Deep Dense and Convolutional Autoencoders for Machine Acoustic Anomaly Detection

Gabriel Coelho[1 [0000−0001−8352−2637]], Pedro Pereira[1 [0000−0002−6169−8778]], Luis Matos[1 [0000−0001−5827−9129]], Alexandrine Ribeiro[3], Eduardo C. Nunes[1], André Ferreira[2], Paulo Cortez[1 [0000−0002−7991−2090]], and André Pilastri[3 [0000−0002−4380−3220]]

[1] ALGORITMI Centre, Dep. Information Systems, University of Minho, Guimarães, Portugal
a82137,id6927,id6929@alunos.uminho.pt, pcortez@dsi.uminho.pt
[2] Bosch Car Multimedia, Braga, Portugal,
andre.ferreira2@pt.bosch.com
[3] EPMQ - IT Engineering Maturity and Quality Lab, CCG ZGDV Institute, Guimarães, Portugal
andre.pilastri@ccg.pt

Abstract. Recently, there have been advances in using unsupervised learning methods for Acoustic Anomaly Detection (AAD). In this paper, we propose an improved version of two deep AutoEncoders (AE) for unsupervised AAD for six types of working machines, namely Dense and Convolutional AEs. A large set of computational experiments was held, showing that the two proposed deep autoencoders, when combined with a mel-spectrogram sound preprocessing, are quite competitive and outperform a recently proposed AE baseline. Overall, a high-quality class discrimination level was achieved, ranging from 72% to 92%.

Keywords: Acoustic anomaly detection · Unsupervised learning · Autoencoders · Convolutional neural network.

1 Introduction

With the advent of the Industry 4.0 phenomenon, the amount of digital data is growing exponentially. In effect, currently there is a widespread usage of interconnected sensors that can capture diverse physical aspects of the productive process (e.g., images, sound, temperatures, torque, energy consumption values). All this data can be used by Artificial Intelligence (AI) and Machine Learning (ML) to extract valuable productive analytics. A particularly relevant ML task is anomaly detection, which intends to distinguish abnormal events from normal ones [18,34]. In industrial processes, the early detection of operating machines with a defects by using ML can potentially [25, 31]: reduce maintenance time and costs; prevent or reduce production stops, and increase the safety of human operators that operate the machines. In this work, we focus on ML methods for Acoustic Anomaly Detection (AAD) [8], which aims to detect abnormal

behaviours using audio data. In particular, we aim to automatically detect, beforehand, if a given industrial machine is not working correctly, by using only the sound produced by it. Several studies addressed this issue as an unsupervised ML task, since data labeling is highly costly and time consuming, requiring great manual human work subject to errors [4].

Over the years, several algorithms were applied to unsupervised AAD problems, including Isolation Forest (IF) [9, 10] and One-Class Support Vector Machines (OCSVM) [4, 29]. Following the success of Deep Learning, there has been a growing usage of neural network architectures for AAD. In particular, AutoEncoders (AE) are becoming popular for unsupervised AAD [15, 24]. When compared with other ML approaches (e.g., IF and OCSVM), AE present the advantage of requiring a lower computational effort [19].

AEs compressed the input features into a lower dimensional space, named latent space, learning their most relevant relationships, and are composed by two main components [5, 22]: an encoder that maps the input vector (features) into the latent space, via a nonlinear transformation; and a decoder that attempts to reconstruct the reverse transformation to the original input signal. The difference between the original input vector and the AE output is called reconstruction error [3]. This error can be used to detect anomalies. AEs assume that normal and anomalous events follow different distributions and it is trained to learn the normal multi-dimensional space of the data, by using only normal event records, aiming to minimize the reconstruction error on such data. When an AE tries to reconstruct new unseen data containing anomalies, the reconstruction errors are higher and by using a predefined threshold, the samples can be signaled as anomalous [5, 22].

Following on good results obtained in previous studies [14, 18, 26, 32], in this work we address unsupervised AAD task in industrial machines using two different AE architectures: deep Dense and Convolutional. In order to use audio as input, it is often necessary to preprocess the raw data by extracting features from the signal. In this work, we use Mel Frequency Energy Coefficients (MFECs), which are a popular sound preprocessing method [7, 28]. Moreover, we use two public datasets [17, 27] to test the proposed AEs that are fed with MFECs. For benchmark purposes, we compare the Dense and Convolutional AEs with a baseline AE architecture that was recently proposed [16].

The paper is organized as follows: Section 2 describes the used datasets, the audio features used and its extraction process, the proposed AE architectures, and the evaluation process. Section 3 presents the experimentation developed and obtained results. Lastly, final conclusions are discussed in Section 4.

2 Materials And Methods

2.1 Dataset

The data used for this task comprises parts of the ToyADMOS [17] and the MIMII [27] datasets, consisting of the normal and anomalous operating sounds of

six types of toy/real machines, as obtained from the DCASE 2020 challenge [16]. The data is divided into two datasets (development and evaluation) for 6 different machine types: ToyCar, ToyConveyor, Slider, Pump, Fan, and Valve.

The ToyCar and ToyConveyor data belong to ToyADMOS dataset. This dataset involved miniature machines (toys) that were damaged deliberately to record anomalous behavior. As for the MIMII Dataset, the sounds were recorded from different industrial machines, aiming to resemble a real-life scenario. In the development datasets, each machine type has 4 different specific machines, except for ToyConveyor, which has only 3. Each machine sound was recorded using only one microphone and sampled at 16 kHz.

The machine sound datasets include normal and anomaly labels that are available for the test data, allowing to estimate the AAD performance of the ML models. Regarding the evaluation data, it contains audio for new machines (new IDs) in each machine type, both for model training and testing. Table 1 summarizes the analyzed datasets. A different number of approximately 10 second Waveform Audio File (WAV) files is used for each machine.

Table 1. Summary of the machine AAD datasets.

	Development			Evaluation		
	Machine	Audio Files		Machine	Audio Files	
	ID	Train	Test	ID	Train	Test
Toy Car	01	1000	614	05	1000	515
	02	1000	615	06	1000	515
	03	1000	615	07	100	515
	04	1000	615			
Toy Conveyor	01	1000	1200	04	1000	555
	02	1000	1155	05	1000	555
	03	1000	1154	06	1000	555
Fan	00	911	507	01	911	426
	02	916	549	03	916	458
	04	933	448	05	1000	458
	06	915	461			
Pump	00	906	243	01	903	2016
	02	905	211	03	606	213
	04	602	200	05	908	348
	06	936	202			
Slider	00	968	456	01	968	278
	02	968	367	03	968	278
	04	434	278	05	434	278
	06	434	189			
Valve	00	891	219	01	679	220
	02	608	220	03	863	220
	04	900	220	05	899	500
	06	892	220			

2.2 Feature Extraction

MFCCs, which are derived from the mel-cepstrum representation of the audio, is one of the best known and most popular audio processing features [30]. However, when computing MFCCs, a Discrete Cosine Transform (DCT) is applied to the logarithm of the filter bank outputs, resulting in decorrelated MFCC features. Therefore, they have the drawback of having non-local features, which makes them unsuitable for Convolutional AE (CAE) processing.

In this paper, we address a feature for audio signal processing named MFECs, which are log-energies derived directly from the filter-banks energies. These are similar to MFCCs, yet they do not include the DCT operation. This feature provided good results in detecting different audio sounds and classification of sounds in previous studies [2, 13, 33].

To prepare the features for the first proposed deep learning architecture, the Dense AE, some operations were made. Audio data are buffered in fixed-length 1-second intervals with 50% overlap. For each audio buffer obtained, the segment is then divided into 64 ms analysis frames, with 50% overlap and 128 MFECs are extracted from the magnitude spectrum of each frame. In this way, 5 time-frames are concatenated to form a 640-dimensional input vector as shown in Figure 1.

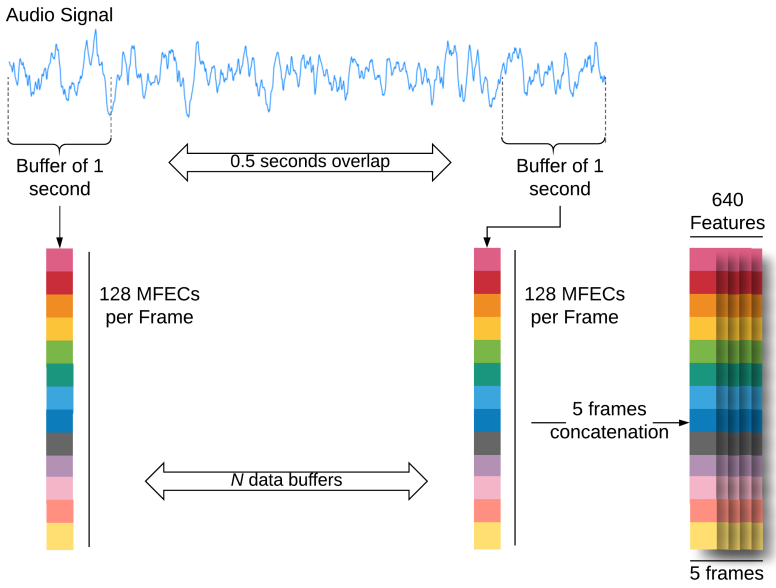

Fig. 1. Feature extraction procedure for the Dense AE.

The second deep learning architecture, the Convolution Neural Network (CNN) AE, requires a different feature extraction method. For each audio, 128 log mel-band energy features were extracted from the magnitude spectrum, con-

sidering 64 ms analysis frames with 50% overlap. Then, each feature was normalized to zero mean and unit standard deviation by using statistics from the training data. Finally, the mel spectrogram was segmented every second into 32 column data, with approximately 100 ms of hop size. This procedure is shown in Figure 2.

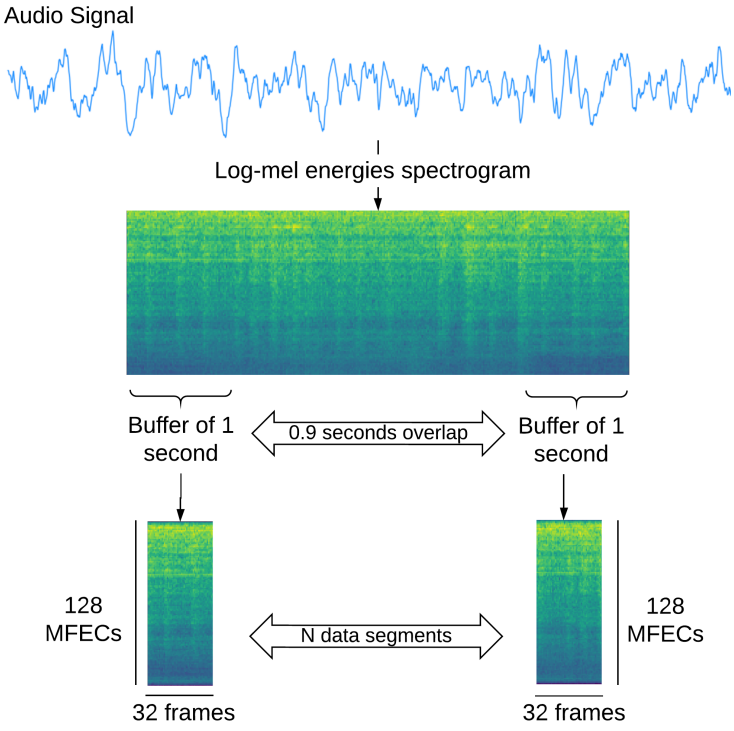

Fig. 2. Feature extraction procedure for the CNN AE.

2.3 Autoencoder Architectures

The two proposed AEs contain a large number of hyperparameters. In order to select the best architectures, we have first conducted several preliminary experiments, in which we only used development data, selecting the best configuration (in terms of the reconstruction error) when varying element such as the number of hidden layers and units per layer. Once the neural architecture was selected, it was fixed and applied to all datasets. For both AEs, the training only uses normal machine sounds.

The first proposed architecture consists of a deep fully-connected AE (top of Figure 3), which was adopted in the baseline AE proposed in [16]. The best

preliminary results were achieved by a Dense AE that includes encoder and decoder components with four fully-connected layers with 512 hidden units, followed by Batch Normalization, with all neural nodes using the popular ReLU as activation function. The Batch Normalization layer allows reduce the internal covariate shift, discarding the need of dropout, and normalizes the inputs for each batch of data [12]. As for ReLU, it presents the advantage of non-saturation of its gradient, which greatly accelerates the convergence of stochastic gradient descent compared to other activation functions, including logistic or hiperbolic tangent [20]. The bottleneck layer is set as one fully-connected layer with 8 hidden units, resulting in an 8-dimensional latent space. To train the AE only normal event audio was used, aiming to learn the data normal event distribution. Turning to the loss function, we adopted the popular Mean Squared Error (MSE), which is more sensitive to extreme errors and that is computed as:

$$MSE_i = \frac{\sum_{k=1}^{I}(x_{i,k} - \hat{x}_{i,k})^2}{I} \tag{1}$$

where i denotes a data instance, $x_{i,k}$ the k-th input value for instance i, $\hat{x}_{i,k}$ the AE predicted output response for the same input and I the total number of inputs of the AE.

Recently, CNNs have achieved promising results on many AAD benchmarks [6, 11, 21]. By integrating 2D convolutional operations in an AE structure, CNN AEs are capable of learning the spatial structure of the input features and reconstruct them while taking into account their spatial structural patterns. Based on this property, the second proposed deep learning architecture for the unsupervised AAD task consists of a deep CNN AE (shown in the bottom of Figure 3). With such an architecture, the AAD task is handled as a computer vision problem by exploring image-like time-frequency representations of audio. The encoder and decoder networks are comprised of convolutional blocks, each consisting of 2D Convolution and Batch Normalization layers, using ReLU as the activation function. The encoder network is composed by a stack of five convolutional layers with 32, 64, 128, 256, and 512 convolutional filters, kernel sizes of 5, 5, 5, 3, and 3 to capture local patterns, and strides of $(1, 2)$, $(1, 2)$, $(2, 2)$, $(2, 2)$, and $(2, 2)$, respectively. The feature size map is reduced throughout the encoder by the convolution operation stride. The bottleneck consists of a layer with 40 convolutional filters, reducing the encoder feature maps to a 40-dimensional compressed input representation. Concerning the decoder network, it starts with a fully-connected layer that increases the latent space dimensionality, equalizing encoder last layer's shape, followed by five 2D transposed convolutional layers that mirror the encoder layers.

Regarding the training algorithm used to train both Dense AE and CNN AE architectures, we employed the Adam optimizer, which also was used in [16], using a learning rate of 0.001. Both AE were trained to minimize MSE between input and its reconstruction (the loss function). The training procedure was iterated up to a maximum of 100 epochs. In each epoch, 10% of training data is randomly divided for validation, which is used for evaluating training process

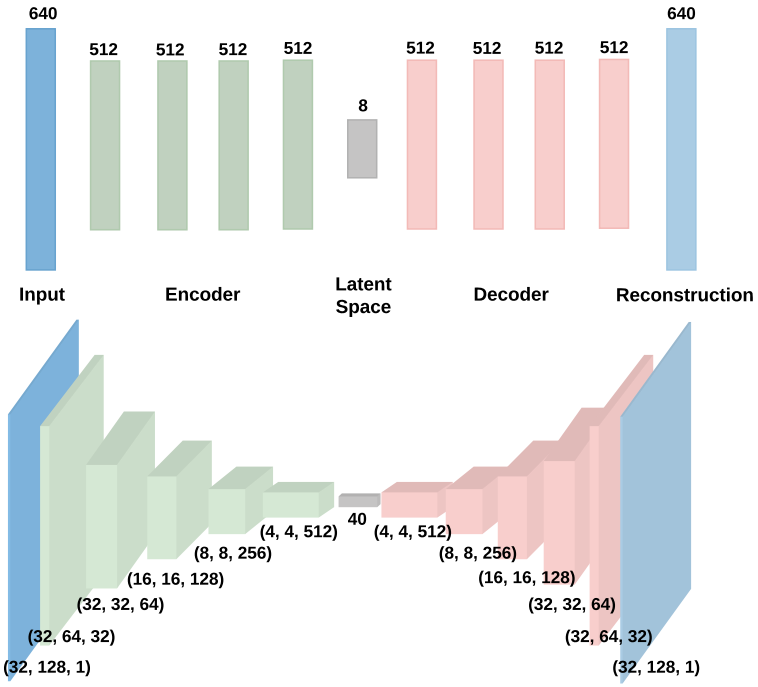

Fig. 3. Proposed AE Network architectures: Dense AE (top) and CNN AE (bottom).

evolution, by computing the reconstruction error on such data. If MSE does not improve on validation data after 10 epochs, an early stopping callback is activated, ending the training process and storing the weights of the model that achieved a lower reconstruction error on validation data. The batch size for both Dense AE and CNN AE architectures was set as 512 and 64, respectively.

Once the AE is trained, the reconstruction error for an unseen sound sample j is used as the decision score (d_j), where $d_j = MSE_j$. A anomaly class label is considered true if $d_j > Th$, where Th is a decision threshold.

2.4 Evaluation

We evaluated our methods, we used two popular metrics on AAD that are based on the Receiver Operating Characteristic (ROC) analysis [17, 27]: Area Under the ROC Curve (AUC) and partial-AUC (pAUC). The ROC curve shows the False Positive Rate (FPR) versus the True Positive Rate (TPR) for different threshold values (Th). In this study, the positive class is the anomaly.

AUC represents the overall ML discrimination performance, while pAUC focuses on a particular range of interest from the ROC curve, defined in this work as the FPR values from 0 to 0.1, which reflects in a model with fewer false alarms. Both metrics are not influenced by unbalanced data, which in occurs in our datasets. The AUC and pAUC values can be interpreted as follows: 50% performance of a random classifier; 60% - reasonable; 70% - good; 80% - very good; 90% - excellent; and 100% - perfect.

As mentioned in Section 2.1, the datasets used in this work contain a total of 6 types of machines, most of them containing 4 specific machine data, except ToyConveyor that contains 3. We used development data to select two first fix AE architectures and then train the models for each specific machine. As for the predictive results, they are measured on the test sets from the evaluation datasets. A single model was created for each machine type and evaluated over each specific machine, using both AUC and pAUC.

3 Results

The proposed dense and CNN AE architectures were implemented in the Python programming language, using the TensorFlow-GPU library [1]. The computational experiments were conducted using two different GPUs (Titan Xp and 1080Ti). To evaluate the model performance, both AUC and pAUC metrics were used, as defined in Section 2.4. Table 2 presents the obtained predictive results for each specific machine, also showing the average value for each machine type. For comparison purposes, the Baseline system results from [16] are also provided in the table.

In terms of the average AUC and pAUC values for each machine type, the Dense AE outperforms the Baseline system in all machine types. Furthermore, the Baseline system only achieved better results in 5 of the 23 analyzed specific machines, namely ToyCar ID 3, pump (IDs of 0, 2, and 4) and slider ID 0. Regarding the CNN AE overall performance, it outperformed the Baseline system, although the latter achieved a higher average AUC values for 2 of 6 machine types (ToyCar and pump).

The two proposed AE architectures[4] are quite competitive in terms of mean AUC and pAUC values, outperforming the Baseline system for all machine types. The Dense AE obtains the best average AUC and pAUC results for ToyCar, ToyConveyor and fan, while the CNN AE achieves the best AAD measures for the slider and valve tasks. Turning to the pump machine, the best AUC value is provided by the Dense AE and the best pAUC is returned by the CNN AE. In particular, when considering the AUC measure, a high quality anomaly class discrimination was achieved by the proposed AEs, since most AUC values are above 70%. Furthermore, the CNN AE architecture obtained excellent results for slider machine type, presenting the highest AUC value (91.77%).

[4] https://github.com/APILASTRI/DCASE_Task2_UMINHO

Table 2. Comparison of AUC and pAUC results for all AE architectures for each machine (best average values are denoted in **bold**)

Machine Type	Machine ID	Baseline		Dense AE		CNN AE	
		AUC (%)	**pAUC** (%)	**AUC** (%)	**pAUC** (%)	**AUC** (%)	**pAUC** (%)
ToyCar	1	81.36	68.40	83.87	72.64	81.59	71.88
	2	85.97	77.72	87.56	80.35	85.46	79.92
	3	63.30	55.21	63.12	55.02	62.73	55.08
	4	84.45	68.97	88.60	76.68	82.38	69.60
	Average	78.77	67.58	**80.79**	**71.17**	78.04	69.12
ToyConveyor	1	78.07	64.25	81.67	69.41	79.90	62.71
	2	64.16	56.01	68.04	58.31	67.78	54.85
	3	75.35	61.03	79.59	63.64	80.11	62.53
	Average	72.53	60.43	**76.43**	**63.79**	75.93	60.03
fan	0	54.41	49.37	56.73	49.72	51.77	49.05
	2	73.40	54.81	79.60	54.00	72.71	55.51
	4	61.61	53.26	70.11	54.11	62.60	52.80
	6	73.92	52.35	81.69	55.15	80.05	53.19
	Average	65.83	52.45	**72.03**	**53.25**	66.78	52.63
pump	0	67.15	56.74	66.94	56.83	66.37	54.95
	2	61.53	58.10	60.77	60.31	54.31	53.58
	4	88.33	67.10	87.00	66.32	94.64	77.26
	6	74.55	58.02	77.53	60.32	76.97	58.05
	Average	72.89	59.99	**73.06**	60.94	72.07	**60.96**
slider	0	96.19	81.44	96.12	82.30	98.86	94.47
	2	78.97	63.68	79.55	64.42	84.06	69.33
	4	94.30	71.98	95.44	76.14	97.69	87.82
	6	69.59	49.02	77.22	49.56	86.46	53.16
	Average	84.76	66.53	87.08	68.10	**91.77**	**76.20**
valve	0	68.76	51.70	74.61	52.28	78.69	52.59
	2	68.18	51.83	76.68	52.72	85.02	55.92
	4	74.30	51.97	79.58	50.96	82.59	53.68
	6	53.90	48.43	57.78	48.73	69.03	50.22
	Average	66.28	50.98	72.16	51.17	**78.83**	**53.10**

4 Conclusions

In this paper, we proposed two AutoEncoder (AE) deep learning architectures for an unsupervised Acoustic Anomaly Detection (AAD) task: a Dense AE and a Convolutional Neural Network (CNN) AE. The two AE architectures were applied to six different real-world industrial machine sound datasets. Using development records from the datasets and sound energy features from mel-spectrograms to preprocess the raw sounds, several preliminary experiments were conducted in order to tune the AE hyperparameters, namely in terms of hidden layers and nodes and activation functions. Then, the selected AE architectures were trained and tested using the evaluation instances from the public domain datasets.

Overall, competitive results were obtained by the Dense and CNN AEs when compared with a recently proposed baseline AE architecture [16]. For two machine types (slider and valve), the best results were achieved by the CNN AE, while the Dense AE provided the best results for the remaining machines (ToyCar, ToyConveyor, fan, and pump). In general, a high anomaly class discrimi-

nation was achieved by both proposed AEs, ranging from 72% (good) to 92% (excellent discrimination level).

As future work, we aim to explore different deep learning architectures for AAD, such as Variational AEs [23]. Furthermore, we intend to study the effect of using audio data augmentation techniques (e.g., pitching, time-shifting, Generative Adversarial Networks) or signal frequency filtering tools, aiming to further improve the AAD results.

5 Acknowledgments

This work is supported by the European Structural and Investment Funds in the FEDER component, through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) - Project n 039334; Funding Reference: POCI-01-0247-FEDER-039334.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems (2016)
2. Afrillia, Y., Mawengkang, H., Ramli, M., Fadlisyah, Fhonna, R.P.: Performance measurement OfMel frequency ceptral coefficient(MFCC) method in learning system of al- qur'an based InNaghamPattern recognition. Journal of Physics: Conference Series **930**, 012036 (dec 2017). https://doi.org/10.1088/1742-6596/930/1/012036, https://doi.org/10.1088/1742-6596/930/1/012036
3. An, J., Cho, S.: Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE **2**(1), 1–18 (2015)
4. Aurino, F., Folla, M., Gargiulo, F., Moscato, V., Picariello, A., Sansone, C.: One-class svm based approach for detecting anomalous audio events. In: 2014 International Conference on Intelligent Networking and Collaborative Systems. pp. 145–151. IEEE (2014)
5. Charte, D., Charte, F., Garca, S., del Jesus, M.J., Herrera, F.: A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines. Information Fusion **44**, 7896 (Nov 2018). https://doi.org/10.1016/j.inffus.2017.12.007, http://dx.doi.org/10.1016/j.inffus.2017.12.007
6. Chen, C., Yuan, W., Xie, Y., Qu, Y., Tao, Y., Song, H., Ma, L.: Novelty detection via non-adversarial generative network. arXiv preprint arXiv:2002.00522 (2020)
7. Chu, S., Narayanan, S., Kuo, C.C.J.: Environmental sound recognition with time-frequency audio features. Audio, Speech, and Language Processing, IEEE Transactions on **17**, 1142 – 1158 (09 2009). https://doi.org/10.1109/TASL.2009.2017438

8. Duman, T.B., Bayram, B., İnce, G.: Acoustic anomaly detection using convolutional autoencoders in industrial processes. In: International Workshop on Soft Computing Models in Industrial and Environmental Applications. pp. 432–442. Springer (2019)
9. Farzad, A., Gulliver, T.A.: Unsupervised log message anomaly detection. ICT Express **6**(3), 229–237 (2020)
10. Harar, P., Galaz, Z., Alonso-Hernandez, J.B., Mekyska, J., Burget, R., Smekal, Z.: Towards robust voice pathology detection. Neural Computing and Applications pp. 1–11 (2018)
11. Hershey, S., Chaudhuri, S., Ellis, D.P., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B., et al.: Cnn architectures for large-scale audio classification. In: 2017 ieee international conference on acoustics, speech and signal processing (icassp). pp. 131–135. IEEE (2017)
12. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. PMLR (2015)
13. Jam, M.M., Sadjedi, H.: Identification of hearing disorder by multi-band entropy cepstrum extraction from infant's cry. In: 2009 International Conference on Biomedical and Pharmaceutical Engineering. pp. 1–5 (2009)
14. Kawaguchi, Y., Endo, T.: How can we detect anomalies from subsampled audio signals? In: 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP). pp. 1–6. IEEE (2017)
15. Kohlsdorf, D., Herzing, D., Starner, T.: An auto encoder for audio dolphin communication. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–7. IEEE (2020)
16. Koizumi, Y., Kawaguchi, Y., Imoto, K., Nakamura, T., Nikaido, Y., Tanabe, R., Purohit, H., Suefusa, K., Endo, T., Yasuda, M., Harada, N.: Description and discussion on DCASE2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring. CoRR **abs/2006.05822** (2020)
17. Koizumi, Y., Saito, S., Uematsu, H., Harada, N., Imoto, K.: Toyadmos: A dataset of miniature-machine operating sounds for anomalous sound detection. In: 2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA). pp. 313–317. IEEE (2019), https://ieeexplore.ieee.org/document/8937164
18. Koizumi, Y., Saito, S., Uematsu, H., Kawachi, Y., Harada, N.: Unsupervised detection of anomalous sound based on deep learning and the neyman–pearson lemma. IEEE/ACM Transactions on Audio, Speech, and Language Processing **27**(1), 212–224 (2018)
19. Koizumi, Y., Saito, S., Yamaguchi, M., Murata, S., Harada, N.: Batch uniformization for minimizing maximum anomaly score of dnn-based anomaly detection in sounds (2019)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25**, 1097–1105 (2012)
21. Li, J., Dai, W., Metze, F., Qu, S., Das, S.: A comparison of deep learning methods for environmental sound detection. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 126–130. IEEE (2017)
22. Liu, Y., Zhuang, C., Lu, F.: Unsupervised two-stage anomaly detection (2021)
23. Ntalampiras, S., Potamitis, I.: Acoustic detection of unknown bird species and individuals. CAAI Transactions on Intelligence Technology (2021). https://doi.org/10.1049/cit2.12007

24. Oh, D.Y., Yun, I.D.: Residual error based anomaly detection using auto-encoder in smd machine sound. Sensors **18**(5), 1308 (2018)
25. Panfilenko, D., Poller, P., Sonntag, D., Zillner, S., Schneider, M.: Bpmn for knowledge acquisition and anomaly handling in cps for smart factories. In: 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA). pp. 1–4. IEEE (2016)
26. Provotar, O.I., Linder, Y.M., Veres, M.M.: Unsupervised anomaly detection in time series using lstm-based autoencoders. In: 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT). pp. 513–517. IEEE (2019)
27. Purohit, H., Tanabe, R., Ichige, T., Endo, T., Nikaido, Y., Suefusa, K., Kawaguchi, Y.: MIMII Dataset: Sound dataset for malfunctioning industrial machine investigation and inspection. In: Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019). pp. 209–213 (November 2019)
28. Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S., Sainath, T.N.: Deep learning for audio signal processing. IEEE J. Sel. Top. Signal Process. **13**(2), 206–219 (2019). https://doi.org/10.1109/JSTSP.2019.2908700, https://doi.org/10.1109/JSTSP.2019.2908700
29. Rovetta, S., Mnasri, Z., Masulli, F.: Detection of hazardous road events from audio streams: An ensemble outlier detection approach. In: 2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS). pp. 1–6. IEEE (2020)
30. Sharma, G., Umapathy, K., Krishnan, S.: Trends in audio signal feature extraction methods. Applied Acoustics **158**, 107020 (2020)
31. Sonntag, D., Zillner, S., van der Smagt, P., Lörincz, A.: Overview of the cps for smart factories project: Deep learning, knowledge acquisition, anomaly detection and intelligent user interfaces. In: Industrial internet of things, pp. 487–504. Springer (2017)
32. Tagawa, T., Tadokoro, Y., Yairi, T.: Structured denoising autoencoder for fault detection and analysis. In: Asian Conference on Machine Learning. pp. 96–111 (2015)
33. Torfi, A., Iranmanesh, S.M., Nasrabadi, N.M., Dawson, J.M.: 3d convolutional neural networks for cross audio-visual matching recognition. IEEE Access **5**, 22081–22091 (2017)
34. Zhu, T., Wang, J., Cheng, S., Li, Y., Li, J.: Retrieving the relative kernel dataset from big sensory data for continuous queries in IoT systems. Eurasip Journal on Wireless Communications and Networking **2019**(1) (dec 2019). https://doi.org/10.1186/s13638-019-1467-4