

# Evaluating Deep Learning Models for Anomaly Detection in an Industrial Transporting System

Kyrylo Kadomskyi

*Taras Shevchenko National University of Kyiv, 64/13, Volodymyrska st., Kyiv, 01601, Ukraine*

## Abstract

Cyber-Physical Production Systems (CPPS) require robust techniques for detecting anomalies and root causes in the system. The model-based diagnosis is a commonly used approach in which a dynamic process model captures spatio-temporal features of the system's behavior. Because of the infeasibility of precise mathematical or expert modeling, algorithms have been developed for learning such models from system observations. These algorithms are characterized by high domain-specialization and yield relatively poor performance in other use cases.

In this paper the CPPS data is used, on which existing models have proven ineffective. The perspective of applying deep learning approach to constructing a process model in such systems is investigated. The main idea is to go from models with fixed structure to more universal techniques for learning optimal structure from dynamic observations. The challenges of evaluating dynamic system models of this class are identified, and evaluation criteria are proposed for representative comparison and benchmarking of the models. It is shown that deep learning models provide increase in anomaly detection score but require additional verification of model robustness.

## Keywords <sup>1</sup>

Anomaly detection, autoencoder, model evaluation, cyber-physical production systems, industrial IoT

## 1. Motivation

Industrial AI is an emergent research field that is actively revolutionizing production plants. Increasing product variety, product complexity and pressure for efficiency lead to systems that contain a growing set of sensors to facilitate automation [1]. In this context diagnosis of complex production processes has gained new attention due to research agendas such as Cyberphysical Production Systems (CPPS) [2, 3]: the initiative of Industrial Internet of Things (IIoT) and Industrie 4.0. In these agendas the most important goals of self-diagnosis are identification of anomalous system behavior, suboptimal energy consumption, or wear in CPPS [4, 5].

The most accepted method is model based diagnosis [4] where the features of normal and anomalous system's behavior are captured by the process model. Modern CPPS are adaptable and changeable, which makes both precise mathematical modelling and manual expert modelling costly and ineffective [6]. Thus, to build the model the process features must be extracted from sensory measurements. As the process often is highly dynamic and variable, the most informative features are spatio-temporal and include sequential events, timing and duration of specific process stages, or the boundaries on observed values specific to each given stage.

To achieve this, novel dynamic modelling techniques are being developed [3, 4, 7, 8] and are currently replacing traditional methods, such as Statistical Process Control (SPC) and Bayesian inference with time dependency. While showing good results in certain applications, this models yield

*IT&I-2020 Information Technology and Interactions, December 02–03, 2020, KNU Taras Shevchenko, Kyiv, Ukraine*

EMAIL: [cyril.kadomskyi@gmail.com](mailto:cyril.kadomskyi@gmail.com) (K. Kadomskyi)

ORCID: 0000-0002-6163-3704



© 2020 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

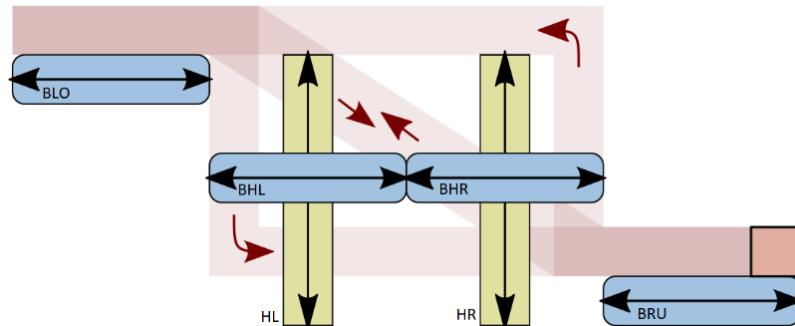
relatively poor performance in other similar use cases [9, 7, 8]. The hypothesis is that this effect is due to limited nature and fixed structure of spatio-temporal features learned by the model, which are imposed by the structure of the model itself. Then the informativeness of learned features will vary in different physical systems, which can explain the observed effect.

In this study Deep Learning (DL) models, such as autoencoders [10], are applied to remove the mentioned limitation by automatically selecting the most relevant features and structure to represent the data. Evaluating these models on the dataset that has proven challenging for applying novel dynamic models is conducted aiming for accurate benchmarking of the two approaches. This in turn provides the possibility to assess the limits of model-based anomaly detection in given class if CPPS.

As results of traditional evaluation techniques in CPPS applications may not be representative [9], the challenges of evaluating dynamic system models in CPPS are identified by analyzing data collected from DL models, and robustness criteria are proposed to increase evaluation representativeness.

## 2. The System and the data

Currently several projects are aimed at utilizing new technical possibilities to meet the challenges of Industrial IoT and Industrie 4.0. Under the European Union's Horizon 2020 research project IMPROVE [11] a number of experiments in industrial systems were made, and environments were designed specifically to test novel methods for self-diagnosis (including monitoring, anomaly detection) and self-optimization [12]. The High Rack Storage System or HRSS is a demonstrator system built in SmartFactoryOWL in Lemgo, Germany. The system transports pallets between its different shelves, as shown in **Figure 1**.



**Figure 1:** A schematic representation of the system. The system consists of two stationary ('BLO', 'BRU') and two movable ('BHL', 'BHR') conveyor belts, as well as vertical rails ('HL', 'HR'). The arrows show three of possible transporting paths.

Source: <https://www.kaggle.com/inIT-OWL/high-storage-system-data-for-energy-optimization>.

Measurements of position, power and voltage are made at each of the system's drives during full transporting cycles. Anomalies in this system include shortening of cycles, pauses, abnormal timing, duration, or sequence of different process stages, as well as increase or decrease in one or multiple signals at certain stages. The task is to detect HRSS anomalies and to localize them with time-step precision by constructing the model of normal system behavior in an unsupervised manner.

A time series dataset [13] was collected in this system under IMPROVE project and is being actively used to test novel approaches to anomaly detection [9, 14]. The data contains 18 real-valued signals sampled 15–20 times per second. It includes time series of 106 normal cycles (25,907 observations) and 111 cycles containing labelled anomalies (23,645 observations). The dataset is unbalanced with 76.0% of negative examples. Statistical distributions of the classes (i.e. normal and anomalous measurements) are not distinguishable in feature space, which excludes direct applying of traditional Machine Learning (ML) methods for anomaly detection (e.g. linear models, decision trees, SVM, etc.). At the same time PCA analysis shows that 10 main principal components cover 98.1% of data variation, so linear dimensionality reduction techniques can be useful. Data quality issues that

may affect model performance include high noisiness, strong outliers, and difference in feature ranges by several orders of magnitude.

### 3. Background research

As the statistical separation of classes is not possible in this task, constructing a model from process measurements involves learning spatio-temporal patterns and events, which are typically characterized by timing and duration of different process stages.

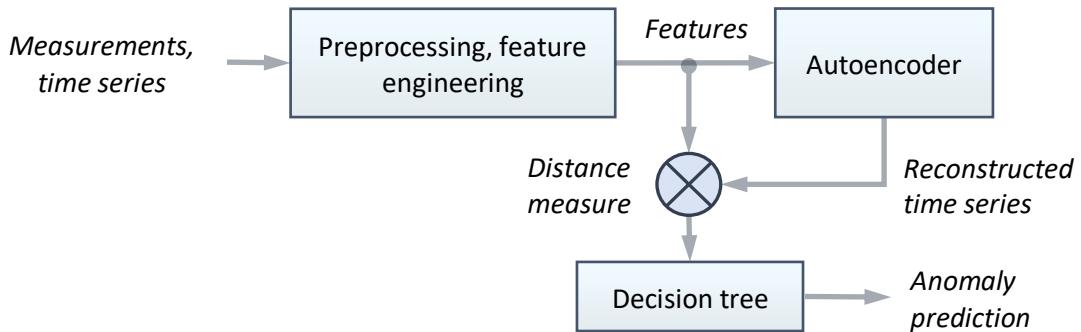
To address this goal the use of dynamic process models such as Hybrid Timed Automata (HTA) has been proposed [9]. To apply a discrete state HTA model to continuous process measurements the unsupervised data preprocessing with self-organizing maps (SOM) and watershed transformations were utilized. This method detects anomalies with timestep precision. Yet, having proven effective in other CPPS applications [7, 8], it yields low performance on HRSS data with 30.76% F1 score and 26.7% recall (1516 true positives).

In another study the Deep Learning architectures were applied to the same data [14]: Siamese LSTM model was used for binary classification of full process cycles into ‘normal’ and ‘anomalous’ classes. Targeting minimal false-positive score this model yields 25.6% F1 measure, 88.2% precision, and 15.0% recall, while being unable to localize anomalies within a cycle.

In both studies anomaly detection rates are low, comparing to other CPPS applications, thus learning a model from the process measurements in HRSS plant remains a challenging task. To address this task, features of HRSS system must be identified that explain observed drop in efficiency. As the results of the two studies are not directly comparable, the perspective of applying DL models in this class of CPSS also remains an open question. Answering it requires strict evaluation of DL models, as well as assessment of the effect of architectural variations. As the representativeness of evaluation results remains unknown [9], additional measures must be developed to assess model robustness.

### 4. The method

In this study a set of autoencoder architectures are applied to the task of anomaly detection [10] in a setup shown in Figure 2. The DL model, i.e. autoencoder, is trained in unsupervised manner to reconstruct normal time series targeting minimal reconstruction loss. Then the trained model is used to reconstruct unseen time series with anomalies, where the reconstruction error is expected to peak at anomalous intervals. To evaluate the model, the distributions of reconstruction error in normal and anomalous intervals are analyzed for being statistically distinguishable. Finally, from the error distributions a decision-rule classifier for anomaly detection is built in a supervised mode.



**Figure 2:** Solution architecture

This method detects anomalies with time step precision, and most of evaluated models can be applied in real time.

5 modifications of LSTM autoencoder and 3 modifications of ConvNet autoencoder were modelled and evaluated in a setup allowing for direct benchmarking against background research. For results to be representative, models' robustness must be assessed. From the analysis of evaluation results, two challenges were identified that must be met to achieve model robustness and the representativeness of evaluation.

1. One distinct feature of HRSS plant is low process variation in normal conditions with 12.6% mean absolute deviation from the averaged process cycle. Under such conditions, an autoencoder model can reach local minima of reconstruction error without reconstructing individual features of distinct cycles (i.e. different process runs). In this case model's output is close to the average training cycle with reconstruction loss close to  $v_{normal}$ . Such model performs well on HRSS data where process variation is low, but it will not be useful in most CPPS applications where process variation is higher.
2. The presence of anomalies may affect model's performance in reconstructing neighboring normal intervals. This is expected behavior in models with internal time-dependency, which are used in this study. In this case model's robustness is limited by the type and the length of anomalies, which typically are not known at training time.

#### 4.1. Robustness criteria

To address the mentioned challenges two robustness criteria are proposed for representative model evaluation.

$RC_1$ . *Reconstructed variation rate* is calculated in unsupervised mode using the training set of normal process cycles, by comparing step-wise standard deviation of reconstructed signal  $s_{reconstructed}^{normal}$  to standard deviation of the model input  $s_{input}^{normal}$ :

$$RC_1 = \frac{\sigma(s_{reconstructed}^{normal})}{\sigma(s_{input}^{normal})} \quad (1)$$

$RC_2$ . *Reconstruction sensitivity to anomalies* is assessed in supervised mode on the set of anomalous cycles (i.e. evaluation set) as the correlation between error of reconstructing normal intervals and the strength of anomalies in the same process cycle or time window:

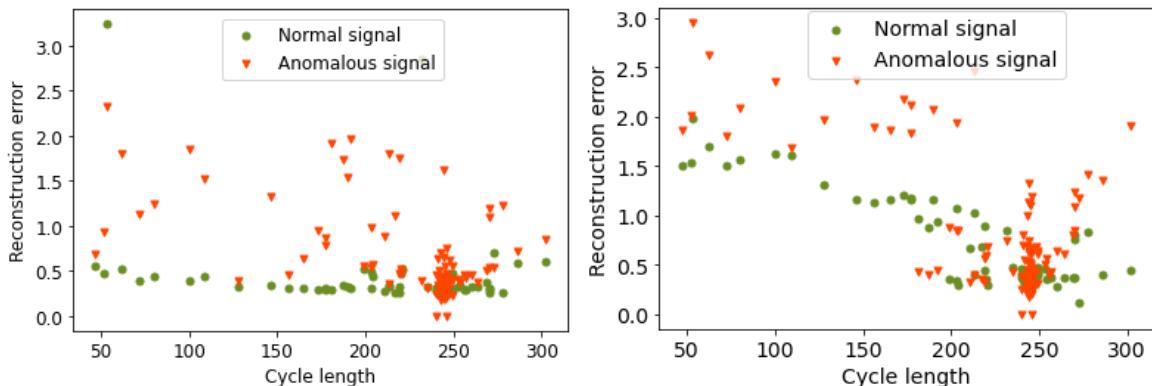
$$RC_2 = \text{corr}(M|s_{reconstructed}^{normal} - s_{input}^{normal}|, \text{anomaly\_estimate}), \quad (2)$$

where *anomaly\_estimate* is domain specific and includes type, time length and strength of the anomaly.

In HRSS plant two distinct types of anomalies are present.

- *Type 1*: amplitude deviations from normal signal
- *Type 2*: deviations in timing, duration, or sequence of process stages

In practice anomalous cycle duration and long-term type-2 anomalies have noticeable effect on  $RC_2$ , as shown in Figure 3.



**Figure 3:** Evaluation results for two different autoencoder models which demonstrate low (left) and high (right) values of  $RC_2$ , respectively. Results were obtained from LSTM 2 and ConvNet 2 models.

## 4.2. Evaluation techniques

To evaluate the DL models, HRSS dataset is split into three parts.

- Training set contains randomly selected 2/3 of normal cycles and is used to train the autoencoder.
- Test set contains remaining normal cycles and is used to validate autoencoder and test it for overtraining.
- Evaluation set contains all cycles with anomalies and is used to assess anomaly detection performance and to justify the selection of decision threshold.

### 4.2.1. Choice of performance measures

The architecture consists of two parts: the autoencoder which is used to reconstruct input time sequence, and the classifier used for anomaly detection. So, two performance indicators are required.

The performance of signal reconstruction was measured with MAE loss function, which is more outlier-resistant and more suitable for high-dimensional data comparing to MSE.

Anomaly detection performance was measured with F1 score and confusion matrix. The F1 score has the advantage of accounting for both false positives and false negatives. Comparing to accuracy and correlation-based measures, which also account for true negatives, F1 score better suits an unbalanced dataset. Also, F1 score with confusion matrix enable direct comparison with the background research.

### 4.2.2. Selecting decision threshold

In anomaly detector the threshold must be set for the signal reconstruction error. Let  $L_n$  be the distribution of signal reconstruction loss obtained on the training set (i.e. in normal cycles); let  $L_n^v$  and  $L_a^v$  be the distributions of loss obtained on validation data: in normal intervals and in anomalies respectively. Then optimal value for the classification threshold can be assessed from  $L_n$ ,  $L_n^v$  and  $L_a^v$  in two ways:

- Unsupervised:  $T = E(L_n) + 2\sigma(L_n)$ .
- Supervised:  $T = \text{argmax } S(L_n^v, L_a^v, T)$ , where  $S$  is a performance measure for anomaly detection.

Experiments on HRSS data show that the optimal threshold value for different architectural modifications varies in a broad range. While the first assessment can be far from optimal, the second assessment may not be possible in most applications where labelled anomalous data is not available.

### 4.2.3. Evaluation steps

Evaluation steps include:

1. calculating performance measures.
2. assessing the statistical separation between autoencoder response to normal and anomalous signals ( $L_n$ ,  $L_n^v$  and  $L_a^v$ ).
3. assessing robustness criteria RC1 and RC2.
4. selecting the optimal model by maximal performance, among models that have passed robustness tests.

## 5. Models

The DL models being tested are divided in two groups by DL architecture type: LSTM and Convolutional. In each group the first model is a traditional architecture used for anomaly detection. Other models are built to assess the effect of architectural modifications on model performance.

The choice of the model's hyper-parameters affects both experimental performance and robustness. Hyper-parameters include the number, types and sizes of layers, compression rate of autoencoder, the use of dropouts, as well as internal layer parameters (e.g. kernel size, activation function). As no computationally effective techniques exist for finding the optimal architecture construction through hyper-parameter choices, this task remains tedious and highly intuition driven [15, 16]. In this study a grid search approach was applied for each model type, obtaining the models shown in Table 1.

**Table 1**

Architecture modifications in the autoencoder. All models have decoder layers symmetrical to encoder. Models were compiled with 'tanh' function for layer activation, and 'sigmoid' function for recurrent activation. In the "CR" column the compression rate of the encoder is given.

Model	Description	Model hyper-parameters		
		Layers in encoder	CR	Transformations at the bottleneck
LSTM 1	Classic LSTM architecture [17, 18]	2 LSTM layers (filters: 30, 60)	60	Final LSTM output repeated for each timestep
LSTM 2	Model without time compressing	2 LSTM layers (filters: 60, 4)	3	No transformation
LSTM 3	Model with time pooling	3 LSTM layers (filters: 25, 25, 6), 2 pooling layers (factor 3, 2)	12	No transformation
LSTM 4	Model with time pooling and convolutional layers	3 LSTM layers (filters: 25, 25, 6), 3 pooling layers (factor 3, 3)	24	1 convolutional layer (6 filters, kernel size 3)
LSTM 5	Model with time pooling and locally connected layers	3 LSTM layers (filters: 25, 25, 6), 2 pooling layers (factor: 3, 2)	12	1 locally connected layer (6 filters, kernel size 5)
LSTM 6	Model with time pooling and dense layers	3 LSTM layers (filters: 25, 25, 6), 3 pooling layers (factor: 3, 2, 2), 1 convolutional layer (6 filters, kernel size 3)	48	Flattening, dense layer (75 filters), dense layer (150 filters)
ConvNet 1	Classic convolutional model	3 convolutional layers (filters: 32, 16, 6; kernel size 5), 3 pooling layers (factor 2)	16	1 convolutional layer (6 filters, kernel size 5)
ConvNet 2	Extended convolutional model	3 convolutional layers (filters: 64, 32, 6; kernel size: 5, 10, 20), 3 pooling layers (factor 3, 2, 2)	24	1 convolutional layer (6 filters, kernel size 5)
ConvNet 3	Convolutional model with dense layers	3 convolutional layers (filters: 64, 32, 6; kernel size: 5, 10, 20), 3 pooling layers (factor 3, 2, 2)	48	Flattening, dense layer (75 filters), dense layer (150 filters)

## 6. Experimental setup

The models were implemented using Keras with Tensorow backend. Training was performed using 'Adam' optimizer and MAE loss function with learning rate of  $\alpha = 0.005$ ,  $\beta_1 = 0.9$ ,  $\beta_2 =$

0.999, and fuzzy factor  $\varepsilon = 10^{-7}$  [19]. The time series of complete process cycles, padded to constant length of 300 timesteps, were used as both input and target. Training was run with 130 epochs for LSTM models and 300 epochs for ConvNet models, in mini-batch mode with batch size 32. To rule out the effect of batch-averaging on robustness criteria  $RC_1$ , training was repeated in stochastic mode (batch size 1). In this setup the number of epochs was reduced by the factor of 5, as epochs are more time-consuming in this mode, but epoch-to-epoch convergence is faster. As no significant influence of the batch size on evaluation criteria was observed in experiments, only results obtained in mini-batch mode are presented. As the reconstruction loss fluctuates between training epochs, averaging across last 10 epochs was used for reliable performance estimate.

Data pre-processing included the following steps:

- Introducing velocity features, calculated with second order accurate central differences.
- Dimensionality reduction from 24 to 12 components with PCA, which preserves 98.2% of data variance.
- Normalization and scaling to the range (0,1), which unifies value ranges of features.
- Time smoothing with gaussian kernel of width 15 and standard deviation 3.
- Unifying time series length by padding.

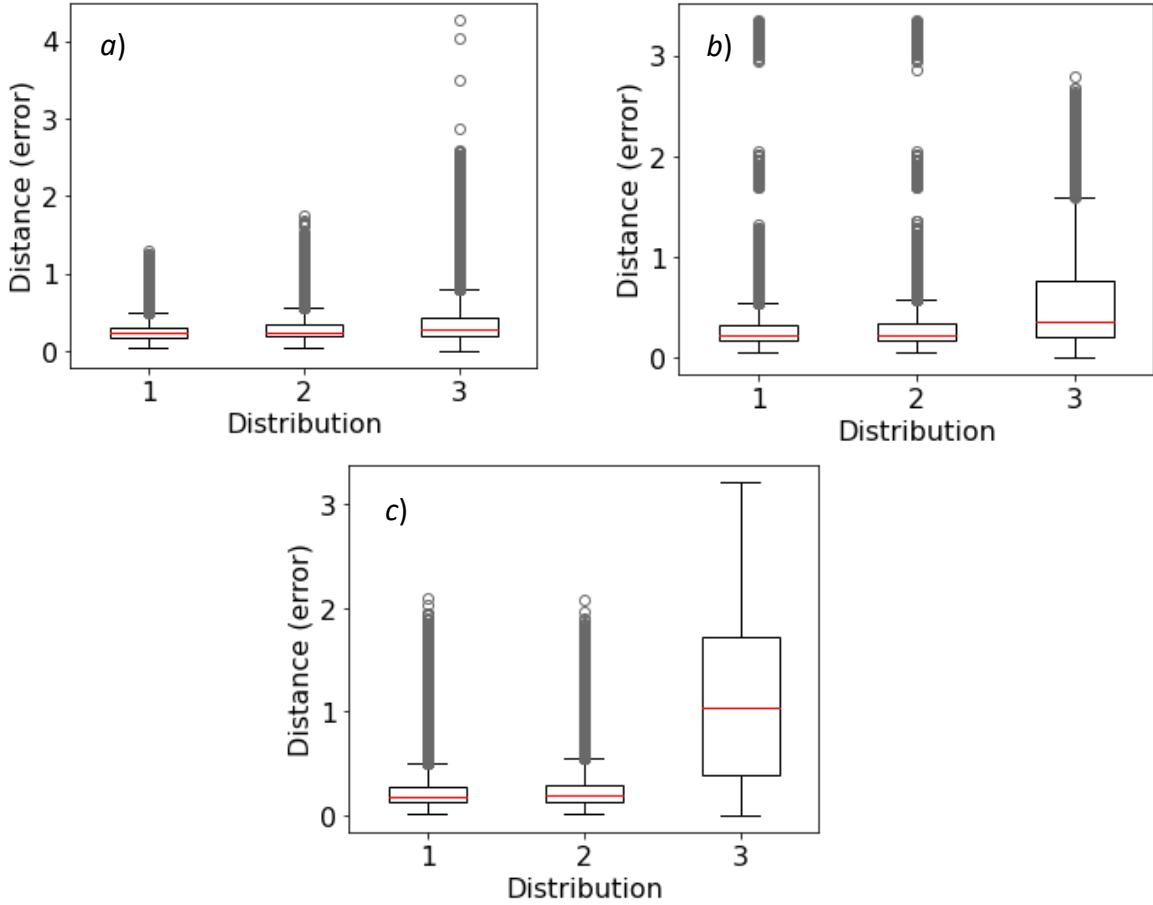
## 7. Results

Reconstruction rates of all models fall into a narrow range, as shown in Table 2, with exception of classic LSTM autoencoder (LSTM 1), which proved unable to accurately reconstruct the process. Thus, reconstruction loss measure cannot be used to assess the efficiency of autoencoder model in CPPS anomaly detection task. Instead, statistical analysis of the loss distributions must be applied.

**Table 2**

Model evaluation results. Column 1 shows anomaly detection score obtained in the case of supervised optimal threshold selection using labelled anomalies, column 2 gives the score obtained with unsupervised threshold estimate. Separate assessments for type 1 and type 2 anomalies are obtained in the case of optimal threshold selection. Reconstruction score is assessed relative to the amplitude variation in normal signal. In some models  $RC_1$  varies significantly during the process cycle, starting with 0.1-7.5%. For such models, the mean value is given, marked with asterisk “\*”, while model’s robustness drops significantly at the beginning of each cycle.

Model	Performance					Robustness	
	Anomaly detection, F1 score, %					Reconstruction, MAE, %	$RC_1$
	Overall estimate		Type 1 anomalies	Type 2 anomalies			
Target value	1	2				100	none or low
LSTM 1	52.6	38.3	25.6	81.8	$42.5 \pm 1.5$	0.0012	none
LSTM 2	46.3	36.3	34.2	59.0	$12.0 \pm 0.53$	51.2	low
LSTM 3	69.2	67.9	37.7	85.9	$13.4 \pm 0.92$	27.14*	low
LSTM 4	62.3	59.2	34.5	78.5	$14.7 \pm 0.83$	36.5*	low
LSTM 5	75.4	75.3	39.3	92.9	$12.63 \pm 0.27$	0.014	none
LSTM 6	75.7	75.2	40.2	93.2	$12.8 \pm 0.41$	0.0064	none
ConvNet 1	38.8	25.8	30.8	55.1	$10.9 \pm 0.81$	0.652	medium
ConvNet 2	48.3	42.7	33.6	62.8	$14.9 \pm 0.75$	26.5	high
ConvNet 3	69.9	68.0	44.3	85.2	$10.4 \pm 0.41$	42.6	high

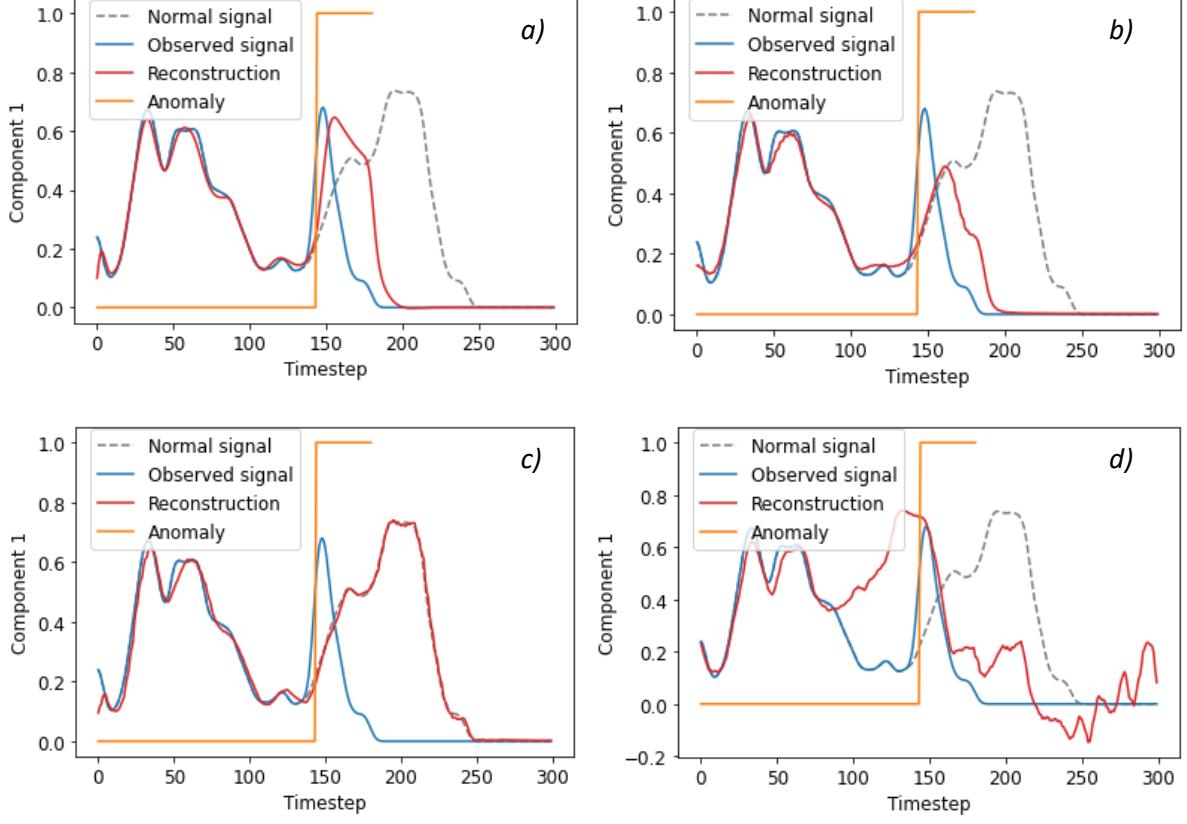


**Figure 4:** Error distributions  $L_n$ ,  $L_n^v$  and  $L_a^v$  for three models with close performance estimates in normal signal reconstruction: a) ConvNet 1 model: the classes are not separated; b) LSTM 2 model: the classes are overlapping; c) LSTM 4: the classes are well separated

Figure 4 shows the statistical distributions of autoencoder's responses to normal and anomalous data. The performance of anomaly detection is defined by the quality of class separation, but it is also highly dependent on the method of selecting decision threshold. Empirically defined optimal threshold for the tested models varies in wide range from 0.20 to 1.12 (Table 2, column “1”). Optimal threshold selection is only possible in a supervised mode with the use of labelled anomalies, while in most practical applications unsupervised threshold selection must be applied (Table 2, column “2”).

While some models achieve high scores in detecting type 2 anomalies, they proved not being sensitive to type 1 anomalies, e. g. 20% amplitude deviations from the normal signal. Then high overall detection score is explained by large relative abundance of type 2 anomalies in HRSS data. For evaluation results to be representative, the detection rates in different anomaly types must be assessed separately.

Figure 5 shows results of reconstructing a single process cycle containing a type 2 anomaly. Graphs *a* and *b* are obtained from robust models (having high  $RC_1$  value and low  $RC_2$  value), graph *c* demonstrates an extreme case of zero  $RC_1$  value, and graph *d* is the case of high  $RC_2$  value. In the cases *a*, and *b* the model captures features of the individual observed cycle, so it is expected to show comparable performance in other typical CPSS applications. In the case *c* the model output follows the averaged train data, regardless of the observed process features. High anomaly detection score in this case is not representative and is only observed due to low variance in training cycles in HRSS. In the case *d* presence of type 2 anomaly strongly affects reconstruction of preceding normal interval, making the task of statistically separating them in time (as well as the resulting performance estimate) inadequate.



**Figure 5:** Reconstruction of a signal containing type 2 anomaly: a) using LSTM 2 model; b) using LSTM 3 model; c) using LSTM 5 model; d) using ConvNet 2 model

Evaluation results indicate that increasing complexity of DL models (top down in Table 2) leads to higher performance measure. However, this is not the case with robustness. Deep LSTM models with heterogeneous layers (LSTM 5 and LSTM 6) tend to average out all variation in the signal (i.e., have low  $RC_1$ ), while deeper convolutional networks lose ability to reconstruct normal signal in presence of type 2 anomalies (i.e., have high  $RC_2$ ). It may be concluded that traditional performance metrics for model evaluation are misleading in case of HRSS, favoring models with low robustness according to criteria  $RC_1$  and  $RC_2$ .

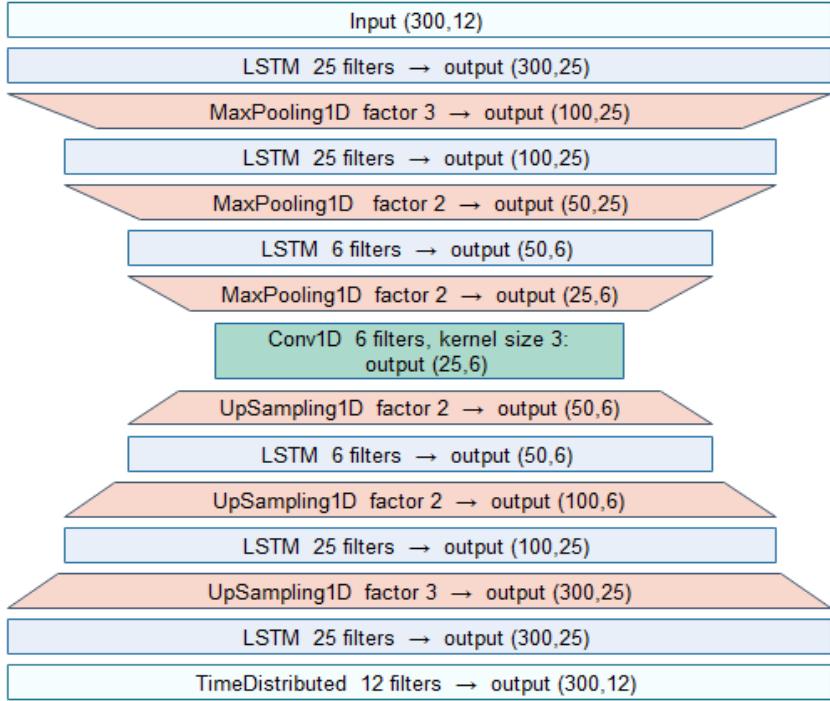
Considering both performance measure and proposed robustness criteria, the LSTM 4 model is selected as the best choice for HRSS data. Model's architecture is demonstrated in Figure 6.

Comparing to traditional LSTM autoencoder architectures [17, 18], this model introduces two distinct architectural features. First, input time-series are not flattened into a vector, and thus the model has lower compression rate. Experimental evidence (Table 2) suggests that preserving time dimension in encoder generally leads to better performance in anomaly detection task. Second, an additional convolution layer is added at model's bottleneck to capture long-term features in input time-series.

The obtained LSTM 4 model provides  $62.3 \pm 2.1\%$  overall anomaly detection rate (F1 score) and 59.1% recall with 3350 true positives, as shown in Table 3. Comparing to the baseline efficiency [9], an increase by 102% in anomaly detection score and an increase by 121% in recall are achieved.

**Table 3**  
Confusion matrix obtained with the selected model.

Labelled	Predicted negative	Predicted positive
Negative	16237	1738
Positive	2320	3350



**Figure 6:** Selected autoencoder architecture

## 8. Conclusions

The problem of the model-based anomaly detection in industrial CPPS was addressed in the Deep Learning paradigm by applying autoencoder architectures. The specific case of HRSS plant was studied, in which construction and evaluation of process models had proven to be a challenging task. The major challenges of applying Deep Learning models were identified as low process variation in the training set, and presence of two distinct types of anomalies, detecting which requires different algorithms or settings.

It was shown that increasing model complexity, both in LSTM and convolution-based models, allow to increase anomaly detection performance but has strong robustness tradeoff. This indicates that model evaluation in systems of this class cannot rely completely on performance metrics. For evaluation results to be representative, detection rates of different anomaly types must be assessed separately, and additional robustness criteria must be considered. Such criteria were proposed based on statistical analysis of both the data and the model output in supervised training context.

In the studied industrial transporting system (HRSS) applying deep learning models and autoencoder techniques allowed for 102% performance gain, F1 score, while preserving model's robustness. Wider assessment of perspectives of CPPS applications requires further experimental research in cases of higher variance in the normal process as well as different types of anomalies.

## 9. Acknowledgements

This research utilizes the data collected at SmartFactoryOWL Lemgo, Germany, under the European Union's Horizon 2020 research project IMPROVE [12]. The data was made publicly available by inIT [13] under a Creative Commons License Attribution-ShareAlike 4.0 International (CC BY-NC-SA 4.0).

## 10. References

- [1] Factories of the future: multi-annual roadmap for the contractual PPP under HORIZON 2020, Publications Office of the European Union, Luxembourg, 2013.

- [2] E. A. Lee, Cyber physical systems: design challenges. In: Proceedings of the 11th IEEE international symposium on Object Oriented Real-Time Distributed Computing (ISORC), Orlando, FL, 2008, pp. 363–369. doi: 10.1109/ISORC.2008.25.
- [3] O. Niggemann, C. Frey, Data-driven anomaly detection in cyber-physical production systems, AT – Automatisierungstechnik, 2015, vol. 63, issue 10. doi: 10.1515/auto-2015-0060.
- [4] L. Christiansen, A. Fay, B. Opgenoorth, J. Neidig, Improved diagnosis by combining structural and process knowledge, in: Proceedings of the 16th IEEE conference on Emerging Technologies Factory Automation, ETFA, Toulouse, France, 2011, pp. 1–8. doi: 10.1109/ETFA.2011.6059056.
- [5] S. Windman, S. Jiao, O. Niggemann, H. Borcherding, A stochastic method for the detection of anomalous energy consumption in hybrid industrial systems, in: Proceedings of the 11th international IEEE conference on Industrial Informatics, INDIN, Bochum, Germany, 2013. doi: 10.1109/INDIN.2013.6622881.
- [6] B. Vogel-Heuser, C. Diedrich, A. Fay, S. Jeschke, M. Kowalewski, S. Wollschlaeger, P. Goehner, Challenges for software engineering in automation, Journal of Software Engineering and Applications 7 (2014) 440–451. doi: 10.4236/jsea.2014.75041.
- [7] N. Hranisavljevic, O. Niggemann, A. Maier, A novel anomaly detection algorithm for hybrid production systems based on deep learning and timed automata, in: Proceedings of the 27th international workshop on Principles of Diagnosis, DX-2016, Denver, Colorado, 2016.
- [8] A. von Birgelen, O. Niggemann, Enable learning of hybrid timed automata in absence of discrete events through self-organizing maps, in: O. Niggemann, P. Schüller (eds.), IMPROVE – Innovative modelling approaches for production systems to raise validatable efficiency. Technologien für die intelligente automation (Technologies for intelligent automation), vol. 8, Springer Vieweg, Berlin, Heidelberg, 2008. doi: 10.1007/978-3-662-57805-6\_3.
- [9] A. von Birgelen, O. Niggemann, Using self-organizing maps to learn hybrid timed automata in absence of discrete events, in: Proceedings of the 22nd IEEE international conference on Emerging Technologies and Factory Automation, ETFA, Limassol, Cyprus, 2017, pp. 1–8. doi: 10.1109/ETFA.2017.8247695.
- [10] C. Zhou, R. C. Paffenroth, Anomaly detection with robust deep autoencoders, in: Proceedings of the 23rd ACM SIGKDD international conference on Knowledge Discovery and Data Mining, KDD '17, Halifax NS, Canada, 2017, pp. 665–674. doi: 10.1145/3097983.3098052.
- [11] IMPROVE. Creating the factory of the future with 4.0 solutions, 2016. URL: <http://improve-vfof.eu/>.
- [12] Physical factory / demonstrators IMPROVE, 2016. URL: <http://improve-vfof.eu/background/physical-factory-demonstrators>.
- [13] inIT, High storage system data for energy optimization, 2018. URL: <https://www.kaggle.com/inIT-OWL/high-storage-system-data-for-energy-optimization>.
- [14] M. Cerliani, Predictive maintenance with LSTM siamese network, 2019. URL: <https://towardsdatascience.com/predictive-maintenance-with-lstm-siamese-network-51ee7df29767>.
- [15] S. R. Young, D. C. Rose, T. P. Karnowski, S.-H. Lim, R. M. Patton, Optimizing deep learning hyper-parameters through an evolutionary algorithm, in: Proceedings of the workshop on Machine Learning in High-Performance Computing Environments, MLHPC '15, Austin, Texas, 2015, article no. 4. doi: 10.1145/2834892.2834896.
- [16] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, The journal of machine learning research, 13 (2012), pp. 281–305.
- [17] A. Sagheer, M. Kotb, Unsupervised pre-training of a deep LSTM-based stacked autoencoder for multivariate time series forecasting problems, Scientific Reports 9, 19038 (2019). doi: 10.1038/s41598-019-55320-6.
- [18] A. H. Mirza, S. Cosan, Computer network intrusion detection using sequential LSTM neural networks autoencoders, in: Proceedings of the 26th Signal Processing and Communications Applications Conference, SIU, Izmir, Turkey, 2018, pp. 1–4. doi: 10.1109/SIU.2018.8404689.
- [19] D. P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: Proceedings of the 3rd international conference for Learning Representations, CoRR, San Diego, CA, 2014, abs/1412.6980.