Towards Highly Efficient Anomaly Detection for Predictive Maintenance

Simon Klüttermann[1,2], Vanlal Peka[1], Philipp Doebler[1], Emmanuel Müller[1,2,3]
[1]*TU Dortmund University*, Dortmund, Germany
[2]*Lamarr Institute for Machine Learning and Artificial Intelligence*, Dortmund, Germany
[3]*Research Center Trustworthy Data Science and Security*, Dortmund, Germany
simon.kluettermann@cs.tu-dortmund.de

***Abstract*—This paper introduces SEAN, a novel anomaly detection algorithm designed for real-time applications in predictive maintenance. SEAN leverages an ensemble-based approach to deliver competitive performance while drastically reducing computational costs. In our comprehensive evaluation across 121 datasets, SEAN consistently outperforms comparable shallow anomaly detection algorithms. Our comparisons reveal that SEAN operates over 20,000 times faster than a similar state-of-the-art deep learning alternative, with negligible sacrifice in detection accuracy. We further demonstrate SEAN's versatility through an ablation study, highlighting how its hyperparameters can be tuned to balance runtime and performance effectively. Additionally, we present a practical C++ export tool that enables the deployment of SEAN on resource-constrained devices, meeting the stringent requirements of on-device predictive maintenance tasks. Our findings underscore SEAN as a powerful and efficient solution for anomaly detection in real-world engineering applications.**

***Index Terms*—Predictive Maintenance, Anomaly Detection, Fault Detection, Predictive Analytics**

I. Introduction

Predictive maintenance leverages machine learning to anticipate equipment failures before they occur, offering a significant advantage over traditional methods like scheduled maintenance. By identifying potential issues early, predictive maintenance minimizes costs, reduces downtime, extends equipment lifespan, and enhances safety [56]. This approach has become indispensable across various industries, including manufacturing [58], aerospace [49], and energy infrastructure [27].

A key technique in predictive maintenance is anomaly detection, as many equipment failures manifest as anomalies [14]. Thus, advancements in anomaly detection algorithms are critical to improving predictive maintenance outcomes. However, while academic research often prioritizes the development of highly accurate but complex neural networks for anomaly detection [43], practical applications have different demands. In real-world scenarios, predictive maintenance systems require algorithms that are not only accurate but also exceptionally fast [16], given that anomalies may only be detectable moments before a critical failure. Additionally, deploying algorithms that rely on large neural networks can be prohibitively expensive and may necessitate external hardware, thereby limiting their scalability and widespread adoption [38]. While faster, non-deep learning-based anomaly detection algorithms exist, many of these methods [25], [37], [43] were developed prior to the deep learning era and have not kept pace with modern advancements.

To address these challenges, we propose an innovative solution by adapting a state-of-the-art neural network-based ensemble anomaly detection method, *DEAN* [7]. We demonstrate that by substituting the neural networks with simple linear regressions, our algorithm becomes several orders of magnitude faster while experiencing only a minor reduction in anomaly detection performance. We introduce this new approach as *SEAN* (Shallow Ensemble ANomaly detection), which not only outperforms comparable methods in both speed and accuracy but also retains DEAN's ensemble structure. This structure allows us to utilize the explainability technique outlined in [31] to efficiently approximate Shapley values [17], enabling the identification of which input features contribute most to an anomaly. This capability is crucial for further understanding and mitigating equipment failures.

Furthermore, we explore the impact of the number of submodels within SEAN that directly influences both its runtime and anomaly detection performance, offering the flexibility to prioritize either based on specific predictive maintenance requirements. Lastly, we present a method for converting the learned model into optimized C++ code, enabling on-device execution. We provide both this conversion method, an example of SEANs interpretability and more generally a Python implementation of SEAN at https://github.com/KDD-OpenSource/SEAN.

II. Related Work

Before presenting our method, we first discuss the relevant literature:

Predictive Maintenance [10], [14], [55] has proven to be a valuable strategy for minimizing operational disruptions, reducing costs, and extending the lifespan of infrastructure by addressing issues as they arise. Equipment failures can often be detected through anomaly detection techniques, as these failures typically present as anomalies [9], [15], [28]. The choice of anomaly detection algorithm varies depending on the application, but common requirements include real-time performance [16], scalability to large datasets [20], [42], and interpretability to simplify maintenance processes [33],

[50]. In this paper, we introduce a new method that excels in accuracy, speed, scalability, and interpretability, providing valuable insights into the causes of anomalies to aid in their resolution.

Anomaly detection [48] involves identifying unusual or unexpected events, with applications ranging from fault detection [18], [40], [45] to network security [29], [41], [52] and scientific research [1], [12], [39]. These tasks can be approached in supervised, semi-supervised [47], [53], or entirely unsupervised manners [30], depending on the availability of labeled anomaly data. In our work, we focus on the unsupervised scenario, which is particularly relevant in engineering applications where not all possible faults are known in advance. Numerous algorithms have been proposed for unsupervised anomaly detection [46], with relatively few performance differences observed among them [23]. This allows us to prioritize secondary objectives beyond detection accuracy. One such objective, often overlooked [43], is the speed of the anomaly detection algorithm—a crucial factor in engineering contexts where real-time decision-making is essential for cost efficiency and safety. In this paper, we propose an algorithm that significantly outpaces existing methods while maintaining competitive performance.

Extremely-fast Anomaly Detection methods are typically simple, often relying on deviations from basic statistical measures like the mean to indicate anomalies. While such approaches are quick, they fall short in performance compared to modern methods that detect deviations from complex normal data. For tasks where safety is critical, more sophisticated yet still fast algorithms are necessary. Shallow methods, such as the Isolation Forest [37], which uses simple random decision trees to isolate anomalies, or CBLOF [25], which clusters normal data and measures deviations from cluster centers, are examples of effective shallow approaches. Due to the linear time complexity of the Isolation Forest, it scales well to larger data. CBLOF is relatively efficient, but not as fast. Because of the ease of implementation, speed, and quality, both are exceptional benchmark methods for ultra-fast AD. However, partially due to limited recent research interest in shallow anomaly detection, these methods are often outperformed by more advanced deep learning techniques. In this paper, we introduce a new shallow method inspired by a recent deep learning approach, which not only surpasses other shallow methods but is also even faster compared to them.

DEAN [7] is a recent anomaly detection algorithm that extracts patterns from normal data and contrasts them against potentially contaminated data. DEAN achieves high performance by combining information from multiple patterns into an ensemble [11], [32], and it is designed to be both highly effective and easily modifiable. The algorithm uses neural networks to encode the pattern extraction task into a loss function. In our work, we leverage DEAN's modularity to replace the neural networks with much faster linear models, resulting in a similarly effective but significantly faster algorithm. Additionally, by building on DEAN, we inherit several of its enhancements, notably the interpretability afforded by polynomial-time Shapley values [31], which is crucial for identifying the root causes of faults in maintenance tasks.

III. Methodology

In this section, we first introduce DEAN and then explain the methodology behind our proposed algorithm, SEAN.

A. DEAN

DEAN (Deep Ensemble ANomaly detection) is a recently proposed anomaly detection algorithm [7]. It functions as an ensemble method that combines several simple anomaly detection models. Each of these models is a neural network trained using the loss function defined in Equation 1.

$$L = \sum_{\mathbf{x} \in X_{normal}} (f_i(\mathbf{x}) - 1)^2 \qquad (1)$$

The loss is non-trivial because functions f_i do not include bias terms (intercepts), i.e., they cannot learn constants. Consequently, the approach learns approximately constant combination of input features, thereby capturing patterns within the normal samples.

By training multiple functions f_i, DEAN extracts various patterns, resulting in a clearer specification of normal behavior. Anomalies are then identified based on an anomaly score, as shown in Equation 2, where the hyperparameter $power$ determines how multiple small deviations are aggregated. The continuous nature of the anomaly score allows for flexible thresholding, facilitating the optimization of false negative and false positive rates depending on the application:

$$score(\mathbf{x}) = \sum_i |f_i(\mathbf{x}) - 1|^{power} \qquad (2)$$

The values $score(\mathbf{x})^{1/power}$ are the vector norms of the discrepancies of the $f_i(\mathbf{x})$ values from 1. For example, $power = 2$ yields the euclidian norm, while $power \to \infty$ results in the maximum norm, i.e., the score is the maximum of all $|f_i(\mathbf{x}) - 1|$. We remove this power, since monotonous transformations do not alter the anomaly ranking.

To enhance model diversity, DEAN employs feature bagging [34], which increases variance among the models by limiting each to a random subset of the available features.

DEAN's ensemble structure is designed for usability and modifiability. For instance, the algorithm can explain which features contribute most to the anomaly score. As shown in [31], this is achieved by calculating the importance of each feature locally at $\mathbf{x}$:

$$importance(\mathbf{x}, feature) = \sum_{i; feature \in f_i} |f_i(\mathbf{x}) - 1|^{power}$$

where in an abuse of notation $feature \in f_i$ indicates that $feature$ is in-bag. This expression is a linear function of the (bagged) Shapley value, a common interpretability method [17] and a crucial aspect for maintenance tasks [49].

B. SEAN

The primary distinction between DEAN and our proposed algorithm, SEAN (Shallow Ensemble ANomaly detection), lies in the replacement of neural networks with linear regression models. In SEAN, each function f_i is a linear regression where the target value (dependent variable, label) is constantly equal to 1. We observe that the loss function in Equation 1 is equivalent to separately fitting mean squared error linear regressions. Let $X_{normal,i}$ denote the feature vectors used for the training of the ith function. The parameters $\boldsymbol{\alpha}_i$ (=coefficients) that minimize the SEAN loss function

$$\min_{\alpha_i} \sum_{x \in X_{normal,i}} (\alpha_i^T \mathbf{x} - 1)^2 \tag{3}$$

are found by standard techniques for least squares regression. For relatively few features the textbook solution to this optimization problem requires only matrix inversion instead of neural network training:

$$\alpha_i = (X_{normal,i}^T \cdot X_{normal,i})^{-1} \cdot X_{normal,i}^T \cdot \mathbf{1}, \tag{4}$$

where $\mathbf{1}$ denotes a vector of ones. Equation 4 is numerically unstable, so singular value decomposition (SVD) is often preferred.

We added a model indice i to the training data, as when the features $X_{normal,i}$ would be identical for all i, the process would be deterministic. So instead, we apply feature bagging [34], similar to DEAN, to ensure diversity among the α_i values. Additionally, we incorporate subsampling [57], training each model on a random subset of samples, further increasing variance by training each submodel on its own slightly different dataset. Both feature bagging and subsampling not only enhance model diversity but also reduce computation time. Feature bagging decreases the number of features used in Equation 4, while subsampling reduces the number of samples. Given the complexity of $O(samples \cdot features^2)$ in Equation 4, these techniques significantly lower computational costs. Additionally, by limiting both the number of training samples used and the number of features, the runtime of SEAN scales well to large datasets.

To further improve performance with only a few submodels, we introduce a variation of feature bagging where instead of selecting a random subset of bag features for a submodel, we multiply the training data matrix (size $samples \times features$) by a matrix F (size $features \times bag$), where each element of F is randomly drawn from a normal distribution with mean zero and variance one, as shown in Equation 5. For this, we are inspired by reservoir computing [54], where a similar approach is used to allow simple models to learn more complicated functions. Because we have seen that variations in the average value of F can affect the model, especially for low features, we normalize F, by subtracting the mean of each row from F and thus setting the mean to be exactly 0 for each generated feature.

$$X_{train} \rightarrow X_{train} \cdot F \tag{5}$$

We will explore the impact of different choices for F, not necessarily with such standard normal entries, in our ablation studies in Section IV-B.

Like DEAN, SEAN identifies anomalies using an anomaly score, as defined in Equation 2. To further enhance speed, we replace the summation with a maximum operator:

$$score(x) = \max_i |\alpha_i \cdot x - 1| \tag{6}$$

C. Hyperparameters

In the following sections, we evaluate SEAN using a specific set of hyperparameters, which represent a balanced trade-off between performance and runtime. Additional hyperparameters are discussed in Section IV-B.

First, similar to DEAN [7], each feature is normalized so that $\max(X_{normal}^{feature}) = 1$ and $\min(X_{normal}^{feature}) = 0$. The SEAN ensemble consists of 50 submodels, each trained on 1000 randomly drawn samples and using $bag = 10\%$ of the available features. If the number of features is too high or too low, we clip bag to ensure it remains within the range $2 \leq bag \leq 100$. Similarly when fewer than 1000 normal samples are available, we use all available samples.

IV. Experiments

In this section, we comprehensively evaluate and benchmark SEAN, adhering to the methodology of a recent anomaly detection benchmark study, ADBench [23]. We conduct comparisons across the 121 datasets recommended by ADBench, including 57 uncorrelated datasets. Additionally, we compare SEAN against various shallow anomaly detection algorithms benchmarked in ADBench, such as distance-based algorithms (e.g., KNN [13] and LOF [6]), density-based algorithms (e.g., COPOD [35], HBOS [21], and ECOD [36]), and other approaches like OCSVM [3], LODA [44], PCA [8], IForest [37], and CBLOF [25]). Given the similarities in methodology, we also compare SEAN with DEAN [7].

In contrast to ADBench, we assume a scenario where a set of uncontaminated samples is available (One-class setting [30]), a common situation in predictive maintenance tasks where normal operation data is abundant. This assumption enhances the performance of anomaly detection models significantly [46]. For performance evaluation, we use the AUC-ROC metric [5], [24], which is the most widely adopted metric in anomaly detection research. Although other metrics like AUC-PR [4] or F1-Score [22] werde studied, results were vastly consistent with AUC-ROC, and we omit them here for brevity.

Moreover, we measure the runtime of each algorithm, restricting evaluations to CPU cores to avoid unfair advantages from GPU-supported implementations. Each evaluation is repeated 10 times, and the results are averaged to ensure robustness.

A. Performance Comparison

We begin by analyzing the runtime-performance trade-off for each algorithm, as illustrated in Figure 1. SEAN demonstrates slightly lower performance compared to DEAN but operates over 20,000 times faster, making it the fastest among all algorithms studied. SEAN outperforms all shallow algorithms, except for distance-based methods. However, distance-based algorithms are known to suffer from the curse of dimensionality [2], limiting their applicability to high-dimensional real-world data, despite their success in low-dimensional benchmark datasets. Additionally, these algorithms typically require calculating the distance between each test and each training sample, and thus, their runtime does not scale well to large datasets.

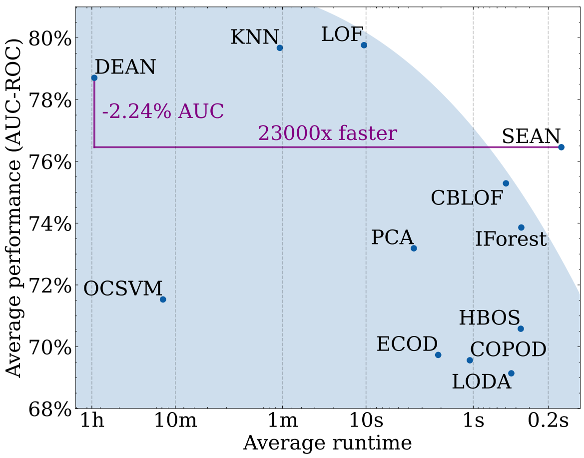

Fig. 1. Runtime (x-axis, logarithmic) and anomaly detection performance (y-axis, in AUC-ROC) averaged across all datasets. An optimal algorithm would be positioned in the top right corner. The blue area indicates the trade-off between speed and performance, suggesting a limit to how fast an algorithm can be while maintaining high performance.

Further performance analysis is provided in the critical difference plot in Figure 2. The plot reveals no significant performance differences between SEAN and other effective competitors, such as DEAN, CBLOF, IForest, and PCA, highlighting that anomaly detection performance alone may not be the most decisive factor in algorithm selection.

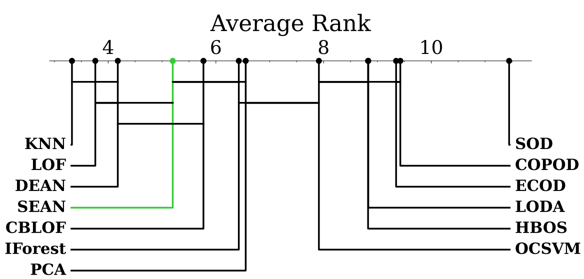

Fig. 2. Critical difference plot comparing the anomaly detection performance of the algorithms in Figure 1. A lower average rank is preferred. A Friedman test [19] is employed to determine significant differences in AUC-ROC performance, with a Wilcoxon test [51] connecting algorithms with no significant differences. Significance is determined using p-values below $p \leq 5\%$, after Bonferroni-Holm correction [26].

We also compare the average runtime required for training and inference by each algorithm, as shown in Figure 3. SEAN emerges as the most efficient, with the ability to train nearly 10 models per second on an average dataset. Its inference cost is similarly low, confirming SEAN as the overall fastest algorithm. Notably, even when excluding deep learning algorithms and the relatively slow OCSVM, the runtime differences between algorithms span multiple orders of magnitude, emphasizing the critical importance of an appropriate algorithm choice for real-time applications like predictive maintenance.

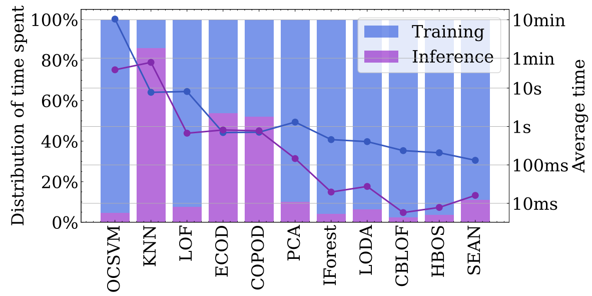

Fig. 3. Comparison of the runtime of algorithms in Figure 1. Training time and inference time are separated, with their relative magnitudes shown in the background.

B. Ablation Study

We investigate the impact of different hyperparameter choices on SEAN's performance, as depicted in Figure 4. The study reveals that hyperparameters allow for a trade-off between runtime and performance. The most influential hyperparameter is the number of submodels in the ensemble. Increasing the number of submodels by a factor of five approximately multiplies the runtime by five but also reduces the performance gap with DEAN to only about 0.6%, while still maintaining a significant runtime advantage.

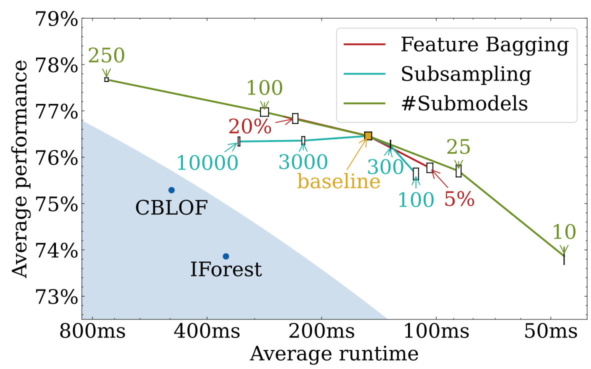

Fig. 4. Runtime and anomaly detection performance for various hyperparameter combinations. Higher values generally increase runtime. Feature bagging is evaluated with 20% and 5% of features (compared to 10% for the baseline case). Models are trained with 100, 300, 3000, and 10000 samples each (compared to 1000), and SEAN ensembles are formed with 10, 25, 100, and 250 submodels (compared to 50).

We also assess different feature bagging techniques in Table I. Our method of multiplying by a random Gaussian matrix with guaranteed zero mean ("baseline") consistently yields the best performance. If we remove our normalization, guaranteeing that the mean of each matrix row is zero (”baseline no normalization”), the performance is slightly lower but still higher than that of every other bagging method. The ”mixture” method no longer draws samples from a normal distribution for the values in the matrix F but draws either a 0 or a 1 from a binomial distribution ($p = 0.5$) for them. It achieves slightly lower performance but offers a marginally faster runtime. The ”classical” feature bagging approach [34], where each sample is randomly assigned a set of features, underperforms. We believe this to be because of the limited number of fitted submodels, which increases the probability of features being entirely omitted by the ensemble. Alternatively, we suggest the ”reverse classical” method, which guarantees that each input sample is assigned and added to exactly one output sample for each submodel. This improves the performance compared to usual feature bagging, but does so at the cost of higher runtime and also without surpassing the baseline approach.

TABLE I
COMPARISON OF DIFFERENT FEATURE BAGGING METHODS.

Algorithm	Performance (AUC)	Time (s)
baseline	$76.46\% \pm 0.08\%$	0.1510 ± 0.0032
baseline no normalization	$75.47\% \pm 0.08\%$	0.1437 ± 0.0026
mixture	$74.42\% \pm 0.05\%$	0.1172 ± 0.0014
classical	$72.40\% \pm 0.14\%$	0.1319 ± 0.0031
reverse classical	$74.16\% \pm 0.05\%$	0.2522 ± 0.0092

C. C++ Export

Many predictive maintenance tasks are executed on-device [38], necessitating not only fast models but also minimal memory usage. These environments often cannot support high-level languages like Python. To address this, we developed a C++ conversion tool that transforms a trained SEAN model into a compact C++ implementation for on-device anomaly detection. We tested this tool on the largest dataset in ADBench, the Census dataset, which contains 37136 test samples with 500 features each. The resulting executable, including all weights, is less than 100 KB in size and requires under 4 MB of RAM.

V. CONCLUSION

This paper presents SEAN as a highly efficient and effective anomaly detection algorithm tailored to the demanding needs of predictive maintenance in real-world engineering settings. Through extensive benchmarking against 121 datasets, SEAN has demonstrated superior speed, outperforming both shallow and deep learning-based algorithms, particularly in high-dimensional contexts. Our results confirm that SEAN achieves a remarkable balance between performance and computational efficiency, making it ideally suited for time-critical applications. The ablation study further underscores the flexibility of SEAN, showing how its performance and efficiency can be fine-tuned through hyperparameter adjustments. Finally, the ability to export SEAN models to C++ for on-device execution ensures that it meets the stringent requirements of memory and processing constraints typical of industrial environments. SEAN stands out as a robust, scalable, and practical solution for anomaly detection, with significant implications for advancing predictive maintenance technologies.

While SEAN has shown strong performance, several avenues for further research are apparent. First, studying the feature bagging process further has shown potential to improve detection accuracy even further.

Expanding SEAN’s capabilities to handle streaming data is another promising direction, allowing for more effective real-time anomaly detection in continuously changing environments by also considering previous time steps. But this also implies further challenges in how to intelligently choose features to balance higher resource consumption and performance, as well as how to handle effects like feature drift.

Finally, optimizing the C++ export tool for deployment on low-resource devices remains an important goal. Enhancing this tool could involve leveraging more efficient libraries or hardware-specific optimizations to better support industrial applications.

VI. ACKNOWLEDGEMENTS

This research was funded by the Research Center Trustworthy Data Science and Security and the Lamarr-Institute for Machine Learning and Artificial Intelligence and we acknowledge the compute time provided by LiDO3.

REFERENCES

[1] Ms. Archana, N ME Student, Ms. S. S. Pawar, and Asst. Prof. Periodicity detection of outlier sequences using constraint based pattern tree with mad. *ArXiv*, abs/1507.01685, 2015.

[2] Richard Bellman. Dynamic programming. *Science*, 153(3731), 1966.

[3] Abdenour Bounsiar and Michael G. Madden. One-class support vector machines revisited. In *2014 International Conference on Information Science Applications (ICISA)*, pages 1–4, 2014.

[4] Kendrick Boyd, Kevin H. Eng, and C. David Page. Area under the precision-recall curve: Point estimates and confidence intervals. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 451–466, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[5] Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[6] Markus Breunig, Peer Kröger, Raymond Ng, and Joerg Sander. Lof: Identifying density-based local outliers. volume 29, 06 2000.

[7] Benedikt Böing, Simon Klüttermann, and Emmanuel Müller. Post-robustifying deep anomaly detection ensembles by model selection. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 861–866, 2022.

[8] Christian Callegari, Loris Gazzarrini, S. Giordano, Michele Pagano, and Teresa Pepe. A novel pca-based network anomaly detection. pages 1 – 5, 07 2011.

[9] Jacinto Carrasco, David López, Ignacio Aguilera-Martos, Diego García-Gil, Irina Markova, Marta García-Barzana, Manuel Arias-Rodil, Julián Luengo, and Francisco Herrera. Anomaly detection in predictive maintenance: A new evaluation framework for temporal unsupervised anomaly detection algorithms. *Neurocomputing*, 462:440–452, 2021.

[10] Thyago P Carvalho, Fabrízzio AAMN Soares, Roberto Vita, Roberto da P Francisco, João P Basto, and Symone GS Alcalá. A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, 137:106024, 2019.

[11] Alvin Chiang, Esther David, Yuh-Jye Lee, Guy Leshem, and Yi-Ren Yeh. A study on anomaly detection ensembles. *Journal of Applied Logic*, 2017.

[12] Nathaniel Craig, Jessica N. Howard, and Hancheng Li. Exploring Optimal Transport for Event-Level Anomaly Detection at the Large Hadron Collider. 1 2024.

[13] Padraig Cunningham and Sarah Delany. k-nearest neighbour classifiers. *Mult Classif Syst*, 54, 04 2007.

[14] Jovani Dalzochio, Rafael Kunst, Edison Pignaton, Alecio Binotto, Srijnan Sanyal, Jose Favilla, and Jorge Barbosa. Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges. *Computers in Industry*, 123:103298, 2020.

[15] Massimiliano De Benedetti, Fabio Leonardi, Fabrizio Messina, Corrado Santoro, and Athanasios Vasilakos. Anomaly detection and predictive maintenance for photovoltaic systems. *Neurocomputing*, 310, 2018.

[16] Leticia Decker, Daniel Leite, Luca Giommi, and Daniele Bonacorsi. Real-time anomaly detection in data centers for log-based predictive maintenance using an evolving fuzzy-rule-based approach. In *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2020.

[17] Theekshana Dissanayake, Tharindu Fernando, Simon Denman, Sridha Sridharan, Houman Ghaemmaghami, and Clinton Fookes. A robust interpretable deep learning classifier for heart anomaly detection without segmentation. *IEEE Journal of Biomedical and Health Informatics*, 25, 2021.

[18] LI Dong, LIU Shulin, and Hongli Zhang. A method of anomaly detection and fault diagnosis with online adaptive learning under small training samples. *Pattern Recognition*, 64, 2017.

[19] Milton Friedman. A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics*, 11:86–92, 1940.

[20] Lorenzo Gigoni, Alessandro Betti, Mauro Tucci, and Emanuele Crisostomi. A scalable predictive maintenance model for detecting wind turbine component failures based on scada data. In *2019 IEEE Power Energy Society General Meeting (PESGM)*, pages 1–5, 2019.

[21] Markus Goldstein and Andreas R. Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. 2012.

[22] Cyril Goutte and Eric Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. volume 3408, pages 345–359, 04 2005.

[23] Songqiao Han, Xiyang Hu, Hailiang Huang, Mingqi Jiang, and Yue Zhao. Adbench: Anomaly detection benchmark. In *NeurIPS*, 2022.

[24] J.A. Hanley and Barbara Mcneil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143:29–36, 05 1982.

[25] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9):1641–1650, 2003.

[26] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

[27] Eda Jovicic, Daria Primorac, Marko Cupic, and Alan Jovic. Publicly available datasets for predictive maintenance in the energy sector: A review. *IEEE Access*, 11:73505–73520, 2023.

[28] Kamat, Pooja and Sugandhi, Rekha. Anomaly detection for predictive maintenance in industry 4.0- a survey. *E3S Web Conf.*, 170:02007, 2020.

[29] Nittaya Kerdprasop, Kacha Chansilp, Kittisak Kerdprasop, and Paradee Chuaybamroong. *Anomaly Detection with Machine Learning Technique to Support Smart Logistics*, pages 461–472. Computational Science and Its Applications – ICCSA 2019, 06 2019.

[30] Shehroz S. Khan and Michael G. Madden. A survey of recent trends in one class classification. In Lorcan Coyle and Jill Freyne, editors, *Artificial Intelligence and Cognitive Science*, pages 188–197, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[31] Simon Klüttermann, Chiara Balestra, and Emmanuel Müller. On the efficient explanation of outlier detection ensembles through shapley values. In De-Nian Yang, Xing Xie, Vincent S. Tseng, Jian Pei, Jen-Wei Huang, and Jerry Chun-Wei Lin, editors, *Advances in Knowledge Discovery and Data Mining*, pages 43–55, Singapore, 2024. Springer Nature Singapore.

[32] Simon Klüttermann and Emmanuel Müller. Evaluating and comparing heterogeneous ensemble methods for unsupervised anomaly detection. In *IJCNN*, 2023.

[33] Rocco Langone, Alfredo Cuzzocrea, and Nikolaos Skantzos. Interpretable anomaly prediction: Predicting anomalous behavior in industry 4.0 settings via regularized logistic regression tools. *Data Knowledge Engineering*, 130:101850, 2020.

[34] Aleksandar Lazarevic and Vipin Kumar. Feature bagging for outlier detection. volume 21, pages 157–166, 01 2005.

[35] Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. Copod: Copula-based outlier detection. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1118–1123, 2020.

[36] Zheng Li, Yue Zhao, Xiyang Hu, Nicola Botta, Cezar Ionescu, and George Chen. Ecod: Unsupervised outlier detection using empirical cumulative distribution functions, 01 2022.

[37] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *ICDM*, 2008.

[38] Irene Niyonambaza Mihigo, Marco Zennaro, Alfred Uwitonze, James Rwigema, and Marcelo Rovai. On-device iot-based predictive maintenance analytics model: Comparing tinylstm and tinymodel from edge impulse. *Sensors*, 22(14), 2022.

[39] Vinicius Mikuni, Benjamin Nachman, and David Shih. Online-compatible unsupervised nonresonant anomaly detection. *Phys. Rev. D*, 105:055006, Mar 2022.

[40] Dubravko Miljković. Fault detection methods: A literature survey. In *2011 Proceedings of the 34th International Convention MIPRO*, pages 750–755, 2011.

[41] Ali Bou Nassif, Manar Abu Talib, Qassim Nasir, and Fatima Mohamad Dakalbab. Machine learning for anomaly detection: A systematic review. *IEEE Access*, 9:78658–78700, 2021.

[42] P. Nunes, J. Santos, and E. Rocha. Challenges in predictive maintenance – a review. *CIRP Journal of Manufacturing Science and Technology*, 40:53–67, 2023.

[43] Madalina Olteanu, Fabrice Rossi, and Florian Yger. Meta-survey on outlier and anomaly detection. *Neurocomputing*, 555:126634, 2023.

[44] Tomáš Pevný. Loda: Lightweight on-line detector of anomalies. *Mach. Learn.*, 102(2):275–304, feb 2016.

[45] Majdi I. Radaideh, Chris Pappas, Mark Wezensky, Pradeep Ramuhalli, and Sarah Cousineau. Fault prognosis in particle accelerator power electronics using ensemble learning. *ArXiv*, abs/2209.15570, 2022.

[46] Lukas Ruff, Jacob Kauffmann, Robert Vandermeulen, Gregoire Montavon, Wojciech Samek, Marius Kloft, Thomas Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, PP:1–40, 02 2021.

[47] Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. *CoRR*, abs/1906.02694, 2019.

[48] Fadi Salo, Ali Bou Nassif, and Aleksander Essex. Dimensionality reduction with ig-pca and ensemble classifier for network intrusion detection. *Computer Networks*, 148:164–175, 2019.

[49] Bibhudhendu Shukla, Ip-Shing Fan, and Ian Jennions. Opportunities for explainable artificial intelligence in aerospace predictive maintenance. In *PHM Society European Conference*, volume 5, pages 11–11, 2020.

[50] Simon Vollert, Martin Atzmueller, and Andreas Theissler. Interpretable machine learning: A brief survey from the predictive maintenance perspective. In *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 01–08, 2021.

[51] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[52] Wen Xu, Julian Jang-Jaccard, Tong Liu, Fariza Sabrina, and Jin Kwak. Improved bidirectional gan-based approach for network intrusion detection using one-class classifier. *Computers*, 11(6), 2022.

[53] Yuki Yamanaka, Tomoharu Iwata, Hiroshi Takahashi, Masanori Yamada, and Sekitoshi Kanai. Autoencoding binary classifiers for supervised anomaly detection. In Abhaya C. Nayak and Alok Sharma, editors, *PRICAI 2019: Trends in Artificial Intelligence*, pages 647–659, Cham, 2019. Springer International Publishing.

[54] Heng Zhang and Danilo Vargas. A survey on reservoir computing and its interdisciplinary applications beyond traditional machine learning. *IEEE Access*, PP:1–1, 01 2023.

[55] Weiting Zhang, Dong Yang, and Hongchao Wang. Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE systems journal*, 13(3):2213–2227, 2019.

[56] Tianwen Zhu, Yongyi Ran, Xin Zhou, and Yonggang Wen. A survey of predictive maintenance: Systems, purposes and approaches, 2024.

[57] Arthur Zimek, Matthew Gaudet, Ricardo J.G.B. Campello, and Jörg Sander. Subsampling for efficient and effective unsupervised outlier detection ensembles. In *19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013.

[58] Zeki Murat Çınar, Abubakar Abdussalam Nuhu, Qasim Zeeshan, Orhan Korhan, Mohammed Asmael, and Babak Safaei. Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. *Sustainability*, 12(19), 2020.