Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 104 (2021) 1547–1552

www.elsevier.com/locate/procedia

54th CIRP Conference on Manufacturing Systems

Scalable anomaly detection in manufacturing systems using an interpretable deep learning approach

Thomas Schlegl[a,c,*], Stefan Schlegl[d], Nikolai West[a], Jochen Deuse[a,b]

[a]*Institute of Production Systems, TU Dortmund University, Leonhard-Euler-Str. 5, 44227 Dortmund, Germany*
[b]*Centre for Advanced Manufacturing, University of Technology Sydney, 11 Broadway, Ultimo NSW 2007, Australia*
[c]*BMW Group, Petuelring 130, 80788 Munich, Germany*
[d]*BotCraft GmbH, Lichtenbergstraße 8, 85748 Garching, Germany*

* Corresponding author. Tel.: +49-151-601-43345. *E-mail address:* thomas.schlegl@bmw.de

Abstract

Anomaly detection in manufacturing systems has great potential for the prevention of critical quality faults. In recent years, unsupervised deep learning has shown to frequently outperform conventional methods for anomaly detection. However, tuning, deploying and debugging deep learning models is a time-consuming task, limiting their practical applicability in manufacturing systems. We approach this problem by developing a deep learning model that learns interpretable shapes that can be used for anomaly detection in temporal process data. Application of the model to assembly tightening processes in the automotive industry shows a significant improvement in model interpretability and scalability.

Keywords: anomaly detection; scalability; interpretability; deep learning; time series data

1. Introduction

Setting. In many manufacturing operations, detailed temporal data of the manufacturing processes is recorded and stored for each part. The unsupervised detection of anomalies in this process data allows to preemptively detect process or product faults, that would otherwise go unnoticed [1, 2]. Approaches based on Deep Learning (DL) have been shown to consistently outperform conventional machine learning (ML) models for anomaly detection in time series data [3, 4]. One of the great challenges in the way of deploying these models is their limited scalability. Although this is one of the most frequently cited non-functional requirements for ML-systems [5, 6], there exists no criteria of what constitutes a scalable model [7, 8]. Nevertheless, it is possible to gain an intuitive understanding of what we mean by a scalable model by considering the practical challenges of model deployment. In essence, a model is scalable if it can be applied to the large number of processes that comprise a manufacturing system [9] without the effort associated with model development and deployment becoming unsustainable.

Motivation. For this effort to remain tractable, the algorithm must either reliably learn an effective anomaly detection model or facilitate rapid improvement cycles. Since a one-size-fits-all model is unrealistic for most real-world settings, scaling a model will inevitably require such improvement cycles. To avoid an iterative process of trial-and-error that usually relies exclusively on model accuracy, we would like to be able to understand what the model has learned. This would allow a vastly more efficient improvement process and make it possible to anticipate model behavior prior to its productive deployment. However, evaluating what a DL-model has learned is extremely challenging. The convergence of model accuracy alone says little about whether or not the model has learned a sensible representation of the data. This is especially true for unsupervised training, where evaluation of the model is made difficult by the absence of labels. To this end, we propose an interpretable DL-model for scalable anomaly detection in manufacturing systems.

Structure. Section 2 recapitulates the rationale behind employing unsupervised deep learning for anomaly detection and gives a brief overview of their application in the manufacturing domain. This is followed by a discussion of the difficulties associated with interpreting neural networks and an overview of

2212-8271
10.1016/j.procir.2021.11.261

different approaches aimed at tackling them. The research gap regarding the development of inherently interpretable DL models is highlighted. Section 3 details the network architecture as well as the modified convolution operation and custom learning objective used by our proposed model. More importantly, we explain the rationale behind these design choices and show how they allow us to learn interpretable representations that can be used to detect anomalies. In section 3 we describe the application of the model to real-world manufacturing data and compare the results with those of a comparable benchmark model in section 4. The implications of these results and the limitations of the proposed model are discussed in section 5. A brief summary of the paper and concluding remarks are provided in section 6.

2. Literature

This section summarizes the rationale behind using unsupervised DL models and gives a brief overview of their industrial application to anomaly detection in time series data. This is followed by a summary of the challenges identified by researchers when attempting to scale these models, highlighting the need for model interpretability. Lastly, different approaches are discussed that have been proposed to improve model interpretability.

2.1. Unsupervised deep learning for anomaly detection

An often-cited advantage of DL models over conventional ML approaches is their ability to learn increasingly abstract hierarchical features from the underlying data [10]. In principle, therefore, these models are well-suited for unsupervised learning tasks. Data in the manufacturing domain is generally characterized by its large size and absence of labels, which often mandates unsupervised learning approaches. To avoid the expensive and time-consuming process of defining domain-specific features [11], conventional ML approaches rely on unsupervised feature extraction and selection algorithms. DL models are able to avoid this intermediate step, that is often accompanied by a loss of information [12]. Empirical results suggest that this advantage is considerable, as DL models tend to outperform conventional machine learning approaches in terms of accuracy [3, 4].

In recent years, researchers have begun to deploy DL-models for the purpose of anomaly detection in time series data in industrial settings [4, 13, 1, 14, 15]. These are sequence-to-sequence models, that learn to model the underlying process either by predicting subsequent time steps or reconstructing the input sequence from a low-dimensional projection [16]. The assumption underlying both approaches is that the model will be unable to replicate an anomalous input sequence if it does not conform to the same rules as most other sequences. While these models tend to outperform conventional ML models, their scalability remains challenging. As explained in section 1, the deployability and operability of a DL model in a manufacturing system depends on the effort required for model debugging and improvement. To do so effectively, developers must understand why the model does not deliver satisfactory performance [17]. A growing number of researchers support the notion that model interpretability is vital for model understanding and thus model improvement [7, 18, 19].

2.2. Interpretability of deep learning models

In practice, DL models are mostly treated as black-box models [20]. This is because the patterns and logic of combining these patterns that are learned by the algorithm are difficult to interpret [21]. Most pertinent work focuses on the aspect of model transparency as defined by Lipton et al. [8], by developing techniques to analyze a trained model to understand the patterns it has learned. A relatively straightforward approach is to visualize the activations of individual cells and relate them to features of the underlying data. By applying this approach to recurrent neural networks (RNN) for language processing tasks, it was possible to show that some cells learn interpretable features, such as keeping track of quotations, while others were activated seemingly indiscriminately [22]. An extension of this approach is the targeted maximization of certain activations via appropriate alterations to the input data. This fictional input data can then be analyzed to understand the feature the cell is detecting. This effectively amounts to running the neural network in reverse and is related to the concept underlying Google's famous DeepDream computer vision program. Another approach is to alter points of the input data and analyze the effect on the prediction of the model to assess the importance of the points under investigation.

Most of the pertinent literature on model interpretation focuses on the retrospective evaluation of cell activations. Recently, however, researchers have suggested that efforts should instead be directed towards designing inherently interpretable models [20, 18]. In this paper, we seek to address this research deficit by proposing a number of design considerations to build an inherently interpretable DL model. Broadly speaking, the lack of interpretability is due to the ability of the algorithm to arbitrarily adjust any set of parameters in order to minimize the training loss. Since conventional cost functions do not contain a mathematical formulation of interpretability, the algorithm has no incentive to learn interpretable representations. The formulation of an effective loss function can be used to address this issue [8]. This paper addresses the proposed research area of designing a DL model that is able to learn interpretable temporal representations from unlabeled time series data. The practical motivation for this research is to facilitate the evaluation of the model and enable a more effective deployment process.

3. Method

In the context of applied research, model interpretability is not an end in itself but instead a means to an end - namely to facilitate an efficient model improvement process. The ultimate goal is to use the model for the detection of anomalies in different manufacturing processes. To be able to detect meaningful

anomalies, the model must learn a good representation of the normal class [17]. Thus, we want the model to learn representations of normal process behavior that can be both interpreted by humans and used for anomaly detection. In this section, we explain the network architecture of the proposed model as well as the motivation underlying its design choices. The model is depicted in figure 1. For the purpose of didactic clarity, we subdivide the model into two networks (A) and (B) that serve the two separate tasks of representation learning and anomaly detection, respectively. The same representations that can be manually evaluated by the human model developer are used by the model itself for anomaly detection. The components pertaining to each subnet are indicated in figure 1.

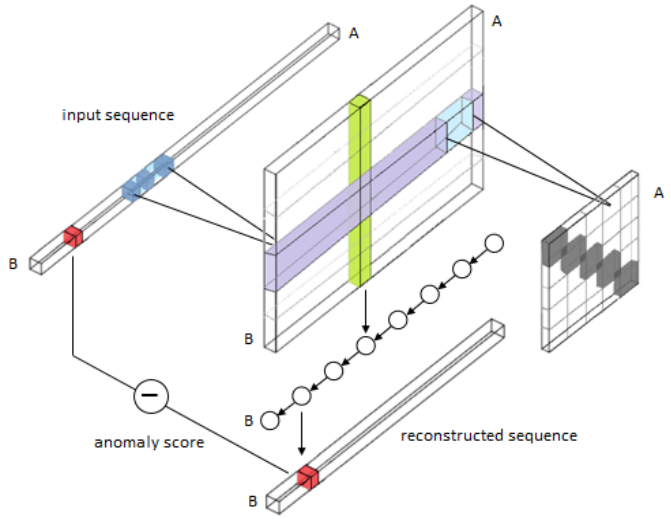

Fig. 1. Network architecture consisting of subnet (A) for learning interpretable representations and subnet (B) for the detection of anomalies in the input sequence

3.1. Learning interpretable representations

The architecture of network (A) is customized to learn interpretable representations of the shapes that constitute the input sequences. Given a set of sequences, the model learns an average representation of the predominant shape for each section of the sequence. When applied to unlabeled manufacturing process data, the model will learn characteristic shapes shared among the majority of fault-free sequences. This is possible, because fault-free sequences make up the vast majority of available data in the manufacturing domain, due to the high repeat accuracy and low fault rate of manufacturing processes. This prerequisite is crucial to the applicability of the model. While learning these shapes may seem trivial, understanding how a class is represented in a model is important to understand what it has learned [17].

The first step is the convolution of a number of k kernels with an input sequence I of length l. The number k corresponds to the number of independent shapes to be learned from the input sequence and is the only parameter of the model architecture intended for customization. The result is an activation map M of size $k \times l$ that shows where each kernel is activated along the time axis. Rather than using a standard one-dimensional convolution, we have defined a custom convolution operation $\star$ that allows the model to learn weights that can directly reflect the learned representations. We call $\star$ a deviation convolution. The output C of a standard convolution operation between kernel weights W and input sequence I is defined as their vector dot product in equation 1

$$c = \mathbf{W} \cdot \mathbf{I} = \sum_j w_j \cdot i_j \tag{1}$$

In comparison, the output of our custom convolution is defined as the sum of the absolute distance between the kernel weights W and the input sequence I in equation 2.

$$c = \mathbf{W} \star \mathbf{I} = \sum_j (w_j - i_j)^2 \tag{2}$$

Intuitively, the weights W will approach the values of the subsequence of I as the loss approaches zero $c \to 0$. The next step is a k min-pooling operation on the activation map M that returns a quadratic pooled activation map m of size $k \times k$. This activation map m summarizes the activation information of each kernel within each section of the input sequence I. Without intervening in the learning process, multiple kernels can be activated simultaneously along the same sections. Intuitively, this makes it extremely challenging to interpret the kernel weights W, as it requires separating the superimposing effects of multiple sets of weights and node activations. This is known as feature entanglement and is what makes neural networks notoriously difficult to interpret [23]. Rather than disentangling these superimposing effects, we want different kernels to be activated exclusively at different sections along I with minimal overlap. This would make it possible to avoid any obfuscating interaction between them. Our model enforces this separating constraint through a special loss function. To calculate the loss, we simple sum the diagonal elements of m. Minimizing this loss via backpropagation forces each kernel to change its weights in order to obtain a minimal deviation from I within its respective section. We call this loss identity loss due its similarity to the diagonal elements of the identity matrix.

3.2. Anomaly detection

Apart from being human-interpretable, the shapes learned by the model provide a meaningful input to the RNN for anomaly detection. The activation matrix M contains information about both the extent and location of the coincidence of the shapes along the input sequence, whereby the element M_{ij} describes the deviation between the i-th kernel around the j-th timestep. The columns of M are used as the input vector for the RNN, which aims to predict subsequent time steps of the input sequence based on this information. Thus, the model

learns to reconstruct the input sequence, based on the strength and order of activations of the k kernels. This model structure, which is trained minimizing the difference between its output and the original input is known as an autoencoder. Thus, network (B) used for the detection of anomalies is a single-layer convolutional-RNN autoencoder. In fact, this structure is often found on its own in literature, as the CNN directs the attention of the RNN to a section of the sequence, aiding it in considering the temporal context of the input sequence. Based on the reasoning previously outlined in section 2.1, the inverse reconstruction error is used as a measure of normality.

4. Results

Application to real-world data. To assess the interpretability of the proposed model, it was applied to a real-world dataset consisting of 10.000 torque sequences of a tightening process in the automotive assembly industry. The data contains 50 anomalies that were individually labeled by process experts. To put the results into context, we constructed a second benchmark model that shared the architecture of subnet (B) as depicted in figure 1, but contained neither the deviation kernel nor the identity loss of the proposed model. As such, it is a simple convoluational-RNN autoencoder trained only on the reconstruction error. Both models, including the custom deviation convolution and the identity loss were implemented using the open source machine learning framework PyTorch. The same model parameters and solver settings were used for training both models. Based on this dataset, it is possible to compare the performance of both models regarding their ability to detect anomalies. In either case, a separating threshold is required to distinguish normal from abnormal sequences based on their reconstruction error. To ensure comparability, figure 2 shows the ROC-curves of both models, constructed by plotting the true positive rate against the false positive rate for various thresholds. Our proposed model shows a consistent improvement over the benchmark model. This supports the claim made in 3, that learning an accurate representation of the normal class greatly improves the ability to detection anomalies.

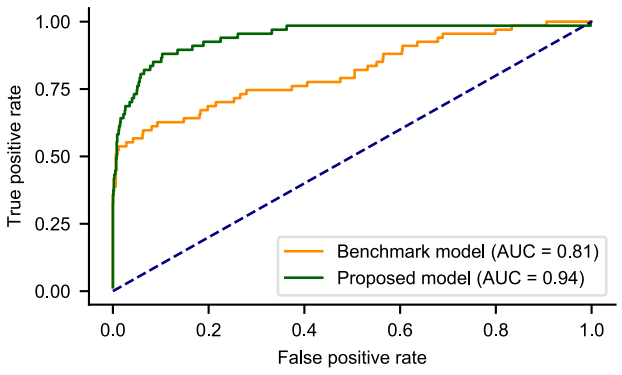

Fig. 2. ROC-curve for the proposed and benchmark models regarding their ability to detect meaningful anomalies

The proposed model successfully adjusts the weights of its kernels to match the majority shapes of the input sequence as discussed in section 3.1. Figure 3 depicts the learned shapes at the points of their respective maximum coincidence along a randomly selected sequence from the dataset.

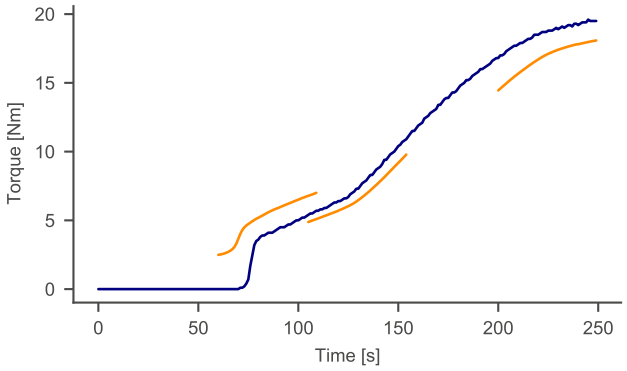

Fig. 3. Interpretable representations learned by the proposed model

In contrast, the weights learned by the benchmark model in figure 4 exhibit no discernible relation to the input sequence. This is unsurprising, as the standard one-dimensional convolution is unsuited for learning interpretable shapes. Nonetheless, the benchmark model is able to successfully reconstruct the input sequence with a reconstruction error similar to the proposed model. In fact, the RNN is able to replicate the input sequence I with only little input from the CNN. This is evident from the fact, that most kernel weights W approach zero as the training progresses, meaning they do not pass on any information to the RNN.

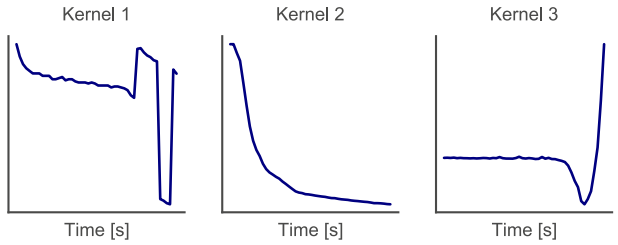

Fig. 4. Visualized kernels of the benchmark model

To understand the information that is available to the RNN for the detection of anomalies, we will take a closer look at the activation maps M of both models, depicted in figure 5. The kernels of the proposed model are sequentially activated along the time axis, forming a diagonal pattern of maximum activation. This is to be expected as a direct consequence of the identity loss. In case of an abnormality in the sequence, the respective kernel is only weakly activated at that location and instead ex-

hibits its maximum activation far from the diagonal. This makes it relatively easy for the RNN to recognize the abnormal behavior.

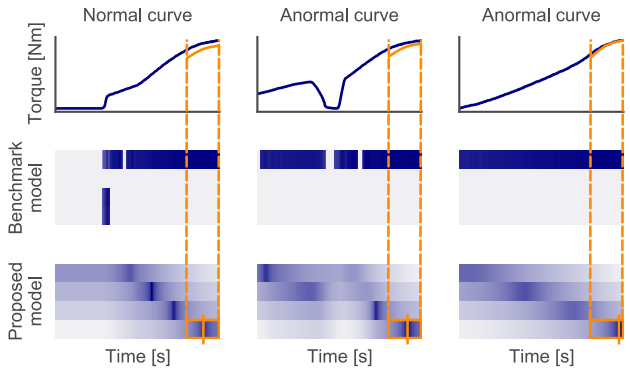

Fig. 5. Activation maps of the benchmark model and the proposed model for both normal and abnormal tightening curves

5. Discussion

In this section, we will discuss the implications of our findings and take a critical look at the limitations inherent in the design of the proposed model. While some were deliberate design choices, others became apparent during our subsequent investigation.

Implications. Our findings show, that the weights of one-dimensional convolutional kernels can be directly interpreted as geometric shapes, if the model is (1) taught to minimize the deviation between the kernel and target shape rather than the dot product and (2) the kernels are confined to separate sections during training to prevent their entanglement. Both were realized through a customized learning objective. This illustrates the importance of a suitable learning objective for solving real-world objectives.

Limitations. A major drawback of the proposed model is the intransparency of the RNN, which somewhat relativizes the advantage of learning interpretable shapes. While all anomalies in the dataset exhibit local deviations from the normal shape of the sequence, the model is not sensitive enough to detect smaller anomalies. Although this information is contained in lower activation values in M, the RNN failed to consider it. A second drawback is the limitations inherent in the learning objective itself. The identity loss allows the model to learn only one shape for each section of the sequence. While this may be sufficient for processes that exhibit only statistical fluctuations, it is unsuitable for processes with multiple modes of operation. In this case, the model will fail to learn a meaningful representation for the less frequent operating modes. For future work, the model needs to be enhanced to allow it to learn multiple representations for each section. One approach is to use the convergence rate of individual kernels to adapt the model during the training process. During the investigation, we noted that some kernels take significantly longer to converge than others for sections where the input data exhibits increased variance. This information could be used to dynamically extend the model. A third drawback is the hyperparameter that sets the kernel length. To decrease the deployment effort, this degree of freedom should be eliminated by having the model learn optimal kernel lengths that are best suited for the detection of anomalies.

6. Conclusion

We developed a DL model that is being able to learn meaningful temporal representations in the input data. The ability to visualize and evaluate what the model has learned enables model developers to evaluate the effectiveness of their training strategy. This, in turn, allows for shorter development and deployment cycles with less time spent on debugging productive models. While the results in section 4 show, that the model is able to beat a comparable benchmark model in terms of accuracy for the provided data set, it is not possible to infer a general model superiority. In this context, it is important to note, that it is not possible to make general statements about the ability of any anomaly detection model to detect quality faults. The question of whether or not an anomaly is indicative of an underlying product or process fault can only be answered by a process expert.

References

[1] R.-J. Hsieh, J. Chou, C.-H. Ho, Unsupervised online anomaly detection on multivariate sensing time series data for smart manufacturing, in: 12th Conference on Service-Oriented Computing and Applications (SOCA), IEEE, 2019, pp. 90–97.
[2] J. Deuse, M. Wiegand, K. Weisner, Continuous process monitoring through ensemble-based anomaly detection, in: N. Bauer, K. Ickstadt, K. Lübke, G. Szepannek, H. Trautmann, M. Vichi (Eds.), Applications in Statistical Computing: Studies in Classification, Data Analysis, and Knowledge Organization, Springer Nature Switzerland, Cham, Switzerland, 2019, pp. 289–304.
[3] R. Chalapathy, S. Chawla, Deep learning for anomaly detection: A survey (2019).
[4] M. Munir, S. A. Siddiqui, A. Dengel, S. Ahmed, Deepant: A deep learning approach for unsupervised anomaly detection in time series, IEEE Access 7 (2019) 1991–2005.
[5] A. Paleyes, R.-G. Urma, N. Lawrence, Challenges in deploying machine learning: a survey of case studies, in: The ML-Retrospectives, Surveys & Meta-Analyses Workshop, NeurIPS 2020, 2020.
[6] L. Baier, F. Jöhren, S. Seebacher, Challenges in the deployment and operation of machine learning in practice, in: Proceedings of the 27th European Conference on Information Systems (ECIS), 2019.
[7] S. Chakraborty, R. Tomsett, R. Raghavendra, D. Harborne, M. Alzantot, F. Cerutti, Srivastava, Mani, A. Preece, S. Julier, R. Rao, T. Kelley, D. Braines, M. Sensoy, C. Willis, P. Gurram, Interpretability of deep learning models: A survey of results, in: Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, UIC-ATC, IEEE, 2017.
[8] Z. Lipton, The mythos of model interpretability, ACM Queue (16) (2018).
[9] N. Laptev, S. Amizadeh, I. Flint, Generic and scalable framework for automated time-series anomaly detection, in: L. Cao, C. Zhang, T. Joachims, G. Webb, D. D. Margineantu, G. Williams (Eds.), 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 15), ACM Press, New York, New York, USA, 2015, pp. 1939–1947.

[10] S. Zhao, J. Song, S. Ermon, Learning hierarchical features from deep generative models, in: Proceedings of the 34th International Conference on Machine Learning, ACM, 2017, pp. 4091–4099.
[11] M. Längkvist, L. Karlsson, A. Loutfi, A review of unsupervised feature learning and deep learning for time-series modeling, Pattern Recognition Letters 42 (2014) 11–24.
[12] M. Maggipinto, A. Beghi, G. Antonio Susto, A deep learning-based approach to anomaly detection with 2-dimensional data in manufacturing, in: 17th IEEE International Conference on Industrial Informatics (INDIN19), IEEE, 2019, pp. 187–192.
[13] D. Kim, H. Yang, M. S. Chung, Squeezed convolutional variational autoencoder for unsupervised anomaly detection in edge device industrial internet of things, in: 4th International Conference on Information and Computer Technologies (ICICT), 2018, pp. 67–71.
[14] J. Liu, J. Guo, P. Orlik, M. Shibata, D. Nakahara, S. Mii, M. Takac, Anomaly detection in manufacturing systems using structured neural networks, in: Y. Liu, Y. Wang (Eds.), 13th World Congress on Intelligent Control and Automation (WCICA 2018), IEEE, 2018, pp. 175–180.
[15] B. Lindemann, F. Fesenmayr, N. Jazdi, M. Weyrich, Anomaly detection in discrete manufacturing using self-learning approaches, Procedia CIRP 79 (2019) 313–318.
[16] M. Mohammadi, A. Al-Fuqaha, S. Sorour, M. Guizani, Deep learning for iot big data and streaming analytics: A survey, IEEE Communications Surveys & Tutorials 20 (4) (2018) 2923–2960.
[17] F. Hohman, H. Park, C. Robinson, D. H. Chau, Summit: Scaling deep learning interpretability byvisualizing activation and attribution summarizations, IEEE Transactions on Visualization and Computer Graphics (2017).
[18] W. Samek, T. Wiegand, K.-R. Müller, Explainable artifical intelligence: Understanding, visualizing and interpreting deep learning models, ITU Journal: ICT Discoveries (1) (2017) 39–48.
[19] M. Kahng, P. Andrews, A. Kalro, D. H. Chau, Visual exploration of industry-scale deep neural network models, IEEE Transactions on Visualization and Computer Graphics (2019).
[20] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature Machine Intelligence (2019) 206–215.
[21] Q.-s. Zhang, S.-c. Zhu, Visual interpretability for deep learning: a survey, Frontiers of Information Technology & Electronic Engineering 19 (1) (2018) 27–39.
[22] T. J. Sejnowski, The unreasonable effectiveness of deep learning in artificial intelligence, Proceedings of the National Academy of Sciences of the United States of America 117 (48) (2020).
[23] Y. Lou, R. Caruana, J. Gehrke, Intelligible models for classification and regression, in: 18th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 12), ACM, 2012.