Evaluation and Adaption of Maintenance Prediction Methods in Mixed Production Line Setups based on Anomaly Detection

Sebastian Soller* and Bastian Fleischmann*
Almanara Research GmbH
Ruhstorf an der Rott, Germany
Email: *forename.surname@almanara-research.de

Matthias Kranz† and Gerold Hölzl†
Faculty of Computer Science and Mathematics
University of Passau, Germany
Email: †forename.surname@uni-passau.de

***Abstract*—Mixed production line setups are common in real world scenarios but are sparsely covered by current research in anomaly detection. A mixed production line setup poses several challenges and properties that make using anomaly detection more complex compared to monotonous production processes. We consider a high variation in the produced products, diluted temporal dependencies, imbalance in the frequency of product types and sensor measurements differing based on the produced product type. We gather contextual information using the OPC UA standard and extract information such as running program name.**

In this work we adapt and evaluate common anomaly detection methods to such a scenario. By building ensembles of anomaly detectors, we account for different setups. Each single work-piece gets evaluated for anomalous behaviour. We evaluate the anomaly detection by using data from a mixed production setup in the automotive domain. The context based approach is compared to a sliding window approach, which we use as a baseline.

The results highlight that our product type-based approach shows a higher precision and recall for all applied detection techniques, by utilizing contextual information. Our experiments additionally show that in the selected industrial case study our approach achieves good results even when only limited data per product type is available.

***Index Terms*—data mining, real-time system, maintenance prediction, adaptive network, reliability evaluation, sustainability**

I. Introduction

In modern production facilities individual type customization and individualization has become an important aspect in production lines. It is common in the food and semiconductor industry that the production process is performed in batch production [1]. Due to the mass-customization these batch sizes are shrinking in size, often down to batch size one [2]. This mixture of product types and change from a single repeating process also challenges maintenance prediction methods. Conventional methods, such as using thresholds to raise machine alarms are becoming harder to implement. Different product types need to be reassessed manually for their limits, what can be a costly process for a small number of produced pieces. Performing updates to the machine settings and maintenance too frequently can result in high cost, high consumption of spare parts and loss of overall equipment effectiveness (OEE) [3].

Predictive maintenance can be performed in multiple ways: optimizing the scheduling policy, predicting the remaining useful life of machines, and monitoring the condition of the machinery [4]. Machine learning systems have opened new opportunities to boost and outperform current predictive systems. This work evaluates a machine's condition by monitoring the deviation of each work piece from the machine's previous behavior. Anomaly detection is successfully applied by researchers in the industrial domain [5], [6]. We apply this method to learn a description of the regular operation of the machinery. The learned model can detect deviations from the expected operation. In previous [7] we used an outlier detection AI based system for the prediction of maintenance. This approach used a feature-based drift model to determine the deviation of each workpiece from its predecessor. It could detect multiple machine errors around ten minutes in advance. This is a short time to take preemptive steps and the goal is to detect more detailed changes in the signal for an earlier detection. Therefore, we apply more sensitive methods for the monitoring of a long-time system deviation.

Often methods for the detection of anomalies inside processes are used on a continuous data stream without type information and will detect anomalies on product type switches. Information about the product mix can be used for an easier windowing approach which is not used for many methods. Instead of a fixed window each product type can be used as window. We propose and implement an anomaly detection applicable to production lines with unknown dependencies. We compare the results for commonly used anomaly detection methods One-Class Support Vector Machine [8], Isolation Forest [9] and Auto Encoder [10]. We analyze the amount of data the learning approach needs for sufficient accuracy. For the parameter tuning and algorithm setup a probabilistic model is used to weight false positives and negatives to maximize an underlying cost function.

We propose a self-configuring distributed system, which is capable to connect, analyze and make a prediction for a wide array of data sources. An information model is built for the prediction setup supplying context information to decide between anomaly detection algorithms and configurations.

978-1-6654-0424-2/21/$31.00 ©2021 IEEE

II. Related Work

Anomaly detection is already applied in multiple domains. An example for such applications is fraud detection in the insurance [11] and finance [12] domains. As in the industrial domain, a description of non-anomalous transactions can be learned using anomaly detection techniques. Training a supervised classifier using previous anomalous transactions limits the model to anomalies that are like the observed ones. Bolton et al. [13] propose performing fraud detection by grouping accounts that perform similar transactions into peer groups. With these groups' anomaly detection can be performed with respect to the corresponding peer group. Therefore, anomaly detection only considers similar operation. By performing anomaly detection on the specific context more fine-grained anomalies can be found.

Similar approaches have been done in the semiconductor industry. In this domain the production process is often exceedingly long. Identifying failures early in the process of producing a microchip can drastically increase the productivity. Puggini et al. [14] and Susto et al. [15] use the Isolation Forest technique for anomaly detection in the plasma etching process of silicon wafers. Susto et al. [16] also compare different outlier detection such as Local Outlier Factor and Angle Based Outlier Detection to identify anomalies in the semiconductor fabrication process. Historic data is used to learn thresholds for non-anomalous operation.

Machine learning techniques are becoming increasingly popular for the task of anomaly detection in the industrial and technical domain [17]. Su et al. [18] use the One-class SVM technique to detect anomalies in temperature readings obtained by sensors. To capture contextual information, they use a sliding window that contains multiple temperature readings. Additionally, they use the K-Nearest Neighbors technique to find anomalies that are like the one that is detected. Therefore, they also perform classification on the type of error that occurs.

Liu et al. [5] propose using an Autoencoder for a manufacturing setup. They consider temporal co-occurrence of different (anomalous) events to refine the neural network. With this co-occurrence information they reduce the neural network to only consider interactions that are present in the observed data. Additionally, Autoencoders are used for applications such as sound machines [6], wind turbines and ozone levels. Different versions of the Autoencoder are used for these setups, such as a Denoising Autoencoders [19] and a Variational Autoencoders [20]. The common advantage these setups have is a uniform signal since the observed processes should remain identical.

Anomalies can also be used to estimate the remaining useful life of machines and devices. Kammerer et al. [21] use matrix distance profiling of time series segments to identify the number of emerging patterns. Their data consists of a single run to failure. In their study they are able to detect a high number of emerging patterns starting around 13 hours before a defect. The number of emerging patterns also has a peak 9-8 hours before the defect. After this the number rapidly decreases as the defect draws closer. Jin et al. [22] identify anomalies by setting thresholds on health indicators. As health indicators they use the vibrations of the bearing. The thresholds are determined by a statistical analysis of historic data. The remaining life is only predicted once an anomaly is detected, otherwise their approach refrains from making a prediction.

Cheng et al. [23] propose monitoring measurements against a health baseline to detect anomalies. Anomalies are identified by performing a sequential probability ratio test using the current measurement and the health baseline. Once an anomaly is detected the features that are responsible for the deviation should be isolated. They propose using techniques such as PCA to isolate the features. Using the responsible features, a definition of the failure is calculated by using physical models and regression can be performed to predict the future values of the features.

We need an evaluation method for the anomaly detection. Lavin et al. [24] propose an evaluation method based on windows around ground truth anomalies and provide rewards for earlier detection. Thus, this method is capable of handling scenarios where the exact time for the occurrence of anomalous behaviour is unclear. Because the scale of this score depends on the concrete problem it is applied to, we use the normalization to a common scale from $-\infty$ to 100. A normalized score of 100 would mean that only perfect detections are made. Further, a score lower than 0 means that the detection is not better than abstaining from making predictions with respect to the used profile. Their rating method allows the usage of different profiles. We will use their profiles low false positives(low FP), low false negatives(low FN) and standard for the evaluation of our setup.

III. SYSTEM STATE AND GOAL

We observe industrial production lines using an OPC UA connection to read data. This sensor data needs to be live monitored for data deviation and anomalies. In a first setup we tried to use previously proposed methods on our sensor data. We choose sliding windows as they are commonly used for anomaly detection with time series data [19] [25] [26] [27], while also providing reliable results. For each data stream gathered by the sensors we used the proposed methods but could not reach sufficient results for a long-time prediction. We extended the simple data drift from our previous work [7] by applying the anomaly detection machine learning methods of the Autoencoder, one-class SVM and Isolation Forest. The best performing feature for every method can be seen in Table I in the baseline columns. The problem with this method is the high variability between each product type produced in the machine we observe. The sliding window picks up these changes and anomalies will be detected, which do not reflect the machine's health. We either get no detections in large portions of the data due to the big mixture of data values and ranges of the different types or get a high number of detections if we detect the data variance as anomalies.

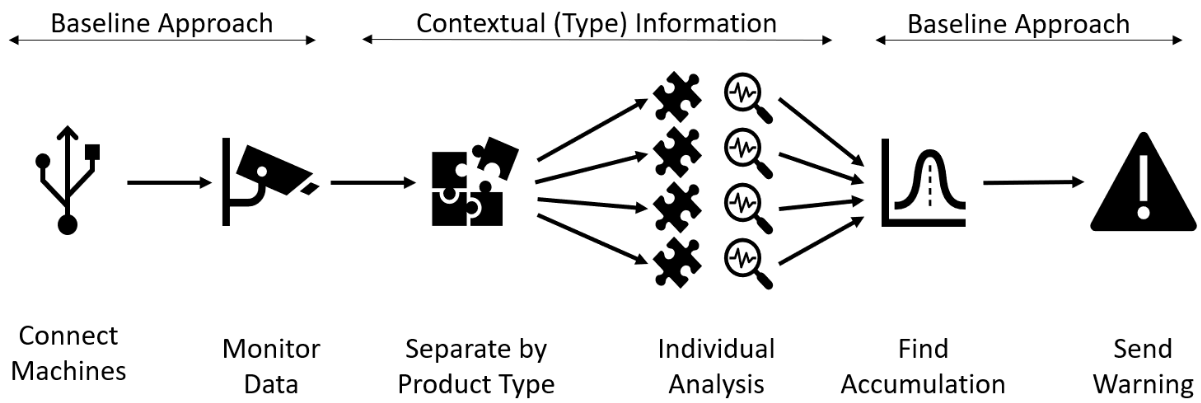

Fig. 1. The pipeline used for the error prediction of each individual machine.

We conduct a pilot study to gather measurements between July 2019 and May 2020. We aim to build up a system as shown in Fig. 1. In this work we present results for a single machine to demonstrate the process steps. The framework created needs to work for a big array of machines. There should be no human interaction with the process beyond the first step of network connection and the last step of interpreting the warning sent by the system. The framework needs to discover the machine endpoints and monitor the data. Each product type produced in the machine needs to be recognized and analyzed individually and an alarm should be triggered if an accumulation of errors is found in a given time frame. We designed an information model depending on each single work piece as schematically presented in Fig. 2 to capture contextual information necessary to dynamically build and adapt the processing chains. Each component of the framework, from preprocessing to the alert message is dependent on its context information, obtained by the machine and product information. The anomaly model is influenced by setups in the framework, such as the focus on low false positives or low false negatives, the seasonality [28] of the time series, which machine and which product were measured. This information for the machine and products is supplied by the communication protocol. Depending on the result of the anomaly model and its predicted failure probability a warning is sent by the framework. In practice, this system is used in a testing phase, where the time series with its number of anomalies can be observed by the foreman. For strong peaks, a notification is sent out.

IV. METHOD

A. Data Collection

In the previous work we obtained the sensor data by using the OPC UA standard. We treated every time series of any machine identical and used a checksum to transfer string data to numerical values for feature extraction. This was done to keep the system on a zero manual configuration level and make analysis possible without human interaction. We further analyzed a machine range of 28 different machine types

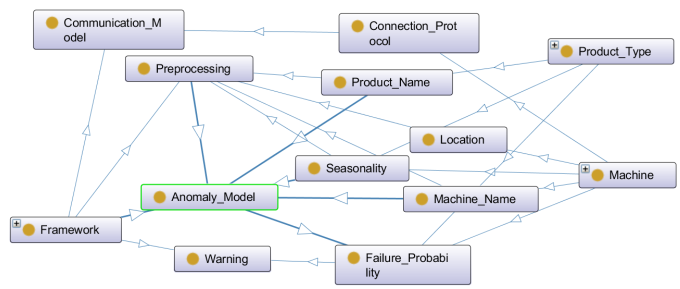

Fig. 2. The framework is autonomously adapted for each analyzed workpiece, depending on contextual information (e.g., machine, product type, manual settings, etc.) The knowledge is preserved in form of an Ontology to enable semantic querying and reasoning.

by three different suppliers controlled by a programmable logic controller (PLC). On any machine we could at a bare minimum of additional information get the program name of the currently running program. As this information is available on any machine observed, we can use this information in a setup without losing the adaptability component of our system. When implemented into a framework, the program name information is used independently of a machine. This aspect of the system is essential for our analysis of differing product types and if we can boost the analysis by separating the data stream by pieces. We can use every single work piece as a semantic segment for the analysis pipeline shown in Fig. 3. In the first step we recognize the product type for each work piece using the program name. For each product type the machine learning model needs a separate training set. We will evaluate the number of instances the different machine learning models need to achieve viable results. If a product type does not have enough instances, the work piece is used for the training set, otherwise it is classified as anomalous or non-anomalous.

B. Preprocesing

The preprocessing we employ for the training and validation data consists of steps for cleaning, normalization and resam-

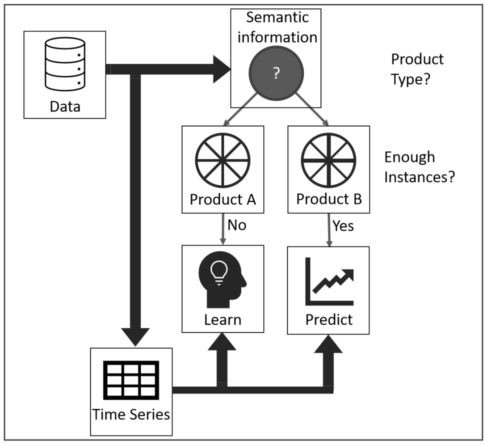

Fig. 3. Procedure for each work piece in a live production environment. Each data segment gets analyzed by its semantic information (schematically presented in Fig. 2) of the product type. The time series gets used for either (i) only learning or (ii) for learning and prediction of the anomaly detection, depending on a sufficient training data size.

pling. The cleaning and resampling steps are optional and depend on the characteristics of the input data. After cleaning the data, we normalize each feature to a value range between -1 and 1 by dividing them by their respective maximum. Each product type is normalized independently as the value range can differ between the various types. The product types can be separated by their program name which is read from the machine. The sensor measurements can also have a varying sampling rate. A varying sampling rate can occur within the measurements of a single feature or across multiple features. Differences can also arise in the duration of the measurements. This results in an inconsistent length of the semantic segments. The anomaly detection techniques we use require inputs of equal length. To this end, we perform mean downsampling to a fixed sampling rate. Each semantic segment is divided into non-overlapping windows with a duration s. The mean of all values within a window is calculated. This mean becomes the measured value at the start of the window. After downsampling each semantic segment has the same sampling rate. Downsampling is used to achieve equal length for all segments with the same duration while being computationally inexpensive. It is also used to make the data more comparable and supports our goal to connect new machines setup free. The preprocessed time series is directly used as a feature vector without feature extraction.

C. Anomaly Detection

As laid out earlier each training sample is a semantic segment that represents the processing of a product. The exact product type is known for each of these products. For our approach we first segregate the learning set by the respective product types. As shown in the schema 2, the learning set then consists of a set for each product type. Each of these sets is used to train an individual model. So, we enforce that only a representation of the corresponding product type is learned. By using the semantic segments, we also achieve that each sample represents the same actions performed by the machine. The result of the training is an ensemble of models for the different product types. To build those models common anomaly detection methods are used. We focus on One-Class Support Vector Machine [8], Isolation Forest [9] and Auto Encoder [10]. Those methods will be adapted to fit the use case by performing cross-validation and parameter tuning. We use hold-out validation as the cross-validation method as it represents the case of ongoing production. The models can be trained by optimizing for Precision, Recall and in addition for evaluation methods focusing on either a low number of false positives or false negatives [24]. The approach is tuned using the standard profile of the evaluation metric in our experiments. This provides a balance between low false positives and negatives.

D. Anomaly Fusion

As we use different anomaly detection methods, we select the best performing method depending on the context information. Additionally, we can perform a fusion of the results for all possible combinations. We can especially combine methods, which need a small number of training samples to reach their highest accuracy with models, which need more instances, but perform better with enough instances. As a simple approach for this work we use a majority vote. In the online setting each method is selected, by the results of the previous detection's score, for the different methods.

V. Data

We collected the data of multiple production lines and different machine types. For the learning set we use the measurements of the products produced from July 2019 to December 2019. The validation set consists of products from January 2020 to March 2020. A relatively long interval for the validation set is chosen as the production in spring 2020 is decreased due to environmental factors. The measurements inherit 55 different product types. Product types are similar in weight and shape, but differ in features, such as design and number and position of drilled holes. During that period, the processing of 85000 products is measured. The sensor system aims at providing measurements every 30 milliseconds, for an average of 15 endpoints for each machine. However, the actual period between sensor measurements is variable. This is due to latency and limited throughput of the industrial network. Because of network and further sensor problems it can also occur that for a product only zero values are measured. We have to clear these values in the pre-processing. Additionally, measurements of a product can be aborted before it is fully processed. Likewise, it can also occur that the completion of a product is not recognized. In this case measurements of multiple products are accounted to one product. This results in a loss of 2897 product samples, which is 3.4 percent of all products produced. The ground truth data consists of detected machine defects. Therefore, it contains labels for events when a machine transitions into a state that requires repair or is no longer operational. We got 14 critical errors, which is attributed to a breaking of machine parts and 82 minor errors.

TABLE I
SCORES OF THE ANOMALY DETECTION USING BASELINE AND THE APPROACH USING THE CONTEXT INFORMATION.

	One-Class SVM		Autoencoder		Isolation Forest		Majority Vote
	Baseline	Context Info	Baseline	Context Info	Baseline	Context Info	Context Info
Standard	8.24	46.97	12.99	45.43	8.64	27.73	46.52
Low FN	17.44	50.37	20.60	51.74	17.70	39.97	50.07
Low FP	-8.36	41.61	1.12	33.92	-7.57	9.20	42.75
Precision	0.20	0.57	0.13	0.34	0.19	0.26	0.59
Recall	0.35	0.57	0.35	0.64	0.35	0.64	0.57

VI. RESULTS

Table I shows the scores of the anomaly detection. The product type-based approach in combination with One-Class SVM has the highest scores for the *Standard* and *Low False Positives* profiles. The One-Class SVM detects slightly less defects than the Autoencoder but is more accurate. In terms of the *Low False Negatives* profile, the product type-based approach with Autoencoder provides a higher score. The recall also shows this. The Autoencoder also has a *Standard* score that is close to the One-Class SVM. It has a lower *Low False Positives* score. The product type-based approach with Isolation Forest has lower scores than the other two. The scores range from 10 to 20 points less than the other two approaches. The recall is equal to the Autoencoder. The applied Majority vote for all anomaly detection methods results in a higher precision of 0.59 but a lower recall of 0.57. The lower false positive score also outmatches each single method with a score of 42.75. The median distance of these anomalies to the corresponding upcoming error is 7.5 hours, while the distance of the closest 25% is less than 45 minutes. We evaluated the results for both the baseline and the product type-based approach for the three anomaly detection techniques previously presented. We used the presented month of April and May for the Evaluation. We only present April, as there were no errors in the machine in May and the anomaly detection did not detect any anomalies during that month.

We evaluated the necessary training set size to achieve the projected results. For the Autoencoder we reach a plateau of only marginal improvements at a training set size of around 430 samples per product type. The One-Class SVM reaches the point of marginal improvements at around 450 samples. While the One-Class SVM needs more samples to reach that point, it performs better than the Autoencoder at lower training set sizes until around 300 samples for a product type in average. The Isolation Forest technique differs from the other two techniques. It has high average scores even with smaller training set sizes close to zero. With an increasing training set size above 250 samples, the average scores decrease for the Isolation Forest.

VII. DISCUSSION

Both the baseline and the product type-based approach can find large accumulations of anomalies near many defects. The product type-based approach finds indicators for up to 64% of all defects and the baseline up to 42%.

The anomalies accumulate before each failure. On average 50% of the detected anomalies are less than 17 hours away and the closest 25% anomalies are within a 1.5 hour range to one of the defects we consider. Therefore, anomalies with a long distance to defects are sparse and rare. The 50% with the longest distance to defects also have a mean distance of 4 days to the next three neighboring anomalies on average. Therefore, large agglomerations of anomalies outside of the defect windows are rare.

A problem we need to consider is the usage of ground truth in this field. For an exemplary accumulation of anomalies without subsequent failure, we inspected the maintenance log of the machine and were not able to find any significant entry as to whether these anomalies correspond to. Further investigation led to a high amount of faulty parts of the product types in the end control several production steps later, which were attributed to processes preceding the machine observed for anomaly detection. Those faulty parts could be the reason behind the anomalous machine behavior, but they also show the strong influence of multiple processes onto each other.

The recall, temporal distribution and scores of the evaluation method way above zero show that we can use anomaly detection to find early indicators for defects in this experimental setup. Especially, highly frequent detections of anomalies are close to defects. Therefore, we can employ anomaly detection to find indicators for defects in this industrial setup. Both approaches also have room for improvement in terms of recall. However, it must be considered that the ground truth also contains preventive actions for critical defects and human interaction with the machinery regularly occurs.

The results show that the product type-based approach already performs well in our industrial setting. This is likely because only limited variation should occur during regular operation in an automated industrial process. Therefore, it can be applied to on-line data in our scenario, as sufficient data can be obtained in a limited time frame. The usage of context information and the ability to obtain it from a large array of machines improves detection results compared to the baseline. More samples could help to build more robust models with less false positives for the Autoencoder and SVM. A larger training set could also help to detect more anomalies in other testing scenarios, as an over proportional representation of anomalies and noise in the training set due to an ill-suited collection period can be reduced. The Isolation Forest performs more stable for a low number of samples, but is outperformed by the Autoencoder for a bigger training set. It finds anomalies by deriving rules to isolate samples in the training set. If the size of the training set increases, more rules to isolate samples can be deduced. Therefore, the rule set becomes stricter. This results in an increasing number of detected anomalies as the size of the training set increases. We use the detection fusion to achieve the best possible result for each produced work piece, by starting the prediction using the Isolation Forest. The first instances of each type are predicted using the Isolation Forest and later we switch to the one-class SVM, as their accuracy starts to outperform the Isolation Forest at a specific point.

VIII. Conclusion and Future Work

Our approach can identify big accumulations of anomalies preceding machine defects. These anomalies are identified as an indicator of machine defects.

Our designed system works well as an indicator for any faulty behavior with either the machine or the produced products thus leading to a more sustainable production in general. All detected accumulations of anomalies were followed by erroneous machine behavior or multiple work piece defects. 75% of predictions can be made at a minimum of 45 minutes preceding the error, up to early prediction of several hours. By combining the different anomaly detection methods, we can select the best possible method for each work piece during runtime and recursively evaluate the prediction and dynamically adapt the used methods for the upcoming work pieces.

Future work has to find a definition of an accumulation of anomalies and the respective metric. Single anomalies in a data stream do not necessarily lead to events. Another aspect is that we did not identify clear patterns for specific defects. We can predict errors, but it can also result in erroneous pieces days later in the end control. An expert's opinion is essential to look into the alert and identify the error. This is still helpful, as it reduces the amount of manual inspections, but we aim to improve the system by means of error identification. This extends in addition to provide warnings, by giving alarms with detailed explanations that are especially useful for inexperienced workers. We are working on using context information to normalize numeric features for a better comparability between different product types. This will result in a more stable monitoring of each machine and bring our approach closer to work on a low amount of each product type down to batch size one. We are currently working on finishing the DataSet and code for publication to allow reproduction of our results and evaluation algorithms.

References

[1] K. Wu, "Taxonomy of batch queueing models in manufacturing systems," *European Journal of Operational Research*, vol. 237, no. 1, pp. 129–135, 2014.

[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] P. Muchiri and L. Pintelon, "Performance measurement using overall equipment effectiveness (oee): literature review and practical application discussion," *International journal of production research*, vol. 46, no. 13, pp. 3517–3535, 2008.

[4] N. Sakib and T. Wuest, "Challenges and opportunities of condition-based predictive maintenance: a review," *Procedia CIRP*, vol. 78, pp. 267–272, 2018.

[5] J. Liu, J. Guo, P. Orlik, M. Shibata, D. Nakahara, S. Mii, and M. Takáč, "Anomaly detection in manufacturing systems using structured neural networks," in *2018 13th World Congress on Intelligent Control and Automation (WCICA)*. IEEE, 2018, pp. 175–180.

[6] D. Y. Oh and I. D. Yun, "Residual error based anomaly detection using auto-encoder in smd machine sound," *Sensors*, vol. 18, no. 5, p. 1308, 2018.

[7] S. Soller, G. Hölzl, and M. Kranz, "Predicting machine errors based on adaptive sensor data drifts in a real world industrial setup," in *2020 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2020, pp. 1–9.

[8] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection," in *Advances in neural information processing systems*, 2000, pp. 582–588.

[9] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

[10] N. Japkowicz, C. Myers, M. Gluck *et al.*, "A novelty detection approach to classification," in *IJCAI*, vol. 1. Citeseer, 1995, pp. 518–523.

[11] M. Kirlidog and C. Asuk, "A fraud detection approach with data mining in health insurance," *Procedia-Social and Behavioral Sciences*, vol. 62, pp. 989–994, 2012.

[12] M. Ahmed, A. N. Mahmood, and M. R. Islam, "A survey of anomaly detection techniques in financial domain," *Future Generation Computer Systems*, vol. 55, pp. 278–288, 2016.

[13] R. J. Bolton, D. J. Hand *et al.*, "Unsupervised profiling methods for fraud detection," *Credit scoring and credit control VII*, pp. 235–255, 2001.

[14] L. Puggini and S. McLoone, "An enhanced variable selection and isolation forest based methodology for anomaly detection with oes data," *Engineering Applications of Artificial Intelligence*, vol. 67, pp. 126–135, 2018.

[15] G. A. Susto, A. Beghi, and S. McLoone, "Anomaly detection through on-line isolation forest: An application to plasma etching," in *2017 28th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*. IEEE, 2017, pp. 89–94.

[16] G. A. Susto, M. Terzi, and A. Beghi, "Anomaly detection approaches for semiconductor manufacturing," *Procedia Manufacturing*, vol. 11, pp. 2018–2024, 2017.

[17] R. P. Ribeiro, P. Pereira, and J. Gama, "Sequential anomalies: a study in the railway industry," *Machine Learning*, vol. 105, no. 1, pp. 127–153, 2016.

[18] J. Su, Y. Long, X. Qiu, S. Li, and D. Liu, "Anomaly detection of single sensors using ocsvm_knn," in *International Conference on Big Data Computing and Communications*. Springer, 2015, pp. 217–230.

[19] G. Jiang, P. Xie, H. He, and J. Yan, "Wind turbine fault detection using a denoising autoencoder with temporal information," *IEEE/Asme transactions on mechatronics*, vol. 23, no. 1, pp. 89–100, 2017.

[20] D. Kim, H. Yang, M. Chung, S. Cho, H. Kim, M. Kim, K. Kim, and E. Kim, "Squeezed convolutional variational autoencoder for unsupervised anomaly detection in edge device industrial internet of things," in *2018 international conference on information and computer technologies (icict)*. IEEE, 2018, pp. 67–71.

[21] K. Kammerer, B. Hoppenstedt, R. Pryss, S. Stökler, J. Allgaier, and M. Reichert, "Anomaly detections for manufacturing systems based on sensor data—insights into two challenging real-world production settings," *Sensors*, vol. 19, no. 24, p. 5370, 2019.

[22] X. Jin, Y. Sun, Z. Que, Y. Wang, and T. W. Chow, "Anomaly detection and fault prognosis for bearings," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 9, pp. 2046–2054, 2016.

[23] S. Cheng and M. Pecht, "A fusion prognostics method for remaining useful life prediction of electronic products," in *2009 IEEE International Conference on Automation Science and Engineering*. IEEE, 2009, pp. 102–107.

[24] A. Lavin and S. Ahmad, "Evaluating real-time anomaly detection algorithms–the numenta anomaly benchmark," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2015, pp. 38–44.

[25] J. Inoue, Y. Yamagata, Y. Chen, C. M. Poskitt, and J. Sun, "Anomaly detection for a water treatment system using unsupervised machine learning," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2017, pp. 1058–1065.

[26] T. G. Dietterich, "Machine learning for sequential data: A review," in *Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR)*. Springer, 2002, pp. 15–30.

[27] D. B. Araya, K. Grolinger, H. F. ElYamany, M. A. Capretz, and G. Bitsuamlak, "Collective contextual anomaly detection framework for smart buildings," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 511–518.

[28] S. Soller, M. Kranz, and G. Hoelzl, "Adaptive error prediction for production lines with unknown dependencies," in *Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics*, 2020, pp. 227–234.