Bayesian optimized autoencoder for predictive maintenance of smart packaging machines.

ARIFEEN, M. and PETROVSKI, A.

2023

Bayesian Optimized Autoencoder for Predictive Maintenance of Smart Packaging Machines

1st Murshedul Arifeen
School of Computing
Robert Gordon University
Aberdeen, Scotland
d.arifeen@rgu.ac.uk

2nd Andrei Petrovski
National Subsea Centre
Robert Gordon University
Aberdeen, Scotland
a.petrovski@rgu.ac.uk

***Abstract*—Smart packaging machines incorporate various components (blades, motors, films) to accomplish the packaging process and are involved in almost all types of the manufacturing industry. Proper maintenance and monitoring of the components over time can help industries to maintain a sustainable production environment. On the contrary, a faulty system may degrade production efficiency and increase the cost. Smart packaging machines comprising several sensors can generate time series data and leverage data driven condition monitoring models to overcome faulty conditions. In this work, we have studied the application of Autoencoder as a data driven condition monitoring tool for the predictive maintenance of packaging machines. The trained Autoencoder on the new system's data can detect worn or degraded components over time. We have also used the Bayesian optimization algorithm to tune the hyper-parameters of the Autoencoder for better predictive performance. Moreover, the reconstruction error is analyzed to identify the worn components in the packaging machine.**

***Index Terms*—Autoencoder, Bayesian Optimization, Predictive Maintenance, Packaging Machine, Fault Detection**

I. Introduction

Packaging is the process of wrapping, boxing, or bottling goods and products for consumers [1]. Packaging plays a principal role in almost all manufacturing industries like clothing, pharmaceuticals, food, and beverage etc [2] [3]. Different types of machines are involved in the packaging process, for instance, labeling machines, filling machines, tape machines, sealers, film wrapping machines, etc [1]. Each component (motors, blades, films) involved with the packaging machinery has a specific lifetime. However, the performance of the components may degrade before reaching their lifetime due to several environmental factors, including changes in temperature, pressure, load distribution, and many more [4]. As a result, the cyber-physical production system may suffer from declined production efficiency, unwanted downtime, and failure to meet the supply-demand [4]. On the contrary, the increased complexity of the packaging machines makes it difficult for human operators to investigate and identify the degraded components through proper maintenance. Therefore, it is required to estimate the component's faulty behavior through inference techniques to maintain a sustainable production environment without the need of any human intervention.

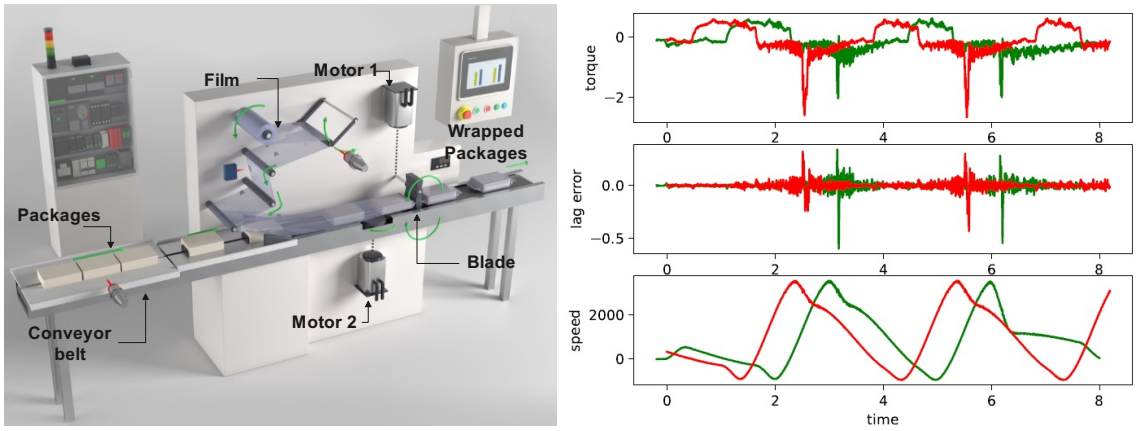
Condition monitoring is a widely applied strategy to overcome such problems [5]. Condition monitoring can be deployed based on the first principal mathematical models (physical based) and data-driven models [5]. Due to the complexity and unknown system dynamics, it is difficult to design a packaging machine through the first principal methods. On the contrary, the 4^{th} industrial revolution made the manufacturing industry autonomous and time efficient with the help of computer-aided technologies [6]. Integration of smart IoT devices like sensors and actuators made predictive maintenance possible by providing enormous amount of data [7] [8]. Therefore, data-driven models are more suitable than the first principal models for condition monitoring and diagnosis. Data-driven models are easy to learn from the data and do not require any system knowledge [4]. Data-driven condition monitoring models can be supervised and unsupervised. Since labeling data instances is a challenging task and generating anomalous data is rare for training a supervised model, unsupervised data driven modeling is more appropriate for condition monitoring.

In this work, we have studied the condition monitoring (components wear detection) of smart packaging machines based on Autoencoder (AE), which is an unsupervised data-driven modeling approach. The AE is trained on the new components data and tested on the degraded components data. The reconstruction error of the AE is used to differentiate new components from worn components. Moreover, we have used the Bayesian optimization process to tune the hyper-parameters of the AE for improving the detection performance.

In section II we have reviewed several recently published papers which show the applicability of AE in tool wear detection. Section III explains the Heat Shrink Wrapper Packaging Machine, the Bayesian optimization process, and the AE model for condition monitoring. In section IV we have discussed the experimental procedure and outcomes. Finally, section V concludes this paper.

II. Literature Review

AE and its variants are widely applied for condition monitoring and tool wear prediction in literature. The unsupervised learning capability of Autoencoder made it an attractive deep learning model for condition monitoring. He et al [10] proposed a Sparse Stacked AE to detect mechanical tool wear from temperature signals. The experimental results show a



(a) (b)

Fig. 1: (a) Schematic diagram of a heat shrink wrapping packaging machine (image taken from [9]). (b) Three features (motor torque, lag error, and blades speed) from the dataset is depicted where the green color represents new blade and red color represents worn blade. The worn blade shows shift in time.

better predictive maintenance outcome than other traditional approaches. Duo et al [11] also used a Sparse Autoencoder to monitor the wear state of a milling cutter from force and vibration data. The mean square reconstruction error is considered to monitor the milling process. On the contrary, Hahn et al [12] combined Variational Autoencoder with Temporal convolutional neural network for milling process monitoring. Similarly, Shi et al [13] proposed a condition-monitoring process for a cutting machine based on Autoencoder with a novel loss function.

Unlike AE based approaches, Birgelen et al [4] showed the application of Self Organized Map (SOM) in systems health prediction. The quantization error of the Self Organized Map is used to determine the worn blades and new blades. Also, they have identified the worn components of the packaging machines. However, the hyper-parameters of the SOM are not optimized. Moreover, if the new blades distribution changes then the SOM may misclassify the new blade as worn blade. In this work we have applied an optimized Autoencoder for condition monitoring (worn component detection) of a smart packaging machine.

III. System Model

A. Heat Shrink Wrapper Machine

A film wrapper machine assembly comprises multiple components to wrap a plastic film around a package. The objective is to cut the plastic film precisely, with the required speed, and at the exact position [14]. Figure 1(a) shows a schematic diagram of a heat shrink film wrapper machine and its working mechanism. The motors (Motor 1 and Motor 2) generates necessary torque (positive and negative) to move the blades up and down at a particular speed for cutting the film. The distance covered by the blades from the initial positions to the final film contact positions can be represented as a lag error. On the contrary, a film spool (as depicted in 1(a)) continuously rotates to provide the necessary plastic films. The film has an initial position and moves at a specified speed. Finally, the packages with the wrapped plastic film pass through a heat shrink tunnel to produce the final product. However, environmental factors like changes in temperature, pressure, or load can degrade the components performance. As a consequence, the blade does not cut at the desired location of the film. Figure 1(b) shows three features of the new and worn blades from the dataset. From the figure, it can be seen that the worn blades lag error and speed have been advanced in time. In other words, the worn blade cut the film before the desired cutting time instant. Also, the motor has been degraded over time as the motor torque is also advanced in time axis. The aim of a predictive maintenance model is to monitor the system degradation and generate alarms for the administrators to replace or fix the degraded elements.

B. Bayesian Optimization

We have used the Bayesian optimization algorithm to tune the hyper-parameters of the AE model [15]. Compared to the Random or Grid search algorithms, Bayesian optimization leverages the past evaluation results to select the next samples from the hyper-parameter space [16]. Therefore, the Bayesian optimization algorithm is more efficient in tuning the hyper-parameters. Since Bayesian optimization keeps track of past search results, it considers narrow regions instead of looking at the whole hyper-parameter space. The Bayesian optimization starts the tuning process by building a surrogate probabilistic model of the objective function. The surrogate model outputs a probability score for a specific combination of hyper-parameters. The best performing hyper-parameters on the surrogate model are applied to the user defined objective function to evaluate its performance on the validation set. The new results from the current iteration are then used to upgrade

the surrogate model. These steps continue until the final trial number is reached.

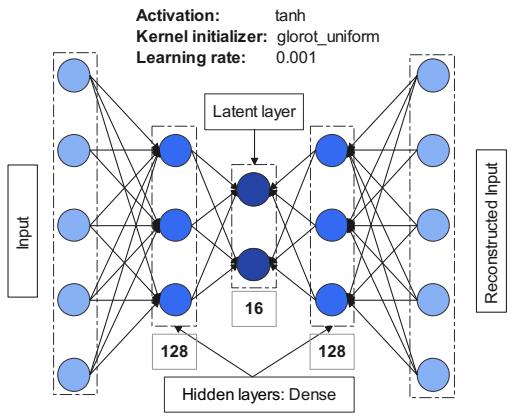


Fig. 2: The proposed AE architecture. The hyper-parameters are optimized using Bayesian optimization algorithm. The input and output layer contains 7 nodes corresponding to 7 features. The first hidden layer consists of 128 nodes, the latent layer has 16 nodes and the final hidden layer also comprises of 128 nodes. All the layers use tanh activation function with kernel initialiser *glorot uniform*. The optimized learning rate returned by the bayesian optimization is 0.001.

TABLE I: Features of the dataset

Features	**Description**
Timestamp	The data are sampled every 8ms.
Motor torque	Torque of the motor for moving the blade
Blades lag error	Blades position error(deviation) from starting to final position
Blades actual position	Actual position of the blade
Blades actual speed	The speed of the blade for cutting the film.
Films lag error	Films position error (deviation) from starting to final position.
Films actual position	Actual position of the film
Films actual speed	The speed of the film to rotate.

C. Autoencoder

An AE is a learning algorithm to learn an informative representation of the data that can be used for other applications by learning to reconstruct the input samples well enough [17]. The basic structure of an AE consists of an encoder, latent representation, and decoder, where the encoders and the decoders are neural networks [18] [19]. But the last layer of the encoder is called the bottleneck layer or the latent representation layer. For reconstructing the input samples, the AE follows unsupervised learning with no label. Lets say we have an unlabeled training dataset D with N number of samples x_i from $i = 1, ..., N$. Mathematically, the unlabeled training data can be represented as, $D = \{x_i | i = 1, ..., N\}; x_i \in \mathbb{R}^n; n \in \mathbb{N}$. The encoder function can be written as $h_i = g(x_i)$, where the h_i is the latent representation layer with the dimension of q, i.e $h_i \in \mathbb{R}^q$. Then the goal of the encoder is to reduce the dimension of the input data from dimension n to q, i.e $g : \mathbb{R}^n \rightarrow \mathbb{R}^q$. On the contrary, the decoder with a function say $f(.)$ reconstructs the input data from h_i. Mathematically it can be denoted as $\bar{x} = f(h_i) = f(g(x_i))$. A loss function representing the reconstructed samples $\bar{x}$ and original samples x is minimized through a learning algorithm by the AE to learn the latent representation of the data. Typically, a deterministic AE follows mean square error (MSE) [18], as a loss function, i.e $Loss = \frac{1}{N}\sum_i^N |x_i - \bar{x_i}|^2$. The hyper-parameters of the AE are tuned through Bayesian Optimization process. The optimized architecture of the AE is depicted in figure 2.

D. Autoencoder for Condition Monitoring

The reconstruction error of the AE can be used to differentiate the new and worn system. We can set a threshold and the reconstruction error below the threshold level indicates new system, while reconstruction error above the threshold indicates a worn system. The reconstruction error based models are already applied in different process monitoring systems as mentioned in the literature review section. However, the hyper-parameter selection of an AE model is a challenging task. In this work we have used the Bayesian optimization to determine the best hyper-parameters of the AE model.

IV. Experiments and Results

A. Dataset

The dataset [20] comes from 3 sets of blades. Each set comprises new and worn packaging machines data with eight features representing the components of the packaging machine and each data file contains 2048 data samples. The features of the dataset are briefly explained in table I.

B. Data Preprocessing

Before applying the AE for condition monitoring, we preprocessed the dataset. The dataset only contains numerical features. The *Timestamp* feature is discarded from the dataset since this feature does not contribute any significant information in model training. We have combined the data files of new blades and worn blades. Therefore, the combined dataset of new and worn blades contain 6144 data samples each. Then, we have used the *MinMaxScaler()* function from the python *Sklearn* library to scale the features.

C. Hyper-parameter tuning

We have used Tensorflow Keras tuner to perform the Bayesian hyper-parameter optimization process. Table II shows the hyper-parameter space. For the first and last layer, we have considered 3 choices of neuron numbers (nodes) that is 32, 64, or 128 and for the bottleneck layer we have selected 8, or 16 neurons. Each layer consists of an activation function. The chosen activation functions [21] are briefly explained below-

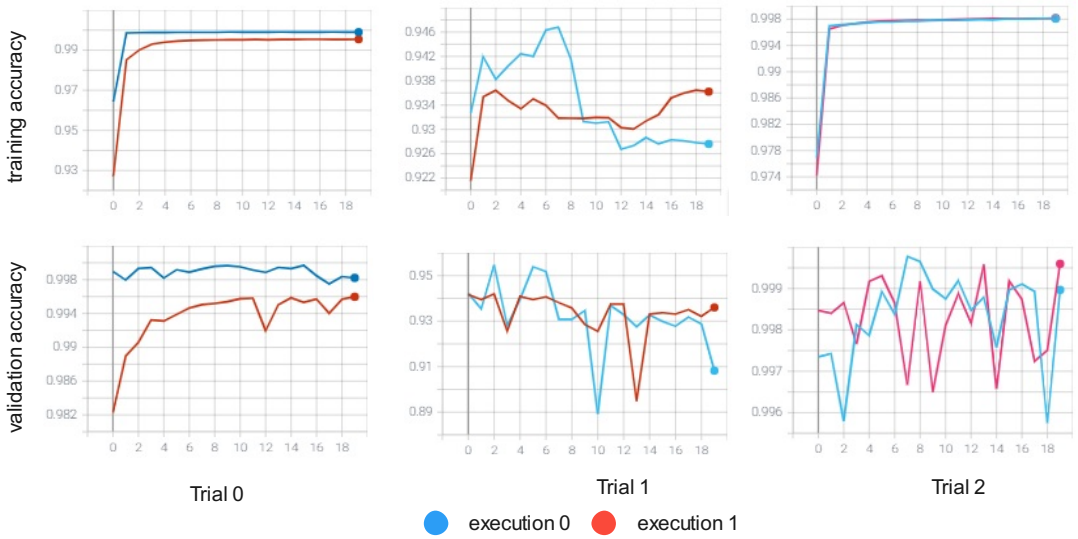


Fig. 3: The results after tuning the hyper-parameters of the AE using Bayesian Optimization process. We have set three trials and in each trial two models are trained. The blue color represents the model 1 (execution 0) and the red color represents model 2 (execution 1) from each trial.

1) ELU: ELU stands for Exponential Linear Unit. This activation function incorporates negative values that enables them to transform mean unit activation near to zero. However, unlike other activation, this function shows lower computational complexity.

$$F(u) = \begin{cases} u & u > 0 \\ \alpha(e^u - 1) & u <= 0 \end{cases}$$

2) ReLU: ReLU stands for Rectified Linear Units. This is a nonlinear activation function like sigmoid but performs better than sigmoid activation functions. This function also eliminates vanishing gradient problem.

$$F(u) = \begin{cases} u & u > 0 \\ 0 & u <= 0 \end{cases}$$

3) Tanh: This nonlinear activation function transforms the input numbers into the range of [-1,1]. The gradient of this activation function is also better than the sigmoid activation function.

$$tan(u) = \frac{\exp(u) - \exp(-u)}{\exp(u) + \exp(-u)}$$

We have also used kernel initializer to assign the weights of the layers instead of random initialization. For the kernel initializer we have used *glorot uniform* which picks up samples from a uniform distribution, *glorot normal* that samples from truncated normal distribution, *he uniform* and finally *he normal*. The learning rate is considered in the range between $1e-4$ to $1e-2$ with log sampling rate.

TABLE II: Hyperparameter Space

Layers	Neurons	Activations	Kernel initializer
Layer 1	32, 64, 128	elu, relu, tanh	glorot uniform, glorot normal, he uniform, he normal
Layer 2	8, 16	elu, relu, tanh	glorot uniform, glorot normal, he uniform, he normal
Layer 3	32, 64, 128	elu, relu, tanh	glorot uniform, glorot normal, he uniform, he normal

D. Model training and validation

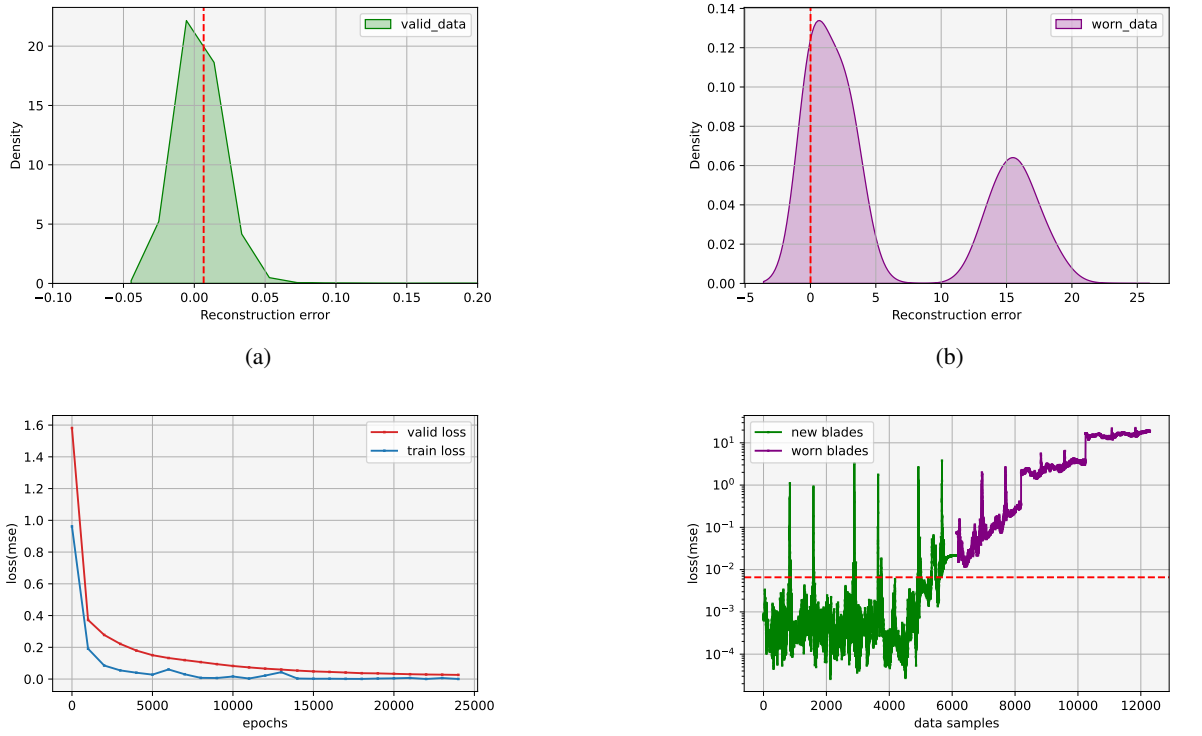
The AE architecture is shown in figure 2. We have built the model depending on the result of the Bayesian optimization process. The first layer is the input layer which contains 7 neurons indicating 7 features of the dataset. The first hidden layer comprises of 128 neurons, $tanh$ activation function and *glorot uniform* kernel initializer. The bottleneck layer comprises of 16 neurons, $tanh$ activation function and *glorot uniform* kernel initializer. The last hidden layer follows the architecture of the first hidden layer. The new blades dataset is splitted into $70:30$ ratio for training and validating the AE with mean squared error loss. Finally, the AE is tested on the worn blades data.

E. Results

For tuning the hyper-parameters of the AE, we have considered 3 trials. In each trial, 2 models are built and the average result is taken as the output. Therefore, after finishing the tuning process we have three scores from 3 trials. Table II presents the Bayesian optimization results. In the first trial, the optimization algorithm chose 128 neurons for first and last layer with kernel initializer *glorot uniform* and activation function *relu*. The bottleneck layer is assigned 8 neurons. However, the validation accuracy for the first trial is 0.99709.

TABLE III: Bayesian Optimization results

Trial	Neuron numbers		Kernel initializer		Activation					
	Layer 1 & 3	Layer 2	Layer 1 & 3	Layer 2	Layer 1 & 3	Layer 2	**train acc**	**valid acc**	**train loss**	**valid loss**
0	128	8	glorot normal	glorot normal	relu	relu	0.99723	0.99709	4.0102E-06	3.0203E-06
1	64	8	he uniform	he normal	relu	relu	0.93191	0.92211	0.00060826	0.00036379
2	128	16	glorot uniform	glorot uniform	tanh	tanh	0.99812	0.99878	3.77E-07	3.52E-07



(a) (b) (c) (d)

Fig. 4: (a) the density plot for the reconstruction error of validation data. (b) the density plot for the reconstruction error of worn data shows. (c) Autoencoder training and testing loss. (d) the reconstruction error for the new and worn blades packaging machines.

Based on the results of the first trial, the Bayesian model selects the hyper-parameters for the second trial as depicted in the second row (Trial 1) of the table II. The validation accuracy for the second trial is 0.92211. Finally, the third trial shows the best result as 0.99878 validation accuracy for the hyper-parameters of 128 neurons of first and second hidden layer, 16 neurons of the bottleneck layer. The activation function and kernel initializer is $tanh$ and *glorot uniform* respectively. Figure 3 demonstrates the training and validation accuracy for 20 epochs of three trials. We can see in the third trial (Trial 2), model 1 (execution 0) and model 2 (execution 1) shows similar performance compared to the previous two trials. Therefore, we have chosen the hyper-parameters from the third trial (trial 2) to build the AE model.

After the optimization process and setting the hyper-parameters from the third trial (Trial 2), we have again trained the AE on the whole dataset from the new blades. Figure 4c shows that the AE training and validation loss, where the training and validation loss almost overlaps each other. To detect the worn system from the new system the reconstruction error of the AE is used. From figure 4d, we can see that the reconstruction error for the worn blades (purple colored) is above the threshold line than the new blades (green colored) on which the AE is trained. The threshold for differentiating worn system from the normal system is defined in equation 1. This equation uses the reconstruction error of validation set from normal data to determine the threshold or limit value which is 0.00659398 . This threshold formula is also known as part average limits defined by Automotive electronics council [22] [23].

$$Threshold = median \pm \frac{inter\_quartile\_range}{1.35} \quad (1)$$

Based on the threshold value from the previous equation, the fault detection rate (FDR) and false alarm rate (FAR) of

the model are computed as 100% and 13% from the following equations-

$$FDR = \frac{f}{F}$$

where, f denotes number of fault data that have been detected as fault and the F refers to total number of faulty samples.

$$FAR = \frac{n}{N}$$

where, n stands for number of normal data that have been detected as fault and N denotes total number of normal samples. The FDR and FAR of the AE is compared with the conventional PCA based approach, where PCA achieved 69.54% FDR and 2% FAR. Although the FAR of PCA is less than AE but PCA can not generalize to detect faults in new data. Therefore, the FDR of PCA is lower than AE.

Moreover, we have analyzed each features contribution to the reconstruction error of the AE for locating the worn system component that is mainly responsible for overall system degradation. The table IV shows the contribution of each of the features in the overall reconstruction error of the AE for the worn system. From the table we can see the *film position* feature is the major component for the total system degradation. *Blades position* feature is also significantly deviated in the worn system.

TABLE IV: Reconstruction error for each feature from the worn blades

Features	**Reconstruction error** $\%$
Motor torque	0.88%
Blades lag error	3.08%
Blades position	13.48%
Blades speed	2.13%
Films position	76.48%
Films speed	2.26%
Films lag error	1.70%

V. Conclusion

In this work, we have studied the application of AE for worn system detection in packaging machines. Since almost all industries utilize packaging machines for wrapping or boxing final products, it is crucial to maintain the packaging system for a sustainable production environment. Any fault may raise unexpected delays or high costs in the production pipeline. This work shows that the AE can be a viable solution to maintain a packaging machine by detecting worn components. Since the components degrade over time, the AE reconstruction error could be a measure to detect the system's abnormal condition. Moreover, by analyzing each feature's contribution to the total reconstruction error, the degraded components within the packaging machine may be replaced to maintain a continuous flow of production.

References

[1] H. May. (2023, Mar.) How does the packaging industry work? webpage. [Online]. Available: https://www.inbusinessmag.com/how-does-the-packaging-industry-work/

[2] Packaging-Gateway. (2020, Dec.) Top ten packaging equipment manufacturing companies. webpage. [Online]. Available: https://www.packaging-gateway.com/features/top-ten-packaging-equipment-manufacturing-companies/

[3] D. Feber, O. Lingqvist, D. Nordigården, and M. Seidner. (2022, Mar.) 2022 and beyond for the packaging industry's ceos: The priorities for resilience. webpage. [Online]. Available: https://www.mckinsey.com/industries/paper-forest-products-and-packaging/our-insights/

[4] A. Von Birgelen, D. Buratti, J. Mager, and O. Niggemann, "Self-organizing maps for anomaly localization and predictive maintenance in cyber-physical production systems," *Procedia cirp*, vol. 72, pp. 480–485, 2018.

[5] G. Serin, B. Sener, A. M. Ozbayoglu, and H. O. Unver, "Review of tool condition monitoring in machining and opportunities for deep learning," *The International Journal of Advanced Manufacturing Technology*, vol. 109, no. 3, pp. 953–974, 2020.

[6] F. Yang and S. Gu, "Industry 4.0, a revolution that requires technology and national strategies," *Complex & Intelligent Systems*, vol. 7, no. 3, pp. 1311–1325, 2021.

[7] M. Arifeen, A. Petrovski, and S. Petrovski, "Automated microsegmentation for lateral movement prevention in industrial internet of things (iiot)," in *2021 14th International Conference on Security of Information and Networks (SIN)*, vol. 1. IEEE, 2021, pp. 1–6.

[8] M. M. Arifeen, D. Bhakta, S. R. H. Remu, M. M. Islam, M. Mahmud, and M. S. Kaiser, "Hidden markov model based trust management model for underwater wireless sensor networks," in *Proceedings Of The International Conference On Computing Advancements*, 2020, pp. 1–5.

[9] Schneider-Electric. (2023, Mar.) Schneider electric packaging automation solutions. webpage. [Online]. Available: https://no.rs-online.com/web/b/schneider-electric/

[10] Z. He, T. Shi, J. Xuan, and T. Li, "Research on tool wear prediction based on temperature signals and deep learning," *Wear*, vol. 478, p. 203902, 2021.

[11] J. Dou, C. Xu, S. Jiao, B. Li, J. Zhang, and X. Xu, "An unsupervised online monitoring method for tool wear using a sparse auto-encoder," *The International Journal of Advanced Manufacturing Technology*, vol. 106, no. 5, pp. 2493–2507, 2020.

[12] T. V. Hahn and C. K. Mechefske, "Self-supervised learning for tool wear monitoring with a disentangled-variational-autoencoder," *International Journal of Hydromechatronics*, vol. 4, no. 1, pp. 69–98, 2021.

[13] C. Shi, B. Luo, S. He, K. Li, H. Liu, and B. Li, "Tool wear prediction via multidimensional stacked sparse autoencoders with feature fusion," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5150–5159, 2019.

[14] Anon. (2020, Aug.) Iiot or industry 4.0 -shrink wrapper - concept. webpage. [Online]. Available: https://www.kaggle.com/code/anshumoudgil/iiot-or-industry-4-0-shrink-wrapper-concept

[15] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, 2012.

[16] V. Nguyen, "Bayesian optimization for accelerating hyper-parameter tuning," in *2019 IEEE second international conference on artificial intelligence and knowledge engineering (AIKE)*. IEEE, 2019, pp. 302–305.

[17] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," *arXiv preprint arXiv:2003.05991*, 2020.

[18] U. Michelucci, "An introduction to autoencoders," *arXiv preprint arXiv:2201.03898*, 2022.

[19] M. Arifeen, T. Ghosh, R. Islam, A. Ashiquzzaman, J. Yoon, and J. Kim, "Autoencoder based consensus mechanism for blockchain-enabled industrial internet of things," *Internet of Things*, vol. 19, p. 100575, 2022.

[20] Anon. (2018, Nov.) Vega shrink-wrapper component degradation. webpage. [Online]. Available: https://www.kaggle.com/datasets/inIT-OWL/vega-shrinkwrapper-runtofailure-data

[21] ——. (2017, Jan.) Machine learning glossary. webpage. [Online]. Available: https://ml-cheatsheet.readthedocs.io/en/latest/activation_functions.html

[22] SemiconductorEngineering. (2021, Feb.) Automotive electronics council (aec). webpage. [Online]. Available: https://semiengineering.com/entities/automotive-electronics-council-aec/

[23] ——. (2011, Dec.) Automotive electronics council (aec). internet draft. [Online]. Available: http://www.aecouncil.com/Documents/AEC_Q001_Rev_D.pdf