Process data based estimation of tool wear on punching machines using TCN-Autoencoder from raw time-series information

Shota Asahi¹, Celalettin Karadogan², Satoshi Tamura¹, Satoru Hayamizu¹ and Mathias Liewald²

¹ Faculty of Engineering, Gifu University, 1-1 Yanagido, Gifu City 501-1193, JAPAN
² Institute for Metal Forming Technology, University of Stuttgart, Holzgartenstr. 17, 70174 Stuttgart, GERMANY
E-mail: asahi@asr.info.gifu-u.ac.jp

Abstract. Tracking the wear states of tools on punching machines is necessary to reduce scrap rates. In this paper, we propose a method to estimate wear state of punches using Temporal Convolutional Network Autoencoder (TCN-Autoencoder), one of the deep learning techniques for learning time-series information with convolutional architecture. Approach involves inputting raw time-series information, such as sensor, vibration and audio data, into TCN-Autoencoder, and calculating the reconstruction error between the output and the input data. The reconstruction error is used as “anomaly score” and indicates the distance from the normal state. By training TCN-Autoencoder only with data annotated as “normal” state, the reconstruction error becomes larger when inputting abnormal state data, which corresponds the wear state of the punch. Performance is evaluated on experimental measurement data that spans various wear states of the punch. The results showed our model can estimate anomalies faster than the conventional machine-learning-based anomaly estimation method, while maintaining the high estimation accuracy. This is due to TCN-Autoencoder being able to learn from both frequency and time domain.

1. Introduction
Lack of online monitoring of punch wear at punching machines causes time and material losses. Punch wear, in the most classic sense, is detected by the operator through deteriorated part quality only after such parts have been produced. In this context, online monitoring of the punch wear condition not only eliminates time and material losses, but also enables accurate control of part quality and timely planning of punch replacement instead of an abrupt production stop. Tool monitoring can be performed by measuring hole quality or punch volume [1]. Alternatively, correlations implemented as soft sensors of indirect quantities may allow an easy monitoring of tool state [2]. Due to the ease of implementation on the shop floor such indirect approaches gained the interest of researchers. Force values and force-displacement curves [3, 6, 7, 8], strain measurements far or nearby the punch [4, 5], acoustic emission [12, 13, 11, 9], sound [5, 14, 15], tool vibration [10] and collective analysis of diverse sensor data [16] have been considered as potential sources of information to estimate the wear state of the tool. The majority of mentioned publications calculated correlations using feature engineering approaches or only a...
few of them [13, 16] considered classification based machine learning as an alternative solution. The authors prefer deep learning techniques to the feature engineering approach because the latter may require time-consuming programming that may not even produce accurate results. With the availability of a sufficient amount of data, deep learning techniques outperform feature engineering approaches and do not require tedious programming of extensive feature extraction routines. This study proposes the use of the correlation between the wear state and the "anomaly score". It is necessary to have classes to classify the input data into groups of wear states. Each wear condition is naturally associated with a quantifiable geometric measure of tool wear. Increasing the number of such classes can even allow a smoother tracking of the geometry of the tool. Approach involves inputting raw time-series information, such as sensor, vibration and audio data, into TCN-Autoencoder, and calculating the reconstruction error between the output and the input data, which then yields the anomaly score for the processed sensor data. The success of this approach is demonstrated at the end by demonstrating not only the prediction capability but also by comparing it with classical classification approaches.

2. Data set
The data set are constructed based on data obtained from various sensors installed in a punching machine. Used sensors measure two vertical displacement values being near and far to the punch, pressure in hydraulic system, sound pressure and finally the punching force. The sampling frequency is 20 kHz for the microphone, and load cell, and 2 kHz for the other sensors. Since all sensor data are used under the same conditions, we perform down-sampling on the series data with the higher frequency so that the sampling frequency of all sensor data is 2 kHz. To generate the data, a punch is used and the entire life of the stamp is observed until it is completely worn out. Not the complete service life of the punch is recorded but only at three phases, namely; new, half-worn and worn, states are recorded each with 2250 punching operations. The number of data for each label, "new", "half-worn" and "worn" are almost the same.

3. State-of-the-art classification of wear states
To have a basis for comparison, we conducted first of all a three-class classification experiment using Support Vector Machine (SVM), a machine learning algorithm used as the classification method.

3.1. Experimental settings
The experimental settings are given in Table 1. The collected data was randomly divided into two halves as training and evaluation data sets. Time-dimensional contextual information in the input data is not included. A metric called F-score is used for evaluation purposes. The F-score is the harmonic mean of precision and recall. In this respect the F-score can make a comprehensive and quantitative evaluation based on both of these indicators.

| Condition                        | Value                                      |
|----------------------------------|--------------------------------------------|
| Classification algorithm         | Support Vector Machine (SVM)               |
| Kernel for SVM                   | RBF                                        |
| Number of the class (label)      | 3 (New, Half-worn, Worn)                   |
| Classification unit              | data point (1 point corresponds to 0.5 ms) |
| Number of training data          | 326,687 (New: 109,041, Half-worn:108,888, Worn: 108,758) |
| Number of test data              | Same as training data                      |
3.2. Result
The results of the classification based estimation is given in Table 2 in the form of a confusion matrix. The F-score of the three-class classification was 0.97538. This result indicates that the classifications was performed with high accuracy. In other words, the labels of the wear state and the values of each data point have a high correspondence even when the time series information is not included. This prediction quality will be used as base for comparison to evaluate the proposed approach.

Table 2. Experimental settings of AEs

| True label | Predicted label | New | Half-worn | Worn | Total |
|------------|-----------------|-----|-----------|------|-------|
| New        | 108,380         | 332 | 329       |      | 109,041|
| Half-worn  | 379             | 102,795 | 5,714 |     | 108,888|
| Worn       | 243             | 1,043 | 107,472 |     | 108,758|
| Total      | 109,002         | 104,170 | 113,515 |     | 326,687|

4. Proposed approach using anomaly detection methods
Authors believe that a method for calculating the degree of anomaly based on deep learning from raw data will outperform classification methods. The conceptual flow of the proposed approach is demonstrated in Figure 1. The proposed approach makes use of Temporal Convolutional Networks based Autoencoders (TCN-AE), which is explained below. Reader is suggested to refer the most actual information on the alternative anomaly detection methods used in this study for comparison purposes [23].

Figure 1. The flow of the proposed method: Use of Autoencoders to calculate anomaly score.

4.1. Autoencoder
The most classical autoencoder (AE), as shown in Figure 2, is an encoder-decoder neural net structure consisting of nodes with symmetric number of Feed-Forward Neural Network (FFNN) layers and dimensions based on a central middle layer, [17]. AE performs supervised learning using the same data in the input and output layers, and reconstructs the input data during inference.

In the field of anomaly detection, the AE structure is trained using a data set consisting only or mostly of normal samples. Having trained to reconstruct the normal data, the AE will be
unable to reconstruct the abnormal data and the reconstruction error will be high. By treating this reconstruction error as the degree of anomaly score, the system can detect abnormal data.

4.2. Temporal Convolutional Networks based Autoencoders (TCN-AE)

A Temporal Convolutional Network (TCN) is an NN structure that learns by repeating one-dimensional convolution in the temporal direction for serial data such as voice and sensor data. TCN-AE, as shown in Figure 3, is an encoder-decoder model consisting of one-dimensional convolutional, pooling and upsampling layers designed so that the number of layers and dimensions of the encoder and decoder parts are linearly symmetric, [18].

Figure 2. The example architecture of Autoencoder (AE).

Figure 3. The structure of the utilised Encoder-Decoder TCN (ED-TCN).

4.3. Training and the calculation of anomaly score

Considered AE-Models are trained on data samples only from the data set in the normal state. With this process, the model learns the distribution of data in the normal state, and is designed to output a low anomaly score when data in the normal state is input, and a high anomaly score when data in the near-failure state is input. The anomaly score is calculated using the vector of the reconstruction error. With an input data with data length $T$ is $X = \{x_1, x_2, ..., x_T\}$ and the output result $Y = \{y_1, y_2, ..., y_T\}$, the anomaly score $a(Y; X)$ is calculated by the following equation (1).

$$a(Y; X) = \sum_{t=1}^{T} (y_t - x_t)^2$$ (1)

5. Experiment

To demonstrate the effectiveness of the proposed deep-learning-based anomaly detection method comparisons with the conventional anomaly detection method, and the classification method conducted in the preliminary experiment are necessary. Here we introduce in short further conventional anomaly detection methods.

5.1. Conventional anomaly detection method

Anomaly score calculations are performed also using four conventional machine learning algorithms: Hotelling’s T-square (T2), Gaussian Mixture Model (GMM), K-Nearest
Neighborhood (KNN) and One Class Support Vector Machine (OC-SVM). However, the number of neighbors in KNN is $k = 3$.

T2[19] is one of the anomaly calculation method based on statistical model. For training the model, the probability density function $N(x|\mu, \Sigma)$ is calculated under the assumption that the training data set contains almost no anomaly data samples. Then, when calculating the anomaly score of new sample $x'$, it is calculated using the equation (2). The equation (2) is based on the Mahalanobis distance calculation method.

$$a(x') = (x' - \mu)^T \Sigma^{-1} (x' - \mu)$$

GMM is a mixture model that represents a given data set by superimposing $K$ normal (Gaussian) distributions $N(x|\mu_k, \Sigma_k)$ ($k = 1, 2, ..., K$), distributions [19]. When used for anomaly detection, as same as T2, we train GMM under the assumption that the training data set contains almost no anomalous data samples. Then, when calculating the anomaly score of new sample $x'$, the inverse of weighted log-posterior probability for each data calculated by the model is output as the anomaly score (3).

$$a(x') = -\ln\left( \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k) \right)$$

KNN is the simplest machine learning method and a non-parametric classification method [20]. When used for anomaly detection, the model is trained based on the assumption that the training data set contains almost no anomalous data samples, and when calculating the anomaly score, the model outputs the sum or average of the distances to the $k$ nearest neighbors. In the experiment, we set the number of nearest neighbors as $k = 3$.

OC-SVM is an anomaly detection method using a pattern recognition algorithm called Support Vector Machine (SVM), [21]. Under the assumption that the training data set contains almost no anomaly data samples, only one class of normal class is used as a label for the training data. The data is mapped into the feature space using a kernel, and the training data is mapped to a location far from the origin. In the mapped feature space, the system learns to set the discriminative boundary to maximize the margin to the origin. For anomaly detection, when data similar to the training data are input, they are clustered far from the origin in the feature space, and when data dissimilar to the training data are input, they are clustered close to the origin. Using this property, we can classify the data as normal or abnormal using discriminative boundaries. We use the inverse of distance as the anomaly score.

5.2. Experimental settings

The conditions for this experiment are shown in Table 3. Each data set is normalized to have mean 0 and variance 1 for each dimension. After that, the data set is cropped into a size of 256 points in length and used as input data. The dimensions and shape of the input data for each method is shown in Table 3. The shape of the input data for TCN-AE is a two-dimensional array with (256 (points), (number of data dimensions)). The input data for AE and OC-SVM are obtained as follows. First, the input data for TCN-AE, a two-dimensional array, is downsampled from 256 points to 128 points in the time direction. Then, it is flattened to a one-dimensional array. Thus, the shape of the input data for AE and OC-SVM is (128 point x dimensionality of the data). The input data for KNN, Hotelling’s T-square method and GMM are obtained as follows. First, the input data of TCN-AE, a two-dimensional array, is flattened to a one-dimensional array, and then conduct dimension reduction by Principal Component Analysis (PCA). Radial Basis Function (RBF) is used as the kernel for SVM classifier. The RBF kernel
is one of the most common kernel functions, which is used to nonlinearize a linear algorithm, such as SVM, that deals only with inner products. The number of dimensions whose cumulative contribution ratio exceeds 95% is treated as the number of dimensions of the input data. The experimental settings for AE and TCN-AE are shown in Table 4, and the structures of TCN-AE and AE are shown in Table 5, and Table 6, respectively.

For training each model, only the data labeled with the state NEW state used. Only the half of the NEW state data was used for training keeping the rest for testing purposes.

The evaluation of the performance is based on two metrics, Area Under Curve (AUC) and anomaly score calculation time. The simple moving average and the simple moving minimum are calculated for each of the 200 data points in order to absorb the effect on anomaly detection caused by the variance among the anomaly levels. The AUC is calculated from the simple moving minimum values of the anomaly levels.

### Table 3. Experimental settings

| Condition                          | Value                  |
|------------------------------------|------------------------|
| Data length after cropping         | 256 point              |
| Data overlap length after cropping | 64 point               |
| Number of data after cropping      | 70,295                 |
| Number of data for training        | 7,029                  |
| Shape of Input data of TCN-AE      | (256,5)                |
| Shape of Input data of AE and OC-SVM| 640                    |
| Shape of Input data of KNN, T2 and GMM| 347                    |
| Dimension reduction standard of PCA| cumulative contribution ratio = 95% |
| Unit of calculation for moving average / minimum| 200 data             |
| Model evaluation metrics           | AUC, anomaly score calculation time |

### Table 4. Experimental settings of AEs

| Condition                                      | Value                                      |
|-----------------------------------------------|--------------------------------------------|
| OS for the experiment                         | Windows 10                                 |
| Framework                                     | Keras[23] (TensorFlow[22] backend)        |
| Activation function for TCN                   | Normalized ReLU                            |
| Activation function for the output of AE      | Linear                                     |
| Loss function                                 | Mean Squared Error                         |
| Optimization algorithm                        | Stochastic Gradient Decent                 |
| Learning rate                                 | 0.01                                       |
| Training epochs                               | 2000 (with Early Stopping)                 |

5.2.1. Area Under Curve  The Area Under Curve (AUC), one of the evaluation metrics in this experiment, is an index for evaluating the performance of binary classification and is calculated from a graph called the Receiver Operatorating Characteristic curve (ROC curve) [19]. When used as an metrics, the larger the AUC, the higher the classification performance of the anomaly detection model, and AUC=1 is used for complete (error-free) classification.

Since there were three classes of labels in this experiment, the following three types of AUCs were calculated and used for evaluation.
Table 5. Architecture of TCN-AE

| Layer             | Filter size | Output shape |
|-------------------|-------------|--------------|
| input             | (256,5)     |              |
| Convolution 1D    | 16          | (256,16)     |
| Max Pooling 1D    | -           | (128,16)     |
| Convolution 1D    | 32          | (128,32)     |
| Max Pooling 1D    | -           | (64,32)      |
| Up Sampling 1D    | -           | (128,32)     |
| Convolution 1D    | 16          | (128,16)     |
| Up Sampling 1D    | -           | (256,16)     |
| Convolution 1D    | 5           | (256,5)      |

Table 6. Architecture of AE

| Layer     | Output shape |
|-----------|--------------|
| input     | 640          |
| Full Connection | 320          |
| Full Connection | 160          |
| Full Connection | 320          |
| Full Connection | 640          |

- AUC₁: The AUC of the moving minimum when binary classification is performed with the NEW state as the first class and the HALF-WORN state as the second class.
- AUC₂: The AUC of the moving minimum when binary classification is performed with the HALF-WORN state as the first class and the WORN state as the second class.
- AUC₃: The AUC of the moving minimum when binary classification is performed with the NEW state as the first class and the WORN state as the second class.

5.3. Result and discussion

In this section, we show tables of the transition of the anomaly levels calculated by the six methods, the AUC based on the moving minimum, and the results of the anomaly score calculation time for all samples. The three introduced types of AUCs and the calculation time for the anomaly scores are shown in Table 7. Tabulated comparison results are discussed here. Performance of presented models are measured with two metrics, AUC and anomaly score calculation time. The AUCs of all three methods, T2, KNN, and TCN-AE, exceeded 0.99, indicating that the three states could be classified almost completely. In addition, despite GMM, anomaly score calculation time is smaller for our AE-based anomaly detection models than the conventional methods. In particular, TCN-AE reduces the anomaly calculation time by a factor of 0.2705 compared to KNN, which has scored similar AUC. This indicates that the model constructed by TCN-AE is able to detect anomalies in this data set faster and more accurate than the other models presented here. We also found that that the proposed model is effective for both aspect: detection performance and real-time detection. Regarding the comparison between AE-based anomaly detection models, the accuracy of the TCN-AE model is higher than that of the AE model. This is because the TCN-AE is able to learn both frequency and time information with the one-dimensional convolutional architecture, and thus is able to calculate the degree of anomaly closer to the actual state of the machine.
Finally, the comparison is done against SVM classifier. All AUC were nearly 1.0 when the three labels, New, Half-worn, and Worn, were classified using the proposed method of TCN-AE. This result means that the 3 class classification accuracy is 100% when the most appropriate threshold is set. On the other hand, when the classification was done by SVM classifier, the F-score was 0.9754. Although these two metrics are not directly comparable, we argue that the proposed anomaly detection method using the raw time-series information is also more accurate than the conventional classification methods.

6. Conclusion
In this paper, we proposed a method for detecting the wear level of the punch of a punching machine. We proposed a method to calculate the anomaly level for detecting anomalies using Autoencoders (AE) from the raw time-series information, such as sound, force and pressure data. We designed the model to input the raw time-series information collected from sensors directly, and output the anomaly score. This architecture eliminates the data preprocessing time and reduces the time to calculate the anomaly score. To apply the proposed method to anomaly detection, only the data set in the normal state is used for training. Since the model learns the distribution of the data in the normal state, it is expected that the anomaly score will be higher than that in the normal state when the data close to the failure is input. To evaluate the performance of the proposed method, accuracy and calculation time were chosen as basic criteria. Regarding the obtained results, we showed that the model constructed by TCN-AE can classify faster and with more accuracy than the alternative approaches, like classical classification and new alternative anomaly detection methods. This can be attributed to the fact that the TCN-AE model is capable of learning by taking into account both frequency and time information, and thus is able to calculate the degree of anomaly closer to the state of the machine. Our future work includes the detection of types of minimum number of sensors when the proposed method is utilised as the anomaly detection tool.

Acknowledgement
Authors would like to thank the TRUMPF GmbH + Co. KG for providing the analysis data.

References
[1] De Silva CW (2007) Vibration monitoring, testing and instrumentation. Taylor & Francis, Boca Raton.
[2] Byrne G, Dornfeld D, Inasaki I, Ketteler G, KonigW, Teti R (1995) Tool condition monitoring(TCM)—the status of research and industrial application. CIRP Ann-Manuf Techn 44:541–567
[3] Jin J, Shi J (2000) Diagnostic feature extraction from stamping tonnage signals based on design of experiments. J Manuf Sci E-T ASME 122:360–369
[4] Li X, Bassiuny AM (2008) Transient dynamical analysis of strain signals in sheet metal stamping processes. Int J Mach Tool Manu 48:576–588
[5] Doege E, Meiners F, Mende T, Strachew W, Yun JW (2001) Sensors in manufacturing: sensors for the punching process. Chap 4.2.1 Toenhoef HK, Inasaki I, Wiley-VCH Verlag GmbH, pp 172–181
[6] Klingenberg W, Singh UP (2004) Principles for on-line monitoring of tool wear during sheet metal punching. Proceedings of the 34th International MATADOR Conference. Springer, London, p 169–174
[7] Lee WB, Cheung C, Chiu WM, Chan LK (1997) Automatic supervision of blanking tool wear using pattern recognition analysis. Int J Mach Tool Manu 37(8):1079–1095
[8] Breitling J, Pfeiffer B, Altan T, Siegert K (1997) Process control in blanking. J Mater Process Tech 71:187–192
[9] Ge M, Zhang GC, Du R, Xu Y (2002) Feature extraction from energy distribution of stamping process using wavelet transform. J Vib Control 8:1023–1032
[10] Sari, D.Y., Wu, TL. & Lin, BT. (2017) Preliminary study for online monitoring during the punching process. Int J Adv Manuf Technol 88, 2275–2285.
[11] Zhang GC, Ge M, Tong H, Xu Y, Du R (2002) Bispectral analysis for on-line monitoring of stamping operation. Eng Appl Artif Intel 15:97–104
[12] Kim BS (1983) Punch press monitoring with acoustic emission (AE) part I: signal characterization and stock hardness effects. J Eng Mater-TASME 105(4):295–300
[13] Wadi I, Balendra R (1997) Using neural networks to model the blanking process. J Mater Process Tech 91:52-65
[14] Schuler G (1998) Metal forming handbook. Springer, Berlin
[15] Ubhayaratne I, Xiang Y, Pereira M, Rolfe B (2015) An audio signal based model for condition monitoring of sheet metal stamping process. Ind. Electronics and Applications, IEEE 10th Conf., 15–17 June 2015
[16] Mardapittas AS, An YHJ (1992) Expert system for tool wear monitoring in blanking intelligent fault diagnosis-part 1: Classification based techniques. IEE Colloq 25:2/1-2/4
[17] Baldi, P., "Autoencoders, unsupervised learning, and deep architectures," Proc. International Conference on Machine Learning, pp.37-49, 2012.
[18] C. Lea, M. D. Flynn, R. Vidal, A. Reiter and G. D. Hager, "Temporal conv. networks for action segmentation and detection," Proc. IEEE Conf. on Computer Vision & Pattern Recognition 2017, pp.156-165.
[19] T. Ide and M. Sugiyama: Anomaly detection and change detection, Kodansha Scientific, 2015.
[20] Cover, T., Peter, H., "Nearest neighbor pattern classification," IEEE transactions on information theory, Vol.13, No.1, pp.21-27, 1967.
[21] Khreich, W., Khosravifar, B., Hamou-Lhadj, A., Talhi, C., "An anomaly detection system based on variable N-gram features and one-class SVM," Information and Software Technology, Vol.91, pp.186-197, 2017.
[22] Abadi, M., Barham, P., et al. "TensorFlow: a system for large-scale machine learning," Proc. 12th USENIX conference on Operating Systems Design and Implementation, pp.265-283, 2016.
[23] "Keras Documentation," https://keras.io/.