Machine Learning in Applications
Detection of Anomalous Behaviour in Industrial Robot

Can Karaçomak (s287864)
Christian Dal Pozzolo (s303406)
Joao Gomes Maurício (s318170)

Politecnico di Torino

Professors:
Santa Di Cataldo
Alessio Mascolini **(Project Owner)**

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Machine Learning in Applications Detection of Anomalous Behaviour in Industrial Robot

***Abstract*—Efficient operation and reliability are critical features that must be maintained in industrial robots, while detecting anomalous behaviour promptly. Anomalies can arise due to mistakes like configuration errors or ageing, leading to reduced precision as well as slower motion, compromising productivity while posing safety risks if unchecked. Our work proposes a comprehensive approach utilizing time series data collected from various sensors for robust detection of anomalous robotic behaviour.**

We present the use of an adversarial autoencoder (AAE) model capable of identifying unusual and unwanted production patterns and differentiating them from established normal activity modes for improved anomaly detection capabilities beyond traditional autoencoder models' performance.

To establish the effectiveness and efficiency of our proposed method, we examine an innovative evaluation protocol considering overestimation or underestimation issues in performance metrics while establishing the baseline model: an untrained and a trained model (both initialized with Xavier initialization) – as suggested in the original paper [4]. We evaluate our method on a real world dataset obtained from an industrial robot, detecting its capability to identify even slight anomalous robotic behaviour convincingly.

**** Our results show that our proposed methodology provides improved detection accuracy and much better performance than existing methods. Furthermore, visualization techniques provide a comprehensive analysis of anomalous behaviours that prove beneficial in understanding the underlying issues associated with industrial robot operations, leading to effective problem resolution.**

**** These findings contribute to the field of industrial robotics by providing an effective solution for the detection of anomalous behaviour. This enables timely identification and mitigation of potential issues, enhancing the overall performance, reliability, and safety of industrial robot systems.**

I. Introduction

The final objective of this project is to implement an adversarial autoencoder (AAE) for anomaly detection on a dataset obtained from a Kuka industrial robot and expose the issues and their possible solutions during the development of the project. The dataset comprises time-series data collected from various sensors of the robot, including joint angle positions, velocity, current, and power usage values. The main challenge is to identify instances when the robot's movements deviate from normal behaviour due to configuration errors or ageing, resulting in reduced precision or slower motion.

To tackle this problem, we will train an AAE model using the provided training data. The model will learn the regular movement patterns exhibited by the robot. By capturing the normal behaviour, the AAE should be capable of recognizing any deviations from the learned patterns and flagging them as anomalies. The project's objective is to assess whether the inclusion of an adversarial component in the autoencoder improves its performance for anomaly detection, in comparison to a traditional autoencoder.

In order to evaluate the effectiveness of the AAE model, appropriate metrics will be selected. These metrics will measure the model's ability to accurately identify anomalies while minimizing false positives and false negatives. The choice of these evaluation metrics will play a crucial role in assessing the performance and determining the success of the implemented AAE for anomaly detection on the Kuka industrial robot dataset.

II. Background

For this project, it was done meticulous research about this topic. Started from reading a paper about Time-Series Anomaly Detection, which is the process of identifying unusual patterns or events in a sequence of date that is ordered by time. This is usually used for:

1) Fraud Detection
2) Network Intrusion Defection
3) Predictive maintenance to detect anomalies in real-time data.

A. Temporal Anomaly Detection (TAD)

To establish a new baseline for Temporal Anomaly Detection (TAD), the authors propose using the F1-score obtained from the prediction of a randomly initialized reconstruction model with a simple architecture, such as an untrained autoencoder consisting of a single-layer LSTM. This baseline serves as a reference point for comparison, similar to how baseline accuracy is used in classification tasks.

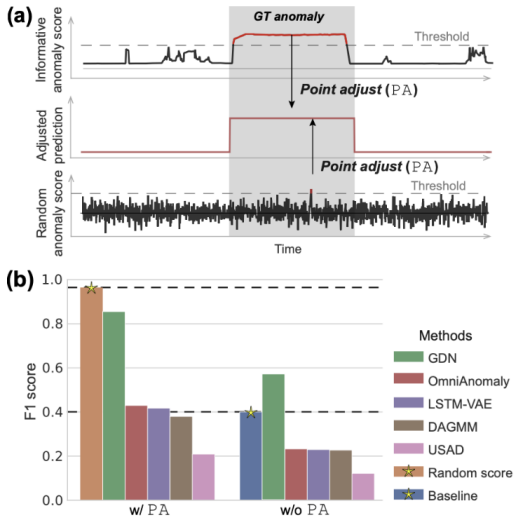
Alternatively, the authors suggest defining the anomaly score as the input itself, which represents an extreme case where the model consistently outputs zero regardless of the input.

If the performance of the new TAD model does not surpass this established baseline, it indicates that the model's effectiveness should be reevaluated. This means that the TAD model needs to demonstrate improvement over the random reconstruction model or show a higher anomaly score than the input-based baseline.

By comparing the performance of the TAD model against these baselines, the authors can assess the effectiveness and value of their proposed method in detecting anomalies in time series data.

B. Evaluation Methods

There are many evaluation methods that can be used, and the paper referred about **Point Adjustment (PA)**. This protocol is a set of rules used to determine when and how to modify scores (errors or bias in the scoring process). It identifies when adjustments are needed, establishes a decision-making process, provides clear guidelines for adjustments, emphasizes communication and transparency, and emphasizes documentation and record-keeping. This protocol ensures fairness and accuracy in adjusting points and maintains the integrity of competitions or assessments.



The paper also introduced us to a new evaluation method, the **PA%**. The authors highlight that using traditional performance measures like F1-score in combination with point adjustment (PA) can lead to overestimation of detection performance. However, without PA, F1-score can sometimes underestimate the detection capability depending on the distribution of the test data.

NOTE: F1-SCORE is a metric used to evaluate the performance of a binary classification model. It combines precision and recall into a single value, providing a balanced measure of the model's accuracy.

To investigate this issue, the authors employ t-distributed stochastic neighbour embedding (t-SNE) on a test dataset from secure water treatment (SWaT). The t-SNE visualization reveals that while most anomalies form a distinct cluster away from normal data, there are abnormal instances that are closer to normal data points. The authors provide visualizations of signals corresponding to these instances.

To address the potential underestimation of detection capability, the authors suggest setting a baseline and adjusting the parameter K manually based on prior knowledge. A larger K is allowed if the test set labels are reliable. Alternatively, to remove dependency on K, they propose measuring the area under the curve of F1 PA%K by incrementally increasing K from 0 to 100.

The PA%K evaluation protocol is proposed as a solution to mitigate overestimation or underestimation issues in the detection performance evaluation, particularly when using F1-score with point adjustment, by adjusting the parameter K based on prior knowledge or by measuring the area under the curve of F1 PA%K.

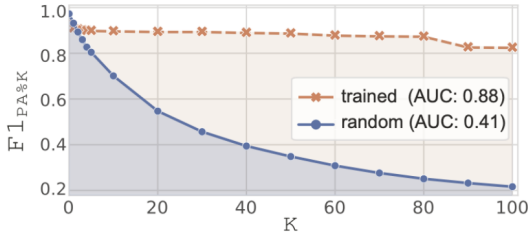


Fig. 1. F1-scores in different K values for trained and randomly initialized models.

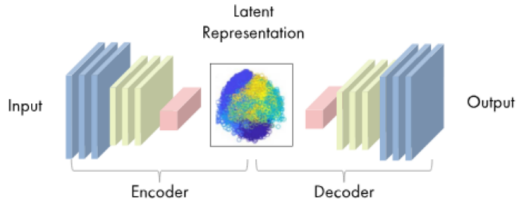
In Conclusion, by addressing the pitfalls in TAD evaluation, we can avoid misguided rankings and properly assess the improvement of TAD methods. It is important to consider the following steps:

1) Avoid using the point adjustment (PA) protocol: The PA protocol can overestimate the detection performance of TAD methods. Instead, evaluate TAD methods without applying PA.
2) Establish a proper baseline: A baseline should be discussed and established for TAD. Future methods should be evaluated based on their improvement compared to this baseline.
3) Compare against better datasets: Several papers have proposed better datasets for TAD evaluation. Utilise these datasets to assess the performance of TAD methods accurately.

By following these steps and considering the limitations of current evaluation practices, we can rigorously evaluate TAD methods and make meaningful comparisons between different approaches.

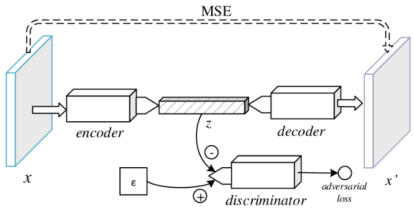
C. Autoencoder [1]

An autoencoder is a specific form of artificial neural network employed for acquiring compact representations of unlabelled data. It consists of two primary functions: an encoder, which converts the input data into a compressed form, and a decoder, which reconstructs the original data using this condensed representation.



D. Adversarial Autoencoder [2]

The concept behind an adversarial autoencoder involves implementing ideas derived from both autoencoders and gen-



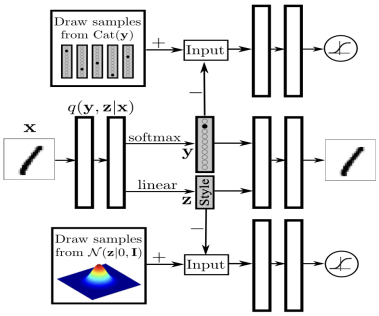
erative adversarial networks (GANs) neural network architectures. By investing high levels of complexity between these domains such as producing quality samples from reduced/symbolic representations while also retaining significant insights regarding input structures during creation, AAE architectures keep decipherable latent space characterizations of their inputs. Within this design framework are competing tools known as generators (represented by the autoencoder portion) and discriminators, helping distinguish original cases from fabricated versions like those seen in GAN models. Through rigorous training efforts involving mimicking genuine real-world examples via artificially generated ones, AAE's incentive and identify stronger feature co-dependencies essential in grasping meaningful interpretations of datasets unlike that seen with classical more basic forms like an Autoencoder produce.

Adversarial autoencoders have various applications:

1) Anomaly Detection*
2) Image Generation
3) Provide a framework for unsupervised learning (the model can learn from unlabelled data without relying on explicit labels or annotations)
4) Data Synthesis
5) Disentangled Representation Learning

Semi-Supervised Adversarial Autoencoder

In semi-supervised AAE, "There are two separate adversarial networks that regularize the hidden representation of the autoencoder. The first adversarial network imposes a Categorical distribution on the label representation. This adversarial network ensures that the latent class variable y does not carry any style information and that the aggregated posterior distribution of y matches the Categorical distribution. The second adversarial network imposes any kind of distribution on the style representation, which ensures the latent variable z is a continuous distribution variable.".



Important note: In this experiment, only the first adversarial network will be used to have learning to construct more repetitive samples than scare occurred samples, which may be temporal anomalies.

III. Materials & Methods

In this section, the implementation details of the examined models and found solutions against faced issues during implementation will be underlined. The latent layer size is determined as 10 for all the models to demonstrate difference clearly.

A. Method

Our expectation from an autoencoder model is able to reconstruct given time-series. Since the model is fed with only "normal" data, it must reproduce the "normal" data by lower difference while reconstructing "slow" data by higher difference. Then, this natural gap can be used for anomaly detection. Before the implementation details of models, having insight into data would be better for good understanding.

Data

Data Normal: The data that collected during production with normal velocity.

Data Slow: The data that collected during production with slow velocity.

Both normal and slow data has 86 features that from action to sensor values. Permanent anomalies are the anomalies effects production velocity (just in data slow). Temporal anomalies are the anomalies by reading abnormal values from sensor (in both data). However, categorical features which has less than 31 different values is extracted, since those may be misleading in reconstruction based TAD. Therefore, there are 79 features in training and test sets. Finally, readings from sensors sampled as a time window which includes 47 previous readings. (Window size: 48)

B. Autoencoder Model

The reference paper [4] suggests a single-layer LSTM autoencoder to adopt as a model. However, considering the complexity of our data, a more complex model is a better option for our case. Therefore, a three-layer LSTM autoencoder has been adopted as a base model. Our base model has estimated 470,000 trainable parameters, which is more than the minimum number (estimated 300,000) of needed parameters to be able to reconstruct our time series.

During development, exploding gradients problem is faced, and the loss was going to infinity during training. Therefore, a standard scaler has been applied to data before processing. Moreover, batch normalization layers are added to the model. Facing the exploding gradients problem must not be strange, since ReLU activation function is used.

Training Details

Trained and untrained base models exhibit the importance of evaluation method selection. Both models' trainable parameters are initialized by Xavier initialization. The standard

training function of TensorFlow is used to train the trained model in the evaluation method selection. And the model trained for 50 epochs with learning rate 10^{-4} and batch size 128.

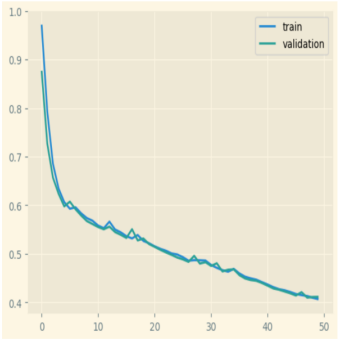


Fig. 2. MSE loss during training

In the figure 2, the convergence does not seem saturated. Therefore, further or other trainings were tried. However, a better model could not be trained.

C. Adversarial Autoencoder Model

An adversarial autoencoder with the same autoencoder baseline is adopted to exhibit and investigate differences between the standard training method and adversarial learning.

The discriminator of the adversarial autoencoder is constructed with four convolutional layers with 2,900 trainable parameters.

Training Details

Since the tasks of discriminator and generator are relatively easier than AE at the beginning, they could start to fool themselves, which can influence AE negatively. Therefore, the weighted training technique is adopted to prevent the problem and provide enough time for the AE model's training. Additionally, the discriminator is learning faster than the generator, which affected the AE negatively again. Because of that, a lower learning rate is selected for the discriminator.

The AAE model is trained for 100 epochs with a learning rate 10^{-4} for the AE and generator 10^{-5} for the discriminator. The batch size is 1024. The generator and discriminator trained with weight 0.2 and AE with 1.

In most cases, discriminators' loss must not decrease during training. However, since the accuracy of the discriminator is kept around 50%, both networks look like they are learning. They can even learn more because losses of discriminator and generator may continue to decrease.

IV. Results

In this section, the results will be presented in two topics. Firstly, the results of trained and untrained baseline models will demonstrate the effect of point adjustment (PA) by comparing F1-scores in different K values and F1-scores without PA. Secondly, the results and differences observed during the migration from traditional training methods to adversarial learning will be demonstrated.

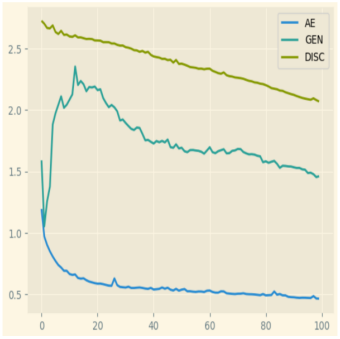


Fig. 3. MSE loss for AE, cross-entropy loss for generator and discriminator during training

A. Evaluation Method

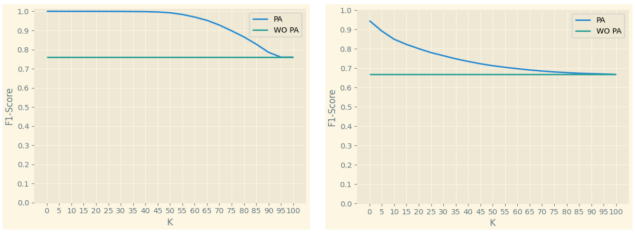


Fig. 4. Best F1-scores for different K.

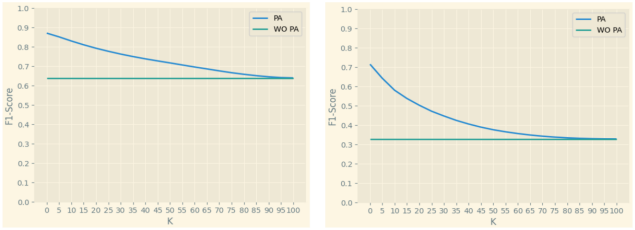


Fig. 5. Mean F1-scores for different K.

(Left: trained model, Right: untrained model)

As we can see on the both graph, PA with lower K values overestimates the model performance, obviously.

Especially in the best F1-scores graphs 4, two things should be focused. Primarily, best F1-score with K equals 1, could reach almost full score. However, mean F1-score is not that high with same K value, and it gives more insight about model performances. Even if mean F1-score overestimates the performance.

Another thing that must be focussed on is the best F1-score without PA for the untrained model can reach very similar results to the trained model. Even F1-score without PA can overestimate the performance on **our task**.

Furthermore, for the graph 4, it must be mention that F1-score of untrained model dramatically decrease by K values. However, trained models resist to decrease in K values. This would be good insight about the performance of models.

B. Adversarial Learning

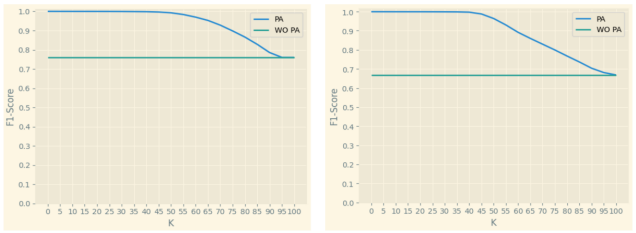


Fig. 6. Best F1-scores for different K.

Fig. 7. Mean F1-scores for different K.

(Left: AE model, Right: AAE model)

Unfortunately, AAE model could not perform better than AE model. As it can be seen in the graphs 6 and 7, F1-scores are slightly lower for AAE model.

V. Discussion & Further Improvements

In this section, issues faced during the project development will be exhibited and discussed. Then, possible solutions to these issues will be shown.

A. Temporal Anomalies

Autoencoder models often become able to well reconstruct also the anomalies in the data. This phenomenon is more evident when there are anomalies in the training set. In particular when these anomalies are labelled, a setting called semi-supervised, the best way to train autoencoders is to ignore anomalies and minimize the reconstruction error on normal data. And model AE-SAD offers solution for this issue according to it. [3]

As we know, the normal data includes many temporal anomalies (abnormal readings). Therefore, the performance of the model will have been increased, if temporal anomalies could be avoided in training. Because of that, the loss function will be changed as following, to avoid or learn less from temporal anomalies or learn how to not construct temporal anomalies.

$$\mathcal{L}(\mathbf{x}) = ||\mathbf{x} - \hat{\mathbf{x}}||_2^2 \quad \Rightarrow \quad \mathcal{L}_F(\mathbf{x}) = (1-y)\cdot||\mathbf{x}-\hat{\mathbf{x}}||^2 + \lambda \cdot y \cdot ||F(\mathbf{x}) - \hat{\mathbf{x}}||^2$$

Fig. 8. Left: Mean Squared Error loss, Right: suggested loss function [3]

If λ equals 0, the temporal anomalies are avoided during training. If λ is between 0 and 1, and function $F(x) = x$, the model learns less from temporal anomalies. And lastly, if λ is between 0 and 1, and function $F(x) = 1 - x$ or $-x$, the model learns how to not construct temporal anomalies.

However, even if loss function has changed, a preprocessing step is needed to determine temporal anomalies. Therefore, most radical 1% of readings will be labelled as temporal anomaly for each feature. And if any of the feature value is in the 1%, the sample will be labelled as anomaly.

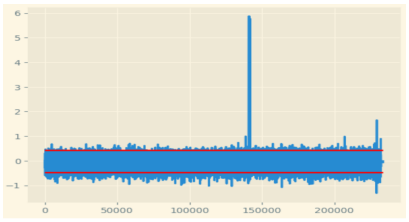


Fig. 9. The samples outside red lines are labelled as temporal anomaly. Representation stands for one feature.

At the end, estimated 5% of the samples are labelled as temporal anomaly.

Training Details

The AE-SAD model is trained for 102 epochs, with a learning rate 10^{-4}. The batch size is 256. λ of loss function is 0.1 and $F(x) = x$.

Fig. 10. Loss during training.

As we can see in graph 10, the training phase looks successful since test loss is kept similar or increasing and other losses decrease.

Results

Fig. 11. Best F1-scores for different K.

Fig. 12. Mean F1-scores for different K.

(Left: AE model, Right: AAE model)

As it can be observable in both graphs 11, 12, AE-SAD model could perform better than the baseline model. Which means, it can be said that abnormal readings must be taken care of in the training phase.

B. Mean vs Best F1-Score for Evaluation

An evaluation method should show the difference of performances clearly. However, best F1-score with point adjustment (PA) would be bad choice for clarity of difference in many K.

As it is obvious in 11, for many K values, especially below 50, it is hard to make an evaluation with PA between models.

VI. Conclusions & future works

This project proved that even randomly initialized models can be looked like a well-performed model, especially in best f1-score. PA can cause an overestimation of models. Furthermore, the selection of the K value of the PA%K technique is crucial to maintaining clarity of evaluation. Even PA%K with low K values may cause the untrained model to look like a well-performed model.

About AAE, hyperparameter selection and keeping the balance of models during the training are the most crucial factors that affect the model's performance. Even though our AAE model could not converge well with the baseline model, it could perform very similarly to the baseline. With better training, the AAE model may perform better than the baseline model.

The task of anomaly detection can be diverse, such as detecting temporal abnormalities or permanent problems. Our mission was to detect permanent issues during production. And the effect of the temporal anomalies was exhibited for our task. The quality of training data and the temporal anomalies should be considered during the training phase for final performance on the test set. This project proved that learning ten times less from temporal anomalies positively affected performance significantly.

Finally, as a discussion, the best F1-score may affect the clarity of evaluation negatively, especially for well-performed models. The mean F1-score may be an excellent alternative to increase the transparency of differences between models.

References

[1] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," *Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook*, pp. 353–374, 2023.

[2] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[3] F. Angiulli, F. Fassetti, and L. Ferragina, "Reconstruction error-based anomaly detection with few outlying examples," *arXiv preprint arXiv:2305.10464*, 2023.

[4] S. Kim, K. Choi, H.-S. Choi, B. Lee, and S. Yoon, "Towards a rigorous evaluation of time-series anomaly detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, 2022, pp. 7194–7201.