Automated Anomaly Detection and Diagnosis of the Environmental Control System of the ISS

*Lukas Moddemann, Henrik Sebastian Steude, Oliver Niggemann
Professorship Computer Science in Mechanical Engineering
Helmut Schmidt University
Hamburg, Germany
*lukas.moddemann@hsu-hh.de

Philipp Grashorn
Ground software development
Airbus Defence and Space
Bremen, Germany
philipp.grashorn@airbus.com

***Abstract*—Maintaining the operation of the International Space Station (ISS) is highly relevant, both socially and scientifically. To this end, corresponding telemetry data is continuously sent to Earth for monitoring and analysis. Due to the importance of life support systems for the survival of the astronauts, the data is currently monitored in laborious manual processes and analyzed for diagnosis in case of failures. Automated anomaly detection and diagnosis of these systems offer the potential not only to increase the efficiency of the engineers, but also to optimize the error-free operation of the space station and thus create more capacity for what the ISS was built for: science in space.**

In this paper we present the current state of development of Anomaly Detection (AD) and Diagnostic (DX) algorithms using data of the european part of the ISS, Columbus, as an example. We describe the state-of-the-art algorithms we implemented from the field of time series anomaly detection and model-based diagnosis and discuss the results in the concrete case. In addition, we discuss the respective advantages and disadvantages as well as the next steps in the project.

***Index Terms*—Machine Learning, Anomaly Detection, Cyber-Physical Systems, Diagnosis**

I. Introduction

The International Space Station ISS is a complex, technological system. In the event of a fault, e.g. a malfunction in the life support system, rapid and targeted identification of the cause of the fault is essential. Currently, data from roughly 20,000 sensors & signals within the European Columbus module is transmitted to the Columbus Operations Center (COL-CC) in Oberpfaffenhofen, where it is analyzed manually by a team of technicians. In case of occurrence of complex failures engineers of Airbus, the European industrial lead for the maintenance, commercialization and evolution of the Columbus, are involved into the resolution process. Maintaining the functionality of the ISS is of very high importance for scientific, political and economic reasons. The ISS currently represents the best opportunity for Europe to participate actively in space research. A central element for this is the automatic detection of dangerous or unwanted situations on the ISS and corresponding quick reactions. Examples of such anomaly detection and diagnosis problems in ISS operations are: (i) anomaly detection for the life support system and in resource consumption (power, data, cooling), (ii) diagnosing failure causes and proposing countermeasures. Such capabilities can directly help to reduce shift operations in control centers from 24x7 to 8x5 hours with on-call duty, a cost saving of about 75 %. Instead of time-consuming manual processes for analyzing the causes of errors, which take days, weeks or even months, intelligent assistance systems should point out possible correlations within seconds, make errors explainable and suggest solutions. This not only saves valuable time, but also sustainably improves the operational availability of the complex system. Overall, this will create more budget, time and availability for the actual purpose of the ISS: science. For this reason, two key components of the (K)ISS Project are to develop a software that is capable of detecting anomalies, i.e. symptoms, in the telemetry data of the Columbus module, as well as running a diagnosis, i.e. finding the root cause. A high level overview of the system to be developed is shown in FIGURE 1.

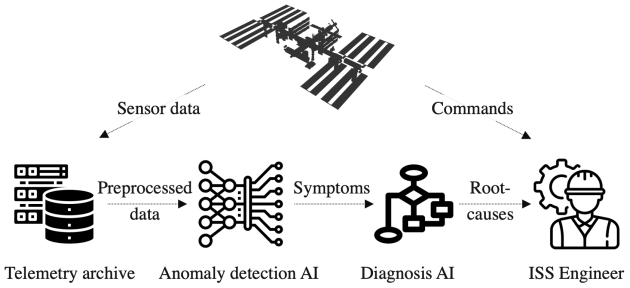

FIGURE 1. OVERVIEW OF THE (K)ISS PROJECT SCOPE

In this contribution, we present the corresponding challenges, the current development status of the AD and DX use cases, discuss the results, and highlight the next steps and potential future research questions. The remainder of this paper is organized as follows: In the following Section, we present the two use cases in more detail and discuss the respective challenges, followed by a Section on the current state of research (Section III) regarding the two use cases and their challenges respectively. Section IV provides an overview of the solutions we are currently using to address the use case challenges, followed by the results in Section V. Finally, we conclude this paper in Section VI.

https://doi.org/10.24405/14540

II. Use Cases and Challenges

From an artificial intelligence (AI) perspective, AD and DX are two fundamentally different problems with currently incompatible approaches. As can be seen in FIGURE 1, sensor signals are the starting point for both problems. Based on these signals, indications of dangerous situations, i.e. anomalies, are identified. This is usually implemented by a neural network learning a model of normal behavior from the historical data. For new sensor data, anomalies can then be detected by comparing them to the prediction of the learned model.

However, often, a failure causes an anomaly at completely different locations in the system. For example, a malfunction in the ventilation system may mean that astronauts are no longer supplied with sufficient oxygen, but also that smoke detectors do not seem to work. Since the cabin air does not circulate independently in micro gravity, both the oxygen-rich air and the smoke must be actively distributed, otherwise neither CO2 saturation, which is dangerous for astronauts, can be detected in time, nor can smoke reach the smoke detector. Anomalies can also be due to various causes of failures. In the example of the ventilation system mentioned above, insufficient air circulation can be caused not only by too little fan power, but also by too much - for example, it has happened in the past that a redundant fan was mistakenly switched on and generated too much suction in the intake area, so that the protection grid was clogged by sucked-in parts. The task of the diagnosis is now to determine the root causes of the fault based on the anomalies. This allows faults to be repaired quickly and dangers to the ISS and crew to be avoided. However, this can no longer be done solely on the basis of data and simple limit monitoring. Hence, more advanced technologies like machine learning must be taken into account. Furthermore, it requires knowledge of system causalities, time dependencies and system structures: How does an error affect the overall system? How long does this take? Which system parts are independent of other system parts? Both the implementation of the AD as well the DX pose their own challenges, which we will describe in the following.

A. Anomaly detection

From a technical perspective, the task of anomaly detection in the example of the Columbus module is that timestamps in the multivariate sensor time series must be assigned to the groups “normal” and “not normal”. The output of the AD serves as input for the DX. That’s why it is important to know which signals or components behave the least normal, in the event of a fault or anomaly. In the concrete case, these requirements poses several challenges from the algorithmic point of view: First, the dataset is very large and needs to be analyzed with special tools such as distributed computing and time series databases. The Environmental Control and Life Support (ECLS) system alone, which is the first Columbus subsystem of interest, emits more than a thousand signals sampled at one Hertz. In total the Columbus module records and sends more than 20 thousand signals to ground. Secondly, the data are very sparsely labeled. A database of anomaly reports can be accessed, however, no accurate start and end times are given and sometime reoccurrences are not recorded in the database. Finally, and thirdly, the time series data are very unbalanced in two respects: anomalies in the Columbus module are (fortunately) very rare, and furthermore, the entire system is in a variety of states at different times, with very different frequencies.

B. Diagnosis

Tied to anomaly detection is the task of diagnosis to determine the root cause from anomalies or more precisely from symptoms. This task is performed by use of a logical solver in satisfiability (SAT) or satisfiability modulo theories (SMT) configuration. These solvers need a DAG describing the causal relations between system components as input. The anomaly detected by the previously performed anomaly detection algorithm is subsequently included as a symptom (not healthy) for the specific component in the rule base of diagnosis. Finally, the statements, that represent the rule basis of the diagnosis written in propositional logic will be checked for satisfiability using a logic solver where the output of the diagnostic service will be a minimal diagnosis for all conflict sets. The final outputs are important for the further investigation and to initiate a reconfiguration of the process or to be able to fix the system. According to the requirements listed above and from the perspective of model-based diagnosis the following challenges arise: First, to be able to build up a representation of the system in first order logic, the system description and the components’ causal relationships need to be known. This results in a high modeling effort and research from the design documentation, which is only partially available in digital form. Secondly, due to the complexity and the size of the ECLS system, different levels of a diagnosis have to be investigated. The levels can differ in diagnostics based on states, components or even on the lowest level of measurement sensor systems that have a causal relation on each other. Lastly, the engineering analysis reports that are part of the anomaly reports are very rare, which means that artificially generated anomalies have to be created within a simulation in order to be able to validate the diagnostic algorithm, including the check of health components for consistency and violation of intervention bounds.

III. State-of-the-Art

In this section, we provide an overview of the state of the art with respect to the challenges we described for the two use cases in the previous chapter.

A. State-of-the-art anomaly detection

Regarding the first challenge within anomaly detection, the handling of large data sets, there are some tools in the python ecosystem that have been developed exactly for this purpose, e.g. Spark [1], Dask [2], and Ray [3].

The detection of anomalies in multivariate time series with statistical models is a well established research field [4]. To address the second challenge, namely the lack of available labels, which are not sufficient for supervised learning, we

limit ourselves to a group of models that can be trained without labels. For the detection of point anomalies in time series, the literature offers a variety of methods from different research areas of statistics and computer science, which have proven to be extremely well-performing. Well-known examples are the Gaussian Mixture models optimized with expectation maximization [5], one class support vector machines [6] and K-Means clustering, all of which use different measures or scores to distinguish normal data points from non-normal ones. The famous Autoencoder (AE) architecture for neural networks was first introduced in 1986 and is also well suited to detect point anomalies based on the reconstruction error [7]. Over the past few years, many new algorithms have been developed, which were specifically designed to identify anomalies in multivariate time series (sequence anomalies), many of which are based on deep neural networks. Among them are, for example, LSTM-based AEs [8], TCN-based [9] AEs, LSTM-base variational AEs [10] and BeatGAN [11], just to name a few.

As for the challenges of varying system modes, there is some work that attempts to extract the modes from the time series of data [12], [13]. If the system modes cannot be derived from the non-categorical signals, these models can be used to train individual models per system mode based on the mode prediction.

B. State-of-the-art diagnosis

To tackle the challenge of performing diagnosis on cyber-physical systems, that differ greatly in how detailed the system under consideration must be modeled in advance. Strong-fault models [14] specify faulty behavior modes for its system components and due to this, a change in the system results in an elaborate process to model the new fault signature combinations. In contrast, weak-fault models [15] make use of the normal behavior and are sometimes called a system with ignorance of abnormal behavior [16]. For this reason the description of the system needs to be formulated solely in the healthy state and therefore the approach is suitable for use cases with a huge amount of components like the ECLS system. Within diagnosis, a rule base for the identification of a root cause can be set up by different logical solvers, which are specifically designed to make a statement about whether a combination of propositional logical statements is satisfiable or unsatisfiable. Sympy [17] as a python library for symbolic mathematics offers an experimental assumptions system and uses a SAT solver for deduction of the exponentially increasing search space of component combinations. Z3 [18] is another state-of-the-art solver that makes use of SMT to be able to solve more complex formulas than solely boolean problems. To formulate the rules in first order logic, the system with its dependencies must be modeled. For this purpose, the research field of causalities has to be considered. According to our first challenge, a data-driven determination of the system dependencies has to be prioritized (e.g. Granger Causality [19], Transfer Entropy [20] or as a combination [21]). Another challenge is the determination of the level of diagnosis. In the field of cyber-physical systems, like ECLS, Struss [22] published a paper about the fundamentals of model-based diagnosis of dynamic systems. Within dynamic systems he proposed to capture the overall behavior of a hybrid system in a set of modes (or states) to be able to model the system. As mentioned above in the state of the art of anomaly detection, there are attempts to discretize hybrid data into modes [12], [13], which offer the opportunity to formulate logical rules according to the system modes for a diagnosis rule base.

IV. Solution

Although the (K)ISS project is still in an early phase, the first approaches have already been implemented and show promising results. As mentioned above, we focus on the ECLS system for validation and demonstration of the solutions. If the implementation is successful, the models will then be transferred to further subsystems (and eventually other systems, such as future space mission vehicles). In this section, we present the current development status of our AD and DX algorithms and discuss the results using the ECLS subsystem as a concrete example.

A. Solution anomaly detection

The general approach to detect anomalies in the telemetry data is, to train a model that "learns" the normal behavior of the subsystem. In this section we describe the dataset, the preprocessing steps and the model we use as a baseline for further developments.

1) Dataset: The data set of the ECLS subsystem, as well as the data sets of the other subsystems, consists of a number of continuous and categorical signals sampled at a frequency of one Hertz. The continuous signals represent sensor readings while the categorical ones represent discrete information such as switch positions, power status or commands. We call the time series dataset $\mathbf{X} \in \mathbb{R}^{T \times d}$ with T being the total number of timestamps (we use a time span of 2 years for the model training) and $d = c + k$ being the number of continuous signals c and the number of categorical signals k combined. In addition to the telemetry data, we collected the label information in order to validate our results. These labels were provided by the subject-matter-experts in form of a table holding the anomalies of the ECLS subsystem and the corresponding start and end timestamp. We use this table to split the dataset into a training set $\mathbf{X}_{train}$ and a validation set $\mathbf{X}_{val}$, such that $\mathbf{X}_{train}$ only holds "normal" time stamps. For the purpose of additional validation, other time windows were added to $\mathbf{X}_{val}$ at random, which do not contain known anomalies. Only $\mathbf{X}_{train}$ was used to update the model weights, so that only the "normal" state of the subsystem was "learned" by the model.

2) Preprocessing: For the preprocessing of the continuous signals, we use a default standard scaling. However, since we have a very large number of categorical signals $(k > 500)$ the preprocessing of the corresponding signals is not as trivial. Applying the one-hot encoding [23], for example, would create over 1000 additional columns and therefore be very inefficient,

in terms of storage, model weights and training time. Instead, we developed our own solution based on the frequencies of the unique combinations of the categorical values: Each unique combination of categorical entries that we observe in the training dataset is assigned to a unique ID. The unique IDs observed in less than 0.5% of the timestamps are assigned to a dummy category. We then apply one-hot encoding to the unique IDs and end up with only 300 categorical columns in the one hot encoded matrix. This dimensionality reduction is due to the fact, that there is a large number of different and rare unique combinations of the categorical signals. These combinations, however, would very likely be ignored by the model, since they do not contribute heavily to the negative average loss, which most ML models use as a training objective. All the preprocessing steps were implemented using Dask [2], such that it could run out of memory on several worker nodes in parallel.

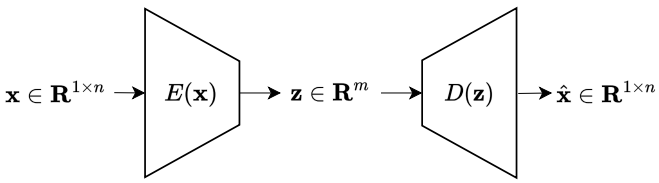

FIGURE 2. AUTOENCODER MODEL ARCHITECTURE

3) Model: According to the research questions of the (K)ISS-project, we aim at developing a deep learning based solution for the AD. For this reason we choose a vanilla AE [7] as a baseline model. The high-level overview of the model architecture is visualized in FIGURE 2. For every timestamp t the encoder E performs a highly non-linear dimensionality reduction $\mathbf{z} = E(\mathbf{x})$, such that $\mathbf{z} \in \mathbb{R}^m$ with $m \ll n$. The Decoder D maps the latent space variable $\mathbf{z}$ back to the original data space: $\hat{\mathbf{x}} = D(\mathbf{z})$. Both E and D are implemented as fully connected neural networks with LeakyReLU activations [24]. The model weights ($\sim$ 800k) were adjusted according to the mean squared error loss $\mathrm{MSE}(\mathbf{X}, \hat{\mathbf{X}})$.

The model was trained on graphical processing units for 100 epochs.

B. Solution diagnosis

The first approach in solving the task of diagnosing the ECLS system of the Columbus module will be presented in the following. The key steps are the definition of a rule base, the check for health states and the output of possible diagnoses.

1) Rule base: The definition of the rule base will be done by system experts and on the basis of a system description given in FIGURE 3. To be able to use the qualitative model of the ECLS System a conversion in well-formed SAT expressions needs to be done. For this, directed acyclic graphs (DAG) [25] are useful for representing the causal relationships, including the processing flows within the system. The main advantage of DAGs is that multiple paths of processing streams can be modeled, expressing the ability for data to be processed in multiple ways. Based on the created DAG the components' relationships will be stored to a list of rules as the above-mentioned rule base.

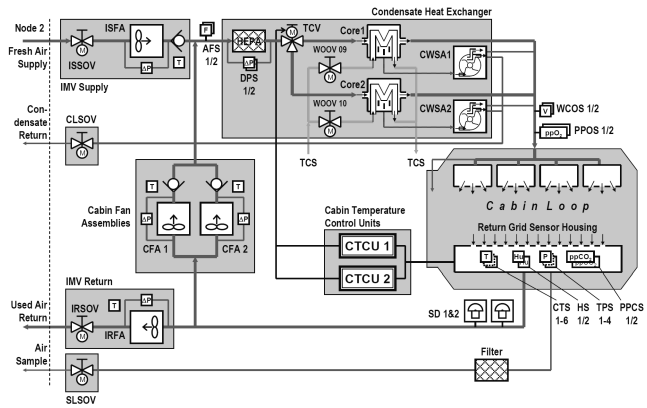

FIGURE 3. OVERVIEW OF THE ECLS SYSTEM [26]

2) Health states: As a next step to a diagnosis algorithm, the health states of the observable (measured) components will be investigated on its completeness according to the formulated rules. Therefore, a comparison between the information about detected anomalies, health states and list of rules will be carried out. However, in case of intentionally deactivated components, such as e.g. a deactivated Cabin Fan Assembly 1 (CFA1) according to FIGURE 3, this component must be excluded from the further diagnostic process. The reason here lies in the consideration of a possible automated reconfiguration on the basis of healthy and available components, to return the system back to an alternative working state and avoid a fatal downtime in e.g. ECLS. For this purpose, an exploration of the system in the actual operating state in case of a detected anomaly is performed. But not only this is under investigation. Another part of the algorithm takes also the check of the upper and lower bounds into consideration. If an anomaly is detected, the diagnosis algorithm will also check all other sensor measurements against its defined bounds to achieve an all-encompassing audit of the health states.

3) Diagnosis: Finally, in model-based diagnosis the formulated rule expressions will be checked on satisfiability. The result of consistency-based diagnosis is a set of possible fault hypothesis. The hypothesis will afterwards be discriminated by calculating the diagnosis of minimal faults.

V. RESULTS

This section describes the results obtained by the current versions of the AD and DX algorithms.

A. Results anomaly detection

The goal of the anomaly detection in the sense of our baseline model is to differentiate between "normal" and "non-normal" telemetry values or time points in the telemetry data stream of the ECLS system. The reconstruction error $e_t = \hat{\mathbf{x}}_t - \mathbf{x}_t = D(E(\mathbf{x_t})) - \mathbf{x}_t$ can be used as a suitable measure for this differentiation. Alerts might be generated as soon as the error hits a pre-defined threshold $e_t \geq e_{thresh.}$.

However, since e is usually rather noisy, further time series modelling, such as Kalman filtering [27] or simply applying a moving average on the time series e_t with $t \in [0, \ldots, T]$ might be necessary as a preprocessing step for the alerting algorithm.

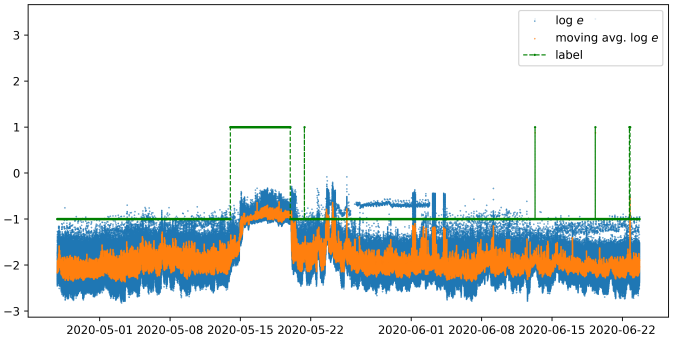

FIGURE 4. VISUALIZATION OF THE PERFORMANCE OF THE BASELINE MODEL FOR ANOMALY DETECTION

FIGURE 4 shows $\log(e)$ and its 5min moving averages over time for a random section of the dataset. The plot also holds the label information encoded as -1 for "normal" and 1 for "anomaly". We clearly see a correlation between the label signal and the moving average of the reconstruction error, especially at the longest anomaly around May 17th. This anomaly could have been detected e.g. by applying a simple threshold against the moving average. However, as expected, many of the very short anomalies can not be detected base on the moving average, even though the corresponding time stamps show a high reconstruction error. Another weakness of the baseline model is that it is not well suited to identify those signals that are primarily involved in the anomaly. However, this feature is needed to generate the output relevant for the diagnosis.

B. Results diagnosis

As a trigger event, the detected anomaly starts the diagnosis algorithm. The algorithm will first perform a preprocessing of rules, representing the system, and afterwards carry out a diagnosis on the symptoms of the system. In an exemplary investigation, a fault in the inter module ventilation supply (IMV Supply) was simulated. To identify the possible root causes in this example, two iterations were necessary. Within the first iteration it was possible to identify a partial diagnosis, namely the Condensate Heat Exchanger (CHX), that was accordingly excluded in a second iteration from the solution set of possible components to be diagnosed. The second iteration of diagnosis pointed out the remaining minimal diagnoses to explain the fault represented by CFA1 and CFA2. A discussion with the ECLS system expert validated the root cause analysis and confirmed it as a possible engineering analysis outcome.

VI. CONCLUSION

In this article, we have presented the current research status of the (K)ISS project. In particular, we focused on the areas of anomaly detection and diagnosis. For the anomaly detection, we introduced our base-line model, which is an autoencoder neural network. We showed that even this neural networks with the relatively simple architectures already deliver relatively good initial results in regard to the identification of faults. However, this baseline model also shows limitations, especially in the localization of the faults. For this reason, further steps are planned for the development of more advance models, such as (i) sequence models, (ii) system state aware models, and (iii) ensemble models to cover both short-term and log-term patterns. In addition, the rule-based alerting as well as the deployment strategy are also future research topics.

For diagnosis, we presented a first approach of model-based diagnosis. At this phase of the project, we showed an automated root cause analysis can be carried out, which causal dependencies of the system as input. These causal dependencies were extracted manually by use of the system description. However, the testing of the algorithm has to be investigated through additional failure simulations as test scenarios. Besides that, further steps into the proposals for diagnostic hypothesis based on past engineering analysis and confirmed root causes are needed, in order to give a user or a possible automated reconfiguration service a comprehensive database for the initiation of countermeasures. Besides that, further steps in the development to a recommender service for diagnostic hypothesis for a future user should be investigated, that matches the proposed diagnoses with those performed in the past. The recommender system can be seen as an additional service that takes past engineering analysis and approved diagnosis from the algorithm to be able to select the most promising diagnosis on an even larger information basis.

ACKNOWLEDGMENT

This research paper out of the project (K)ISS is funded by dtec.bw – Digitalization and Technology Research Center of the Bundeswehr.

REFERENCES

[1] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, and Others, "Apache spark: a unified engine for big data processing," *Commun. ACM*, vol. 59, no. 11, pp. 56–65, 2016.

[2] M. Rocklin, "Dask: Parallel computation with blocked algorithms and task scheduling," in *Proceedings of the 14th python in science conference*, vol. 130, 2015, p. 136.

[3] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M. I. Jordan, and Others, "Ray: A distributed framework for emerging {AI} applications," in *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, 2018, pp. 561–577.

[4] A. Garg, W. Zhang, J. Samaran, R. Savitha, and C.-S. Foo, "An evaluation of anomaly detection and diagnosis in multivariate time series," *IEEE Trans Neural Netw Learn Syst*, vol. PP, Aug. 2021.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via theEMAlgorithm," *J. R. Stat. Soc.*, vol. 39, no. 1, pp. 1–22, Sep. 1977.

[6] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," *Adv. Neural Inf. Process. Syst.*, vol. 12, 1999.

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.
[8] P. Barmpoutis, P. Papaioannou, K. Dimitropoulos, and N. Grammalidis, "A review on early forest fire detection systems using optical remote sensing," *Sensors*, vol. 20, no. 22, p. 6442, 2020.
[9] S. Bai, J. Zico Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," http://arxiv.org/abs/1803.01271v2, 2018.
[10] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for Robot-Assisted feeding using an LSTM-Based variational autoencoder," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1544–1551, Jul. 2018.
[11] B. Zhou, S. Liu, B. Hooi, X. Cheng, and J. Ye, "BeatGAN: Anomalous rhythm detection using adversarially generated time series," in *IJCAI*, 2019, pp. 4433–4439.
[12] N. Hranisavljevic, A. Maier, and O. Niggemann, "Discretization of hybrid CPPS data into timed automaton using restricted boltzmann machines," *Eng. Appl. Artif. Intell.*, vol. 95, p. 103826, Oct. 2020.
[13] V. Fortuin, M. Hüser, F. Locatello, H. Strathmann, and G. Rätsch, "Deep self-organization: Interpretable discrete representation learning on time series," in *ICLR*, 2019.
[14] P. Struss and O. Dressler, ""physical negation" integrating fault models into the general diagnostic engine." in *IJCAI*, vol. 89, 1989, pp. 1318–1323.
[15] J. De Kleer, A. K. Mackworth, and R. Reiter, "Characterizing diagnoses and systems," *Artificial intelligence*, vol. 56, no. 2-3, pp. 197–222, 1992.
[16] A. Feldman, G. Provan, and A. Van Gemund, "Approximate model-based diagnosis using greedy stochastic search," *Journal of Artificial Intelligence Research*, vol. 38, pp. 371–413, 2010.
[17] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh *et al.*, "Sympy: symbolic computing in python," *PeerJ Computer Science*, vol. 3, p. e103, 2017.
[18] L. d. Moura and N. Bjørner, "Z3: An efficient smt solver," in *International conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2008, pp. 337–340.
[19] C. W. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica: journal of the Econometric Society*, pp. 424–438, 1969.
[20] T. Schreiber, "Measuring information transfer," *Physical review letters*, vol. 85, no. 2, p. 461, 2000.
[21] C. Kühnert, *Data-driven methods for fault localization in process technology*. KIT Scientific Publishing, 2013, vol. 15.
[22] P. Struss, "Fundamentals of model-based diagnosis of dynamic systems," in *IJCAI (1)*, 1997, pp. 480–485.
[23] K. Potdar, T. S. Pardawala, and C. D. Pai, "A comparative study of categorical variable encoding techniques for neural network classifiers," *Int. J. Comput. Appl. Technol.*, vol. 175, no. 4, pp. 7–9, 2017.
[24] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.
[25] J. Pearl, *Causality*. Cambridge university press, 2009.
[26] J. Doyé, "An advanced columbus thermal and environmental control system," in *SpaceOps 2012*, 2012, p. 1292285.
[27] G. Welch, G. Bishop *et al.*, "An introduction to the kalman filter," 1995.