*; Artificial Neural Network	0		1	[81]
	Clustering; *Deep Learning*	Centroid-based Clustering; *Decision Tree & Recurrent Neural Networks*	1	[82]	0	

* Further information to the papers referenced can be found in the Appendix: Conceptual Matrix Table 4 | Table 5.

illustrate both, *reactive* and *proactive* approaches, using exemplary use cases. In sum, this section sheds light on RQ2: *What potential can be derived from combining CEP and ML for smart factories in the IIoT?*.

5.2 Reactive and Proactive Real-time Data Analytics in Smart Factories

As visualized in Tab. 2, research efforts are generally balanced between the two paradigms. However, specific learning approaches and techniques show trends. Slightly more contributions classified as supervised learning are reactive, specifically regression is predominantly proactive. Unsupervised learning, on the other hand, is reactive without exception, which may indicate an unsuitability for proactive approaches using CEP. Contributions combining the two approaches are mostly proactive. Lastly, we also identified contributions, which we could not place in our classification scheme, as they represent unique cases or do not mention classifiable ML techniques. Further details are available in Tab. 5 in the Appendix.

Reactive. CEP is declarative by nature. A domain expert must define rules and patterns in advance, which involves a high amount of manual work, in-depth domain knowledge, and high technical expertise (Akbar et al. 2015a; Turchin et al. 2009). In our literature review, we found, that by harnessing ML, these tasks are at least partially automatable. Specifically, reactive measures improve CEP by enabling a higher degree of descriptive and diagnostic analytical power.

In complex environments, not all event occurrences, which feed low-level event data into the CEP system, are known in advance and, thus, unambiguously processable by manually specified rules. Naturally, unspecified events will occur and cannot be taken into consideration. Against this backdrop, research on ML proposes to circumvent these downsides of CEP. Depending on the analytical task, ML technique can automatically learn from the data and derive correlations, anomalies, or patterns in the event stream. Later, the generated knowledge can be fed into the CEP system as new rules or patterns to identify novel future events. Researchers coined this automated rule or pattern learning (Lee and Jung 2017; Margara et al. 2013; Metz et al. 2012; Petersen et al. 2017).

In the following, we present the approach proposed by Metz et al. (2012), to illustrate a reactive application of ML in CEP. As depicted in Fig. 9, the approach is positioned in the manufacturing domain. Production resources on the shop floor generate low-level event data (e.g., about process, quality, performance), which is collected by a data collection engine. Subsequently, the data is aggregated with data from enterprise applications (e.g., product data), which results in enriched complex events. Next, the CEP engine analyzes the incoming event streams and matches it with predefined patterns. If a critical situation occurs, employees are notified. Up until this step, traditional CEP operations take place. However, there are instances in which novel situations occur that would also require a reaction from employees on the shop floor. In this case, the CEP system initiates a process in a so-called rule induction manager, which improves the rule base and refines defined event patterns automatically. First, data is transferred from the process database, which is restricted to a sample size consisting of related data (e.g., timespan, specific machine or product). Second, a suitable algorithm is selected, based on predefined selection and parameterization criteria (e.g., rule accuracy, rule coverage). Third, rules are generated by the selected algorithm. Finally, the rules are added to the rule store to supplement the existing rule base and can now be identified if the pattern occurs.

Proactive. By default, future events are not considered in CEP (Christ et al. 2016). However, the perceived value of a CEP system would increase through predictive analytics by enabling proactive reactions to events (Schwegmann et al. 2013). By extending the system with adequate ML techniques, forecasting and predictive power are achievable in CEP.

To demonstrate proactiveness in CEP, we present the scenario of Christ et al. (2016) who propose a generic architecture to integrate predictive analytics within a CEP system (cf. 9). Therefore,

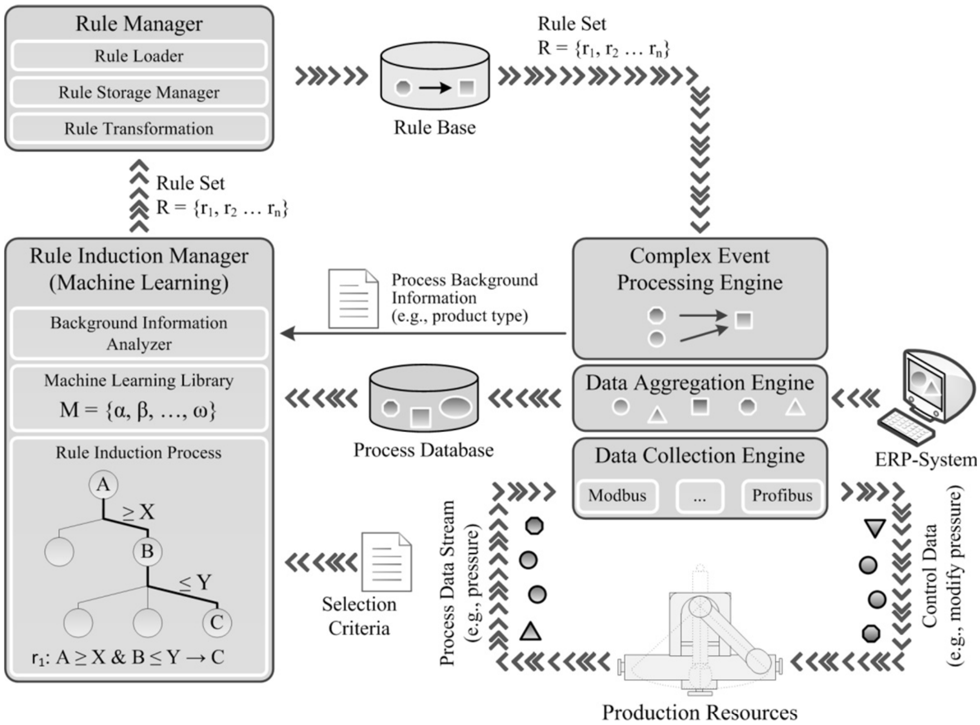

Figure 9: Scenario for Reactive ML and CEP (Metz et al. 2012)

the formerly reactive system is advanced by a ML-based proactive component with an evaluation and learning loop. This requires three connections between the predictive analytics component and the CEP engine: First, (a) (complex) events from the global event source are transferred to the predictive analytics component, serving as a basis for training and prediction of the predictive analytics model. The input is extended by a (b) prediction trigger to initiate the calculation of event predictions. Further, the predictive analytics component (c) calculates a Conditional Event Occurrence Density Estimation (CEODE) as output. This probability density function is fed back into the CEP engine as another incoming event. Inside the CEP system itself, despite the higher power, CEODE does not work for all types of events. Thus, predictive EPAs do not replace the existing reactive EPAs. Both are (d) loosely coupled with one another and complement each other. While the reactive EPAs are implemented in the traditional way, the dedicated predictive EPAs require special (e) proactive event processing rules. This creates higher-quality predefined (complex) events when certain patterns occur. Further, if a certain probability threshold is exceeded a (f) proactive action is triggered. After triggering, the predictive analytics component calculates (h) predictions on basis of the likelihood of event occurrences and transfers the CEODE function back into the event stream, which acts as a novel event source to which the predictive EPA listens. The (c) calculated CEODE functions are passed as events themselves. Closing the loop, the historical event data stream is used in the predictive analytics component for (g) learning, to (re)train the predictive analytics model.

5.3 Synthesis of Potential Solutions Using CEP and ML

In the following, we highlight further exemplary application scenarios of combining CEP and ML, specifically in smart factories. The section is structured according to reactive and proactive potentials as identified previously (cf. Tab. 2).

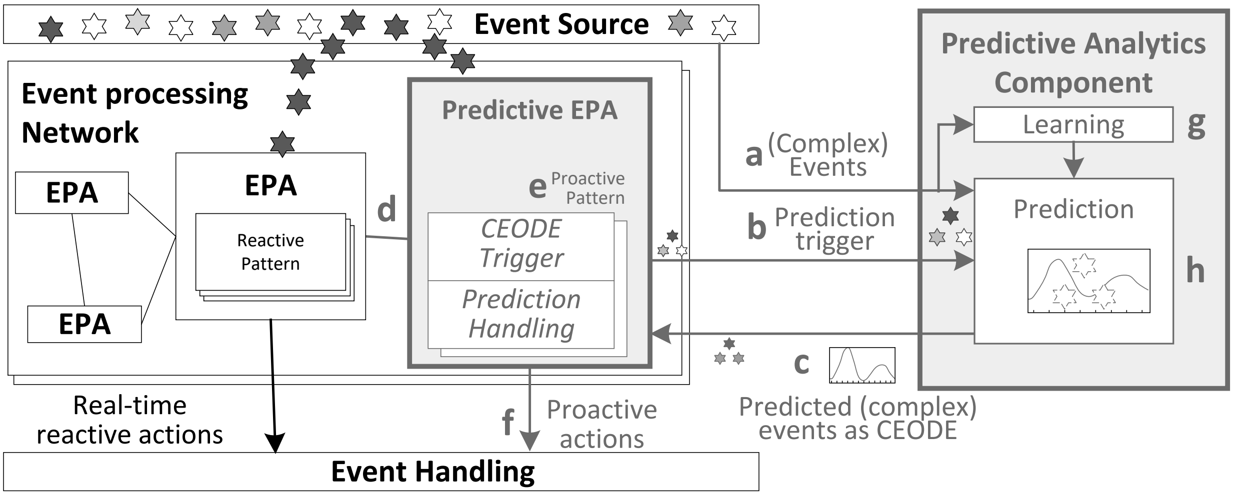

Figure 10: Scenario for Proactive ML and CEP (Christ et al. 2016)

Reactive measures. Reactive measures of CEP integrated with ML should enable more efficient processes concerning the management of faults in manufacturing. The objective is to create transparency and uncover information regarding fault localization and subsequent correction. When first implementing CEP, typically not all possible manufacturing incidents are known. Thus, a continued development of patterns for recognition is essential. Ideally, previously unknown event patterns can be detected automatically and included for future processing (Petersen et al. 2017; Widder et al. 2007).

As described above, Metz et al. (2012) offers a specific application for smart factories. Production data in combination with process data is used to train an ML algorithm to classify new rules via decision trees, which are subsequently used in CEP. Further approaches use camera systems and image processing sensors for smart manufacturing (Christ et al. 2016). These also provide valuable information for smart factories to achieve more precise pattern recognition and adaptation. For example, Barsocchi et al. (2018) employs deep learning (specifically a convolutional neural network) to automatically learn rules of device use and access control in smart environments. Abdallah et al. (2018) also use deep learning to learn and adapt security and surveillance policies automatically based on camera footage (e.g., unauthorized intrusion).

Furthermore, sensor networks are used in combination with classification algorithms (Shahad et al. 2016). Anomalous behavior is detected by analyzing events from pressure, vibration, temperature, and proximity sensors (Gungor and Hancke 2009). Approaches from other research areas also show the potential for problem solving. Some of the authors rely on video surveillance records to learn activity or event patterns (Katzouris et al. 2015; Skarlatidis et al. 2015). Typically, this is achieved with inductive logic programming. In contrast, Skarlatidis et al. (2015) employ a Markov model. The application takes place outside smart factories though but is potentially transferable.

Other authors use accelerometer-based event streams to detect motion patterns of people automatically (Mehdiyev et al. 2016; Preuveneers et al. 2016). For example, Mehdiyev et al. (2016) use the Fuzzy Unordered Rule Induction Algorithm (FURIA) for classification. Preuveneers et al. (2016) use a combination of clustering and classification based on naive Bayes and an unspecified clustering approach. In production-specific scenarios, this offers the possibility of monitoring and controlling industrial robot arms (Neto et al.

2009). A manual implementation of the motion profiles as events in a CEP system would require a great deal of time and technical expertise yielding a cost increase.

Further possible application scenarios for ML and CEP, relate to increasing the sustainability and energy efficiency of smart manufacturing operations. Ta-Shma et al. (2018), for example, employ an unsupervised *k*-means clustering approach for energy management of smart devices interconnected through the Internet of Things.

Another application domain is network security. The use of cyber-physical systems opens up new cyber-attack areas that can potentially result in operational disruption or production downtime (Mitchell and Chen 2014). To prevent resulting delays and costs (Bhandari 2012; Turchin et al. 2009), applications rely on CEP-based intrusion detection systems, which are extended by ML components. Turchin et al. (2009) use discrete Kalman filters, whereas Bhandari (2012) use a combination of Markov logic and data mining approaches. In both cases, the dynamic and adaptive adaptation of the attack patterns is possible without the need for manual intervention and recurring, time-consuming refinement.

The measures outlined above will primarily eliminate manual monitoring activities and the implementation of declarative rules in the CEP system. This results in improved reactivity in production facilities and saves time and cost.

Proactive measures. In the following, we outline proactive scenarios and measures of integrating ML and CEP in smart manufacturing.

Christ et al. (2016) propose an integration of ML and CEP based on conditional density estimation, which receives event data from camera systems and sensors. The production operations are evaluated in real-time for abnormal changes and compared with probabilities of occurrence. The corresponding analytical task is also the central theme of the ACM DEBS 2017 Grand Challenge (Rivetti et al. 2017). Here, events are to be grouped based on discrete states (e.g., temperature or pressure of the production machine), for preprocessing, employing a clustering algorithm. Subsequently, a Markov model was trained in real-time to compute the probability of occurring anomalies. For all solutions, state transitions between the clusters were used. Finally, the system notifies an employee when the probability exceeds a predefined threshold value.

Mousheimish et al. (2017) employ a shapelet-based technique and combine automatic rule-learning with predictive analytics. Engel and Etzion (2011) propose a proactive architectural standard in industrial environments. By combining Bayesian networks with decision tree classification, the system tackles scheduled maintenance processes without considering the condition of the machine. The system supports spare parts management by indicating the actual demand of parts depending on the remaining useful lifetime of the machine. Subsequently, the maintenance intervals can be optimized.

Further applications are situated in logistics, such as material handling at the production site, by monitoring conveyor belts (Beamon 1999) as well as external material movements such as road traffic (Akbar et al. 2015b, 2018; Y. Wang et al. 2018). In the latter, event data is continuously recorded and transmitted via sensors, camera systems, or induction loops and is used to predict traffic congestion among other examples. Akbar et al. (2015) rely on Adaptive Moving Window Regression (AMWR) to ensure continuous learning of the CEP system. Y. Wang et al. (2018) use Bayesian nets to predict probabilistic complex events. The ACM DEBS 2018 Grand Challenge was also situated in the logistics domain. More specifically the goal was to predict arrival ports and destination times of shipping vessels. These insights are transferable to a smart factory setting to optimize just in time and just-in-sequence operations. The contestants employed various ML techniques to tackle the challenge. Nguyen et al. (2018), for example, combine random forests with a recurrent neural network in a supervised fashion for prediction. Kammoun et al. (2018), on the other hand, apply dimensionality reduction, clustering, and an artificial neural network for classification, which constitutes a combined learning approach.

The approaches described above show that proactivity primarily leads to the optimization of the flexibility of manufacturing, maintenance, and goods flow. Predictive power in smart factories also results in cost reduction by enabling proactive maintenance processes.

6 Discussion and Research Agenda

Our study shows that the use of ML has vast and diverse potential for real-time data analytics based on CEP. Dayarathna and Perera (2018) confirm this and describe the use of ML as a significant advancement in real-time event processing. However, despite the importance of the research domain for the IIoT, a thorough synthesis of the various techniques and their combinations has not been undertaken so far.

Based on our literature review, we identified that supervised learning techniques are dominant in CEP so far. Recently, there has been a growing research interest in combining supervised learning and unsupervised learning approaches. Reaping the benefits of both approaches carries significant potential for implementing adaptable and proactive systems. Reinforcement learning and other alternatives, on the other hand, have hardly been investigated to this point. Future research could start here to investigate the effectiveness of reward-based approaches for CEP as reinforcement learning is applied effectively already in multi-agent systems. Since event processing agents are regarded conceptually as autonomous EPA of a more comprehensive EPN, it makes sense to investigate the effectiveness of this approach for real-time data analytics in the IIoT as well.

In a second synthesis, we explored application potentials of ML in CEP for smart factories in the IIoT. We identified both (1) reactive and (2) proactive approaches, which help to achieve economic target objectives. We found that combining ML with CEP mainly addresses the reduction of time and costs (i.e., in terms of automation). Previous research was targeted primarily at reactive measures, although in the last two years in particular, we identified more research efforts in proactivity.

As identified in our synthesis, supervised learning is mainly employed for reactive measures in smart factories. Here particularly SVM and artificial neural networks (as classification techniques) are used. Unsupervised learning, on the other hand, is underrepresented, but initial research has been carried out in the domain of CEP. However, it is only applied in reactive applications, which may suggest a lack of suitability for proactive applications. Combined learning is predominantly proactive. In particular, a combination of *k*-means and Markov models seems to be promising. Lastly, in supervised learning probabilistic models and regression are employed mainly in proactive contributions.

Based to the exploration and discussion of our literature review, we can derive the following issues and future potential regarding the use of ML in CEP. Following our research method, these issues and potential correspond to a developing research agenda (RA) as proposed by vom Brocke et al. (2009):

RA1: Further optimization and adaptation of ML techniques to the dynamic nature of event-driven real-time data analytics. Despite the expanding integration of CEP and ML to address real-time data analytics problems in smart factories, there is a lack of further possibilities and of practical evaluation to prove their suitability. An example is the use of regression approaches from the area of supervised learning for reactive measures that has not yet been evaluated (cf. Tab. 2).

RA2: Integration of both reactive and proactive applications in a single CEP system to reap the benefits of both and harness synergies. So far, there is no effort to combine ML approaches from the reactive and the proactive domain. All previous publications are either focusing on reactive or proactive systems (cf. Tab. 2). However, developments in recent years show that there are already combinations of different ML techniques, but so far no approach combines both reactive and proactive approaches. Future research could address this gap, that is by automated rule learning with the prediction of future events. Mousheimish et al. (2017) propose a first step in this direction by first learning predictive rules or patterns.

RA3: Dynamic adaptation of the CEP systems knowledge base to (external) circumstances and changes fueled by online learning approaches. In our synthesis, we also identified first approaches harnessing *online learning* (Katzouris et al. 2016; Petersen et al. 2016), where data becomes available in sequential order and is used to update the best predictor for future data at each step (Agarwal et al. 2008). Due to CEP's real-time focus, employing ML in an online fashion could potentially achieve further synergies such as more flexible EPA but will necessitate further research efforts.

RA4: Application of reinforcement learning to reap prescriptive power for CEP systems. The analytical approaches we identified are of descriptive, diagnostic, or predictive nature. Prescriptive analytics builds on top of this. It expands the other levels by recommending concrete measures or alternative courses of action to choose the best action alternative (Delen and Zolbanin 2018). Thus, it could be of great value for CEP systems. Traditionally this is performed by optimization techniques. But recently, researchers have been applying reinforcement learning for prescriptive problems (Qu et al. 2016; X. Wang et al. 2016). Compared to other learning paradigms, reinforcement learning is exploratory and thus, explicitly generates new knowledge (Ishii et al. 2002). In contrast, supervised and unsupervised learning, are partly exploitative. They analyze already existing knowledge to derive insights or transfer it to new data inputs. Therefore, reinforcement learning is increasingly being used for prescriptive purposes (Shroff et al. 2014).

7 Conclusion and Outlook

Several applications of integrating ML for the continuous real-time analysis of events in CEP are currently being examined in research and practice. In the area of smart factories as a key application field of event processing in the IIoT, first implementations are employed in reactive as well as proactive scenarios. Examples range from information-supported repair due to machine faults or warnings about a low remaining useful lifetime of machinery parts. The integration of CEP and ML is also regarded as promising concerning big data analytics (Flouris et al. 2017), in particular regarding high velocity date. Further, we can observe a trend towards probabilistic or uncertain event models. In contrast to deterministic event models, in these models not all attributes are known or accurate (Flouris et al. 2017). However, first approaches using Bayesian and Markov networks have only been applied with moderate success (Alevizos et al. 2017).

As pointed out in our research agenda, future work could start here and lead to further improvements through new approaches such as online learning (Petersen et al. 2016) and the combination of reactive and proactive approaches. The integration of CEP with reinforcement learning is also promising. As presented in the smart factory scenarios, machine uptime and the associated maintenance processes can be optimized as well.

Finally, using prescriptive analytics, the machine should not be regarded as an individual component. The IIoT-based cyber-physical system, which it is embedded in, should be considered as a whole for the best business decision in a corporate context (G. Wang et al. 2016).

References

Abdallah H. B., Abdellatif T., Chekir F. (2018) AMSEP: Automated Multi-level Security Management for Multimedia Event Processing. In: Procedia Computer Science 134, pp. 452–457

Agarwal S., Saradhi V. V., Karnick H. (2008) Kernel-based online machine learning and support vector reduction. In: Neurocomputing 71(7-9), pp. 1230–1237

Akbar A., Carrez F., Moessner K., Sancho J., Rico J. (2015a) Context-aware stream processing for distributed IoT applications. In: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). IEEE, pp. 663–668

Akbar A., Carrez F., Moessner K., Zoha A. (2015b) Predicting complex events for pro-active IoT applications. In: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). IEEE, pp. 327–332

Akbar A., Kousiouris G., Pervaiz H., Sancho J., Ta-Shma P., Carrez F., Moessner K. (2018) Real-time probabilistic data fusion for large-scale IoT applications. In: IEEE Access 6, pp. 10015–10027

Alevizos E., Skarlatidis A., Artikis A., Paliouras G. (2017) Probabilistic complex event recognition: A survey. In: ACM Computing Surveys (CSUR) 50(5), pp. 1–30

Alpaydin E. (2004) Introduction to machine learning. MIT press

Amin A. (2003) Recognition of hand-printed characters based on structural description and inductive logic programming. In: Pattern recognition letters 24(16), pp. 3187–3196

Apache Software Foundation (2019) Apache Kafka: A Distributed Streaming Platform https://kafka.apache.org/ Last Access: 2019-05-05

Argyriou A., Evgeniou T., Pontil M. (2008) Convex multi-task feature learning. In: Machine Learning 73(3), pp. 243–272

Artikis A., Paliouras G., Portet F., Skarlatidis A. (2010) Logic-based representation, reasoning and machine learning for event recognition. In: Proceedings of the 4th ACM International Conference on Distributed Event-Based Systems. ACM, pp. 282–293

Bacciu D., Lisboa P. J., Sperduti A., Villmann T. (2015) Probabilistic Modeling in Machine Learning. In: Springer Handbook of Computational Intelligence. Springer, pp. 545–575

Barsocchi P., Calabrò A., Ferro E., Gennaro C., Marchetti E., Vairo C. (2018) Boosting a Low-Cost Smart Home Environment with Usage and Access Control Rules. In: Sensors 18(6), pp. 1–22

Bauernhansl T. (2017) Die vierte industrielle Revolution–Der Weg in ein wertschaffendes Produktionsparadigma. In: Handbuch Industrie 4.0. Springer, pp. 1–31

Baumann F. W., Sekulla A., Hassler M., Himpel B., Pfeil M. (2018) Trends of machine learning in additive manufacturing. In: International Journal of Rapid Manufacturing 7(4), pp. 310–336

Beamon B. M. (1999) System reliability and congestion in a material handling system. In: Computers & industrial engineering 36(3), pp. 673–684

Bhandari S. (2012) An adaptive event stream processing environment. In: Proceedings of the on SIGMOD/PODS 2012 PhD Symposium. ACM, pp. 57–62

Bruns R., Dunkel J. (2015) Complex event processing: komplexe Analyse von massiven Datenströmen mit CEP. Springer

Buczak A. L., Guven E. (2015) A survey of data mining and machine learning methods for cyber security intrusion detection. In: IEEE Communications Surveys & Tutorials 18(2), pp. 1153–1176

Christ M., Krumeich J., Kempa-Liehr A. W. (2016) Integrating predictive analytics into complex event processing by using conditional density estimations. In: 2016 IEEE 20th International Enterprise Distributed Object Computing Workshop (EDOCW). IEEE, pp. 1–8

Comar P. M., Liu L., Saha S., Tan P.-N., Nucci A. (2013) Combining supervised and unsupervised learning for zero-day malware detection. In: 2013 Proceedings IEEE INFOCOM. IEEE, pp. 2022–2030

Cooper H. M. (1988) Organizing knowledge syntheses: A taxonomy of literature reviews. In: Knowledge in society 1(1), pp. 104–126

Dayarathna M., Perera S. (2018) Recent advancements in event processing. In: ACM Computing Surveys (CSUR) 51(2), pp. 1–36

De Raedt L., Kersting K. (2008) Probabilistic inductive logic programming. In: Probabilistic Inductive Logic Programming. Springer, pp. 1–27

Delen D., Zolbanin H. M. (2018) The analytics paradigm in business research. In: Journal of Business Research 90, pp. 186–195

Deng L., Yu D. (2014) Deep learning: methods and applications. In: Foundations and Trends in Signal Processing 7(3–4), pp. 197–387

Drucker H., Burges C. J., Kaufman L., Smola A. J., Vapnik V. (1997) Support vector regression machines. In: Advances in neural information processing systems, pp. 155–161

Engel Y., Etzion O. (2011) Towards proactive event-driven computing. In: Proceedings of the 5th ACM international conference on Distributed event-based system. ACM, pp. 125–136

Etzion O., Niblett P., Luckham D. C. (2011) Event processing in action. Manning Greenwich

Flouris I., Giatrakos N., Deligiannakis A., Garofalakis M., Kamp M., Mock M. (2017) Issues in complex event processing: Status and prospects in the big data era. In: Journal of Systems and Software 127, pp. 217–236

Gelb A. (1974) Applied optimal estimation. MIT press

Ghahramani Z. (2003) Unsupervised learning. In: Summer School on Machine Learning. Springer, pp. 72–112

Ghahramani Z. (2015) Probabilistic machine learning and artificial intelligence. In: Nature 521(7553), pp. 452–459

Gungor V. C., Hancke G. P. (2009) Industrial wireless sensor networks: Challenges, design principles, and technical approaches. In: IEEE Transactions on industrial electronics 56(10), pp. 4258–4265

Howley T., Madden M. G., O'Connell M.-L., Ryder A. G. (2005) The effect of principal component analysis on machine learning accuracy with high dimensional spectral data. In: International Conference on Innovative Techniques and Applications of Artificial Intelligence. Springer, pp. 209–222

Ishii S., Yoshida W., Yoshimoto J. (2002) Control of exploitation–exploration meta-parameter in reinforcement learning. In: Neural networks 15(4-6), pp. 665–687

Janiesch C., Diebold J. (2016) Conceptual Modeling of Event Processing Networks. In: 24th European Conference on Information Systems (ECIS). AIS, pp. 1–15

Janiesch C., Fischer M., Winkelmann A., Nentwoch V. (2019) Specifying Autonomy in the Internet of Things: The Autonomy Model and Notation. In: Information Systems and e-Business Management 17(1), pp. 159–194

Janiesch C., Matzner M., Müller O. (2012) Beyond process monitoring: a proof-of-concept of event-driven business activity management. In: Business Process Management Journal 18(4), pp. 625–643

Kagermann H., Wahlster W., Helbig J. (2013) Recommendations for implementing the strategic initiative INDUSTRIE 4.0. acatech - National Academy of Science and Engineering

Kalman R. E. (1960) A new approach to linear filtering and prediction problems. In: Journal of basic Engineering 82(1), pp. 35–45

Kammoun A., Raynaud T., Gillani S., Singh K., Fayolle J., Laforest F. (2018) A scalable framework for accelerating situation prediction over spatio-temporal event streams. In: Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems. ACM, pp. 221–223

Katzouris N., Artikis A., Paliouras G. (2015) Incremental learning of event definitions with inductive logic programming. In: Machine Learning 100(2-3), pp. 555–585

Katzouris N., Artikis A., Paliouras G. (2016) Online learning of event definitions. In: Theory and Practice of Logic Programming 16(5-6), pp. 817–833

Kotsiantis S. B., Zaharakis I., Pintelas P. (2007) Supervised machine learning: A review of classification techniques. In: Emerging artificial intelligence applications in computer engineering 160, pp. 3–24

Kubat M. (2017) An introduction to machine learning. Springer

Lee D. D., Seung H. S. (2001) Algorithms for non-negative matrix factorization. In: Advances in neural information processing systems, pp. 556–562

Lee O.-J., Jung J. E. (2017) Sequence clustering-based automated rule generation for adaptive complex event processing. In: Future Generation Computer Systems 66, pp. 100–109

Luckham D. (2002) The Power of Events. Addison-Wesley Reading

Luckham D. (2011) Event processing for business: organizing the real-time enterprise. John Wiley & Sons

Ludvig E. A., Bellemare M. G., Pearson K. G. (2011) A primer on reinforcement learning in the brain: Psychological, computational, and neural perspectives. In: Computational neuroscience for advancing artificial intelligence: Models, methods and applications. IGI Global, pp. 111–144

Margara A., Cugola G., Tamburrelli G. (2013) Towards automated rule learning for complex event processing. Technical Report

Marsland S. (2011) Machine learning: an algorithmic perspective. Chapman and Hall/CRC

Mehdiyev N., Krumeich J., Werth D., Loos P. (2015) Sensor event mining with hybrid ensemble learning and evolutionary feature subset selection model. In: 2015 IEEE International Conference on Big Data (Big Data). IEEE, pp. 2159–2168

Mehdiyev N., Krumeich J., Werth D., Loos P. (2016) Determination of event patterns for complex event processing using fuzzy unordered rule induction algorithm with multi-objective evolutionary feature subset selection. In: 2016 49th Hawaii International Conference on System Sciences (HICSS). IEEE, pp. 1719–1728

Metz D., Karadgi S., Müller U., Grauer M. (2012) Self-learning monitoring and control of manufacturing processes based on rule induction and event processing. In: 4th International Conference on Information, Process, and Knowledge Management (eKNOW 2012), pp. 88–92

Mitchell R., Chen I.-R. (2014) A survey of intrusion detection techniques for cyber-physical systems. In: ACM Computing Surveys (CSUR) 46(4), pp. 1–29

Mousheimish R., Taher Y., Zeitouni K. (2016) Automatic learning of predictive rules for complex event processing: Doctoral symposium. In: Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems. ACM, pp. 414–417

Mousheimish R., Taher Y., Zeitouni K. (2017) Automatic learning of predictive CEPules: bridging the gap between data mining and complex event processing. In: Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems. ACM, pp. 158–169

Muggleton S. (1991) Inductive logic programming. In: New Generation Computing 8(4), pp. 295–318

Mutschler C., Philippsen M. (2012) Learning event detection rules with noise hidden markov models. In: 2012 NASA/ESA Conference on Adaptive Hardware and Systems (AHS). IEEE, pp. 159–166

Neely A., Gregory M., Platts K. (1995) Performance measurement system design: a literature review and research agenda. In: International Journal of Operations & Production Management 15(4), pp. 80–116

Neto P., Pires J. N., Moreira A. P. (2009) Accelerometer-based control of an industrial robotic arm. In: RO-MAN 2009-The 18th IEEE International Symposium on Robot and Human Interactive Communication. IEEE, pp. 1192–1197

Nguyen D.-D., Le Van C., Ali M. I. (2018) Vessel destination and arrival time prediction with sequence-to-sequence models over spatial grid. In: Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems. ACM, pp. 217–220

Olsson L., Janiesch C. (2015) Real-time Business Intelligence und Action Distance: Ein konzeptionelles Framework zur Auswahl von BI-Software. In: 12. Internationale Tagung Wirtschaftsinformatik (WI), pp. 691–705

Pang S., Ozawa S., Kasabov N. (2005) Incremental linear discriminant analysis for classification of data streams. In: IEEE transactions on Systems, Man, and Cybernetics, part B (Cybernetics) 35(5), pp. 905–914

Petersen E., To M. A., Maag S. (2016) An online learning based approach for CEP rule generation. In: 2016 8th IEEE Latin-American Conference on Communications (LATINCOM). IEEE, pp. 1–6

Petersen E., To M. A., Maag S. (2017) A novel online CEP learning engine for MANET IDS. In: 2017 IEEE 9th Latin-American Conference on Communications (LATINCOM). IEEE, pp. 1–6

Phyu T. N. (2009) Survey of classification techniques in data mining. In: Proceedings of the 1st International MultiConference of Engineers and Computer Scientists, pp. 18–20

Preuveneers D., Berbers Y., Joosen W. (2016) SAMURAI: A batch and streaming context architecture for large-scale intelligent applications and environments. In: Journal of Ambient Intelligence and Smart Environments 8(1), pp. 63–78

Qiu J., Wu Q., Ding G., Xu Y., Feng S. (2016) A survey of machine learning for big data processing. In: EURASIP Journal on Advances in Signal Processing 2016(1), pp. 1–16

Qu S., Wang J., Govil S., Leckie J. O. (2016) Optimized adaptive scheduling of a manufacturing process system with multi-skill workforce and multiple machine types: an ontology-based, Multi-agent reinforcement learning approach. In: Procedia CIRP 57, pp. 55–60

Rivetti N., Busnel Y., Gal A. (2017) Flinkman: Anomaly detection in manufacturing equipment with apache flink: Grand challenge. In: Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems. ACM, pp. 274–279

Salipante P., Notz W., Bigelow J. (1982) A matrix approach to literature reviews. In: Research in organizational behavior 4, pp. 321–348

Samuel A. L. (1959) Some studies in machine learning using the game of checkers. In: IBM Journal of research and development 3(3), pp. 210–229

Schwegmann B., Matzner M., Janiesch C. (2013) preCEP: Facilitating predictive event-driven process analytics. In: International Conference on Design Science Research in Information Systems. Springer, pp. 448–455

Shahad R. A., Bein L. G., Saad M. H. M., Hussain A. (2016) Complex event detection in an intelligent surveillance system using CAISER platform. In: 2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES). IEEE, pp. 129–133

Shalev-Shwartz S., Ben-David S. (2014) Understanding machine learning: From theory to algorithms. Cambridge University Press

Shepherd C., Günter H. (2010) Measuring supply chain performance: current research and future directions. In: Behavioral Operations in Planning and Scheduling. Springer, pp. 105–121

Ta-Shma P., Akbar A., Gerson-Golan G., Hadash G., Carrez F., Moessner K. (2018) An ingestion and analytics architecture for iot applied to smart city use cases. In: IEEE Internet of Things Journal 5(2), pp. 765–774

Shroff G., Agarwal P., Singh K., Kazmi A. H., Shah S., Sardeshmukh A. (2014) Prescriptive information fusion. In: 17th International Conference on Information Fusion (FUSION). IEEE, pp. 1–8

Skarlatidis A., Paliouras G., Artikis A., Vouros G. A. (2015) Probabilistic event calculus for event recognition. In: ACM Transactions on Computational Logic (TOCL) 16(2), pp. 1–35

Srivastava N., Salakhutdinov R. R. (2012) Multimodal learning with deep boltzmann machines. In: Advances in neural information processing systems, pp. 2222–2230

Sutton R. S., Barto A. G. (2018) Reinforcement learning: An introduction. MIT press

Tokic M. (2013) Reinforcement Learning: Psychologische und neurobiologische Aspekte. In: KI-Künstliche Intelligenz 27(3), pp. 213–219

Turchin Y., Gal A., Wasserkrug S. (2009) Tuning complex event processing rules using the prediction-correction paradigm. In: Proceedings of the Third ACM International Conference on Distributed Event-Based Systems. ACM, pp. 1–12

Verband der Hochschullehrer für Betriebswirtschaft e.V. (2019) Teilrating Wirtschaftsinformatik https://vhbonline.org/vhb4you/jourqual/vhb-jourqual-3/teilrating-wi/ Last Access: 2019-05-05

vom Brocke J., Simons A., Niehaves B., Riemer K., Plattfaut R., Cleven A. (2009) Reconstructing the giant: on the importance of rigour in documenting the literature search process. In: Proceedings of the 17th European Conference on Information Systems, pp. 2206–2217

Wang G., Gunasekaran A., Ngai E. W., Papadopoulos T. (2016) Big data analytics in logistics and supply chain management: Certain investigations for research and applications. In: International Journal of Production Economics 176, pp. 98–110

Wang S., Chaovalitwongse W., Babuska R. (2012) Machine learning algorithms in bipedal robot control. In: IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42(5), pp. 728–743

Wang X., Wang H., Qi C. (2016) Multi-agent reinforcement learning based maintenance policy for a resource constrained flow line system. In: Journal of Intelligent Manufacturing 27(2), pp. 325–333

Wang Y., Gao H., Chen G. (2018) Predictive complex event processing based on evolving Bayesian networks. In: Pattern Recognition Letters 105, pp. 207–216

Watt J., Borhani R., Katsaggelos A. K. (2016) Machine learning refined: foundations, algorithms, and applications. Cambridge University Press

Webster J., Watson R. T. (2002) Analyzing the past to prepare for the future: Writing a literature review. In: MIS Quarterly, pp. xiii–xxiii

Widder A., Ammon R. v., Schaeffer P., Wolff C. (2007) Identification of suspicious, unknown event patterns in an event cloud. In: Proceedings of the 2007 Inaugural International Conference on Distributed event-based systems. ACM, pp. 164–170

Widder A., von Ammon R., Hagemann G., Schönfeld D. (2009) An Approach for Automatic Fraud Detection in the Insurance Domain. In: AAAI Spring Symposium: Intelligent Event Processing. AAAI, pp. 98–100

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

Xu D., Tian Y. (2015) A comprehensive survey of clustering algorithms. In: Annals of Data Science 2(2), pp. 165–193

Yin S., Kaynak O. (2015) Big data for modern industry: challenges and trends [point of view]. In: Proceedings of the IEEE 103(2), pp. 143–146

Appendix

A1 Table 3: Concept Matrix for Supervised Learning

A2 Table 4: Concept Matrix for Unsupervised and Combined Learning

A3 Table 5: Publications of Category *Others* Excluded from Concept Matrices

Table 3: Concept Matrix for Supervised Learning

Supervised Learning																									
				Classification									Probabilistic Method			ILP	Regression			Supervised Clustering		Statistical Estimator	Deep Learning		Data Type
#	Reference	Reactive	Proactive	Support-vector Machines	Shapelet-based Algorithms	Decision Trees	Artifical Neural Networks	Markov Model	Discriminant Analysis	Nearest Neighbor Search	Naive Bayes Classifier	Own Development	Bayesian Networks	Markov Model	Conditional Density Estimation	Own Development	Support-vector Regression	Symbolic Regression	Own Developement	Centroid-based Clustering	Own Develpment	Kalman Filter	Deep Neural Network	Convolutional Neural Network	
[1]	Gad et al. 2014	•		•		•				•	•														network data
[2]	Peterson et al. 2017	•		•																					network data
[3]	Peterson et al. 2016	•		•																					sensor data
[4]	Jayan and Rajan 2014	•																							network data
[5]	Schwegmann et al. 2013		•	•													•								process data
[6]	Patri 2016	•			•																				sensor data
[7]	Patri et al. 2016	•			•																				sensor data
[8]	Mousheimish et al. 2016a		•		•																				sensor data
[9]	Mousheimish et al. 2016b		•		•																				sensor data
[10]	Mousheimish et al. 2017		•		•																				sensor data
[11]	Metz et al. 2012	•				•																			sensor data
[12]	Sun et al. 2019	•				•																			sensor data
[13]	Fülöp et al. 2012		•			•																			sensor data
[14]	Engel and Etzion 2011		•			•							•												sensor data & process data
[15]	Nguyen et al. 2018		•			•	•																		sensor data & process data
[16]	Duenas et al. 2018		•			•																			network data
[17]	Bodunov et al. 2018		•			•																	•		sensor data
[18]	Widder et al. 2009	•					•		•																process data
[19]	Widder et al. 2008	•					•		•																process data
[20]	Widder et al. 2007	•					•		•																process data
[21]	Yang et al. 2015	•					•																		language data
[22]	Borkowski et al. 2017		•				•																		process data
[23]	Shahad et al. 2016	•										•													sensor data & image data
[24]	Mutschler and Philippsen 2012	•						•																	sensor data
[25]	Baldoni et al. 2015		•					•																	network data
[26]	Roşca et al. 2018		•							•															sensor data
[27]	Sabahi-Kaviani and Ghassemi 2019	•										•													network data
[28]	Mehdiyev et al. 2016	•										•													sensor data
[29]	Mehdiyev et al. 2015b	•										•													sensor data
[30]	Mehdiyev et al. 2015a	•										•													sensor data
[31]	Zhu et al. 2013		•										•												simulation data
[32]	Wang and Kuang 2015		•										•												simulation data
[33]	Wang and Cao 2014		•										•												simulation data
[34]	Wang et al. 2018a		•										•												simulation data
[35]	Akbar et al. 2018		•										•												sensor data & social data
[36]	Skarlatidis et al. 2015	•												•											image data
[37]	Christ et al. 2016		•												•										sensor data & image data
[38]	Artikis et al. 2010	•														•									sensor data
[39]	Katzouris et al. 2017a	•														•									image data
[40]	Katzouris et al. 2017b	•														•									image data
[41]	Artikis et al. 2017	•														•									process data
[42]	Katzouris et al. 2016	•														•									sensor data
[43]	Katzouris et al. 2015	•														•									image data
[44]	Katzouris et al. 2019	•														•									sensor data
[45]	Ziekow et al. 2013		•														•								sensor data
[46]	Lang and Capik 2014		•														•	•							N/A
[47]	Cuzzocrea et al. 2018		•																•						process data
[48]	Akbar et al. 2017		•																•						sensor data & social data
[49]	Akbar et al. 2015b		•																•						sensor data & social data
[50]	Roledene et al. 2016		•																	•					process data
[51]	Lee and Jung 2017	•																			•				process data
[52]	Cuzzocrea et al. 2019		•																		•				Process data
[53]	Turchin et al. 2009	•																				•			network data
[54]	Abdallah et al. 2018	•																					•		video data
[55]	Lin et al. 2018		•																				•		sensor data
[56]	Barsocchi et al. 2018	•																						•	sensor data
	Amount	**30**	**26**	**4**	**5**	**8**	**6**	**2**	**3**	**2**	**1**	**5**	**6**	**1**	**1**	**7**	**3**	**1**	**3**	**1**	**2**	**1**	**3**	**1**	

Table 4: Concept Matrix for Unsupervised and Combined Learning

#	Reference	Reactive	Proactive	Unsupervised Learning: Clustering: Density-based Clustering	Clustering: Hierarchical Clustering	Clustering: Centroid-based Clustering	Dimensionality Reduction: Principal Component Analysis	Unsupervised Classification: Own Development	Data Type
[57]	Junior et al. 2017	•		•					simulation data
[58]	Terroso-Sáenz et al. 2015	•		•					simulation data
[59]	Sugiura et al. 2017	•			•				sensor data
[60]	Gander et al. 2013	•				•			network data
[61]	Akbar et al. 2015a	•				•			sensor data
[62]	Ta-Shma et al. 2018	•				•			sensor data
[63]	Petersen et al. 2018	•				•		•	sensor data
[64]	Teixeira et al. 2010	•					•		network data
	Amount	**8**	**0**	**2**	**1**	**4**	**1**	**1**	

#	Reference	Reactive	Proactive	Combined Learning: **Clustering; *Classification***: Centroid-based Clustering; *Markov Model*	**Clustering; *Classification***: Centroid-based Clustering; *Gaussian Mixture Model*	**Clustering; *Classification***: Density-based Clustering; *Markov Model*	**Clustering; *Classification***: Hierarchical Clustering; Centroid-based Clustering; SVM; *Decision Trees*; *Naive Bayes Classifer*	**Clustering; *Classification***: Unspecified Clustering Algorithm; *Naive Bayes Classifier*	**Clustering; *Probabilistic Method*; Topic Mining**: Density-based Clustering; *Markov Model*; Bag-of-words	**Clustering; *Classification*; Collaborative Filtering**: Centroid-based Clustering; *Naive Bayes Classifier*; Matrix Factorization	**Clustering; *Probabilistic Method***: Densitiy-based Clustering; *Bayesian Inference*	**Regression; *Deep Reinforcement Learning***: Logistic Regression; *Deep Belief Network*; *Markov Decision Processes*	**Clustering; *Dimensionality Reduction*; Classification**: Centroid-based Clustering; *Space-filling Curve*; Artificial Neural Network	**Clustering; *Deep Learning***: Centroid-based Clustering; *Decision Tree & Recurrent Nueral Networks*	Data Type
[65]	Akram et al. 2017		•	•											sensor data
[66]	Amariei et al. 2017		•	•											sensor data
[67]	Choi et al. 2017		•	•											sensor data
[68]	Jankov et al. 2017		•	•											sensor data
[69]	Mayer 2017		•	•											sensor data
[70]	Rivetti et al. 2017		•	•											sensor data
[71]	Zaarour et al. 2017		•	•											sensor data
[72]	Archetti et al. 2014		•	•											sensor data & image data
[73]	Chandrathilake et al. 2016	•			•										sensor data
[74]	Bachar et al. 2018		•			•									sensor data
[75]	Sejdovic et al. 2016		•				•								sensor data
[76]	Preuveneers et al. 2016	•						•							sensor data
[77]	Terroso-Sáenz et al. 2016		•						•						sensor data & social data
[78]	Zhang et al. 2016		•							•					social data
[79]	Jung et al. 2018		•								•				sensor data
[80]	Wang et al. 2018b		•									•			simulation data
[81]	Kammoun et al. 2018		•										•		sensor data
[82]	Song et al. 2018	•												•	N/A
	Amount	**3**	**16**	**8**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	

Table 5: Publications of Category Others Excluded from Concept Matrices

Description	**Publication(s)**
Contribution partly uses own developments which learn from events: *Windows Learner*; *Event/ Attributes Learner*; *Constraints Learner*; *Aggregates Learner*; *Parameters* Learner; Sequences Learner; Negations Learner	(Margara et al. 2013; Margara et al. 2014)
Contribution uses different algortihms, partly belonging to the research field of Data Mining: *Markov Logic*; *Association Rule Mining*; *Sequence Mining*; *Frequent Itemset Mining*	(Bhandari 2012)
Contribution uses Supervised ML for predictions, but does not specify the algorithm chosen.	(Tóth et al. 2010)
Framework for the application of Supervised ML in CEP (area of IoT), with no naming of specific algorithm used.	(Soto et al. 2016)
Contribution ueses semi-Supervised ML to also learn from unknown data, with a method based on *Graph Cut Minimization*.	(Michelioudakis et al. 2018)
Contribution(s) use(es) Supervised ML algortihms based on F*requent Sequence Mining*, which automatically learn pattern of interest.	(Gay et al. 2015; George et al. 2016)