Novelty Detection algorithms and their application in Industry 4.0

Christoph Stemp
University of Passau
Passau, Germany
stemp04@gw.uni-passau.de

***Abstract*—Novelty detection is a very important part of Intelligent Systems. Its task is to classify the data produced by the system and identify any new or unknown pattern that were not present during the training of the model. Different algorithms have been proposed over the years using a wide variety of different technologies like probabilistic models and neural networks. Novelty detection and reaction is used to enable self*-properties in technical systems to cope with increasingly complex processes. Using the notion of Organic Computing, industrial factories are getting more and more advanced and intelligent. Machines gain the capability of self-organization, self-configuration and self-adaptation to react to outside influences.**

This survey paper looks at the state-of-the-art technologies used in Industry 4.0 and assesses different novelty detection algorithms and their usage in such systems. Therefore, different data-sources and consequently applications for potential novelty detection are analyzed. Three different novelty detection algorithms are then present using different underlying technologies and the applicability of these algorithms in combination with the defined scenarios is analyzed.

***Keywords*—Novelty detection, Outlier detection, Organic Computing, Machine learning, Industry 4.0**

I. Introduction

In computer science and especialy machine learning, theoretical and experimental progress advances with great strides, but implementation of those results in industrial settings is often lacking and behind the state-of-the-art research [1].

One concept that is not yet widely adapted is novelty detection for autonomous systems. Novelty detection aims at discovering and also reacting to outliers and newly arising processes in continuous data streams, e.g. produced by sensors in autonomously acting systems, like surveillance networks or industrial machines in big, modern factories. Machines used in an Industry 4.0 setting are equipped with sensors which measure a substantial amount of manufacturing parameters and are connected with other machines in the factory to enable self-organization, self-configuration, self-adaptation and decentralized management [2].

These features of Industry 4.0 machinery ties in with the concept of Organic Computing (OC) which aims to cope with the increasing complexity of modern IT-technologies by shifting the management of tasks from one central entity to its smaller subsystems which act autonomously and inherit some combination of self-* properties ([3], [4]).

In this sense of OC, the self-organization and self-adaptation of such machines could benefit from the notion of novelty detection. Therefore, this paper analyzes different data-sources in Industry 4.0 and novelty detection algorithms to assess, which of them would perform best in a modern industrial environment. The key contribution of this paper is the theoretical evaluation of novelty-detection algorithms on theoretical industrial data-sets to better understand the concepts and dynamics of novelty-detection and possible fields of application.

The paper is organized as follows. Firstly, the terms and basic concepts of Industry 4.0 and novelty detection are discuss in Section II. In Section III, different data-sources that occur in Industry 4.0 factories are described and afterwards, three different novelty detection algorithms are described in Section IV. Section V evaluates the usage of the described novelty detection algorithms with the data-sets described in Section III. Section VI summarizes the results and findings of this article and gives a brief outlook to further work that could be done in this filed.

II. Basic concepts

For a better understanding of this article, this section briefly introduces term definitions, the technologies used in Industry 4.0 and novelty detection.

A. Term definitions

Because the terms anomaly, outlier and novelty are often used interchangeably, a finer definition is required. However, because of different definitions used in literature, this is not an easy task. For this paper, these terms are defined as follows:

An *anomaly* is typically a sample that is not explained by a model describing the data, as it differs from the expected samples. *Outliers* are similar, they are often described as data-points that deviate from a given density model. Outliers should be removed from the data-set to increase the performance, whereas anomalies should trigger actions, like novelty or fault detection. A *Novelty* is a clusters of anomalies, typically associated with a newly arising process or change of distribution. *Noise* is also part of every data-set and commonly consists of randomly scattered samples.

B. Industry 4.0

Over the recent centuries, manufacturing went through several industrial revolutions, which each lead to a paradigm shift in production techniques and management. The current effort is to merge the industrial production of goods with information and communication technologies (ICT), which is not only pursued by large corporations, but is also a core objective in the high-tech strategy of the German government [5]. Lasi et al. described Industry 4.0 as ”*[t]he vision of future production [that] contains modular and efficient manufacturing systems and [that] characterizes scenarios in which products control their own manufacturing process.*” [2]

These visions are implemented with the use of different emerging technologies. As described by [5], three key concepts for enabling the shift to Industry 4.0 are Smart Factories, Cyber-Physical Systems (CPS) and Internet of Things (IoT). The concept of Smart Factory was introduced by Lucke at al. 2008 in [6]. The authors described it as a factory that is combined with ubiquitous computing capabilities to realize context awareness, meaning that the components of the system communicate with each other as well as gather and share contextual information.

Such a combination of physical production hardware and software elements are often described as Cyber-Physical Systems. These systems are comprised of a large-scale network of embedded systems like sensors, actuators and computing nodes, which are spatially separated and have to fulfill certain criteria like robustness, self-awareness and self-maintenance [7]. H. Cheng explored a large variety of papers focusing on CPS to gain knowledge about the current state-of-the-art research [8]. He grouped his findings into ten research areas which range from algorithms over modeling to system design and wireless sensor networks, which are one of the key technologies for bridging the gap between the physical and cyber world. Because all these networks are heavily reliant on interconnectedness and communication, IoT can be considered as the backbone of CPS [7].

IoT devices are used in a wide variety of application, mostly for data collection, ranging from object tracking with the use of active or passive RFID tags, over automatic product inspections by continuously measuring the production status with sensors, to optimizing material acquisition through monitoring and prediction of stock and usage [9].

C. Novelty detection

As mentioned in the introduction, novelty detection aims at finding outliers of potentially novel processes in sensor data. Extensive research was conducted in recent years to develop new algorithms and approaches for novelty detection. In early review papers written by Markou and Singh [10], [11] in 2003, novelty detection algorithms are grouped in statistical approaches and neural network based approaches. Algorithms which fall under the statistical category are based on models built with the statistical properties of the underlying data. Data is tested against a trained model of ”normal” data and a novelty score is determined for each point and if the score is greater than a defined novelty threshold, it is considered abnormal. Statistical approaches can be further divided into parametric and non-parametric, depending on the model used. One example of a novelty detection algorithm using a parametric approach is presented by Bishop in [12]. In this article, a Gaussian Mixture Model (GMM) is trained and optimized by using the EM algorithm. New, previously unseen data is then shown to the model and evaluated to return the degree of novelty.

Neural network approaches can also be categorized further by the architecture and methods used, like multi-layer perceptrons (MLP), support vector machines (SVM) or oscillatory networks [11]. In [13], Singh and Markou implemented novelty detection for video sequences using an MLP as classifier. They used a rejection filter to categorize the data either as known or unknown. The known data is then classified by the neural network and the rejected data is collected and clustered to identify new classes, which are then labeled and used for training in the next iteration.

In more recent research however, this categorization in statistical and neural networks is no longer used, as described by Pimentel et al. in [14]. Instead, categories like probabilistic, distance-based and domain-based techniques are used. The category of probabilistic algorithms contains parametric approaches, using for example a GMM ([12], [15]) or a Hidden Markov Model (HMM) ([16], [17]), and non-parametric approaches like algorithms using kernel density estimators like the Parzen windows estimator ([18]). Distance-based algorithms, like the name suggests, use well known distance metrics to distinguish between known and novel data points. One such approach is k-Nearest Neighbor (k-NN) using Euclidean or Mahalanobis distance measures. Used for novelty detection, k-NN is based on the premise that ”normal” data points have close neighbors but the distance to novel points is much greater [19]. Domain-based novelty detection uses methods to create a boundary between different classes like SVMs. Ma and Perkins, for example, use an one-class SVM to detect novelties in time-series data [20].

III. Data sources in Industry 4.0

To select appropriate novelty detection algorithms for an Industry 4.0 setting, the different data sources that can occur in such a factory need to be identified and analyzed. It is important to select the algorithm based on the type of data that should be analyzed because some algorithms might not be able to handle certain types of input. For example, if the data source is high dimensional and the approach can only process a small amount of dimensions, possible outliers could remain undetected. In these modern factories, a great variety of processes gather data. In the scope of this paper, the focus lies on production, logistics and communication processes.

A. Production

One data source in Industry 4.0 is the production process. Machines and industrial robots record lots of different sensor data and parameter about the manufacturing process, like pressure, spindle speed, machine load or parts per hour. Furthermore, RFID chips can be attached to the workpiece to track the position throughout the whole assembly line. Video surveillance can also be used for detecting the position of objects [21]. Data produced by machines can further be distinguished by usage.

1) Machine Health: Machine health is one important reason to record data. High cost industrial machinery deteriorates over time and maintenance and repairs become necessary. Sensors are installed to measure machine usage parameters like vibration and temperature to increase the lifetime of the machine, optimize performance and reduce downtime costs [22]. This data is typically low to medium dimensional, numeric time-series data.

2) Production Parameters: Production parameters are also an important data source in Industry 4.0. Every produced part is modeled to specific dimensions, which have to be adhered to by the machine. Therefore, it is important to measure available parameters during manufacturing like motor load, fixture pressure and spindle load (e.g. in CNC mills). Finished machined dimensions can either be measured by touch probes or via visual means with 3D laser scanning [23]. These values are measured by sensors built into the machine and are simple, low to medium dimensional numerical data.

3) Object Tracking: In factories, workpieces move between different machines along an assembly line for the different production steps. The pieces are tagged with RFID chips to enable tracking between machining stations and factory locations as well as monitor the states and identities of the objects [24]. The acquired data is of complex nature, it includes different data types like a location of some sort, a state or state identifier and identity information.

B. Logistics

Ensuring the timely delivery of material and workpieces is an essential part of management in Industry 4.0. Therefore, logistics is another important data source. Material has to be transported to and from machines and finished products have to be transferred to and from storage and to the end user. Like with production object tracking, RFID tags as well as GPS and GMS based systems can be used ([25], [26]). Data gathered by this system contains position and location as well as numerical data in a time-series manner.

C. Communication

A critical aspect of modern Industry 4.0 inspired factories is the connectedness of every part, from machines to small IOT sensors. All the network communication between these entities can be considered another data source. This communication and sensor network has to be secured against outside attacks, for it is the backbone of the factory. The data transferred over the network can be considered connection and package data, containing continuous numerical and categorical attributes, like the packet size, duration, different kinds of flags and protocol type and thus is of medium to high dimensionality.

IV. ALGORITHMS

In the scope of this paper, three algorithms are reviewed, one using a probabilistic, the second neuronal network and the third a graph based approach. In the category of probabilistic novelty detection, the *2 Stage Novelty Detection and Reaction (2SNDR)* algorithm presented by Gruhl et al. in [27] is discussed. For neuronal networks, the novelty detection algorithm proposed by Hawkins et al. in [28] using Replicator Neural Networks (RNN) is reviewed. As a third algorithm a Link-based Outlier and Anomaly Detection in Evolving Data Sets (LOADED) proposed by Ghoting et al. in [29] is presented.

A. 2SNDR

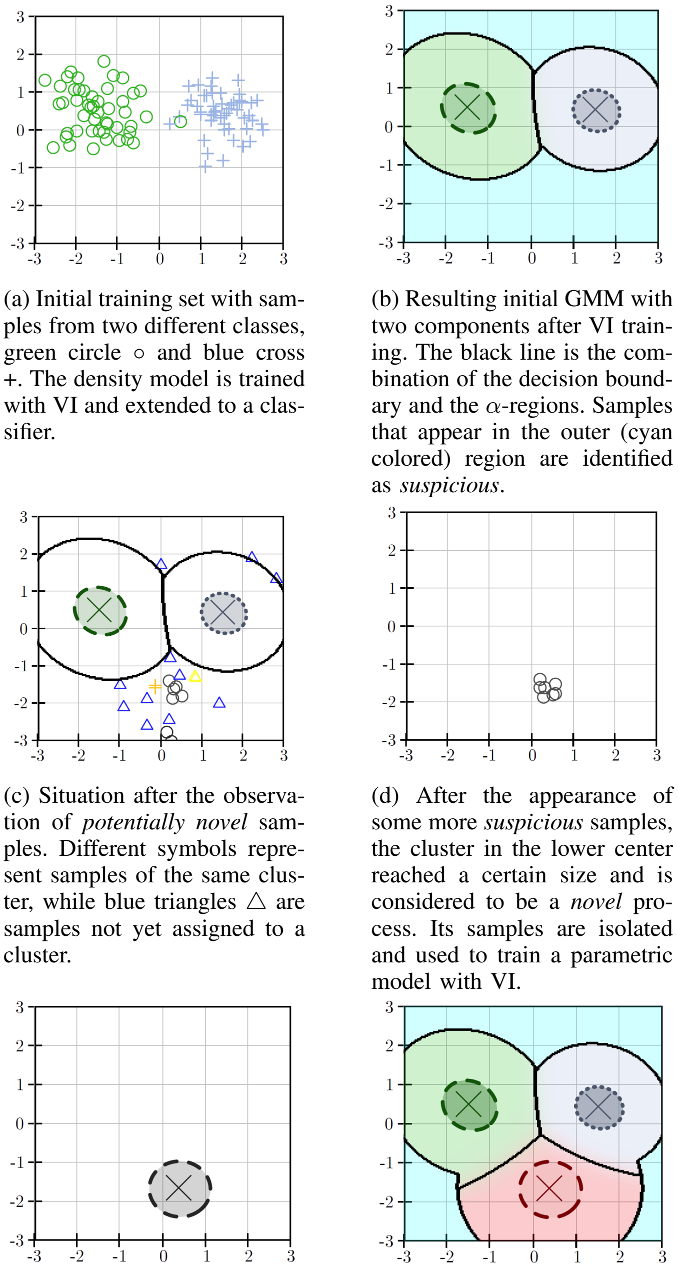
Gruhl et al. introduced 2SNDR as an extension of existing classifiers or Gaussian Mixture Models adding novelty detection and model adaption techniques [27]. 2SNDR has three stages. First, suspicious data points are identified, then in the second stage, novel processes are identified and in the third stage, the model is adapted to include new processes.

1) Algorithm: Figure 1 shows the process of novelty detection and model adaption in different steps as described below. The figure is taken from [27]. Initially, a model is trained with VI on the underlying data (1a, 1b).

a) First stage - suspicious sample detection: Suspicious samples are identified by their distance to the mean of the nearest Gaussian. This is done by exploiting the fact that the squared Mahalanobis distances between samples and the mean of Gaussians is χ^2_D-distributed and thus, the Mahalanobis distance can be defined with the quantile function $F^{-1}_{\chi^2_D}$ of the χ^2_D distribution as $\rho = F^{-1}_{\chi^2_D}(\alpha)$. The parameter α can now specify how close a sample has to be to a Gaussian center to count as a normal *non-suspicious* sample. Samples that have a greater Mahalanobis distance from the centers of all Gaussians than samples in the α-region are classified as *suspicious* (1c).

b) Second stage - novel process detection: Now, that outliers are detected, the algorithm evaluates if the samples belong to a novel, previously unseen process. Therefore all suspicious samples are cached in a circular buffer of size $\tilde{b}$ and compared to each other. If the distance between two samples is smaller than a predefined distance ϵ, these points are added to a cluster, similar to the DBSCAN algorithm. If a cluster reaches a threshold of $minPts$ points, it counts as a novel process.

c) Model adaption: If a novel process is detected, the underlying model has to be adapted. The first step of adaption is to isolate the samples belonging tho the new process (1d) and preforming a VI training so the samples are represented by another GMM (1e). Then, the main GMM and the new GMM are merged together by fusing their respective hyperparameters. Non-overlapping parts are simply added to the main model and overlapping components are fused if their pairwise divergence is greater then 0.5. Finally, the hyper-distribution of the mixing-coefficients have to be adjusted to still form a distribution in the final GMM. After the fusing is performed, the corresponding samples are removed from the ring-buffer and the new model is used from there on out (1f).



(a) Initial training set with samples from two different classes, green circle ○ and blue cross +. The density model is trained with VI and extended to a classifier.

(b) Resulting initial GMM with two components after VI training. The black line is the combination of the decision boundary and the α-regions. Samples that appear in the outer (cyan colored) region are identified as *suspicious*.

(c) Situation after the observation of *potentially novel* samples. Different symbols represent samples of the same cluster, while blue triangles △ are samples not yet assigned to a cluster.

(d) After the appearance of some more *suspicious* samples, the cluster in the lower center reached a certain size and is considered to be a *novel* process. Its samples are isolated and used to train a parametric model with VI.

(e) After the VI training converges, the *novel* process is represented by a GMM which consists of a single component. The newly acquired knowledge will be fused with the initial model shown in (b).

(f) Updated GMM and classifier after the *novel* process is integrated. The updated decision regions are shown as well. The red component and region corresponds to the *novel* process.

Fig. 1: Illustration of the proposed technique. In the training set only samples of two processes are present. In the operational phase a third process emerges and starts to generate samples. After enough *potentially novel* samples are observed, the model and the classifier are updated [27].

2) Properties: Because this algorithm is based on GMMs, it is trained in an unsupervised manner, which means that no labels are required for it to function. Furthermore, the use of VI permits prior knowledge about the data to be included in the training process and thus the usage of already collected data. Uniformly distributed noise in the input data does not affect the approach, because the novel process detection mechanism needs a dense grouping of samples. Despite using GMMs which need time for fitting, 2SNDR is designed to work in an online mode, which makes this approach suitable for continuous operating systems with low to medium dimensional data. On the other hand, probabilistic models require good, dense training data to perform well. Their performance decreases with small, higher dimensional data-sets, because the samples are more spread out in the hyperspace.

B. RNN Outlier Detection

Replicator Neural Network Outlier Detection was first proposed by Hawkins et al. in [28].

RNNs are multi-layer perceptron neural networks and a variant of usual regression models. They are designed with three hidden layers and the same number of output neurons as input neurons, with the middle layer having fewer neurons than output and input. Figure 2 depicts a fully connected RNN with three neurons in the middle layer.

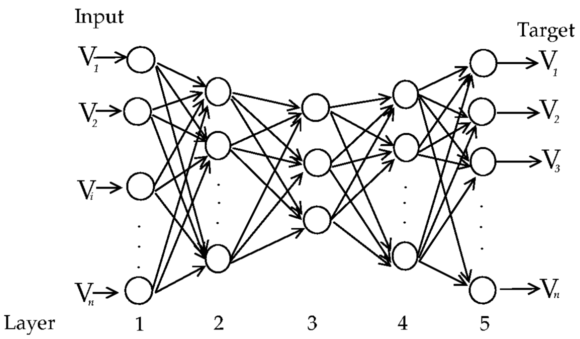


Fig. 2: A schematic view of a fully connected Replicator Neural Network. [28]

RNNs are designed to reproduce the input vectors at the output layer with minimal reconstruction error. As activation function, the Sigmoid function is used for the outer layers. The activation function at the middle layer is specifically designed to create a staircase shape to quantise the continuously distributed data points into discrete bins to achieve data compression.

1) Algorithm: To enable the RNN to detect outliers, the notion of outlyingness is used. Therefore, an *Outlier Factor (OF)* of is defined as

$$OF_i = \frac{1}{n}\sum_{j=1}^{n}(x_{ij} - o_{ij}) \tag{1}$$

the average reconstruction error over all features, with i being the ith sample and n being the number of features. If a sample is over a given threshold, it is considered an outlier. If it is smaller than the threshold, it is considered normal. The neural network is trained on existing data samples using an adaptive learning rate.

2) Properties: Because of the nature of neural networks, this approach is best applicable for data-sets with low to medium dimensionality. They suffer form the curse of dimensionality and training becomes increasingly hard, time- and resource-intensive. Moreover, the performance of the network increases with the training time, which is often not available in an online fashion. Another downside of this approach is that it is not very usable for mixed-attribute data-sets. For categorical data, networks have to be trained per category to circumvent the drawback that only non-categorical data can be used for training, which increases the time and memory requirements as well as loosing information about the dependencies between the categorical and continuous attributes.

C. LOADED

Distance and density based algorithms are often used for anomaly detection, but they have major drawbacks in that they can't normally cope with mixed-attribute data. Therefore, the graph-based algorithm LOADED is proposed by Ghoting et al. [29], which takes the dependencies between categorical and non-categorical attributes into account.

1) Algorithm:

a) Categorical attribute space: For this algorithm to detect outliers, the notion of similarity has to be defined in the categorical attribute space. A data points in categorical attribute space is comprised of a number of N attribute-value pairs. Two points are linked, if they have at least one attribute-value pair in common, with the link-strength being the number of shared pairs.

An outlier in categorical attribute space is then defined as a point that has either very few links to other points, or has a very low link-strength. Therefore, a scoring function is defined which grants a score inversely proportional to the sum of the strengths of all links, meaning that a point with no links gets the highest score and a point with many links to others will have a low score.

b) Mixed attribute space: For the mixed attribute space, a correlation matrix filled with Pearson correlation coefficients is incrementally maintained. A data point is considered linked to another point in the mixed attribute space, if they are linked in the categorical attribute space and if the continuous attributes comply with the corresponding covariance as listed in the correlation matrix. If a sample violates at least one of these properties, it is considered to be outliers.

2) Properties: As intended by the authors, LOADED is capable of dealing not only with numerical data but also with mixed type data-sets. But the approach suffers from the curse of dimensionality. The memory and calculation requirements increase rapidly with higher dimensional data. On the other hand, the algorithm is designed to detect outliers in one pass, which makes it suitable for applications with time constraints.

V. Evaluation

With an overview over the data sources and types of data-sets produced in an Industry 4.0 environment and three different novelty detection algorithms discussed, the potential applicability of these algorithms can be evaluated. For each data-source, an application case for the novelty detection algorithm can be specified. The machine health data can be used for machine health monitoring and production parameter data can be used to detect problems and faults during the manufacturing process. The data collected during the tracking of objects can be used for localization anomaly detection to improve production efficiency and lower costs. Logistics data can be used in logistics quality management and communication data can be used to detect faulty sensors or possible intrusions in the network. The results of the evaluation is shown in Table I, where a check-mark is present, if the algorithm is considered suitable for a given application in this Industry 4.0 setting.

A. Machine health monitoring and production fault detection

Machine health monitoring and production fault detection requires the algorithm to perform online in a time sensitive way to detect potential damage to the machine and possible faults in the production parameters quickly. The data in these data-sets contain no labels, but older, already saved data may be available to assist the algorithm. 2SNDR is a good candidate for these applications, because it is built to perform in an online fashion. The algorithm operates in an unsupervised way, therefore, no labels have to be provided. Also, the VI training process can utilize the old data to gain more knowledge about the processes. RNNOD works well with low dimensional data and also needs no a priori knowledge, but the training of the neural net needs more time that may not be available in these settings. LOADED also works in low dimensional data-sets and is designed to work in an online fashion because it can detect outliers in one pass. On the other hand, this algorithm is best suited for mixed type data-sets and therefore not the perfect candidate for these applications.

B. Localization anomaly detection

The data-set of this application is a medium dimensional data-set comprised of complex, mixed-type data. To be able to react to possible anomalies in the object tracking data, the algorithm probably has to work in an online fashion, but the data could also be processed offline. 2SNDR has the advantage of performing in an online mode, but is not very good in working with mixed-type data. RNNOD has the same problem as 2SNDR and cannot handle continuous and categorical data at once very well. LOADED on the other hand is built specifically for such data-sets and is the pest candidate for this application because the algorithm can also work in an online fashion.

C. Logistics quality management

For this data-set, containing location data, distance based approaches would probably be best suited, but all three presented algorithms can be used for this application. For logistics

Data source	Data type	Application	Algorithm		
			2SNDR	RNNOD	LOADED
Machine Health	Low / medium dimensional, numeric time series	Machine health monitoring	✓		
Production Parameters	Low / medium dimensional numeric	Fault detection in production	✓		
Object Tracking	Medium dimensional, mixed-type	Localization anomaly detection			✓
Logistics	Low dimensional, numerical, location	Logistics quality management	✓	✓	✓
Communication	Medium to high dimensional, streaming, mixed-type	Sensor failure detection	✓		✓
		Intrusion detection	✓	✓	✓

TABLE I: This table shows the evaluation results of the different novelty detection algorithms theoretically applied to Industry 4.0 applications arising from observed data-sources.

quality management, the data is processed offline after it is collected and therefore RNNOD can also be used because the algorithm has no constraints in time and memory. 2SNDR can also be used if the location data is numerical. Like with machine health and monitoring, LOADED is not the best choice because of its focus on mixed-type data, but it can also be used with only continuous numerical data.

D. Sensor failure detection

Sensor failure detection is a complex problem because data produced by a faulty sensor can look like more noise (which is already present plentiful enough) in the communication data. It is critical to detect broken sensors as quickly as possible to ensure that other parts of the factory don't get damaged because of it. 2SNDR has the advantage that it can handle uniformly distributed noise very well which is beneficial in this application to distinguish between real noise and a faulty sensor. This approach is also capable of performing under tight time and resource constraints. RNNOD again has the problem of requiring time to train the neural network, which makes it unsuitable for time and mission critical applications. LOADED also can function well under time constraints, but needs more memory because of the covariance matrix in this medium to high dimensional data set. Because of the online operation capabilities and because the data is of a mixed-type, this algorithm is a viable candidate for this application.

E. Intrusion detection

Network intrusion detection is well documented field in novelty detection and all of the described algorithms can handle this task. The authors of the respective papers that introduced these algorithms, all tested their approach on the KDD 1999 cup data-set with good performance [27]–[29]. However, LOADED has still the benefit of making use of the mixed-type data-set, which the two other algorithms can only do to an extent. 2SNDR also has the advantage over RNNOD to be designed to work in an online manner, with RNNOD, again, needing more time to train the neural network on the medium to high dimensional data-set.

VI. Conclusion and Outlook

This paper discussed different novelty detection algorithms and their application in certain domains of Industry 4.0. Different data-sources emerging from processes in Industry 4.0 inspired factories and theoretically applied three novelty detection algorithms are analyzed and their potential eligibility was assessed. As we have seen, no single novelty detection method is the best for all tasks. Different data-sets require different concepts as well as different situations require different properties. It was assessed, that the probabilistic 2SNDR algorithm can be applied to machine health monitoring, fault detection in production, logistics quality management as well as sensor failure detection and intrusion detection. The neural network based RNNOD can be used for logistic quality management and intrusion detection and the graph based LOADED approach can be used for localization anomaly detection and intrusion detection because of its capability to handle mixed-type data-sets. The covered data-sources presented in this paper are by no means complete and a lot of research has to be done in the field of novelty detection in Industry 4.0.

For future work, this paper could be extended to describe more state-of-the-art novelty detection algorithms and categories. Also, more data-sources and applications in an Industry 4.0 setting could be described to form a comprehensive taxonomy for novelty detection in Industry 4.0. The results of this paper could be tested on real data-sets to confirm or refute the findings of this paper.

All in all, the area of intelligent systems, especially seen from the perspective of Organic Computing is a very interesting topic and great innovation and improvements will be made in the next decades.

References

[1] M. Sharp, R. Ak, and T. Hedberg, "A survey of the advancing use and development of machine learning in smart manufacturing," *Journal of Manufacturing Systems*, vol. 48, pp. 170 – 179, 2018, special Issue on Smart Manufacturing.

[2] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & Information Systems Engineering*, vol. 6, no. 4, pp. 239–242, 2014.

[3] S. Tomforde, B. Sick, and C. Müller-Schloer, "Organic computing in the spotlight," 2017.

[4] C. Müller-Schloer and S. Tomforde, *Organic Computing - Technical Systems for Survival in the Real World*. Birkhäuser, 2017. [Online]. Available: https://doi.org/10.1007/978-3-319-68477-2

[5] H. Kagermann, J. Helbig, A. Hellinger, and W. Wahlster, *Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry; final report of the Industrie 4.0 Working Group*. Forschungsunion, 2013.

[6] D. Lucke, C. Constantinescu, and E. Westkämper, "Smart factory - a step towards the next generation of manufacturing," in *Manufacturing Systems and Technologies for the New Frontier*, M. Mitsuishi, K. Ueda, and F. Kimura, Eds. London: Springer London, 2008, pp. 115–118.

[7] L. Esterle and R. Grosu, "Cyber-physical systems: challenge of the 21st century," *e & i Elektrotechnik und Informationstechnik*, vol. 133, no. 7, pp. 299–303, Nov 2016.

[8] H. Chen, “Theoretical foundations for cyber-physical systems: A literature review,” *Journal of Industrial Integration and Management*, vol. 02, p. 1750013, 10 2017.

[9] F. Tao, Y. Wang, Y. Zuo, H. Yang, and M. Zhang, “Internet of things in product life-cycle energy management,” *Journal of Industrial Information Integration*, vol. 1, pp. 26 – 39, 2016.

[10] M. Markou and S. Singh, “Novelty detection: a reviewpart 1: statistical approaches,” *Signal Processing*, vol. 83, no. 12, pp. 2481 – 2497, 2003.

[11] ——, “Novelty detection: a reviewpart 2:: neural network based approaches,” *Signal Processing*, vol. 83, no. 12, pp. 2499 – 2521, 2003.

[12] C. M. Bishop, “Novelty detection and neural network validation,” *IEE Proceedings - Vision, Image and Signal Processing*, vol. 141, no. 4, pp. 217–222, Aug 1994.

[13] S. Singh and M. Markou, “An approach to novelty detection applied to the classification of image regions,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 4, pp. 396–407, April 2004.

[14] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, “A review of novelty detection,” *Signal Processing*, vol. 99, pp. 215 – 249, 2014.

[15] D. P. Filev and F. Tseng, “Real time novelty detection modeling for machine health prognostics,” in *NAFIPS 2006 - 2006 Annual Meeting of the North American Fuzzy Information Processing Society*, June 2006, pp. 529–534.

[16] D.-Y. Yeung and Y. Ding, “Host-based intrusion detection using dynamic and static behavioral models,” *Pattern Recognition*, vol. 36, no. 1, pp. 229 – 243, 2003.

[17] S. Ntalampiras, I. Potamitis, and N. Fakotakis, “Probabilistic novelty detection for acoustic surveillance under real-world conditions,” *IEEE Transactions on Multimedia*, vol. 13, no. 4, pp. 713–719, Aug 2011.

[18] D.-Y. Yeung and C. Chow, “Parzen-window network intrusion detectors,” in *Object recognition supported by user interaction for service robots*, vol. 4, Aug 2002, pp. 385–388 vol.4.

[19] V. Hautamaki, I. Karkkainen, and P. Franti, “Outlier detection using k-nearest neighbour graph,” in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 3, Aug 2004, pp. 430–433 Vol.3.

[20] J. Ma and S. Perkins, “Time-series novelty detection using one-class support vector machines,” in *Proceedings of the International Joint Conference on Neural Networks, 2003.*, vol. 3, July 2003, pp. 1741–1745 vol.3.

[21] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *ACM Comput. Surv.*, vol. 38, no. 4, Dec. 2006.

[22] J. Lee, H. D. Ardakani, S. Yang, and B. Bagheri, “Industrial big data analytics and cyber-physical systems for future maintenance & service innovation,” *Procedia CIRP*, vol. 38, pp. 3 – 7, 2015, proceedings of the 4th International Conference on Through-life Engineering Services.

[23] P. Zheng, H. wang, Z. Sang, R. Y. Zhong, Y. Liu, C. Liu, K. Mubarok, S. Yu, and X. Xu, “Smart manufacturing systems for industry 4.0: Conceptual framework, scenarios, and future perspectives,” *Frontiers of Mechanical Engineering*, vol. 13, no. 2, pp. 137–150, Jun 2018.

[24] J. Brusey and D. C. McFarlane, “Effective rfid-based object tracking for manufacturing,” *International Journal of Computer Integrated Manufacturing*, vol. 22, no. 7, pp. 638–647, 2009.

[25] K. Prasanna and M. Hemalatha, “Rfid gps and gsm based logistics vehicle load balancing and tracking mechanism,” *Procedia Engineering*, vol. 30, pp. 726 – 729, 2012, international Conference on Communication Technology and System Design 2011.

[26] C. Yuqiang, G. Jianlan, and H. Xuanzi, “The research of internet of things’ supporting technologies which face the logistics industry,” in *2010 International Conference on Computational Intelligence and Security*, Dec 2010, pp. 659–663.

[27] C. Gruhl, B. Sick, A. Wacker, S. Tomforde, and J. Hähner, “A building block for awareness in technical systems: Online novelty detection and reaction with an application in intrusion detection,” in *2015 IEEE 7th International Conference on Awareness Science and Technology (iCAST)*, Sept 2015, pp. 194–200.

[28] S. Hawkins, H. He, G. Williams, and R. Baxter, “Outlier detection using replicator neural networks,” in *Data Warehousing and Knowledge Discovery*, Y. Kambayashi, W. Winiwarter, and M. Arikawa, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 170–180.

[29] A. Ghoting, M. E. Otey, and S. Parthasarathy, “Loaded: link-based outlier and anomaly detection in evolving data sets,” in *Fourth IEEE International Conference on Data Mining (ICDM’04)*, Nov 2004, pp. 387–390.