IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 13 February 2025, accepted 14 March 2025, date of publication 17 March 2025, date of current version 25 March 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3552219

RESEARCH ARTICLE

Optimization of Adversarial Reprogramming for Transfer Learning on Closed Box Models

ALEXANDER BOTT, MORITZ SIEMS, ALEXANDER PUCHTA, AND JÜRGEN FLEISCHER
wbk Institute of Production Science, Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany

Corresponding author: Alexander Bott (alexander.bott@kit.edu)

This work was supported by KIT-Publication Fund of Karlsruhe Institute of Technology.

ABSTRACT In this work, we optimise a transfer learning approach for predicting the Remaining Useful Life (RUL) of ball bearings, particularly in scenarios with limited data availability. Accurate RUL prediction is crucial for improving maintenance strategies, reducing downtime and improving machine reliability, making it highly relevant to industry. We use the Black Box Adversarial Reprogramming (BAR) algorithm to process target domain data in a source domain model through adversarial reprogramming. While it has been shown that this concept can work well in image classification, and a further modification has been developed to classify time series features, this work focuses primarily on the remaining key challenges such as the selection and comparison of appropriate loss functions, the optimisation of hyperparameters using Bayesian methods, and data labelling in the absence of ground truth. Our results show an increase in the performance of the BAR algorithm on the macro f1 score of 0.23 on the training set and up to 0.21 on the test set.

INDEX TERMS Adversial reprogramming, closed box transfer learning, machine learning, RUL-classification.

I. INTRODUCTION

Predictive maintenance (PdM) is a proactive approach that assesses the health of machines by continuously monitoring their condition and employing analytics and machine learning methods to predict when maintenance will be required. This strategy aims to minimize unplanned downtimes by anticipating failures before they occur, potentially saving a company significant penalties and reputation loss [1], [2], [3]. Predictive maintenance can be implemented using various methodologies, including knowledge-based systems, traditional machine learning approaches, and deep learning techniques, each offering distinct advantages and limitations suited to specific scenarios [4]. In recent years, predictive maintenance has gained significant traction due to the rapid growth in sensor-generated data from production plants [5]. Machine learning, in particular, has emerged as a powerful tool for extracting valuable insights and supporting decision-making processes from large datasets, proving especially effective in applications such as fault diagnosis [6] and Remaining Useful Life (RUL) prediction [1], [7]. However, gathering the necessary run-to-failure data in real-world scenarios can be economically challenging due to the high costs, time consumption, and high component variance [2], [4].

Transfer learning solves this problem by enabling a model developed and trained in a "data-rich" source domain to be repurposed for a related task in a "data-scarce" target domain [8]. However, in real-world scenarios, access to the source domain model is not always possible, complicating the reprogramming of the model for new tasks [9]. To address this challenge [9] introduced the Black Box Adversarial Reprogramming (BAR) approach. BAR treats the source domain model as a closed box, operating without access to its internal parameters. This is particularly important because of potential probity issues and also to provide a high degree of freedom in the choice of the source domain model. Building on the results of [10] this research will utilize the BAR algorithm to predict the RUL label of time series features extracted from vibration signals of ball bearings. Although [10] demonstrated the potential usage of BAR in feature classification of time series, there

The associate editor coordinating the review of this manuscript and approving it for publication was Haidong Shao.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

are still some limitations in their approach leading to considerable performance fluctuations of the BAR algorithm. This paper addresses key limitations by utilizing a larger dataset with a more applicable domain gap and a state-of-the-art optimizer for more efficient training and faster convergence. We perform Bayesian optimization to find optimal hyperparameters and assess the impact of three state-of-the-art loss functions on the BAR algorithm's f1 score. To account for stochastic fluctuations in gradient estimation, we analyze the distribution over 100 runs. Results are compared to benchmarks from [10], the BAR algorithm's initial performance, and an alternative transfer learning method, highlighting BAR's effectiveness in time series classification.

II. RELATED WORK

Transfer learning has gained significant attention in recent years. Reference [11] provided foundational definitions and an initial taxonomy that has shaped the field. Reference [12] introduce a novel approach inspired by adversarial reprogramming, which allows attackers to reprogram a target model for a different task. To use this concept for transfer learning, an algorithm reuses deep learning-based closed box models to classify out-of-domain images into completely different categories. A translation function is combined with the closed box model. While originally applied to image classification tasks, [10] extended the BAR approach to the classification of time series features in the context of predictive maintenance. Their study focuses on predicting the Remaining Useful Life class of rolling bearings by adapting the adversarial reprogramming concept to closed box time series classifiers. They developed a comprehensive machine learning pipeline, including feature engineering and model fine-tuning, and compared the BAR method with traditional transfer learning approaches that were also Feature-based.

Feature-based transfer learning is an active area of research, with notable contributions such as Domain adaptation which has received considerable attention [13]. Reference [14] introduced further a data augmentation method of source and target data. Additionally [15] addressed class weight bias in domain adaptation, proposing a solution to issues overlooked by maximum mean discrepancy methods. These methods address the different issues that are also limiting the performance in [10].

The BAR algorithm's learning process of [10] and [12] is based on minimizing a loss function using gradient estimation via zeroth-order optimization. This technique is notable for estimating gradients without direct access to the underlying model. The training process starts by initializing the weight vector W as a one-vector. To compensate for the imbalance of the dataset, alternating oversampling and undersampling are applied, with the number of iterations controlled by the parameter n_split. After sampling from the dataset, the *one-sided Averaged Gradient Estimator* is used to estimate gradients by averaging several gradient estimates at a given point W_i, where i represents the current training iteration. The gradient estimates depend on the chosen loss function, which is a modified cross-entropy loss in the case of [10]. At each iteration i, q random vectors U_j are generated at point W_i, and a simple gradient g_j is calculated along each vector U_j on the Loss-surface L.

$$g_j = (L(W + U_j) - L(W)) \cdot U_j \tag{1}$$

The arithmetic mean of these q different gradients provides the estimated gradient $\bar{g}$.

$$\bar{g} = \frac{1}{q} \sum_{j=1}^{q} g_j \tag{2}$$

Using this estimated gradient, the weight vector $W^{(i)}$is updated in the direction of $\bar{g}$ with a learning rate α.

$$W_{i+1} = W_i - \alpha \cdot \bar{g} \tag{3}$$

The training concludes after $n = n_{splits}$ * epochs iterations, and the weight vector W_i with the lowest loss L is selected as the final weight vector. Although this approach is showed learning capability with an achieved weighted f1 score of 0.77 on the training data, The generalization to the test data only showed a score of 0.58. The research of [10] highlights the potential of the BAR algorithm for time series classification in predictive maintenance, while also emphasizing limitations related to hyperparameter tuning, domain gap, and dataset size.

Because of the existing limitations in [10] and the potential solutions in the state of the art we want to apply their method to a larger ball-bearing dataset with a more fitting domain gap while using a state-of-the-art optimizer and Bayesian hyperparameter search. Further we want to to investigate the response of the BAR-approach to different loss functions which can compensate class imbalance. This class-inbalance is especially in RUL-prediction or classifications an unavoidable system behavior. Through this we want to further improve the application of BAR in the context of time series classification in the contextof Predictive Maintenance.

III. METHOD

A. STRUCTURE

The general methodology of this approach can be seen in Figure 1. The dataset described below is divided into two parts: the source domain, which comprises most time series, and the target domain, which contains the remaining series. The classification model is trained in the source domain and serves as the base model for the transfer learning task, whereas the BAR algorithm is employed in the target domain. The effectiveness of this approach is evaluated by assessing the macro f1 score of the predictions.

B. DATASET

In this work, we use the XJTU-SY bearing dataset developed at Jiaotong University, which consists of 15 complete run-to-failure time series for rolling element bearings. The data

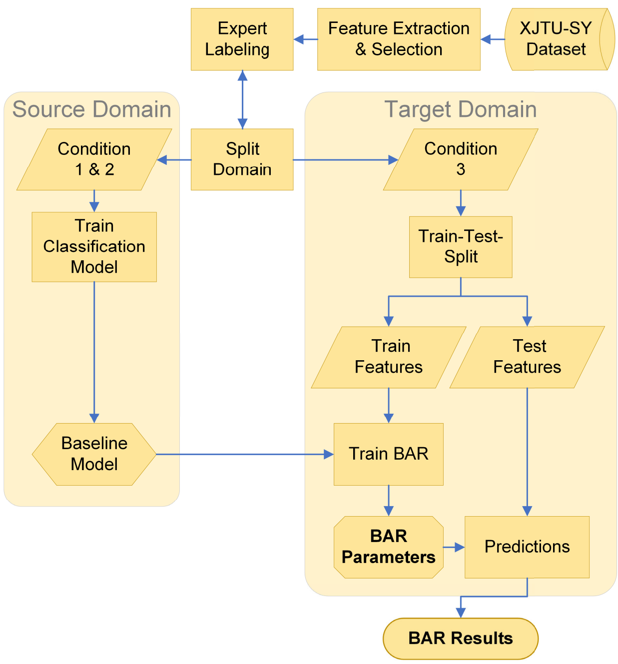


FIGURE 1. Schematic method applied in this work.

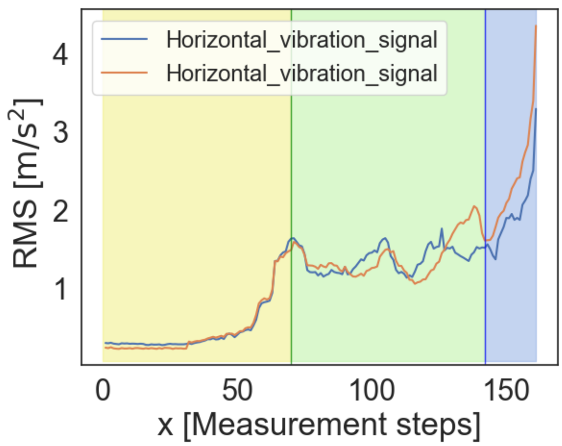


FIGURE 2. RMS of Bearing 1_2 with colorized damage stages.

TABLE 1. Source and target domain split of the XJTU-SY bearing dataset.

Source Domain		Target Domain	
Training	Test	Training	Test
bearing_1_1, bearing_1_2, bearing_1_5, bearing_2_1, bearing_2_2, bearing_2_5	bearing_1_3, bearing_2_4	bearing_3_1, bearing_3_3	bearing_3_4

was collected through accelerated degradation experiments conducted under three different operating conditions, varying in radial force and rotating speed. In this study, we focus exclusively on data from bearings that failed due to inner or outer race issues, excluding those that failed due to cage-related problems. For more detailed information about the dataset and the experiments, see [16].

The dataset does not document the specific damage states within the bearing's lifetime. This lack of ground truth makes it inevitable to assign labels explicitly, since we use the data to train a supervised classification model. To address this, an expert labeling process was conducted. The dataset reveals that the time series exhibit three visually distinct stages during their lifetime, marked by a significant increase in vibration signals as shown in Figure 2. These stages occur at different points in the bearing's life, necessitating individual labeling to more accurately capture the RUL for each bearing. The three stages defined by [17] describe a relatively static system behavior with the fault occurrence (yellow) in the beginning. The next stage describes the damage propagation (green) until the component fails in the damage stage (blue).

For feature extraction and selection, we employ the tsfresh library. Initially, the time series are given as fixed intervals by the measurement set up in itself. In the applied dataset a measurement is taken in every minute of the accelerated degradation test for $1s$. These measurement segments form the timeseries from which a comprehensive set of computationally efficient features is extracted using the *EfficientFCParameters* function of tsfresh. This allows a feature vector to represent a measurement segment without overlap between time segments. Subsequently, a hypothesis test-based feature selection is applied to assess the relevance of each feature in predicting the target label. For each of the three classes, the 20 most relevant features are retained. Finally, the intrinsic feature ranking function of a random forest classifier is employed to evaluate the importance of each feature, with the highest-ranking features being selected. The final feature set is further enhanced by integrating key time-frequency features as proposed by [18]. Further details on the extraction and selection process of tsfresh are provided in the work of [19].

The features from conditions one and two are normalized to the range [0,1] and then used to train a Random Forest classification model, which serves as the source domain model in the transfer learning framework. For the target domain, we use features from condition three. The classification model is designed to take a feature vector and predict class probabilities for the RUL classes.

C. CLOSED BOX ADVERSARIAL REPROGRAMMING

The Closed Box Adversarial Reprogramming algorithm operates by transforming the target domain data1 into the input space of the source domain model through an adversarial programming approach. The principle of operation is depicted in figure 3. This transformation enables the source domain model to process the target domain data and generate meaningful predictions for RUL class probabilities. In contrast to equation 1, used in [10] approach, we utilize formula 4 suggested by [20] to calculate the gradients, where β is a smoothing parameter, and b is a scaling parameter

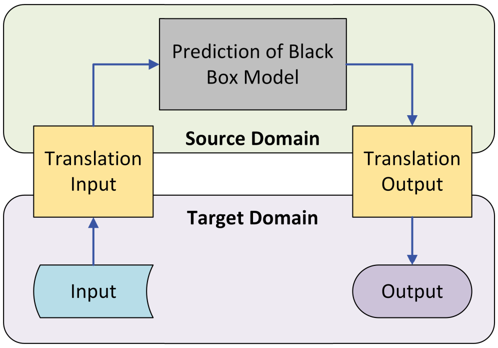


FIGURE 3. Operating principle of BAR algorithm after [10].

which we set to one.

$$g_j = b \cdot \frac{(L(W + \beta \cdot U_j) - L(W))}{\beta} \cdot U_j \tag{4}$$

D. LOSS FUNCTIONS

In this work, the dataset exhibits a significant class imbalance, with a strong skew towards class one. This is imbalance in RUL-classification and prediction tasks is a typical system behavior. Because the first phase is relatively static and the introduction of wear particles leads to an exponential increase in wear. This imbalance can cause the model to become biased towards the majority class, resulting in poor performance on the minority classes. To address this issue, various methods can be employed, such as resampling techniques to balance the dataset or algorithmic adjustments that assign greater importance to the minority class through weighted loss functions. We utilize three different loss functions, which can compensate for the likely class imbalances, to compare and evaluate their performances using the macro f1 score.

1) CATEGORICAL CROSS-ENTROPY LOSS

The first loss function under investigation is a modified version of the categorical cross-entropy loss. It comprises two components: the cross-entropy loss L_{CE} and a penalty term L_{PEN} which is balanced by a factor δ.

$$L = L_{CE} + \delta \cdot L_{PEN} \tag{5}$$

The cross-entropy loss penalizes deviations between the probability prediction p_i and the true label y_i, ensuring the model's predictions align with the ground truth.

$$L_{CE} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \log(p_{ij}) \tag{6}$$

The penalty term calculates the sum of differences between the probability assigned to the true class and the highest probability assigned to any class. It penalizes incorrect predictions but imposes no penalty for correct predictions, regardless of the confidence level.

$$L_{PEN} = \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \cdot (\max_{j \in J} p_{ij} - p_{ij}) \tag{7}$$

2) FOCAL LOSS

The focal loss function is a modified version of the categorical cross-entropy loss, designed specifically to tackle the class imbalance problem. As depicted in (8), it introduces a scaling factor $(1 - p_t)^\gamma$ that automatically down-weights the contribution of easily classified examples, directing the learning process towards harder, misclassified examples [21].

$$L_{FL}(p_t) = -\alpha_t (1 - p_t)^\gamma \log(p_t) \tag{8}$$

3) KULLBACK-LEIBLER DIVERGENCE

The Kullback-Leibler (KL) divergence is a metric that quantifies the difference between two probability distributions. In machine learning, the KL loss is one of the most widely used loss functions and finds application in various scenarios such as knowledge distillation or adversarial training [22]. In this work, KL divergence is employed as a loss function to measure the discrepancy between the predicted probability distribution and the true distribution. A higher KL divergence score indicates a larger difference between the two distributions. This loss function is especially useful in training models that generate or approximate probability distributions, as it directly evaluates how closely the model's predictions align with the target distribution. Notably, KL divergence requires no additional hyperparameters, simplifying the optimization process.

$$D_{KL}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{9}$$

E. HYPERPARAMETER TUNING

The BAR algorithm requires careful tuning of several hyperparameters to achieve optimal performance. Further details regarding the hyperparameters can be found in the work of [10].

We utilize Bayesian hyperparameter tuning which is explored in more depth by [23] in combination with cross validation techniques. Unlike traditional methods such as grid search or random search, Bayesian hyperparameter tuning uses a probabilistic model to make more informed decisions about which hyperparameters to try next. The key idea is to build a surrogate model that approximates the objective function, then use this model to select hyperparameters that are likely to perform well. The surrogate model is easier to evaluate than the true objective function, making it computationally cheaper to explore the hyperparameter space. The selected hyperparameters can be observed in table 2.

IV. RESULTS & DISCUSSION

A. BASE CLASSIFICATION MODEL

Table 3 summarizes the performance of the random forest classification model trained on source domain features from the XJTU-SY dataset. The model performed well on training data, correctly classifying 1281 of 1327 instances, including minority classes. However, test performance was lower, with

TABLE 2. Optimized Hyperparameters.

Hyperparameter	Approach		
	Cross-entropy	Focal	KL divergence
Learning Rate α	0.082	0.078	0.058
Smoothing Factor β	0.09	0.09	0.046
q	43	50	42
Punishment L_{PEN}	0.15	-	-
Factor	1	3	3

TABLE 3. Performance metrics of the base model on the train data in the source domain.

	Train Data			
	Precision	Recall	f1 Score	Support
Class 1	0.98	1.00	0.99	737
Class 2	0.99	0.91	0.95	458
Class 3	0.81	0.98	0.89	132
Accuracy			0.97	1327
Macro avg	0.93	0.96	0.94	1327
Weighted avg	0.97	0.97	0.97	1327

TABLE 4. Performance metrics of the base model on the test data in the source domain.

	Test Data			
	Precision	Recall	f1 Score	Support
Class 1	0.98	0.87	0.92	68
Class 2	0.91	0.83	0.87	111
Class 3	0.54	1.00	0.70	21
Accuracy			0.86	200
Macro avg	0.81	0.90	0.83	200
Weighted avg	0.90	0.86	0.87	200

172 of 200 instances correctly classified. The model struggled with minority class 3, misclassifying 18 class 2 instances as class 3. This issue is reflected in the macro f1 score, which dropped from 0.94 on the training data to 0.83 on the test data in table 4.

Compared to the base model in [10], which achieves a macro f1 score of 0.95 on training data and 0.61 on test data, our model generalizes better to unseen data while maintaining similar training performance. Despite some overfitting, the model performs strongly on training data and shows good test performance.

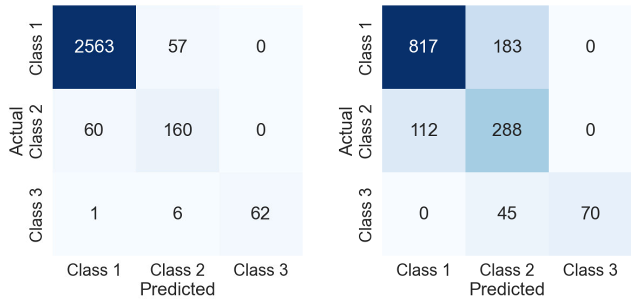
B. BAR

Table 5 compares the best results of the BAR approaches. The cross-entropy approach, shown on the left, achieves a macro f1 score of 0.88 on the training set and 0.74 on the test set. The focal loss approach, in the center, scores 0.73 on training data and 0.79 on test data, while the KL divergence approach, on the right, achieves 0.79 on training data and 0.73 on test data. For comparison, the internal benchmark (W=1), representing a baseline scenario with no input data adjustments, is also shown. This benchmark, where the source domain model directly classifies the target domain data, achieves a macro f1 score of 0.63.

The Cross-Entropy approach achieved a macro f1 score of 0.88 on the training set but dropped to 0.74 on the test set, indicating a moderate degree of overfitting. This suggests that, while the model performs well on the training data, it struggles to maintain this level of accuracy on unseen data.

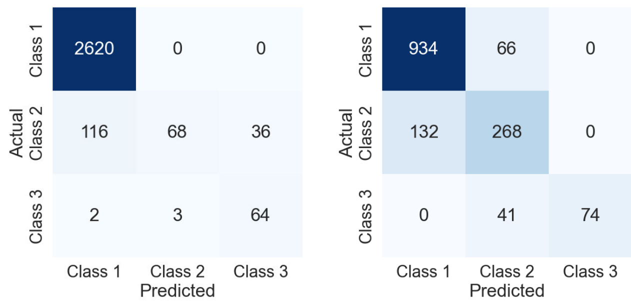
TABLE 5. Comparison of best f1 scores for test and train data and the internal benchmark of W=1.

	Benchmark	Investigated		
	Internal	F-Loss	CE-Loss	KL-Div.
train	0.63	0.73	0.88	0.79
test	0.63	0.79	0.74	0.73



(a) Train data (b) Test data

FIGURE 4. Confusion matrices of the cross entropy loss approach.



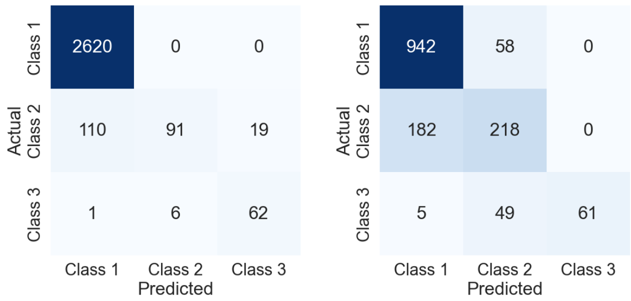
(a) Train data (b) Test data

FIGURE 5. Confusion matrices of the focal loss approach.

In contrast, the Focal Loss approach yielded macro f1 scores of 0.73 on the training data and 0.79 on the test set. Interestingly, this model's performance on the test set surpasses its training performance, indicating that Focal Loss may improve generalization, particularly in scenarios with class imbalances. This outcome aligns with Focal Loss's design, which down-weights easy negatives and emphasizes harder, often underrepresented classes, thus helping the model to generalize more effectively.

The KL divergence achieved a macro f1 score of 0.79 on the training set but dropped to 0.73 on the test set, indicating a smaller degree of over-fitting compared to the cross-entropy approach. This suggests that, while the model performs well on the training data, it struggles to maintain this level of accuracy on unseen data as well.

During the training of each approach, the weight matrix W is adapted to align the target domain input data with the source domain. Each approach demonstrates improved performance compared to the internal benchmark, with weights



(a) Train data (b) Test data

FIGURE 6. Confusion matrices of the KL divergence loss approach.

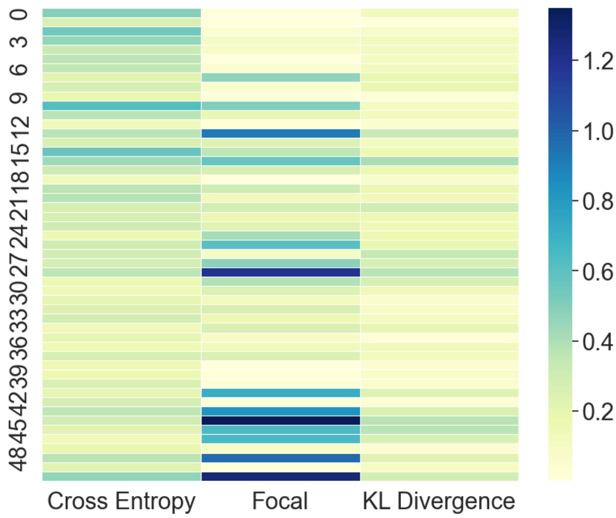


FIGURE 7. Variance of the 51 values of W over 100 runs.

that effectively perform this translation. The cross-entropy approach achieves the highest training score, surpassing the benchmark by over 30%. Meanwhile, the focal loss approach achieves a macro f1 score of 0.79 on the test set, exceeding the benchmark by more than 40% and demonstrating the strongest generalization to new data.

The internal baseline is, as described above, an application of the source domain model with no modification of the input via the weighting matrix W. If the loss surface on which the ideal weights for W are searched is convex and leads to a clear optimum, the entry in W should be the same over all three loss functions investigated and be significantly different from 1.

Figure 7 shows the resulting weights corresponding to the confusion matrixes above. Here, a clear difference between the entries in W can be seen, which leads to the assumption of a nonconvex loss surfaces which vary significantly from each other. To further investigate this, the BAR approaches training process was run 100 times using the corresponding optimal hyperparameters found by the Bayesian optimization. We see stochastic fluctuations in the macro f1 score due to the random sampling of vectors used for gradient estimation. As a result, the weight vector W is influenced by this stochastic element and, consequently, the performance of the approach itself. Figure 8 shows the distribution of the f1 scores obtained on the test data by each approach displaying ranges from 0.35 to 0.74 for the cross entropy combination, 0.52 to 0.79 for the focal loss and 0.27 to 0.73 for the model using the KL divergence loss function. The initialization of the weight vector is static over the different approaches chosen as 1. The main driver of this spread in the resulting quality is the stochastic of the sampling process for the gradient estimation because the used adam-optimizer is without stochastically in its gradient determination and controlled random seeds deterministic in its optimization [24]. The non-convex loss surface leads in interaction with the stochastically of the gradient estimation to such a wide variety in the performance.

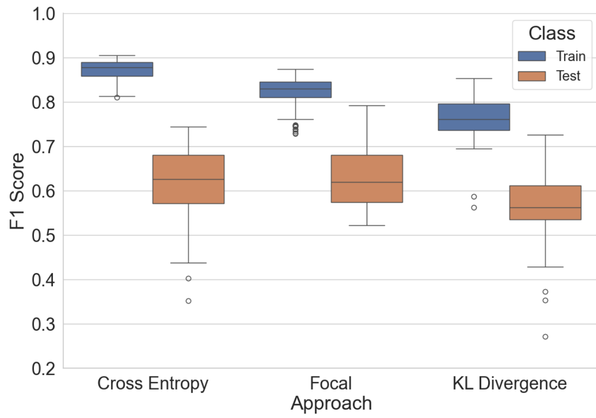


FIGURE 8. Distribution of f1 scores.

V. LIMITATIONS

One limitation of this study is the absence of ground truth labels, which can introduce inaccuracies in the data. Mislabeling may lead the model to learn incorrect relationships between input features and target classes, resulting in reduced predictive accuracy. In cases where a significant portion of the data is mislabeled, the model may become biased toward these inaccuracies, limiting its ability to discern true patterns. Another constraint is the small sample size in RUL-prediction tasks, which may restrict the generalizability of the findings. If more extensive datasets become available or this approach is applied to other more data extensive predictive maintenance tasks, the proposed approach is expected to scale considerably well. The resulting loss surface in this case would be expected to be a better representation of the underlying situation because of reduced data driven side optima. Although the sample size was sufficient for the methodology employed in this study, larger datasets with established ground truth labels are necessary to validate these results and assess their applicability to broader scenarios. The study's results are also subject to stochastic fluctuations due to variability in gradient estimation. This stochasticity can reduce the reliability of hyperparameter optimization and cause performance differences between the training and testing phases, thereby complicating reproducibility. Finally, this work exclusively investigated a homogeneous transfer learning framework, which assumes a shared feature space between source and target domains. The findings may not extend directly to heterogeneous transfer learning scenarios, where source and target domains differ significantly in feature space. The results of [9] for complicated transfer tasks in image classification show promising results. This leads to the assumption that the modified BAR approach could handle more heterogeneous datasets and transfer tasks by further optimizing the applied loss function or gradient estimation. However, further research is needed to investigate this.

VI. CONCLUSION & OUTLOOK

This study demonstrates the effectiveness of the Closed Box Adversarial Reprogramming algorithm for predicting the Remaining Useful Lifetime of ball bearings in "data-scarce" transfer learning scenarios. By optimizing hyperparameters, selecting effective loss functions, and addressing labeling challenges, we improved the BAR algorithm's macro f1 score by up to 0.21 on the test set. Among the loss functions, focal loss showed the best generalization, emphasizing the importance of handling class imbalances. Bayesian optimization further enhanced training efficiency and reliability.

Despite these advances, performance variability due to stochastic gradient estimation and nonconvex loss surfaces remains challenging. Future work should focus on larger datasets with reliable ground truth labels such as synthetic simulation data, heterogeneous transfer learning scenarios, and methods to mitigate stochastic effects for improved reproducibility. Here especially the approach from [25] could enable a significant reduction of the stochastic variance through an additional combination of the revised BAR with advanced optimizers to better address the non-convex loss surfaces would offer promising opportunities to improve performance further.

This work provides a foundation for using BAR in predictive maintenance, with significant potential for broader transfer learning applications.

ACKNOWLEDGMENT

(Alexander Bott and Moritz Siems contributed equally to this work.)

REFERENCES

[1] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. D. P. Francisco, J. P. Basto, and S. G. S. Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," *Comput. Ind. Eng.*, vol. 137, Nov. 2019, Art. no. 106024.

[2] M. S. Azari, F. Flammini, S. Santini, and M. Caporuscio, "A systematic literature review on transfer learning for predictive maintenance in industry 4.0," *IEEE Access*, vol. 11, pp. 12887–12910, 2023.

[3] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li, "Predictive maintenance in the industry 4.0: A systematic literature review," *Comput. Ind. Eng.*, vol. 150, Dec. 2020, Art. no. 106889.

[4] T. Zhu, Y. Ran, X. Zhou, and Y. Wen, "A survey of predictive maintenance: Systems, purposes and approaches," 2019, *arXiv:1912.07383*.

[5] J. Yan, Y. Meng, L. Lu, and L. Li, "Industrial big data in an industry 4.0 environment: Challenges, schemes, and applications for predictive maintenance," *IEEE Access*, vol. 5, pp. 23484–23491, 2017.

[6] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: A review and roadmap," *Mech. Syst. Signal Process.*, vol. 138, Apr. 2020, Art. no. 106587.

[7] V. Mathew, T. Toby, V. Singh, B. M. Rao, and M. G. Kumar, "Prediction of remaining useful lifetime (RUL) of turbofan engine using machine learning," in *Proc. IEEE Int. Conf. Circuits Syst. (ICCS)*, Dec. 2017, pp. 306–311.

[8] S. Niu, Y. Liu, J. Wang, and H. Song, "A decade survey of transfer learning (2010–2020)," *IEEE Trans. Artif. Intell.*, vol. 1, no. 2, pp. 151–166, Oct. 2020.

[9] Y. Y. Tsai, P. Chen, and T.-Y. Ho, "Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2020, pp. 9614–9624.

[10] A. Bott, F. Schreyer, A. Puchta, and J. Fleischer, "Black box adversarial reprogramming for time series feature classification in ball bearings' remaining useful life classification," *Mach. Learn. Knowl. Extraction*, vol. 6, no. 3, pp. 1969–1996, Aug. 2024.

[11] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2009.

[12] G. F. Elsayed, I. Goodfellow, and J. Sohl-Dickstein, "Adversarial reprogramming of neural networks," 2018, *arXiv:1806.11146*.

[13] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. 24th Int. Conf. Mach. Learn.*, Jun. 2007, pp. 759–766.

[14] H. Daumé, "Frustratingly easy domain adaptation," 2009, *arXiv:0907.1815*.

[15] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 945–954.

[16] B. Wang, Y. Lei, N. Li, and N. Li, "A hybrid prognostics approach for estimating remaining useful life of rolling element bearings," *IEEE Trans. Rel.*, vol. 69, no. 1, pp. 401–412, Mar. 2020.

[17] L. Cui, X. Wang, H. Wang, and H. Jiang, "Remaining useful life prediction of rolling element bearings based on simulated performance degradation dictionary," *Mechanism Mach. Theory*, vol. 153, Nov. 2020, Art. no. 103967.

[18] W. Mao, J. He, and M. J. Zuo, "Predicting remaining useful life of rolling bearings based on deep feature representation and transfer learning," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1594–1608, Apr. 2020.

[19] M. Christ, A. W. Kempa-Liehr, and M. Feindt, "Distributed and parallel time series feature extraction for industrial big data applications," 2016, *arXiv:1610.07717*.

[20] C. Tu, P. Ting, P. Chen, S. Liu, H. Zhang, J. Yi, C. Hsieh, and S. Cheng, "AutoZOOM: Autoencoder-based zeroth order optimization method for attacking black-box neural networks," in *Proc. AAAI Conf. Artif. Intell.*, Jan. 2019, vol. 33, no. 1, pp. 742–749.

[21] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[22] D. Wu, S. Xia, and Y. Wang, "Adversarial weight perturbation helps robust generalization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, Jan. 2020, pp. 2958–2969.

[23] J. Wu, X. Y. Chen, H. Zhang, L. D. Xiong, H. Lei, and S. Deng, "Hyperparameter optimization for machine learning models based on Bayesian optimization," *J. Electron. Sci. Technol.*, vol. 17, no. 1, pp. 26–40, Mar. 2019.

[24] R. Bock and J. Weiß, "Local convergence of the Adam optimization algorithm under deterministic conditions," in *Proc. IEEE Conf. Decis. Control*, 2019, pp. 8852–8859.

[25] S. Liu, B. Kailkhura, P. Chen, P. Ting, S. Chang, and L. Amini, "Zeroth-order stochastic variance reduction for nonconvex optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., May 2018, pp. 3727–3737.

ALEXANDER BOTT received the M.Sc. degree in mechanical engineering from Karlsruhe institute of Technology, in 2022. He is currently pursuing the Ph.D. degree in machine, plant, and process automation in the subject areas of model transfer and prediction uncertainty in data-driven models. Since 2022, he has been a Research Associate with the wbk Institute of Production Science, working on different projects in the domain of data science for manufacturing.

MORITZ SIEMS received the B.Sc. degree in industrial engineering and management from Karlsruhe Institute of Technology (KIT), in 2023, where he is currently pursuing the master's degree in information systems. Since 2023, he has been a Research Assistant with the wbk Institute of Production Science, KIT, with research interests in the area of data science in the context of production.

ALEXANDER PUCHTA received the M.Sc. degree in mechanical engineering from Karlsruhe institute of Technology, in 2020. He is currently pursuing the Ph.D. degree in machine, plant, and process automation, focusing on the topic of autonomous modeling of feed axes for machine tools. He has been the Head of the Intelligent Machines and Components Group, wbk Institute of Production Science, since 2023.

JÜRGEN FLEISCHER received the degree in mechanical engineering from Universität Karlsruhe (TH), and the Ph.D. degree from the Institute for Machine Tools and Industrial Engineering (iwb), in 1989. Since 1992, he has been holding several leading positions in industry before being appointed as a Professor and the Head of the wbk Institute for Production Technology, Karlsruhe Institute of Technology (KIT), in 2003. In addition, he has been a Visiting Professor with Tongji University, Shanghai, since 2012. His current scientific research focuses are intelligent components for machine tools and handling systems, the automation of immature processes, and agile production facilities. As a recognized member of the scientific community, he is active in German Academy of Science and Engineering (acatech), and is a member of several scientific and industrial advisory boards.