

Article

Not peer-reviewed version

Declarative Programming Approaches for Robust Anomaly Detection in HPDC Process Data

Tomasz Piotr Michno^{*}, Roxana-Maria Holom, Sebastian Schmalzer, [Pauline Meyer-Heye](#), Giulia Scampone, Elias Riegler, Matthias Hartmann, Urban Repanšek, Nejc Košir, Peter Šifrer

Posted Date: 2 February 2026

doi: 10.20944/preprints202602.0035.v1

Keywords: declarative programming; anomaly detection; HPDC process data



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Declarative Programming Approaches for Robust Anomaly Detection in HPDC Process Data [†]

Declarative Programming for HPDC Anomaly Detection

Tomasz Piotr Michno ^{1,*} , Roxana-Maria Holom ² , Sebastian Schmalzer ² , Pauline Meyer-Heye ¹ , Giulia Scampone ^{1,3} , Elias Riegler ^{1,3}, Matthias Hartmann ^{1,3} , Urban Repanšek ⁴, Nejc Košir ⁴ and Peter Šifrer ⁴

¹ AIT Austrian Institute of Technology GmbH, Austria

² RISC Software GmbH, Austria

³ LKR Light Metals Technologies, Austria

⁴ LTH Castings d.o.o., Slovenia

* Correspondence: Tomasz.Michno@ait.ac.at

[†] Current address: This preprint has not undergone peer review (when applicable) or any post-submission improvements or corrections. The Version of Record of this contribution is published in Lecture Notes in Networks and Systems, and is available online at https://doi.org/10.1007/978-3-032-05745-7_4

Abstract

The increasing demand for lightweight components with complex geometries and high performance has significantly driven the use of High Pressure Diecasting technologies in the transportation sector. However, the production of high-quality HPDC components necessitates precise control of process parameters. Within this scope, the Data and Metadata for Advanced Digitalization of Manufacturing Industrial Lines (metaFacturing) project, funded by the EU Horizon program, is trying to solve some of the challenges. The focus is on a digitalized toolchain that will optimize the use of raw materials, incorporating recycled ones, reduce operator costs and effort, as well as waste caused by out-of-specification production results. In this paper, a solution for the challenge of the structured data characteristics is presented using two differently trained classifiers for anomalous and non-anomalous data in order to improve classification performance. To additionally increase efficiency and utilize process data characteristics, different declarative programming methods are investigated, such as ILP, ASP, CLP, and CP.

Keywords: declarative programming; anomaly detection; HPDC process data

1. Introduction

In recent decades, the demand for lightweight components in the transportation sector has driven the widespread use of High Pressure Die Casting (HPDC). This casting process enables the high-volume production of complex components with thin wall thicknesses and good mechanical properties. However, a precise control of casting parameters is essential to ensure the production of high-quality parts. In order to improve the process settings and product quality, shorten the production time, reduce material waste and environmental impact, manufacturers, as well as researchers are seeking for new solutions, especially in the area of Industry 4.0. Within this scope, the Data and Metadata for Advanced Digitalization of Manufacturing Industrial Lines (metaFacturing) project funded by the EU Horizon program addresses some of the challenges. metaFacturing focuses on the development of a digitized toolchain for metal part production which will optimize the use of raw materials incorporating recycled ones, reduce the operator cost and effort, as well as waste caused by out-of-specification production results.

To validate the digitalized toolchain for strategic applications, a HPDC production line-based demonstrator was prepared at LTH Castings factory, and the formation of defects was monitored

under varying casting parameters. Although several microstructural defects can be formed during the HPDC process [7], the most common defects leading to failure of the component include gas porosities, shrinkage porosities, and cold runs. Several methods can be employed to detect these defects, such as X-ray inspection, ultrasonic testing, dye penetrant testing, and leakage testing. The latter is particularly relevant for HPDC components required to withstand different operating pressures and is typically performed through water immersion, air or helium leak detection.

Central aspects of the development of the casting demonstrator included data generation and fusion of all relevant data throughout the production chain. Data available from the HPDC process is summarized as follows:

- Process control parameters and machine measurements - structured data (one value for each parameter/measurement per die casting shot): biscuit thickness, maximum metal pressure, average piston speed, level of melt, leakage of chamber, etc.;
- Machine configuration parameters - time series data reported by the HPDC machine for each die casting shot: configured plunger velocity and configured metal pressure;
- Actual profiles data - time series data reported by the HPDC machine for each die casting shot: actual plunger position, velocity, and metal pressure;
- Thermal images (image data) generated by an infrared camera for each die casting shot;
- Quality control information for each metal part - structured data (one value per die casting shot).

The quality status of a cast part can be either determined automatically by the HPDC machine, if the process control parameters/measurements are out of tolerance, through visual inspection by machine operators and by the leak testing machine (at a different stage in the production). Leakage testing efficiently detects defects that could lead to component failure. Furthermore, it facilitates the assessment of casting parameters suitable for the production of high quality components. Because the cost of scrap increases as the production process progresses, it is important to eliminate defective parts as early as possible. Since casting is the first operation in serial production and leakage test occurs near the end, there is an opportunity to reduce scrap costs. Predicting the quality status during the casting process would help to achieve this goal.

HPDC is a complex process with a large number of process variants, non-linear correlations and differences between simulation and production. Monitoring and optimizing it is challenging for the human operator. In this paper, we focus on analyzing the structured data from the process (i.e., process control parameters, machine measurements, and the corresponding quality control information). The other types of data are being investigated separately in the project.

One of the challenge in the structured data analysis is the fact that defects occur in various forms and can have different characteristic, both connected to parameters deviations and other factors. To address this problem, we propose to use two separately trained classifiers for anomalous and non-anomalous data in order to improve the classification performance.

In order to find anomalies in the data, also called outliers, many techniques can be used. For example, for high-dimensional data in an Euclidean space, unsupervised methods can be used, especially with taking into account subspaces [23], like Axis-Parallel Subspaces [10]. There are also methods that find anomalies using information about the neighborhood [23], for example kNN [17] or LoOP [11]. Some of them also use the distance measurement from the defined number of neighbors or sliding windows [17,20]. Another method is to ensemble different anomaly detection results [23], for example, using different subsets of subsamples and features [14]. Other techniques use Support Vector Machines or Deep Belief Networks [5,20] or consider mixed attributes anomaly detection [23].

All of these methods have advantages and disadvantages. Subspace-based approaches are very efficient, however it is not easy to find the right subspace. Neighborhood information-based algorithms are not affected by the data distribution, but mostly they do not perform well and are sensitive to the parameters used. Methods based on ensembling results may have high accuracy but may be inefficient. With mixed attributes, anomaly detection algorithms can achieve high accuracy. However, they usually have high complexity and it is difficult to obtain correlation structures of features [23].

Characteristic of the HPDC process data can be also taken into consideration, like information about correlated sensors as well as e.g. their known "safe" ranges. One of the methods that can be used to incorporate this information is declarative programming, which is successfully used in many areas like project resource allocation [21], carrying out multimodal processes flow [3], but also errors diagnosis [4] and anomaly detection. In the area of anomaly detection different declarative programming methods can be used, like for example Answer Set Programming (ASP) [2], Constraint Programming methods [12] or Logical Programming [1]. The advantages of using declarative programming are: robust constraints handling [8,18], optimization-driven anomaly detection [9,22] and advanced logical reasoning [6].

2. Anomalies Detection for Sensors Data

The anomalous data in the HPDC production process can indicate situations that are less standard and, in some cases, may lead to defects in the produced samples. However, due to the specificity of the process, not all outliers lead to defects, so they also have to be classified.

The main idea and motivation of this paper is to provide a solution, which will improve the overall precision of data classification by training separate classifiers for anomalous and non-anomalous subsets. Such an approach should allow the use of different classifiers for each subset, or even additional data sources (like profile data or thermal images) in order to provide more insight into the process.

2.1. Data Preprocessing

In order to analyze the data gathered from experiments further preprocessing is needed. In the training phase, the first step includes normalization of the values, then finding in the training dataset the samples that are classified as good, followed by computing the average value for each sensor. As a result, a *master* set of reference values is created. Next, the initial thresholds that define whether a value may be an anomaly or not need to be selected. Two approaches can be used here: one based on experts knowledge [19] or one calculated from the reference values. In order to reduce human subjective bias, the second approach has been chosen in this research, with standard deviation as a measure. The last step incorporates calculating the differences for each sample with master values. The procedure for prediction includes only the difference calculation step, using master data from the training.

2.2. Declarative Programming Approaches for HPDC Sensors Data Anomaly Detection

After the data preprocessing step, anomaly detection can be performed using one of the Declarative Programming approaches. For this research, the following methods have been selected: Constraint Programming with solver that uses SAT (satisfiability) methods (CP-SAT), Answer Set Programming (ASP), Constraint Logic Programming (CLP), and Integer Linear Programming (ILP). The advantage of using such methods is, for example, the possibility to define constraints for each sensor, support correlations between sensors, as well as use the advantages of optimization methods. For ASP and CLP, different versions were considered - one with less and one with more strict constraints.

The process of anomaly detection is similar for all methods and can be defined as follows:

- Given:
 - S - Set of sensors; M_s, T_s - Master and test readings for $s \in S$.
 - $\Delta_s = |T_s - M_s|$ - Absolute difference between readings; τ_s - Anomaly threshold.
 - $b_s \in \{0, 1\}$, $x_s \in [0, 1]$ - Binary and decimal anomaly indicators for s .
 - $w_s = \Delta_s \cdot \gamma$ - Scaled severity weight; γ - Scaling factor.
 - C - Set of correlated sensor pairs (s, s') ; α - Max allowed anomaly ratio.
 - N - Total number of sensors.
- Threshold-Based Constraints - for each sensor, a check is made against its predefined threshold. If the value is lower than the threshold, the sensor value is treated as non-anomalous, if it exceeds

the threshold, it can be a possible anomaly:

· CP-SAT, ASP (version 1), CLP (version 1):

$$b_s = 0, \quad \text{if } \Delta_s < \tau_s, \quad \forall s \in S. \quad (1)$$

· ILP (version 1: $c = 3$, version 2: $c = 1.5$):

$$x_s \geq 0.5, \quad \text{if } \Delta_s \geq c \cdot \tau_s, \quad \forall s \in S. \quad (2)$$

· ASP (version 2: $d = 3$, version 3: $d = 10$, $d = 35$), CLP (version 2: $c = 1$, $d = 4$):

$$b_s = 1, \quad \text{if } c \cdot \Delta_s \geq d \cdot \tau_s, \quad \forall s \in S. \quad (3)$$

- Correlation Constraints - if sensors s and s' are correlated, then their anomaly status should be the same:

· CP-SAT, ASP, CLP:

$$b_s = b_{s'}, \quad \forall (s, s') \in \mathcal{C}, \quad (4)$$

· ILP:

$$x_s = x_{s'}, \quad \forall (s, s') \in \mathcal{C}. \quad (5)$$

- Anomaly Limit Constraints - the total number of detected anomalous sensor values cannot exceed the predefined maximum number (if it exceeds, it may, for example be caused by defective sensor readouts):

· CP-SAT, ASP, CLP:

$$\sum_{s \in S} b_s \leq \alpha N. \quad (6)$$

· ILP:

$$\sum_{s \in S} x_s \leq \alpha N. \quad (7)$$

- Objective functions:

· CP-SAT - maximize the severity of detected anomalies:

$$\max \sum_{s \in S} w_s \cdot b_s, \quad (8)$$

· ASP - all versions - minimize the number of detected anomalies while maintaining the constraints:

$$\min \sum_{s \in S} b_s. \quad (9)$$

· CLP - version 1 and 2 - minimize the inconsistency in anomaly detection while maintaining the constraints:

$$\min \sum_{(s, s') \in \mathcal{C}} |b_s - b_{s'}| \quad (10)$$

· ILP - minimize the weighted number of detected anomalies, prioritizing large deviations:

$$\min \sum_{s \in S} \Delta_s \cdot x_s. \quad (11)$$

- Solvers: CP-SAT - using Google OR-Tools' CP-SAT solver; ASP - using clingo solver from The Potsdam Answer Set Solving Collection developed at the University of Potsdam; CLP - using Prolog with CLP(FD) library for finite domain constraints; ILP - using PuLP with CBC solver for integer linear programming.

3. Experiments

The experiments have been performed using a prototype application written in Python and utilizing following libraries for anomaly classifiers: sklearn, ortools.sat.python, clingo, pyswip and scipy.optimize. The following methods have been tested using experimental application: CP-SAT, CLP (two versions - one with less strict constraints and one with more strict constraints), ASP (three versions - with less and more strict constraints), ILP (two versions - one with less strict constraints and one with more strict constraints), and for comparison Local Outliers Factor (LOF). For training, 4093 production HPDC process data points has been used (3707 with class "good" and 386 with class "bad" - a 0.10 "bad" to "good" proportion). For testing the performance of the trained classifiers, 1024 sample data points have been used (926 with class "good" and 98 with class "bad" - a 0.11 "bad" to "good" proportion).

Firstly the performance of the anomaly detection algorithms has been tested. The train and test dataset results for each algorithm are shown in Table 1. As it can be seen, each method divided the data differently, with CP-SAT, CLP (both versions), ASP version 1 and LOF having the closest proportions to the classes proportions in the train dataset and ASP version 1 and LOF for the test dataset. An interesting result has been obtained for the CP-SAT method, where for the test data all elements were classified as anomalous.

Table 1. Comparison of anomalous and non-anomalous subsets detection for each method for training and test dataset.

Method	Train			Test		
	Anom.	Non-anom.	Proportions	Anom.	Non-anom.	Proportions
CP-SAT	1816	2277	0.80	1024	0	-
CLPv1	1839	2254	0.82	953	71	13.42
CLPv2	384	3709	0.10	75	949	0.08
ASPv1	1721	2372	0.73	457	567	0.81
ASPv2	171	3922	0.04	45	979	0.05
ASPv3	154	3939	0.04	40	984	0.04
ILPv1	4031	62	65.02	1015	9	112.78
ILPv2	3980	113	35.22	856	168	5.10
LOF	633	3460	0.18	164	860	0.19

Further experiments incorporated the following classification algorithms for each subset: SVM, Random Forest, K-Nearest Neighbors, Gaussian Naïve Bayes, Logistic Regression, Genetic Algorithm, XGBoost, and LightGBM. Due to limited space only the best scores are shown in Table 2, with a focus on the "bad" class, but the experiments were performed for all mentioned methods and classifiers. Because the datasets from the production HPDC process are imbalanced, the following performance metrics have been selected: F1-score, Balanced Accuracy (BA) and Matthews Correlation Coefficient (MCC).

As can be seen in Table 2, for anomalous data, four methods obtained quite high values for each metric: ILP version 1, LOF, ASP versions 2 and 3. The highest values were obtained by ASP version 2. For non-anomalous data high values were achieved by: ILP version 1, LOF, ASP version 2 and 3. The highest values were obtained by ILP versions 1 and 2, but since all the metric values were equal to 1, this may suggest overfitting of the classifier. Apart from ILP, the ASP version 3 performance was the second highest.

Taking into consideration the data classification algorithms, the highest values were obtained using the XGBoost and LightGBM algorithms. In most cases, the worst results were achieved using Gaussian Naïve Bayes, which for this process data varied between 0.20 and 0.76, with an average of 0.45 for the MCC.

Table 2. Comparison of methods for anomalous, non-anomalous, and all training dataset

Anomalous Data					
Method	Algorithm	F1-Score Good	F1-Score Bad	BA	MCC
ILPv1	LightGBM	0.9793	0.7347	0.7956	0.7398
ASPv1	LightGBM	0.9601	0.7143	0.7804	0.7109
CLP	LightGBM	0.9590	0.7424	0.7975	0.7332
CP-SAT	XGBoost	0.9735	0.6835	0.7723	0.6821
LOF	LightGBM	0.9582	0.8308	0.8609	0.8019
ASPv2	XGBoost	0.9434	0.9189	0.9250	0.8712
CLPv2	LightGBM	0.9517	0.8372	0.8600	0.8085
ILPv2	LightGBM	0.9743	0.6522	0.7484	0.6650
ASPv3	XGBoost	0.9362	0.9091	0.9167	0.8563
Non-Anomalous Data					
ILPv1	LightGBM	1.0000	1.0000	1.0000	1.0000
ASPv1	XGBoost	0.9863	0.6154	0.7222	0.6576
CLP	LightGBM	0.9878	0.3158	0.5938	0.4278
CP-SAT	-	0.0000	0.0000	0.0000	0.0000
LOF	LightGBM	0.9773	0.5747	0.7043	0.6126
ASPv2	XGBoost	0.9771	0.6557	0.7542	0.6645
CLPv2	XGBoost	0.9762	0.6154	0.7274	0.6386
ILPv2	LightGBM	1.0000	1.0000	1.0000	1.0000
ASPv3	LightGBM	0.9788	0.6880	0.7676	0.7003
Trained on all training dataset for specific test data					
Test Data	Algorithm	F1-Score Good	F1-Score Bad	BA	MCC
All	LightGBM	0.9767	0.7179	0.7846	0.7245
ASPv2 Anom.	LightGBM	0.9259	0.8889	0.9000	0.8305
ASPv2 Non-anom.	LightGBM	0.9782	0.6667	0.7553	0.6823
ASPv3 Anom.	LightGBM	0.9565	0.8333	0.8571	0.8092
ASPv3 Non-anom.	LightGBM	0.9783	0.6833	0.7651	0.6958

The most important comparison was made with other anomaly detection and data classification algorithms. In order to test how the presented approach performs in comparison with the other outlier detection method, Isolation Forest and Local Outliers Factor (LOF) were used. However, reasonable results were obtained only for LOF, whereas the Isolation Forest, even with different parameter setups was not able to divide the data into two subsets, so it was omitted.

For the anomalous dataset, ASP versions 2 and 3 obtained higher F1-Score for "bad" class, as well as higher BA and MCC (e.g. MCC higher by 8.4% for ASPv2 and 6.79% for ASPv3). For the F1-Score for "good" class, the values were a bit lower than for LOF, but as mentioned before, because of the imbalanced dataset, the BA and MCC are more meaningful. For the non-anomalous dataset, when the ILP method is omitted, the highest values are obtained by ASP versions 1, 2 and 3 for F1-Score for "bad" class, as well as BA and MCC, and only slightly lower F1-Score for "good" class for ASP version 2. In case of the MCC metric, accordingly for ASP versions 1, 2 and 3, 7.35%, 8.47 and 14.32% higher values than for the LOF were achieved.

Another test was conducted to evaluate the performance in comparison with the classifier trained on the whole dataset. The highest performance for such a classifier was obtained by LightGBM, and in order to compare results, it was tested with anomalous and non-anomalous datasets obtained from ASP versions 2 and 3. Compared to the ASP version 2 datasets, for the anomalous subset, the ASP version 2 classifier was 2.78% (BA) and 4.90% (MCC) better. However, for the non-anomalous subset, the results were 0.15% (BA) and 2.62% (MCC) worse. Compared to ASP version 3 datasets, for the anomalous subset, the ASP version 3 classifier was 6.94% (BA) and 5.83% (MCC) better, with a slightly lower F1-score for the "good" class (by 2.13%) and 9.09% higher F1-score for the "bad" class. Additionally, for the non-anomalous subset, the results were only slightly better, accordingly by 0.34% (BA) and 0.65% (MCC).

4. Conclusion and Future Work

In this paper different methods of declarative programming have been presented as a solution for anomaly detection in HPDC production process data. The following methods have been considered: CP-SAT, CLP, ASP and ILP as anomaly detectors, together with SVM, Random Forest, K-Nearest Neighbors, Gaussian Naïve Bayes, Logistic Regression, Genetic Algorithm, XGBoost, and LightGBM, as classifiers. Presented results prove that the main idea of this research allows for obtaining at least not worse results for both anomalous and non-anomalous subsets than the classifier trained for the whole dataset. Additionally, for anomalous subset, the metrics can achieve noticeably higher values, with only slightly lower performance for the non-anomalous set for ASP version 2, and mostly equal performance for ASP version 3. The most promising method in the performed tests was ASP version 3, which will be developed in future research.

However, presented approach requires further investigation, especially testing with more process data collected during the project execution. Apart from the extended testing, more research should be done on the specific characteristics and capabilities of each method. Furthermore, incorporating results from other data sources, for example, profile data (using different approaches like Dynamic Time Warping [13] or forecasting models [16]) or thermal images (using e.g. deep learning [15]) is part of the next steps and will be covered in next publications.

Acknowledgments: Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them. The work described in this paper is supported by the metaFacturing project (GA 101091635), which has received funding under the Horizon Europe programme. This preprint has not undergone peer review (when applicable) or any post-submission improvements or corrections. The Version of Record of this contribution is published in Lecture Notes in Networks and Systems, and is available online at https://doi.org/10.1007/978-3-032-05745-7_4.

Conflicts of Interest: The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Angiulli, F., Greco, G., Palopoli, L.: Discovering anomalies in evidential knowledge by logic programming. *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)* **3229**, 578–590 (2004). doi:10.1007/978-3-540-30227-8_48
2. Bellusci, P., Mazzotta, G., Ricca, F.: Modelling the outlier detection problem in asp(q). *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **13165 LNCS**, 15–23 (2022). doi:10.1007/978-3-030-94479-7_2
3. Bocewicz, G., Nielsen, I., Wójcik, R., Banaszak, Z.: Declarative models of periodic distribution processes. *Lecture Notes in Mechanical Engineering* pp. 116–129 (2022). doi:10.1007/978-3-030-99310-8_10
4. Ceballos, R., Gasca, R.M., Valle, C.D., Borrego, D.: Diagnosing errors in dbc programs using constraint programming. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **4177 LNAI**, 200–210 (2006). doi:10.1007/11881216_22
5. Erfani, S.M., Rajasegarar, S., Karunasekera, S., Leckie, C.: High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition* **58**, 121–134 (10 2016). doi:10.1016/J.PATCOG.2016.03.028
6. Fichte, J.K., Hecher, M., Szeider, S.: Inconsistent cores for asp: The perks and perils of non-monotonicity. *Proceedings of the AAAI Conference on Artificial Intelligence* **37**, 6363–6371 (6 2023). doi:10.1609/AAAI.V37I5.25783, <https://ojs.aaai.org/index.php/AAAI/article/view/25783>
7. Gariboldi, E., Bonollo, F., Rosso, M.: Proposal of a classification of defects of high pressure diecast products. *La Metallurgia Italiana* pp. 39–46 (2007)
8. Hentenryck, P.V.: Constraint logic programming. *The Knowledge Engineering Review* **6**, 151–194 (1991). doi:10.1017/S0269888900005798
9. Kotecha, P.R., Bhushan, M., Gudi, R.D.: Efficient optimization strategies with constraint programming. *AIChE Journal* **56**, 387–404 (2 2010). doi:10.1002/AIC.11972

10. Kriege, H.P., Kröger, P., Schubert, E., Zimek, A.: Outlier detection in axis-parallel subspaces of high dimensional data. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **5476 LNAI**, 831–838 (2009). doi:10.1007/978-3-642-01307-2_86
11. Kriegel, H.P., Kröger, P., Schubert, E., Zimek, A.: Loop: Local outlier probabilities. *International Conference on Information and Knowledge Management, Proceedings* pp. 1649–1652 (2009). doi:10.1145/1645953.1646195
12. Kuo, C.T., Davidson, I.: A framework for outlier description using constraint programming. *Proceedings of the AAAI Conference on Artificial Intelligence* **30**, 1237–1243 (2016). doi:10.1609/AAAI.V30I1.10174
13. Müller, M.: Dynamic time warping. *Information Retrieval for Music and Motion* pp. 69–84 (2007). doi:10.1007/978-3-540-74048-3_4
14. Pasillas-Díaz, J.R., Ratté, S.: Bagged subspaces for unsupervised outlier detection. *Computational Intelligence* **33**, 507–523 (8 2017). doi:10.1111/COIN.12097
15. Pierdicca, R., Paolanti, M., Felicetti, A., Piccinini, F., Zingaretti, P.: Automatic faults detection of photovoltaic farms: solair, a deep learning-based system for thermal images. *Energies* 2020, Vol. 13, Page 6496 **13**, 6496 (12 2020). doi:10.3390/EN13246496
16. Poczeta, K., Papageorgiou, E.I.: Energy use forecasting with the use of a nested structure based on fuzzy cognitive maps and artificial neural networks. *Energies* 2022, Vol. 15, Page 7542 **15**, 7542 (10 2022). doi:10.3390/EN15207542
17. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. *ACM SIGMOD Record* **29**, 427–438 (5 2000). doi:10.1145/335191.335437
18. Rossi, F., van Beek, P., Walsh, T.: Chapter 4 constraint programming. *Foundations of Artificial Intelligence* **3**, 181–211 (1 2008). doi:10.1016/S1574-6526(07)03004-0
19. Rötzer, F., Göbel, K., Liebetreu, M., Strommer, S.: Knowledge graph extraction from retrieval-augmented generator: An application in aluminium die casting. In: *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics - Volume 2: ICINCO*. pp. 365–376. INSTICC, SciTePress (2024). doi:10.5220/0012951000003822
20. Thudumu, S., Branch, P., Jin, J., Singh, J.J.: A comprehensive survey of anomaly detection techniques for high dimensional big data. *Journal of Big Data* **7**, 1–30 (12 2020). doi:10.1186/S40537-020-00320-X/TABLES/6
21. Wikarek, J., Sitek, P., Banaszak, Z.: Decision support model for the configuration of multidimensional resources in multi-project management. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **12876 LNAI**, 290–303 (2021). doi:10.1007/978-3-030-88081-1_22
22. Wolsey, L.A.: *Integer programming*. Wiley (2021)
23. Xu, X., Liu, H., Yao, M.: Recent progress of anomaly detection. *Complexity* **2019**, 2686378 (1 2019). doi:10.1155/2019/2686378

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.