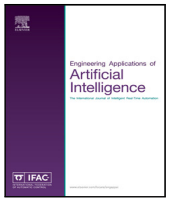
Engineering Applications of Artificial Intelligence 104 (2021) 104354

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



VASP: An autoencoder-based approach for multivariate anomaly detection and robust time series prediction with application in motorsport ☆

Julian von Schleinitz [a,*], Michael Graf [a], Wolfgang Trutschnig [b], Andreas Schröder [b]

[a] BMW AG Motorsport, Germany
[b] University of Salzburg, Austria

ARTICLE INFO

Keywords:
Variational autoencoder
Anomaly detection
Time series prediction
Motorsport
Deep learning
LSTM

ABSTRACT

The aim is to provide a framework for robust time series prediction in the presence of anomalies. The framework is developed based on a data set from motorsport but is not limited to this specific area. In motorsport, the usage of sensors during races is generally restricted. Estimating the outputs of these missing sensors therefore provides an advantage over the competition. Deep learning approaches such as long short-term memory (LSTM) neural networks have proven to be useful for that task, however, their accuracy decreases significantly if anomalies occur in the input signals. To overcome this problem, we propose the variational autoencoder based selective prediction (VASP) framework which combines the tasks of anomaly detection and time series prediction. VASP consists of a variational autoencoder (VAE), an anomaly detector and LSTM predictors. Depending on the anomaly detector, a subset of the inputs may be replaced by the VAE, allowing a more robust prediction. To the best of our knowledge the approach of using a VAE to only selectively replace anomalous input data before prediction has not yet been published. Our contributions are clear implementation guidelines and a comparison to other VAE-based methods and a LSTM approach as baseline. We simulate anomalies with three approaches and show that VASP outperforms other methods by having no trade-off between accuracy and robustness. VASP is as accurate as the baseline for regular data, but for anomalous inputs the error is reduced by 13% to 33% on average and up to 70% in special cases.

1. Introduction

1.1. Background

In motorsport, the main objective of a race engineer is to prepare the car setup in such a way that the driver is able to win the race. Each driver prefers a different car setup to maximize the performance, which is reflected by different driving styles (Wörle et al., 2018, 2019; Schleinitz et al., 2019). In several race series the rules only allow the usage of various sensors during testing but not during race events. In order to optimize the car setup for the individual drivers the engineers need reliable signal estimations to compensate for the missing information.

Fig. 1 depicts the basic process of data analytics in motorsport. The goal is to get information about the vehicle state A, which is an abstract term for all relevant information that an engineer requires about the vehicle. For that purpose a measurement process is conducted leading to the measured data D. However, D is insufficient to describe A fully, for example because of missing sensors. Therefore, the model M is used to transform and complement the measured data. The model contains prior knowledge, e.g. from domain expertise, principles of vehicle dynamics, or machine learning models which were trained on relevant data. Applying the model to D leads to the processed data D* which contains information in a format for inferring the vehicle state A directly from D*.

An example for a signal that is not in the measured data D is the side slip angle of the vehicle, as the required sensor is not allowed during races. However, information about the side slip angle is essential in races as it is an important parameter for understanding and optimizing tire behavior. Therefore, it is estimated by a model M based on prior knowledge with the available data D as input. The processed data D* then includes the side slip angle estimation which the engineers can use to understand the current vehicle state.

Apart from being incomplete the measured data D can also contain anomalies, e.g. because of broken sensors. Due to the hostile environment for sensors in race cars (high temperatures, strong vibrations and hard road impacts) failures occur regularly. Detecting these failures as quickly as possible is vital to avoid making wrong decisions caused by false data. As a consequence, we develop a method which combines

☆ This work was generously supported by BMW AG, Germany.
* Corresponding author.
E-mail address: julian.von-schleinitz@bmw-motorsport.com (J. von Schleinitz).

https://doi.org/10.1016/j.engappai.2021.104354
Received 16 October 2020; Received in revised form 7 April 2021; Accepted 14 June 2021
Available online 23 June 2021

anomaly detection and robust prediction. The scope of this work focuses on the path from D to D* via the model M. The objective is that the obtained processed data set D* is as precise as possible, even in the case of anomalies in measured data D.

1.2. Problem statement

Given a data set of time series we address the following tasks:

- Robust time series prediction by using supervised learning
- Multivariate anomaly detection and correction of the affected time series by using unsupervised learning
- Integration of anomaly detection and robust prediction into a modular framework

Time series should be predicted from the available sensor data to allow the engineers to draw conclusions about the vehicle state. The prediction should be as precise as possible even if the input data contained anomalies. Therefore, an anomaly detection method is used to identify and replace anomalies in the input data. Furthermore, the framework is required to be modular in the sense that it should be possible to add new prediction modules without changing or retraining the entire framework. This leads to the requirement that the predictors need to be trained individually and separately from the anomaly detection module.

1.3. Related work

Time series prediction with deep learning methods, especially long-short term memory neural networks (LSTM), which are a kind of recurrent neural networks (RNN), have scored significant achievements in recent years (Li et al., 2019). Over the past years there has been considerable progress in RNNs (Wielgosz et al., 2018). The LSTM network was introduced by Hochreiter and Schmidhuber (1997) on the basis of the RNN (Williams and Zipser, 1989). LSTMs can detect patterns over the temporal dimension and avoid the vanishing /exploding gradient problem which stands as a difficult issue to be circumvented when training recurrent or very deep neural networks (van Houdt et al., 2020). For time series forecasting, which can be seen as a branch of time series prediction, hybrid LSTM approaches achieved very good results for applications in various fields. Altan et al. (2021) proposed LSTM models for forecasting wind speed in combination with a grey wolf optimizer or forecasting the price of digital currency with empirical wavelet transform (Altan et al., 2019). It has to be noted that other approaches exist for this type of problem which we do not consider in this work, such as e.g. support vector regression (Karasu et al., 2020). To sum up, LSTMs are used for a large variety of tasks, since they are able to model non-linear dependencies. However, also because of their non-linear nature, anomalies in the input data can increase the prediction error significantly. LSTM networks can also be sensitive to the choice of parameters (Krstanovic and Paulheim, 2017). For our application, a prediction method robust to anomalies is needed.

Our approach does not aim to make the LSTM prediction network itself more robust but instead to ensure that its inputs contain as few anomalies as possible. Therefore, as stated in Section 1.2, a method for multivariate, unsupervised anomaly detection in time series is required. Goldstein and Uchida (2016) proposed different types of anomalies, showing that even the definition of multivariate anomalies in two dimensions can be difficult and not always unambiguous. They suggested that an anomaly score will therefore be more useful than binary labels. In this work, we use a distance metric as an anomaly score. One possibility to detect anomalies is to create hand-engineered criteria by defining an interval within an instance is assumed to be regular. This works well for univariate anomalies, i.e. in one dimension. However, even for two dimensions defining appropriate intervals can become difficult since the underlying dependence of the data may imply that building the Cartesian product of intervals does not yield a reliable bivariate regular region. In general, creating useful hand-specified criteria can involve significant engineering efforts and domain expertise, especially for high dimensional data.

As a consequence, we opt for developing purely data driven methods. We use reconstruction based anomaly detection methods, such as autoencoders (Park et al., 2017). An autoencoder compresses and reconstructs high dimensional inputs. The idea behind the anomaly detection is that the autoencoder cannot reconstruct unforeseeable patterns or noise as well as regular data. Ideally, this process eliminates anomalies, however, some potentially useful information may also be lost due to the compression. By comparing the inputs to the autoencoder outputs, an anomaly score can be calculated using a distance metric. There are many recent engineering applications for autoencoder-based anomaly detection, e.g. machinery fault detection (Shen et al., 2018), statistical process monitoring (Lee et al., 2019), integrated modular avionics (Gao et al., 2018), high performance computing (Borghesi et al., 2019).

The variational autoencoder (VAE) is a special kind of autoencoder and was introduced by Kingma and Welling (2013) as a stochastic variational inference and learning algorithm. A variational autoencoder is a probabilistic graphical model that combines variational inference with deep learning (An and Cho, 2015). The VAE framework has a wide array of applications from generative modeling to semi-supervised learning and representation learning (Kingma and Welling, 2019). The VAE reconstruction error can be used to determine an anomaly score (Chen et al., 2019b,a; Marchi et al., 2015a; Sakurada and Yairi, 2014; Borghesi et al., 2019). Moreover, there is a whole branch of LSTM-based anomaly detectors which exploit a property of inconsistent signal reconstruction in the presence of anomalies (Marchi et al., 2015b). An and Cho (2015) used a reconstruction probability instead of a reconstruction error, which they considered a more objective anomaly measure than a reconstruction error. However, their procedure needs multiple runs of a VAE, making it computationally expensive on a data set considered in this paper, which is why we use the standard reconstruction error approach.

VAE models were also combined with LSTM or RNN networks (Principi et al., 2017; Cho et al., 2014; Malhotra et al., 2016). Bao et al. (2017) proposed a stacked autoencoder with wavelet transforms and LSTM networks for stock price predictions. Essien and Giannetti (2019) suggested a similar approach for univariate time series prediction with the addition of convolutional neural networks. Zhang and Chen (2019) presented an unsupervised model-based anomaly detection approach, which assumes that anomalies are objects that do not fit perfectly with the model. They used a LSTM model as encoder and decoder in a VAE to capture temporal dependencies. Also Park et al. (2017) described a LSTM based autoencoder. They introduced a LSTM-VAE-based detector using a reconstruction-based anomaly score for identifying when the current execution of a robot arm differs from past successful experiences.

However, the combination of time series prediction and anomaly detection is sparsely documented. For web-based applications, Chen et al. (2019b) proposed a sequential VAE-LSTM network to integrate anomaly detection and also trend prediction under one framework. The models performed better on prediction and anomaly detection than a VAE or LSTM alone. They used the VAE output as the input for an LSTM for trend prediction and trained both simultaneously using a single loss function.

Our proposed VASP framework processes input data through a VAE, whose output is used for both anomaly detection and signal reconstruction: If an anomaly is detected the anomalous part in the data will be replaced by the VAE. This cleaned data set is then utilized for time series prediction using a LSTM neural network. Compared to the proposed methods from literature, our VASP approach has several advantages. First, the anomaly detection which makes the method robust is independent of the predictor. Consequently, both can be trained independently and exchanged if needed. Second, the prediction

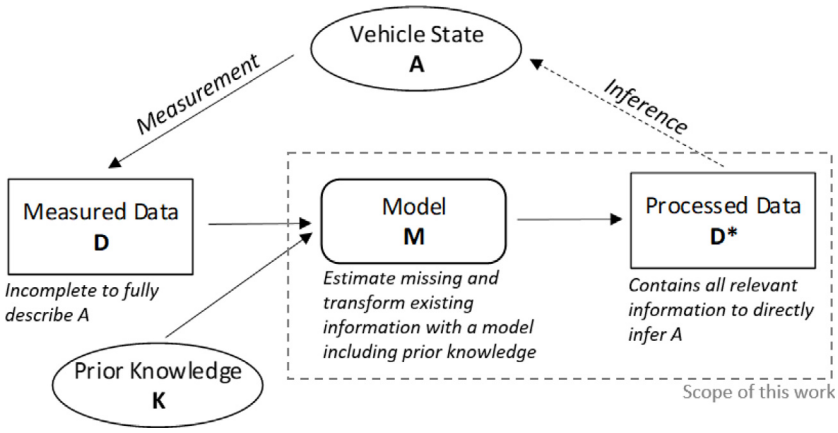


Fig. 1. The underlying process of data processing in motorsport. Incomplete information about a vehicle state A is measured, leading to D. Using a model M which includes prior knowledge results in the processed data D*, reflecting the relevant information to infer A.

model does not even have to be a machine learning model, it can be for example a physical model, making the application of the VASP method versatile. Third, our comparison of VASP to a simple LSTM approach (Baseline) and other different LSTM-VAE architectures (VAE1, VAE2 and VAE3) shows that VASP reduces the prediction error in the case of anomalies on average by 13% to 33%. VAE2 is similar to the VAE-LSTM network of Chen et al. (2019b) apart from the VAE and LSTM model not being trained simultaneously due to our constraint that the framework has to be modular. Chen et al. (2019b) state that using the VAE output makes the block robust to noise and possible anomalies. Although we can confirm that the approach is more robust than using a LSTM without VAE (Baseline) our results show that the accuracy decreases in the presence of a low number of anomalies as described in Section 4.

Summarized, our contributions in this field of research are:

- Combining a VAE with LSTM prediction models to the modular VASP framework, whose selective anomaly replacement before prediction is innovative to the best of our knowledge
- An experimental study of the VASP approach on a large data set from top-level motorsport and a comparison to other methods
- Clear guidelines to implement the proposed methods

The rest of this paper is organized as follows. Section 2 introduces the data which we used to train and evaluate our method. The data comes from multiple seasons of professional motorsport and contains approximately 20 million observations from vehicle sensors, which we refer to as input signals. Additionally, it includes the target signals, which are not always available and should be predicted by the model. These prediction signals are the longitudinal velocity of the vehicle at the center of gravity (vLong) and the side slip angle (aSlip).

Section 3 describes VASP, a modular deep learning architecture based on a VAE and LSTM aiming at solving the aforementioned problem. The basic principles of both methods are detailed as well. Four architectures other than VASP are considered to allow a comparison between these methods. For that purpose, we simulate anomalies. Single or multiple input signals are disturbed by artificial anomalies to evaluate how the different methods cope with anomalies in the input signals. The data sets containing the simulated anomalies are called $AS1$, $AS2$ and $AS3$. The root-mean-square error (RMSE) between reference and predicted signal determines the ranking of the models. The anomaly detection is compared by receiver operating characteristic (ROC) curves and the area under the ROC-curve (AUC).

The experimental results of this paper are presented in Section 4. The prediction tasks are evaluated for two different target signals (vLong and aSlip) and the three data sets with simulated anomalies $AS1$–$AS3$. Comparing the different models, VASP reaches the lowest overall error in all cases. While the other approaches have to compromise in either accuracy (VAE2, VAE2, VAE3) or robustness (Baseline), VASP can combine both due to its selective anomaly replacement.

2. Data

The basis of this study is a data set from BMW Motorsport. It consists of time series signals from a professional motorsport racing series and depicts important race car parameters for vehicle dynamics. In total there were approx. 20 million observations corresponding to almost five days of uninterrupted driving on race tracks. The data were collected over four years from ten different race tracks and 17 different drivers. Since this data set was used for driver and vehicle development in top level motorsport, it was thoroughly checked for anomalies and maintained by experts.

We divided the data into training, validation and test set with a $80-10-10\%$ split. The split was conducted on a per-lap basis, the percentage assigned to validation and test set was chosen to contain enough diversity over the different tracks and drivers. The training set was used to train the models, the validation set for hyperparameter tuning and model selection. The test set was utilized for evaluating the performance of the final results.

Table 1 lists time series signals from the data set with descriptions. The sample rate of all signals was 50 Hz. The signals which were always available (and therefore do not have to be predicted) are referred to as *input signals*. Fig. 2 gives an exemplary overview on the input signals in the time domain.

In contrast, Table 2 lists two signals of sensors which are not allowed in races. If no sensor data are available, these signals will be predicted from the input signals and are therefore referred to as *prediction signals*.

2.1. Notation

Throughout this paper, the set of all input signals are denoted as a matrix $\boldsymbol{X}$

$$\boldsymbol{X} = \begin{bmatrix} X_1^{(1)} & \cdots & X_n^{(1)} \\ \vdots & \ddots & \\ X_1^{(m)} & \cdots & X_n^{(m)} \end{bmatrix} \in \mathbb{R}^{m\times n}.$$

For an entry $X_i^{(t)} \in \mathbb{R}$ in $\boldsymbol{X}$, the index $i = 1,\dots,n$ denotes the feature dimension and the index $t = 1,\dots,m$ the temporal dimension. A column of $\boldsymbol{X}$, which corresponds to n univariate time series is denoted by

$$\boldsymbol{x}_i := \left[X_i^{(1)}, \quad X_i^{(2)}, \quad \dots, \quad X_i^{(m)}\right]^T \in \mathbb{R}^m.$$

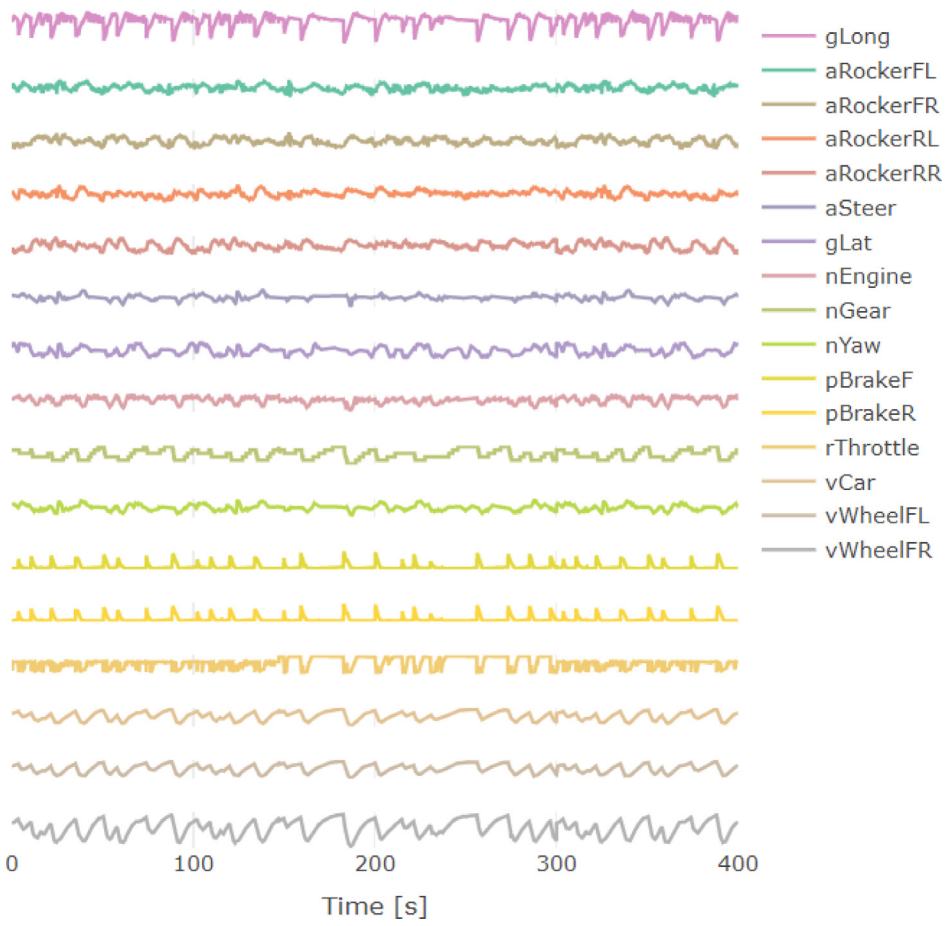


Fig. 2. Overview about the input signals in the time domain.

Table 1
Input signals: Time series signals, which were always available in the data set.

Signal name	Description
gLat	lateral acceleration COG[a]
gLong	longitudinal acceleration COG
aRockerFL	Rocker angle[b] front left
aRockerFR	Rocker angle front right
aRockerRL	Rocker angle rear left
aRockerRR	Rocker angle rear right
aSteer	Steering wheel angle
nYaw	Yaw rate
nEngine	Engine speed
nGear	Gear
pBrakeF	Brake pressure front axle
pBrakeR	Brake pressure rear axle
rThrottle	Throttle displacement
vWheelFL	Wheel speed front left
vWheelFR	Wheel speed front right
vCar	Vehicle speed, calculated from wheel speeds

[a]COG: Center of gravity
[b]Rocker angle: indicates the suspension movement.

Table 2
Prediction signals: Time series signals which should be predicted using the input signals.

Signal name	Description
aSlip	Side slip angle of the vehicle at the rear axle
vLong	longitudinal velocity COG

Similarly, all input time series at the time t, i.e. a row of $\boldsymbol{X}$ is denoted as

$$\boldsymbol{x}^{(t)} := \left[X_1^{(t)}, \quad X_2^{(t)}, \quad \ldots, \quad X_n^{(t)}\right] \in \mathbb{R}^n.$$

2.2. Data normalization

All time series were normalized: Let $\boldsymbol{w}_i \in \mathbb{R}^m$ be a raw time series signal. Then, $\boldsymbol{x}_i \in \mathbb{R}^m$ is calculated by

$$\boldsymbol{x}_i = \frac{1}{\sigma(\boldsymbol{w}_i)}(\boldsymbol{w}_i - \mu(\boldsymbol{w}_i)\boldsymbol{e}),$$

with the all-ones vector $\boldsymbol{e} \in \mathbb{R}^m$, whereby $\mu(\boldsymbol{w}_i) \in \mathbb{R}$ is the sample mean and $\sigma(\boldsymbol{w}_i) \in \mathbb{R}$ the sample standard deviation of $\boldsymbol{w}_i$. Notice that this normalization increases the numerical stability for the later use in artificial neural networks.

2.3. Correlations

Fig. 3 clearly shows that some input and prediction signals are strongly Pearson-correlated. These correlations can be exploited by the VAE, however, it is also capable of detecting and using non-linear relations between the signals. For example vLong and vWheelFL or vWheelFR have a Pearson correlation coefficient very close to one which is natural since the wheel speeds should be approximately similar to the vehicle speed. However, they differ in the case of tire slippage, which occurs regularly in motorsport as the cars are driven on their dynamic limit. Then both vLong and the wheel speeds need to be analyzed to infer the vehicle state. What is more, the vCar calculation based on the wheel speeds is not reliable in this case. These extreme driving situations are, however, the most interesting for engineers, emphasizing the need for the prediction signal vLong. Furthermore, signals which have a weak Pearson correlation can be important as well, since there can be a non-linear relation. For example the longitudinal acceleration gLong and vLong have a correlation close to zero,

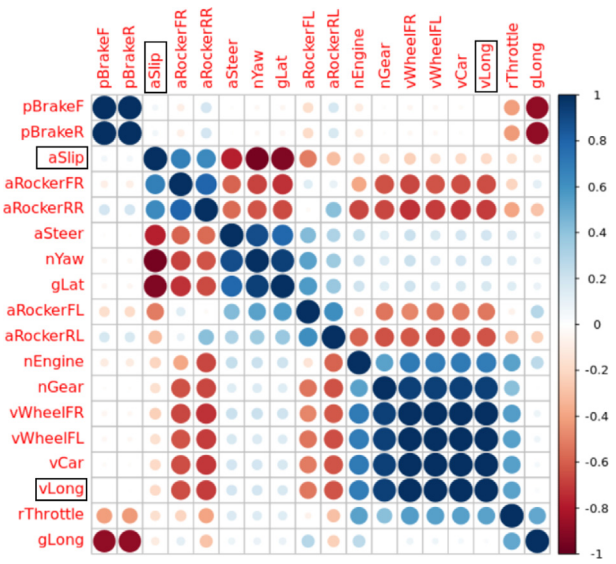


Fig. 3. Pearson correlation matrix of all signals which are used as model inputs and outputs.

however, acceleration and velocity are closely correlated through a differential equation.

For aSlip the situation is similar. The side slip angle is highest in corners and therefore closely correlated to aSteer, gLat and nYaw.

In summary, all of the input signals listed in Table 1 are used as model inputs. No feature selection is applied to provide as much information as possible to the VAE to reconstruct anomalous signals.

3. Methods

3.1. Overview

Fig. 4 sketches the VASP framework. It consists of three kinds of modules, the VAE, anomaly detection and predictors. The input signals are processed through the VAE module V and compared to the raw inputs $\boldsymbol{x}_i$ in the anomaly detector A. If the difference is higher than a predefined threshold, the input is assumed to be anomalous. Sequences classified as anomalous are replaced by the VAE outputs, yielding $\tilde{\boldsymbol{x}}_i$ which are passed on to the predictor modules $\{P_1, \dots, P_h\}$.

In the rest of this section we describe all parts of VASP, including the underlying principles of the applied methods. What is more, the anomaly simulation procedure and different models to benchmark VASP are detailed.

3.2. Variational autoencoder

The variational autoencoder (VAE) is a stochastic variational inference and learning algorithm which uses a neural network for the recognition model. This section summarizes the VAE method described in the original publication from Kingma and Welling (2013).

Considering a data set $\boldsymbol{X} = [x_1, \dots, x_n]^{\mathrm{T}} \in \mathbb{R}^{n\times 1}$ consisting of n independent and identically distributed samples of a continuous variable X. It is assumed that the data was generated by an unobserved random variable Z. The process has two steps:

1. Z is generated from a distribution $p^*_\theta(Z)$, called the prior
2. X is generated from a conditional distribution $p^*_\theta(X|Z)$, called the likelihood.

We assume that the prior and likelihood come from parametric families of distributions with densities $p_\theta(Z)$ and $p_\theta(X|Z)$. The true parameter θ^* and the values of the latent variable Z are unknown to us. The goal is to derive information on the latent variable and on the conditional density $p_\theta(Z|X)$ given the observed data.

Utilizing the model parameters θ we aim to maximize the marginal likelihood $p_\theta(X) = \int p_\theta(X|Z)p_\theta(Z)dZ$. However, the integral is intractable to compute for complex likelihood functions $p_\theta(X|Z)$, as in ANNs with non-linear hidden layers considered in this work (Kingma and Welling, 2013). To overcome this, we introduce an approximation $q_\phi(Z|X) \approx p_\theta(Z|X)$ for the true posterior $p_\theta(Z|X)$.

$q_\phi(Z|X)$ should be as close as possible to $p_\theta(Z|X)$, whereby closeness can be quantified according to the Kullback–Leibler divergence D_{KL} which can be considered a measure for information loss if one distribution is used to represent another one. For the Kullback–Leibler divergence of $q_\phi(Z|X)$ and $p_\theta(Z|X)$ we have that

$$D_{KL}(q_\phi(Z|X) \parallel p_\theta(Z|X)) =$$
$$\log p_\theta(X) + D_{KL}(q_\phi(Z|X) \parallel p_\theta(Z)) - \mathbb{E}_{q_\phi(Z|X)}[\log p_\theta(X|Z)],$$

whereby $\mathbb{E}$ is the expectation operator. Rearrangement yields

$$\log p_\theta(X) - D_{KL}(q_\phi(Z|X) \parallel p_\theta(Z|X)) = \mathbb{E}_{q_\phi(Z|X)}[\log p_\theta(X|Z)] - D_{KL}(q_\phi(Z|X) \parallel p_\theta(Z)). \tag{1}$$

The left hand side of (1) is what we want to maximize, the (log-) likelihood of generating real data $\log p_\theta(X)$ should be high and the difference between the real data and estimated posterior distributions $D_{KL}(q_\phi(Z|X) \parallel p_\theta(Z|X))$ should be low. Since $D_{KL}(q_\phi(Z|X) \parallel p_\theta(Z|X))$ is non-negative, the right hand side of (1) is an evidence lower bound (ELBO) for $\log p_\theta(X)$

$$\log p_\theta(X) = \log \int p_\theta(X|Z)p_\theta(Z)dZ \geq \mathbb{E}_{q_\phi(Z|X)}[\log p_\theta(X|Z)] - D_{KL}(q_\phi(Z|X) \parallel p_\theta(Z)). \tag{2}$$

The negative of (2) yields a loss function $L_{VAE}(\theta, \phi)$ which has to be minimized with respect to θ and ϕ.

$$\begin{aligned} L_{VAE}(\theta, \phi) &= -\log p_\theta(X) + D_{KL}(q_\phi(Z|X) \parallel p_\theta(Z|X)) \\ &= -\mathbb{E}_{q_\phi(Z|X)}[\log p_\theta(X|Z)] + D_{KL}(q_\phi(Z|X) \parallel p_\theta(Z)) \end{aligned} \tag{3}$$

Additional information and proofs can be found in Kingma and Welling (2013, 2019).

In the case of the variational autoencoder, we use ANNs for the probabilistic encoder $q_\phi(Z|X)$ and decoder $p_\theta(X|Z)$, where the parameters ϕ and θ are optimized simultaneously. We let $q_\phi(Z|X)$ be a multivariate Normal distribution

$$\log q_\phi(Z|X) = \log \mathcal{N}(z; \mu_z, \mathrm{diag}(\sigma_z^2)).$$

Thereby, the mean μ_z and the diagonal covariance matrix $\mathrm{diag}(\sigma_z^2)$ are non-linear functions of X, which are determined by an ANN. z is obtained by sampling from $q_\phi(Z|X)$

$$z \sim q_\phi(Z|X).$$

However, for training a neural network, this cannot be used for backpropagating the gradient since the sampling $z \sim q_\phi(Z|X)$ is a stochastic process. With the *reparameterization trick* (Kingma and Welling, 2013), z is represented as the sum of a deterministic variable and an auxiliary independent random variable ϵ. Using a multivariate Gaussian distribution $\mathcal{N}$

$$z = \mu_z + \sigma_z \odot \epsilon, \text{where } \epsilon \sim \mathcal{N}(0, \boldsymbol{I}). \tag{4}$$

This allows the VAE to be trained with backpropagation and, for example, a gradient descent optimization algorithm.

3.2.1. Implementation

Fig. 5 shows a graphical representation of the VAE which is used in this work. The encoder is an ANN which maps the input $\boldsymbol{X}$ to a latent variable $\boldsymbol{Z}$ and is denoted by $q_\phi(\boldsymbol{Z}|\boldsymbol{X})$. The decoder $p_\theta(\boldsymbol{X}|\boldsymbol{Z})$, also an ANN, tries to reconstruct $\boldsymbol{X}$ given the latent variable $\boldsymbol{Z}$. The resulting

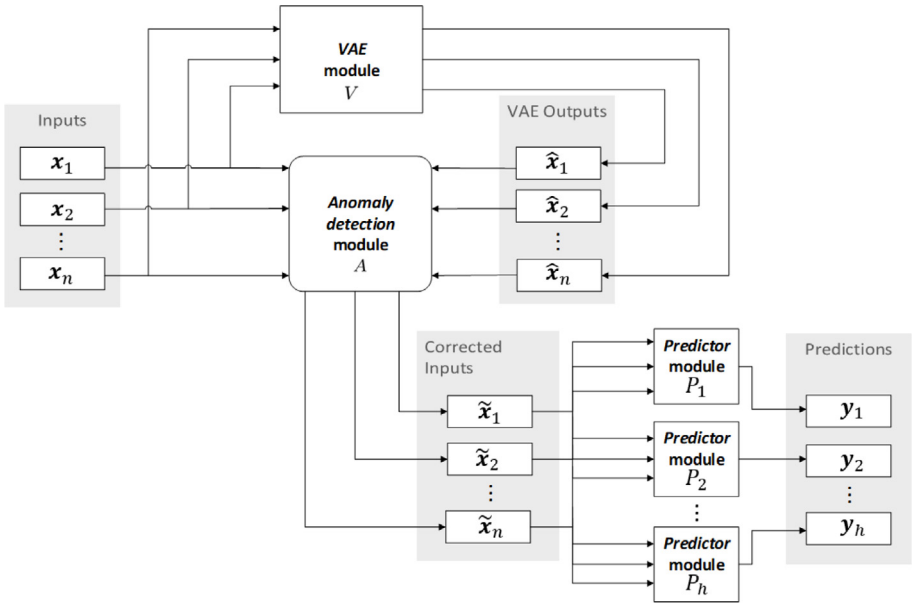


Fig. 4. Proposed VASP framework for anomaly detection and time series prediction. The input time series $\boldsymbol{x}_i \in \mathbb{R}^m$ with m time steps are passed through the VAE, creating $\hat{\boldsymbol{x}}_i \in \mathbb{R}^m$. These signals are compared to the original input signals $\boldsymbol{x}_i$ in the anomaly detection module. If an anomaly is detected, the anomalous part of $\boldsymbol{x}_i$ will be replaced by $\hat{\boldsymbol{x}}_i$. This results in the corrected inputs $\tilde{\boldsymbol{x}}_i \in \mathbb{R}^m$ which are passed to the predictor modules, yielding the predictions $\boldsymbol{y}_j \in \mathbb{R}^m$.

output is $\hat{\boldsymbol{X}}$. The basic principle is that the input is compressed by the encoder into the latent variable and then reconstructed by the decoder. The variable $\boldsymbol{Z}$, can be interpreted as a dimensional reduction of the input $\boldsymbol{X}$.

Note, that in this work the input $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ is a matrix (instead of a vector) containing the time series. The latent variable $\boldsymbol{Z} \in \mathbb{R}^k$ remains a vector. Therefore, the encoder needs to collapse one dimension of $\boldsymbol{X}$ and the decoder needs to reconstruct it. For this we used a LSTM network, the details are described in Section 3.4.

The loss function (3) consists of two parts. For the Kullback–Leibler divergence, we let both $q_\phi(Z|X)$ and $p_\theta(X|Z)$ be multivariate Gaussian distributions. The Kullback–Leibler divergence in that case is

$$D_{KL}(q_\phi(Z|X) \parallel p_\theta(Z)) = -\frac{1}{2}\sum_{k=1}^{K}(1+\log(\sigma_{z_k}^2) - \mu_{z_k}^2 - \sigma_{z_k}^2),$$

where K has the dimension of z and k denotes the kth element of the vector (Kingma and Welling, 2013). To represent the expected reconstruction error, we used the sum of the square error between the input $\boldsymbol{X}$ and output $\hat{\boldsymbol{X}}$ of the VAE. We sum over both the feature and the time dimension to calculate the reconstruction loss. The resulting loss function is

$$L_{VAE}(\theta,\phi) = \sum_{t=1}^{m}\sum_{i=1}^{n}\left(\boldsymbol{X}_i^{(t)} - \hat{\boldsymbol{X}}_i^{(t)}\right)^2 - \frac{1}{2}\sum_{k=1}^{K}(1+\log(\sigma_{z_k}^2) - \mu_{z_k}^2 - \sigma_{z_k}^2) \tag{5}$$

Figs. 7 and 8 depict the model architecture of the utilized encoder and decoder, respectively. The hyperparameters are shown in Table 3 and were determined by a grid search. As described above, during training, $\boldsymbol{z}$ arises by sampling from a normal distribution $\mathcal{N}$ parameterized with $\boldsymbol{\mu}_z$ and $\boldsymbol{\sigma}_z^2$ using the reparameterization trick in (4). When the model is used for inference, i.e. for prediction, $\boldsymbol{\mu}_z$ is passed directly on to $\boldsymbol{z}$. Fig. 6 shows the effect of the VAE on three time series signals. The anomalies in the VAE outputs $\hat{\boldsymbol{x}}_i$ are significantly smaller compared to inputs $\boldsymbol{x}_i$.

3.3. Anomaly detection module

After the VAE, the signals are forwarded to the anomaly detection module. From a reference data set $\boldsymbol{X}^*$ without anomalies, the standard deviation $\boldsymbol{\sigma}^* \in \mathbb{R}^n$ is calculated for every input signal i between the input data $\boldsymbol{x}_i^*$ and the VAE output $\hat{\boldsymbol{x}}_i^*$

$$\sigma_i^* = \sigma(|\boldsymbol{x}_i^* - \hat{\boldsymbol{x}}_i^*|). \tag{6}$$

Given σ_i^*, for every time step t and time series i in $\boldsymbol{X}$ an anomaly score $\boldsymbol{\delta} \in \mathbb{R}^{n\times m}$ is calculated with a distance metric

$$\delta_i^{(t)} = |\, X_i^{(t)} - \hat{X}_i^{(t)} \,| \cdot \frac{1}{\sigma_i^*} \tag{7}$$

and $\boldsymbol{\xi} \in \mathbb{R}^{n\times m}$

$$\xi_i^{(t)} = \begin{cases} 1, & \text{if } \delta_i^{(t)} \geq \alpha \\ 0, & \text{otherwise,} \end{cases}$$

where $\xi_i^{(t)}$ is a boolean variable indicating if the input $X_i^{(t)}$ is anomalous and α serves as a threshold value. Since $\delta_i^{(t)}$ was divided by σ_i^* in (7), the same value for the threshold α is used for all signals. The anomalous parts of the signal are then replaced, resulting in a corrected output $\tilde{\boldsymbol{X}}$

$$\tilde{X}_i^{(t)} = \begin{cases} \hat{X}_i^{(t)}, & \text{if } \xi_i^{(t)} = 1 \\ X_i^{(t)}, & \text{otherwise.} \end{cases}$$

A threshold value $\alpha = 5$ was found to be suitable by a grid search.

Table 3
Hyperparameters for the VAE.

Parameter	Setting
Learning rate	0.001
Batch size	512
Dropout	0.2
Recurrent dropout (LSTM)	0.1
Optimizer	'Adam' (Kingma and Ba, 2014)
Input temporal dimension	5
Input feature dimension	16
Latent space dimension	8
Output temporal dimension	5
Output feature dimension	16

3.4. Long short-term memory networks

Having at hand the anomaly detection module the next step is the prediction process in VASP. The predictors are based on LSTM neural

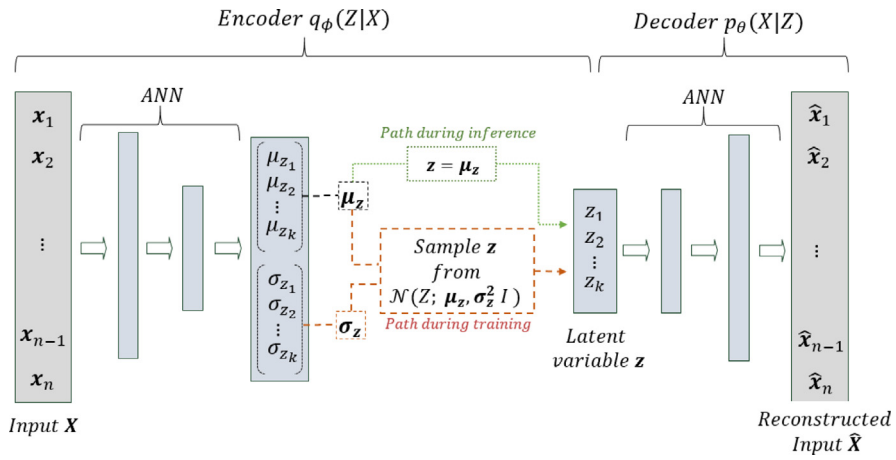


Fig. 5. Simplified structure of the VAE used in this paper. The input matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ is mapped to the latent variable vector $\boldsymbol{z} \in \mathbb{R}^k$ by the encoder. During training, $\boldsymbol{z}$ arises from sampling from a normal distribution $\mathcal{N}$ parameterized with $\boldsymbol{\mu}_z \in \mathbb{R}^k$ and $\boldsymbol{\sigma}_z^2 \in \mathbb{R}^k$. When the model is used for inference, $\boldsymbol{\mu}_z$ is directly passed to $\boldsymbol{z}$. The decoder tries to reconstruct $\boldsymbol{X}$ from $\boldsymbol{z}$ resulting in $\hat{\boldsymbol{X}}$.

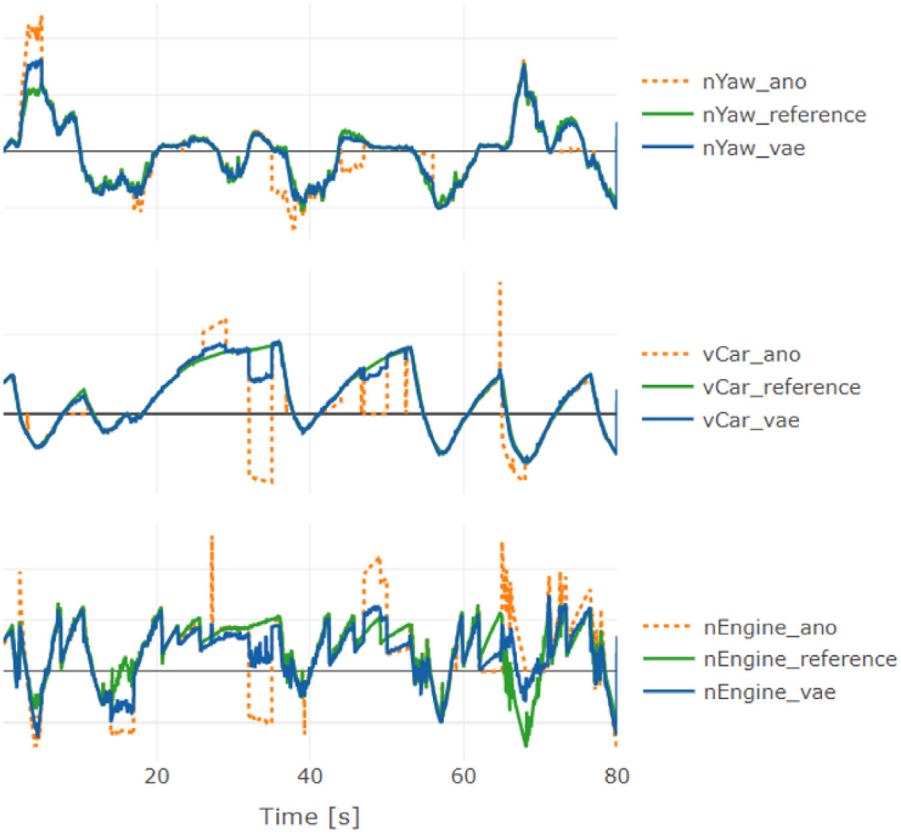


Fig. 6. The effect of the VAE on three input signals. The orange signals are the inputs which are perturbed by anomalies. The blue signals are the VAE outputs and the green signals are the ground truth references which are illustrated for comparison. Clearly, the VAE reduces the anomalies in the orange input signals leading to the blue signals being much closer to the green reference signals. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

networks introduced by Hochreiter and Schmidhuber (1997). A LSTM consists of an *input gate* $\boldsymbol{g}_{in}$, *forget gate* $\boldsymbol{g}_f$ and *output gate* $\boldsymbol{g}_o$. With these, the LSTM decides which information will be forgotten, learned or passed on to the next time step. The outputs of the time step $t-1$ serve as inputs at the time step t. Apart from the output, a LSTM passes the cell state $\boldsymbol{C}$ to the next time step. The cell state makes it easier for information to flow unchanged through multiple time steps, resulting in the ability of the LSTM cells to learn long-term dependencies and avoiding the vanishing gradient problem.

The LSTM can be described by a set of equations (Hochreiter and Schmidhuber, 1997). Let $\boldsymbol{x}^{(t)} \in \mathbb{R}^m$ be the input vector and $\boldsymbol{y}^{(t)} \in \mathbb{R}^M$ be the output vector at the time t, following the notation defined in Section 2.1. Furthermore, let $\boldsymbol{W}_f, \boldsymbol{W}_{in}, \boldsymbol{W}_C, \boldsymbol{W}_o \in \mathbb{R}^{m \times (m+M)}$ be weight matrices and $\boldsymbol{b}_f, \boldsymbol{b}_{in}, \boldsymbol{b}_C, \boldsymbol{b}_o \in \mathbb{R}^M$ the bias vectors for each gate. For the forget gate $\boldsymbol{g}_f^{(t)} \in \mathbb{R}^M$ at the time step t we have

$$\boldsymbol{g}_f^{(t)} = \sigma_{act}(\boldsymbol{W}_f \cdot [\boldsymbol{y}^{(t-1)}, \boldsymbol{x}^{(t)}] + \boldsymbol{b}_f),$$

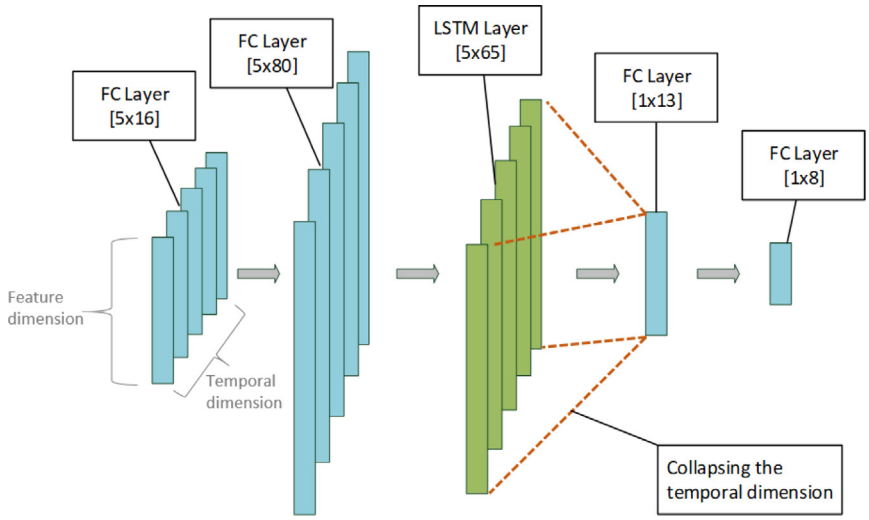


Fig. 7. VAE encoder model architecture. FC layers are fully connected layers. The input consists of 16 signals (feature dimension) over 5 time steps (temporal dimension). The LSTM layer collapses the temporal dimension.

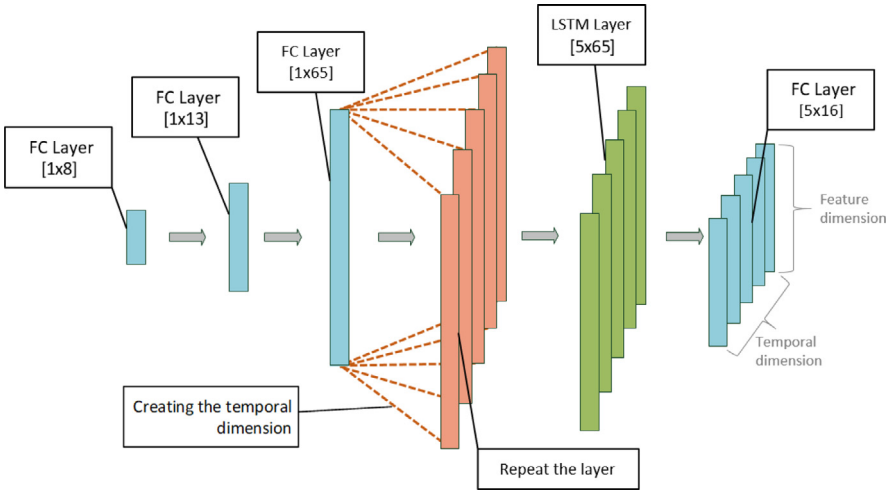


Fig. 8. VAE decoder model architecture. The FC layer is repeated before the LSTM layer to recreate the temporal dimension. The output dimension of 5 x 16 has to be equal to the input dimension of the encoder.

where σ_{act} is an activation function. The input gate $\boldsymbol{g}_{in}^{(t)} \in \mathbb{R}^M$ controls the new vector for the candidate cell state $\tilde{\boldsymbol{C}}^{(t)} \in \mathbb{R}^M$

$$\boldsymbol{g}_{in}^{(t)} = \sigma_{act}(\boldsymbol{W}_{in} \cdot [\boldsymbol{y}^{(t-1)}, \boldsymbol{x}^{(t)}] + \boldsymbol{b}_{in}),$$
$$\tilde{\boldsymbol{C}}^{(t)} = \tanh(\boldsymbol{W}_{C} \cdot [\boldsymbol{y}^{(t-1)}, \boldsymbol{x}^{(t)}] + \boldsymbol{b}_{C}),$$

with tanh being the hyperbolic tangent. The new cell state $\boldsymbol{C}^{(t)} \in \mathbb{R}^M$ comprises of the old state $\boldsymbol{C}^{(t-1)}$ and the candidate cell state $\tilde{\boldsymbol{C}}^{(t)}$

$$\boldsymbol{C}^{(t)} = \boldsymbol{g}_{f}^{(t)} \odot \boldsymbol{C}^{(t-1)} + \boldsymbol{g}_{in}^{(t)} \odot \tilde{\boldsymbol{C}}^{(t)},$$

whereby $\odot$ is the element-wise product. The output $\boldsymbol{y}^{(t)}$ is controlled by the output gate $\boldsymbol{g}_{o}^{(t)} \in \mathbb{R}^M$

$$\boldsymbol{g}_{o}^{(t)} = \sigma_{act}(\boldsymbol{W}_{o} \cdot [\boldsymbol{y}^{(t-1)}, \boldsymbol{x}^{(t)}] + \boldsymbol{b}_{o}),$$
$$\boldsymbol{y}^{(t)} = \boldsymbol{g}_{o}^{(t)} \odot \tanh(\boldsymbol{C}^{(t)}).$$

There are also different variants of LSTMs, though, according to a study from Greff et al. (2017) these did not show significant improvements over the standard LSTM architecture.

3.5. Prediction module

The prediction module contains two bidirectional LSTM layers. In contrast to a standard LSTM, a bidirectional LSTM uses every time series in reverse order as an additional input, which can be more effective than standard LSTM layers (Graves and Schmidhuber, 2005). Both a recurrent dropout in the LSTM and a dropout layer were used to prevent overfitting. The model architecture and the hyperparameters were optimized with a grid search and are summarized in Fig. 9 and Table 4. The Keras implementation of the bidirectional LSTM was employed (Chollet et al., 2015; Allaire and Chollet, 2018).

The module to predict the j^{th} time series $\boldsymbol{y}_j$ with the input $\boldsymbol{X}$ is denoted by P_j.

$$\boldsymbol{y}_j = P_j(\boldsymbol{X})$$

All predictions $\boldsymbol{y}_j$ are denoted by $\boldsymbol{Y}$

$$\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_h].$$

Note, that we used different predictors P_j for each $\boldsymbol{y}_j$. This makes the framework modular, because one predictor can be changed without effecting the others.

3.6. Anomaly simulations

To evaluate the performance of the proposed framework we used the data set described in Section 2. This data set was assumed to contain very few anomalies since it was carefully checked by specifically

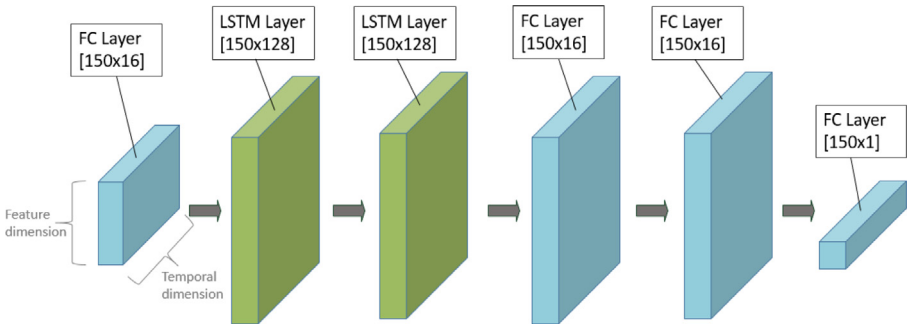


Fig. 9. LSTM-based Predictor module architecture.

Table 4
Hyperparameters for the LSTM-based predictor module.

Parameter	Setting
Learning rate	0.001
Batch size	256
Dropout	0.3
Recurrent dropout	0.1
Optimizer	"Adam" (Kingma and Ba, 2014)
Input temporal dimension	150
Input feature dimension	16
Output temporal dimension	150
Output feature dimension	1

trained engineers. Moreover, anomalies are by definition rare events, making it unfeasible to gather enough anomalous data for a detailed characterization. Also, the ground truth is usually not known. To test the performance of our framework we therefore simulated different kinds of anomalies, the main advantage being that an arbitrary amount of anomalous data can be created and that the original data can serve as a ground truth. A disadvantage of this procedure could be that an anomaly detector could overfit to the simulated anomalies and achieve worse results for real anomalies. Therefore, all models were trained on real data only. The simulated anomalies were used exclusively for model evaluation.

We used three different simulation methods to create the anomalous data sets $AS1$, $AS2$ and $AS3$. In $AS1$ and $AS2$, one signal at a time was perturbed with anomalies, for $AS3$ multiple signals were altered simultaneously. For the anomaly simulations only a fraction of the test set was used since the anomaly simulations enlarged the data set significantly.

Anomalies by permutation (AS1)

A time series signal $\boldsymbol{x}_i$ was randomly permuted in the time dimension

$$\overline{\boldsymbol{x}}_i = \pi(\boldsymbol{x}_i),$$

whereby π denotes the permutation operator. Fig. 10 shows an example of permuted signals. The temporal information is lost, however the mean and the standard deviation over the time dimension remain equal to the original signal. The test set contained 15,000 samples (300 s in the time domain). The 16 input signals and two output signals were permuted randomly one at a time. In addition to the unaltered version this resulted in 19 variations of the test set leading to a data set with 285,000 observations.

Anomalies by injection (AS2)

Permutation in the time dimension is an extreme example of disturbance in a signal. More realistic anomalies were created as well. Especially, gains, offsets and short spikes on sensor signals, which can arise from false calibration or be caused by mechanical wear over a period of time, were reported by engineers frequently. We tried to model these anomalies using the following approach.

Table 5
Anomaly injection parameters for $AS2$.

Method	p	R	d
Gain	0.5	$[-1,\ 2]$	300
Spike	0.001	$[5\mu(\boldsymbol{x}_i),\ 5\mu(\boldsymbol{x}_i)]$	5
Offset	0.1	$[-3\mu(\boldsymbol{x}_i),\ 3\mu(\boldsymbol{x}_i)]$	300

The number of anomalous points is denoted by L and calculated from to the length n of $\boldsymbol{x}_i$ and a given parameter p

$$L = \lfloor np \rceil.$$

Let $\boldsymbol{c} \in \mathbb{R}^L$ be the index vector of selected points from $\boldsymbol{x}_i$

$$c_1, \dots, c_L \in \{1, \dots, n\}$$
$$c_1 < \cdots < c_L.$$

Given an interval $R = [a, b]$, a vector $\boldsymbol{\varrho} \in \mathbb{R}^L$ is drawn randomly using the discrete uniform distribution U

$$\varrho_1, \dots, \varrho_L \in R,$$
$$\varrho_e \sim U(R), e \in \{1, \dots, L\}.$$

Additionally, a parameter $d \in \mathbb{N}$ is defined which allows applying the transformation to the next d time steps in the time series $\{\boldsymbol{x}_i^{(t)}, \dots, \boldsymbol{x}_i^{(t+d)}\}$.

Then, the time series $\overline{\boldsymbol{x}}_i \in \mathbb{R}^m$ is determined through a recursive procedure. Thereby, the second lower index denotes the looping variable

(*i*) $\tilde{\boldsymbol{x}}_{i,0} = \boldsymbol{x}_i$

(*ii*) repeat for $e = \{1, \dots, L\}$ and $t = \{1, \dots, m\}$:

$$\tilde{x}_{i,e}^{(t)} = \begin{cases} T(\tilde{x}_{i,e-1}^{(t)}, \varrho_e), & \text{if } c_e \le t \le c_e + d \\ \tilde{\boldsymbol{x}}_{i,e-1}^{(t)}, & \text{otherwise} \end{cases}$$

(*iii*) $\overline{\boldsymbol{x}}_i = \tilde{\boldsymbol{x}}_{i,L}$.

In doing so, the following two transformations were applied:

- Gain

  $$T(x, y) = x \cdot y$$

- Offset

  $$T(x, y) = x + y.$$

Spikes were also generated using the offset transformation with different parameters. The described procedure can lead to some points being transformed multiple times. We included this possibility on purpose to make the data more diverse.

Fig. 11 shows an example for signals with anomaly injection. Analogous to $AS1$, this procedure results in a data set with 285,000 observations.

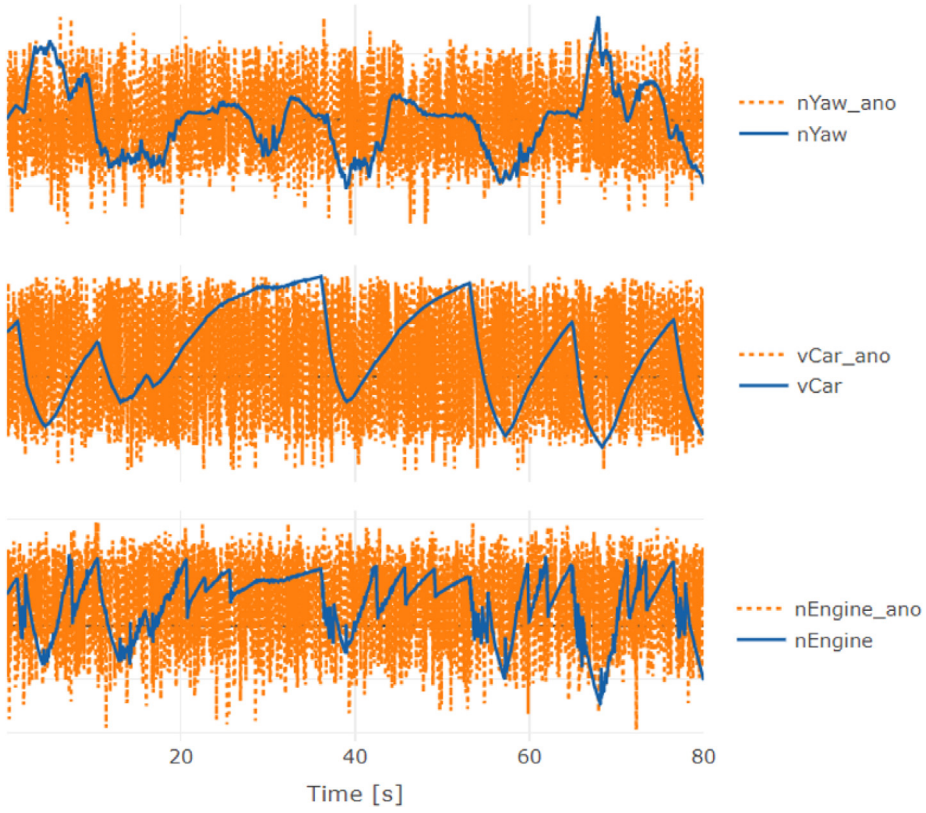


Fig. 10. Examples of normal (blue) and permuted (orange) signals from $AS1$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

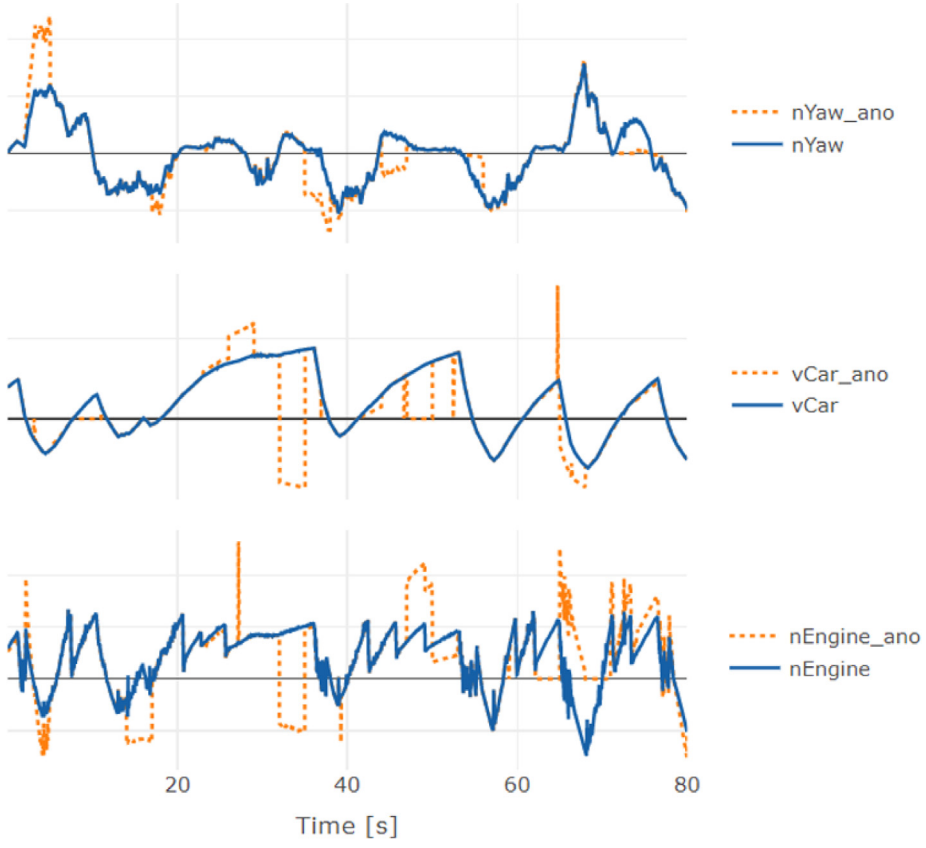


Fig. 11. Examples of normal signals (blue) and signals with simulated anomalies from $AS2$ (orange) using the parameters in Table 5. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Combination of multiple anomalies (AS3)

The effect of multiple anomalies was created with the same method which was used to create the $AS1$ data set. We studied the cases of k out of the n signals being anomalous. There are $\binom{n}{k}$ possible combinations for each value of k. It would be computationally expensive to simulate all possibilities (in our case for $n = 16$). Therefore, we decided to randomly draw m samples from the $\binom{n}{k}$ possible combinations as an approximation. The process was executed for $k = [0, \dots, n]$. The results were then averaged over all m samples for each value of k. For this experiment, the test data set was decreased to 3000 samples, which corresponds to 60 s in the time domain. The number of simulated combinations was set to $m = 100$. Combined with $k = 16$ signals, this resulted in a data set with 4.8 million observations.

3.7. Evaluation metrics

Prediction

For the predicted signals the root-mean square error (RMSE) could be used since the ground truth signals were available in the data. The RMSE between a reference signal $\boldsymbol{yr}_j$ and the predicted signal $\boldsymbol{y}_j$ with m time steps is

$$\text{RMSE}(\boldsymbol{yr}_j, \boldsymbol{y}_j) = \sqrt{\frac{1}{m}\sum_{t=0}^{m}(\boldsymbol{yr}_j^{(t)} - \boldsymbol{y}_j^{(t)})^2}.$$

Anomaly detection

The anomaly detection module $\boldsymbol{\xi}$ was treated as a classifier for evaluation. The ground truth $\boldsymbol{\xi r}$ is 1 if an anomaly was injected and 0 otherwise. The classification results were compared by receiver operating characteristic (ROC) curves and the area under the ROC curve (AUC) using the Precrec package in R (Saito and Rehmsmeier, 2017).

The ROC curve was computed with the true positive rate TPR and false positive rate FPR. Given a continuous random variable U, which serves as a score for binary classification and a threshold parameter α, the instance is classified as 1 for $U > \alpha$ and 0 otherwise. There is a probability density $f_1(u)$ for the instance belonging to the class 1 and $f_0(u)$ for it belonging to the class 0. The TPR and FPR curves are defined by

$$TPR(\alpha) = \int_{\alpha}^{\infty} f_1(u)du,$$
$$FPR(\alpha) = \int_{\alpha}^{\infty} f_0(u)du.$$

The ROC curve is then obtained by plotting $TPR(\alpha)$ over $FPR(\alpha)$ for varying values of α. In contrast, the AUC is a single score and given by

$$AUC = \int_0^1 TPR(FPR^{-1}(u))du.$$

An AUC value of 1 would denote an optimal separation of the classes whereas an AUC value of 0.5 would imply that the model cannot discriminate the classes (an AUC value of 0 would mean that the classes are perfectly separated, however, assigned in reversed order).

3.8. Model comparison

Including VASP, five different frameworks were evaluated on the data set with simulated anomalies.

- VASP (Fig. 4): Selective use of the input signals and VAE outputs. The predictor modules operated on a corrected version of the inputs $\check{\boldsymbol{X}}$

  $\boldsymbol{y}_j = P_j(\check{\boldsymbol{X}})$.

- Baseline (Fig. 12): No anomaly correction on the inputs. The predictor modules used the inputs directly

  $\boldsymbol{y}_j = P_j(\boldsymbol{X})$.

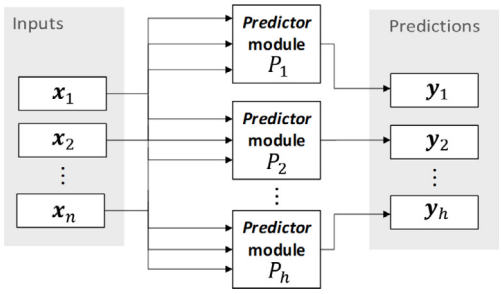


Fig. 12. Baseline model framework. Prediction directly on the basis of the input data.

- VAE1 (Fig. 13): Only the VAE outputs were used for prediction. The predictor module was trained on the input signals $\boldsymbol{X}$

  $\boldsymbol{y}_j = P_j(\hat{\boldsymbol{X}})$

- VAE2 (Fig. 13): Only the VAE outputs were used for prediction. The predictor module $\hat{P}$ was trained on the VAE output signals $\hat{\boldsymbol{X}}$.

  $\boldsymbol{y}_j = \hat{P}_j(\hat{\boldsymbol{X}})$.

- VAE3 (Fig. 14): Similar to Baseline, the predictor modules used the inputs directly to predict $\boldsymbol{y}_j$.

  $\boldsymbol{y}_j = P_j(\boldsymbol{X})$

  All predictions $\boldsymbol{y}_j$ are denoted by $\boldsymbol{Y}$

  $\boldsymbol{Y} = [\boldsymbol{y}_1, \dots, \boldsymbol{y}_h]$.

  Afterward, a VAE V which was trained on the inputs and the predictions, was applied to both $\boldsymbol{X}$ and $\boldsymbol{Y}$

  $\hat{\boldsymbol{Y}} = V([\boldsymbol{X}, \boldsymbol{Y}])$.

4. Results

The accuracy of the anomaly detection module was analyzed. Depending on the anomaly simulation method, the AUC ranged from 0.87 to 0.98.

The main task of the frameworks was time series prediction in the presence of anomalies. Tables 6 and 7 show the overall prediction RMSE for each data set for vLong and aSlip, respectively. These summarized results indicate that the VASP method had no disadvantages, it was the superior method in all scenarios and could reduce the error by 13–33%. The other VAE models were only better than Baseline if the most important input signals had anomalies. In all other cases, too much relevant information was lost in the VAE. Theoretically, a lower rate of compression, i.e. a larger hidden layer z, could decrease the reconstruction error for normal data, however, at the cost of reduced robustness with respect to anomalies. Consequently, there is a trade-off between robustness and accuracy. VASP combined both robustness and accuracy by selecting the VAE outputs only if the input signals showed anomalies, making the architecture very suitable for the applications considered in this work.

In the following section the results are compared and discussed in detail. The experiments were carried out in R 3.6.2 (R Development Core Team, 2008) using the R interface to Keras (Chollet et al., 2015; Allaire and Chollet, 2018) and tensorflow (Allaire and Tang, 2018).

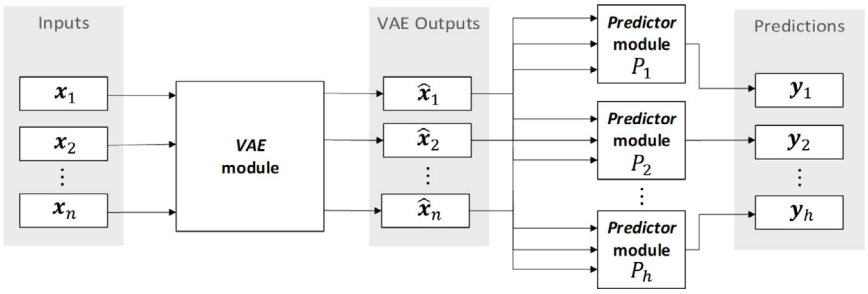


Fig. 13. VAE1 and VAE2 model framework. Only the VAE outputs were passed on to the predictor.

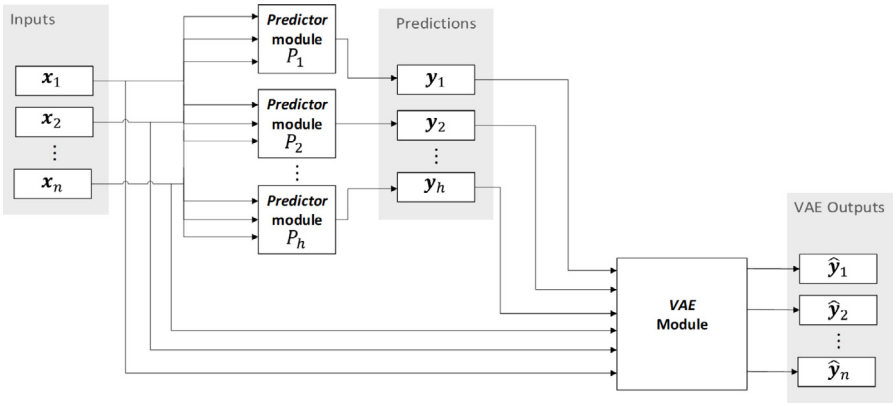


Fig. 14. VAE3 model framework. The first part is equivalent to Baseline, then both the predictions and inputs were forwarded into a VAE module.

Table 6
RMSE comparison for vLong prediction for three different anomaly simulations ($AS1 - AS3$). The columns after the data set label denote the corresponding ranks. VASP had the least error for all scenarios.

Model	*AS1*		*AS2*		*AS3*	
Baseline	0.082	2	0.069	2	0.478	4
VAE1	0.107	4	0.102	4	0.434	3
VAE2	0.105	3	0.096	3	0.428	2
VAE3	0.277	5	0.251	5	0.538	5
VASP	**0.054**	1	**0.060**	1	**0.415**	1

Table 7
RMSE comparison for aSlip prediction for three different anomaly simulations ($AS1 - AS3$). The columns after the data set label denote the corresponding ranks. Again, VASP had the least error for all scenarios.

Model	*AS1*		*AS2*		*AS3*	
Baseline	0.340	2	0.278	2	0.819	5
VAE1	0.399	5	0.331	5	0.759	2
VAE2	0.376	4	0.304	4	0.767	3
VAE3	0.368	3	0.294	3	0.809	4
VASP	**0.276**	1	**0.243**	1	**0.682**	1

4.1. Anomaly detection

The anomaly detection module was evaluated on the data sets $AS1 - AS3$. Fig. 15 depicts the AUC values. The anomaly detection module was equal for VASP, VAE1 and VAE2 since these models used the same VAE. The AUC value of $AS2$ (0.87) was lower than the one of $AS1$ (0.96). This is to be expected since the anomalies arising from time permutations changed the signals to a much larger extent than the ones generated by mechanism $AS2$.

The $AS3$ data set was used for analyzing the influence of the number of anomalous inputs. The highest score of 0.98 was attained for one anomalous signal, the lowest score of 0.59 for the setting in which all signals were anomalous. Not surprisingly, the AUC decreased steadily with increasing number of anomalies. Fig. 16 shows the ROC curves for $AS3$ for 1–16 anomalies.

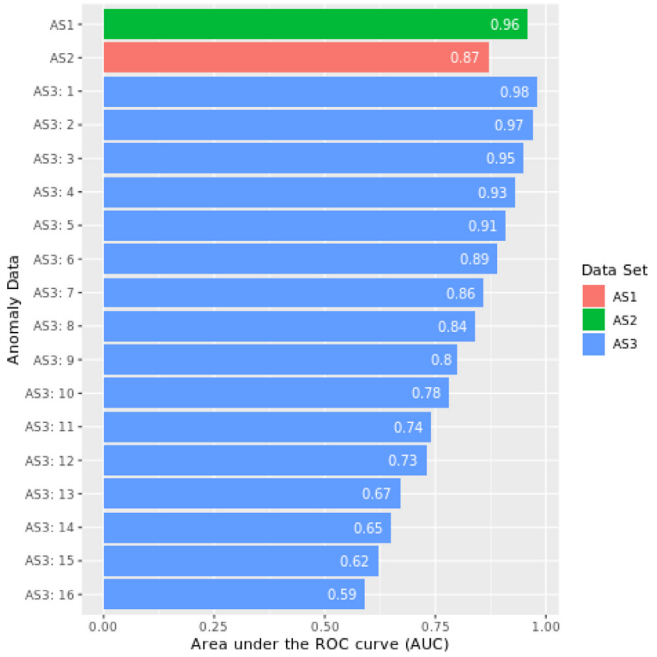


Fig. 15. Area under the ROC curve (AUC) for the simulated data sets $AS1 - AS3$. For $AS3$ also the number of anomalous signals is included on the y-axis.

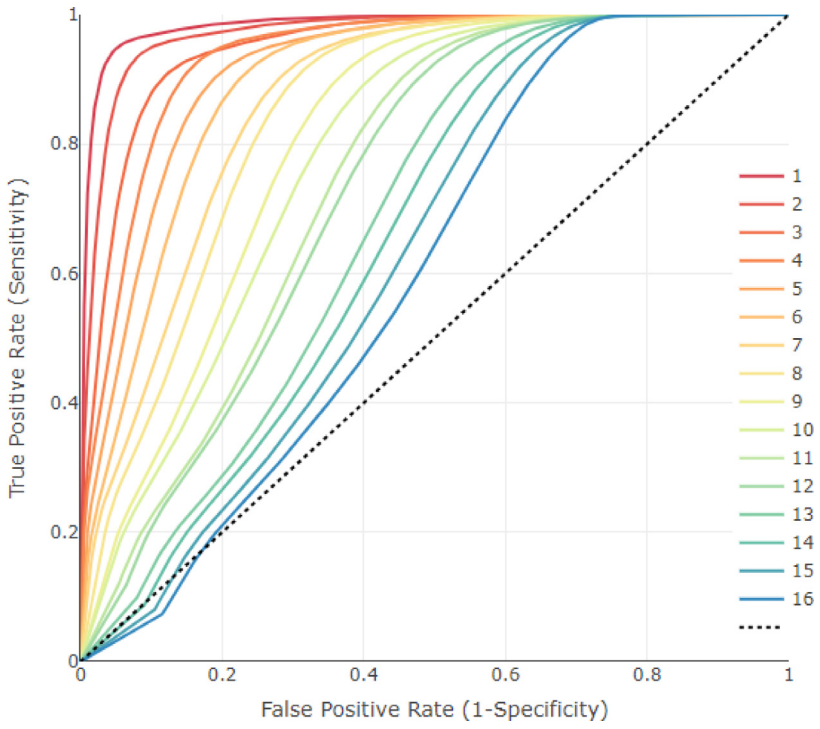


Fig. 16. Receiver operating characteristic (ROC) curves for the simulated data set $AS3$ (1–16 anomalies), according to the description in Section 3.7. The dashed line depicts the line of no-discrimination which would be achieved by randomly guessing.

4.2. Robust time series prediction

The achieved accuracy on the prediction task was considered the main evaluation metric for comparing the different models. We compared the models' predictions for vLong and aSlip on the basis of the anomalous data sets $AS1$, $AS2$ and $AS3$.

4.2.1. AS1

In the AS1 scenario one input signal at a time was permuted. Fig. 17 shows the prediction and the reference for the different models for vLong and aSlip, respectively. Additionally, an anomalous input signal as well as the original input signal is depicted in the lowest panel. For vLong (Fig. 17 a) the Baseline model could not predict the reference as accurately as VASP. This applied particularly for the case of large or small values of vLong. The gray arrows indicate time periods where the differences are maximal.

For aSlip (Fig. 17 b), the VASP and Baseline model are visually relatively close. Nevertheless, a difference between VASP and VAE1 and VAE2 is observable as indicated by the gray arrows.

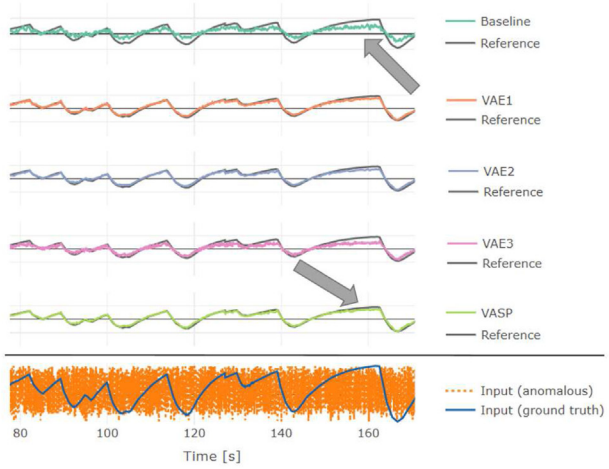
For the complete test set the RMSE was used as an evaluation metric. Fig. 18 illustrates a RMSE comparison of the models. The signal name on the x-axis represents the signal which was anomalous, the y-axis yields the corresponding RMSE. The ordering of the signals was in accordance with the mean RMSE over all models. Notice that the chosen procedure is similar to the permutation feature algorithm proposed by Fisher et al. (2018). The term feature importance was introduced by Breiman (2001) in the context of Random Forests (a method for bagging decision trees) and describes the importance of a feature for a prediction task. However, this method of calculating feature importance is not directly applicable in the ANN context. The permutation feature algorithm is based on the idea that permuting an important signal will decrease the accuracy of a predictor more than permuting an unimportant feature. The difference or ratio between the prediction accuracy on original versus permuted data can be viewed as a measure for feature importance. Consequently, the ordering of the anomalous channels in Fig. 18 can be seen as an ordering in terms of feature importance.

vLong For vLong, vCar followed by vWheelFL and vWheelFR were the most important features, which is reasonable since these signals also have a high Pearson correlation with vCar (Fig. 3). This was most obvious for the Baseline model since it relied almost solely on these three signals leading to a very high error if one of them was disturbed. The range between the best and worst RMSE was large for the Baseline model. On the contrary, for all VAE based models, this difference was smaller. In almost all cases VAE3 exhibited the worst performance, VAE1 and VAE2 produced comparable results. For anomalies in the two most important signals they achieved a lower error than Baseline, however, in all other cases the error was higher. Also when no anomalies were present these models were distinctly worse than the Baseline. VASP exhibited the best overall performance. It was clearly better for anomalies in vCar, vWheelFL and vWheelFR. Only for anomalies in nGear was it distinctly worse than Baseline, in the other cases it was similar. Also for the case of no anomalies being present, it could achieve an error at the level of Baseline.

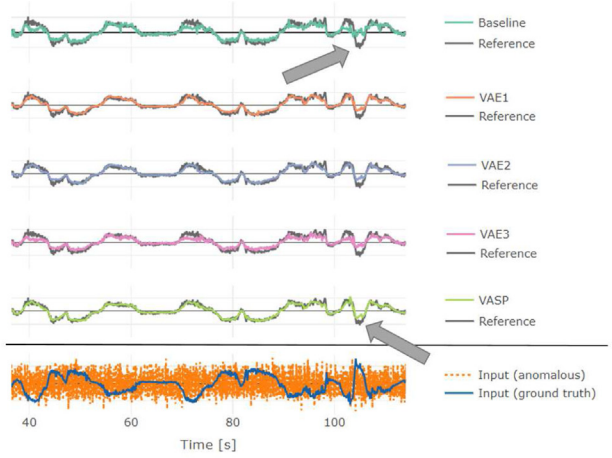
The second column of Table 6 gathers the averaged RMSE over all signals for $AS1$. VASP had the lowest error, followed by Baseline, VAE2, VAE1 and VAE3.

aSlip The models were closer to each other, especially for anomalies in the most important signals (aSteer, nYaw and gLat) — a result not surprising given the physical relation between these signals and aSlip: A vehicle builds up side slip angle when cornering, for which the steering wheel angle, the yaw rate and the lateral acceleration are good indicators. Similar to the case of vLong, VASP was the best performing model. Baseline performed poorly in particular for anomalies in gLat. The trend for VAE1 and VAE2 was again similar to the vLong prediction, for anomalies in important signals they were better than Baseline, however, did not reach the same accuracy in the absence of anomalies. Interestingly, VAE3 performed much better than in the case of vLong prediction.

In the second column of Table 7, the averaged RMSE over all signals for $AS1$ are shown. VASP had the lowest error, followed by Baseline, VAE3, VAE2 and VAE1.



a) **AS1 vLong**: The input signal vCar was disturbed with anomalies. VASP is visually close to the grey reference line, whereas Baseline shows deviations.

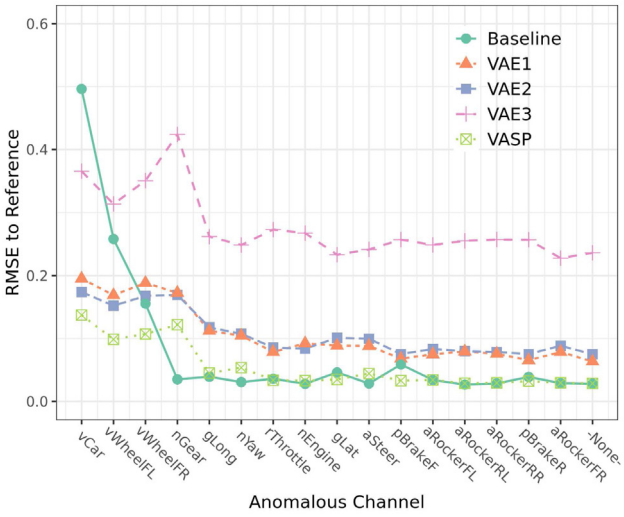


b) **AS1 aSlip**: The input signal nYaw was disturbed with anomalies. The grey arrows show that VASP had still a good accuracy while Baseline could not predict the peaks and dips in the reference signal anymore.

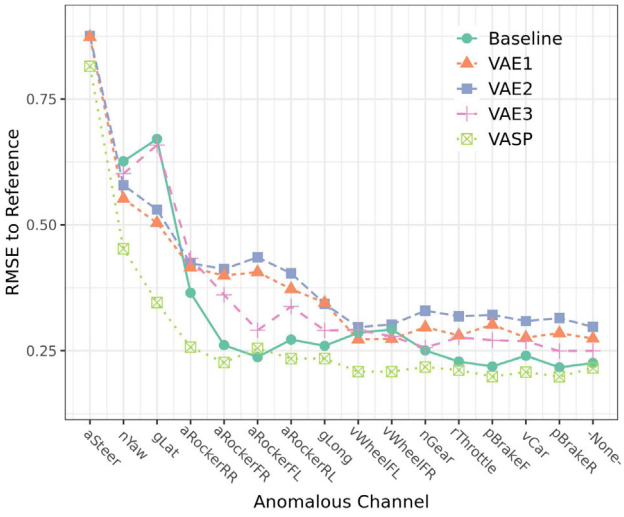
Fig. 17. AS1: Exemplary model comparison in the time domain. The lowest graph in each panel depicts the anomalous as well as the original input signal.

Discussion In general, the VAE models (except VASP) were only better than Baseline if the most important input signals had anomalies. For all other cases, too much relevant information was lost in the VAE. For example a lower rate of compression, i.e. a larger hidden layer z, could decrease the reconstruction error, however, at the cost of reduced robustness with respect to anomalies. The VASP model avoided this issue by selecting the VAE outputs only if the input signals showed anomalies.

An explanation for the difference in performance of VAE3 is that all other models improved significantly for the vLong compared to aSlip prediction. The signal vLong was similar to vCar and it seemed to have a lower rate of change in high frequencies than aSlip, resulting in it being easier to predict. The VAE3 model used a VAE after the prediction, which had a loss function that was optimized to minimize the mean squared error over all signals. As such, it might not have given as much attention to minimizing the already relatively low prediction error for vLong.



a) **AS1 vLong**: The error of Baseline is distinctly higher for anomalies in the first two signals compared to VASP. Only for anomalies in nGear VASP is significantly worse. The other methods show no advantage over VASP.



b) **AS1 aSlip**: VASP is the superior method in almost all cases. However, for anomalies in aSteer, also VASP has a relatively high error. This indicates that this signal could not be reconstructed as well as the other signals by the VAE.

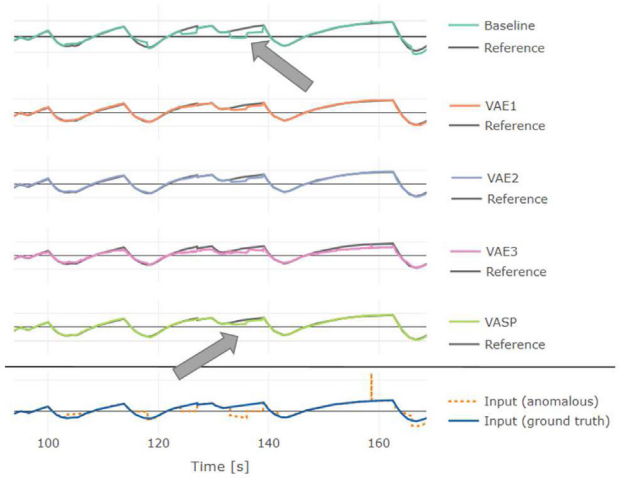
Fig. 18. AS1: RMSE comparison for different anomalous signals. The x-axis label depicts the signal which was anomalous and the y-axis the resulting prediction RMSE to the reference signal. The x-label – None – indicates that no anomalies are present.

4.2.2. AS2

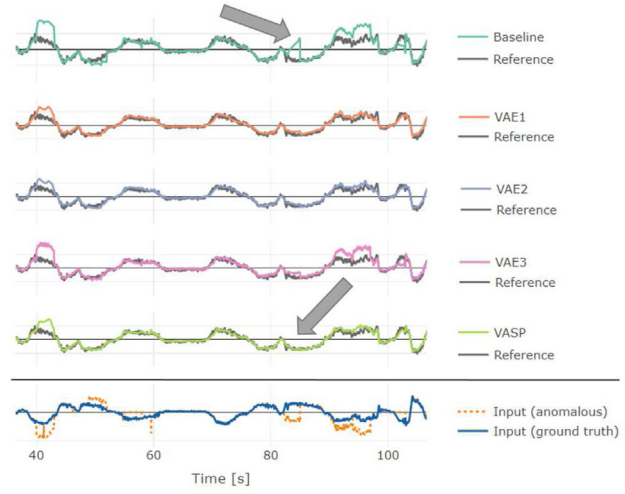
The data set $AS2$ contained anomalies less obvious than $AS1$. Gains and offsets with different strengths were applied. Fig. 19 illustrates examples in the time domain. For both vLong and aSlip the arrows indicate points of large differences between VASP and Baseline. For Baseline an anomaly in nYaw led to a distinct error in the aSlip prediction, whereas the other models' prediction errors were much lower at this point.

Similar to Section 4.2.1 Fig. 20 depicts a comparison of the implemented models.

vLong VAE3 showed the highest error over all anomalous signals. Baseline had a small error for anomalies in unimportant signals, but very large error in the case of important signals being disturbed.

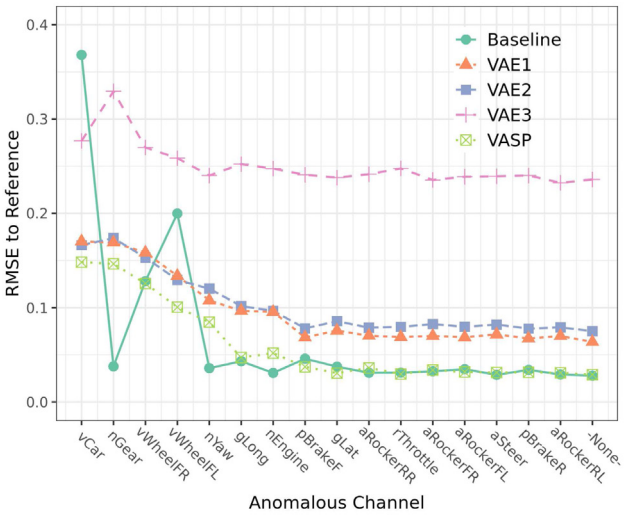


a) **AS2 vLong**: The input signal vCar was disturbed with anomalies. The arrows indicate areas where Baseline suffers much more from the anomalies than VASP.

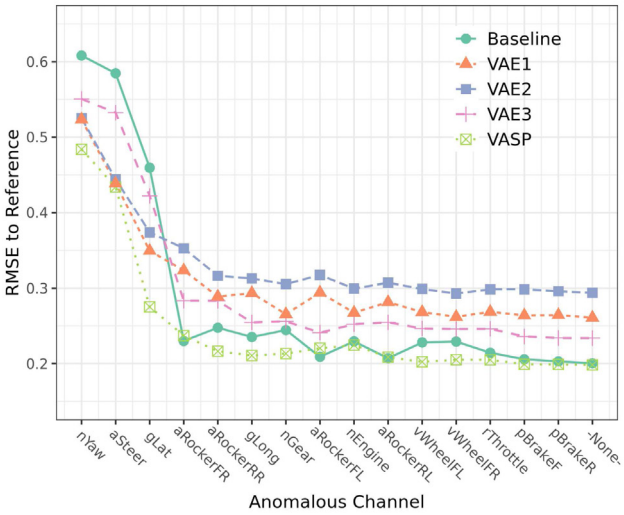


b) **AS2 aSlip**: The input signal nYaw was disturbed with anomalies. The arrows highlight again differences between Baseline and VASP. Whereas VASP was unaffected by the anomalies, Baseline shows significant deviations.

Fig. 19. AS2: Exemplary model comparison in the time domain. The lowest graph in each panel depicts the anomalous as well as the original input signal.



a) **AS2 vLong**: VASP is more consistent than Baseline, especially from vCar to nYaw. Baseline is not affected by anomalies in nGear, there, it has an advantage over VASP. In the other cases, VASP is similar or better.



b) **AS2 aSlip**: VASP has an equal or smaller RMSE than Baseline for all channels. For nYaw, aSteer and gLat, VASP has a clear advantage.

Fig. 20. AS2: RMSE comparison for different anomalous signals. The x-axis label depicts the signal which was anomalous and the y-axis the resulting prediction RMSE to the reference signal. The x-label – None – indicates that no anomalies were present.

VAE1 and VAE2 were more stable, but did not reach the same accuracy as Baseline in the absence of anomalies. VASP was worse than Baseline for anomalies in nGear and nYaw, and performed better or similarly or in all other case.

The third column of Table 6 displays the averaged RMSE over all signals for $AS2$. Again, VASP had the lowest error, followed by Baseline, VAE2, VAE1 and VAE3.

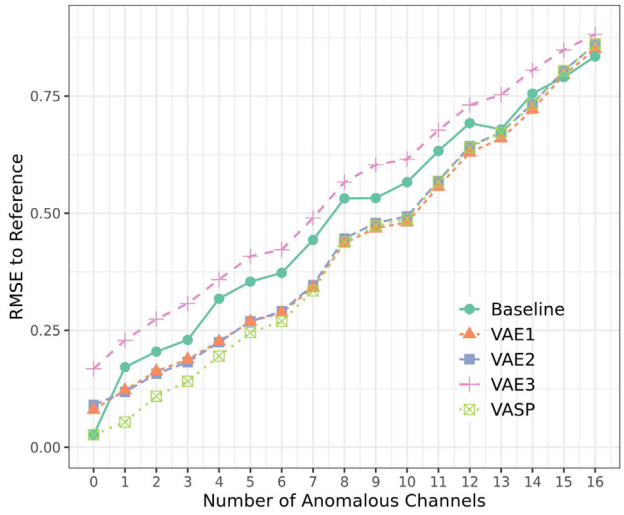
aSlip Here, VASP showed a distinct advantage over all other methods with almost no downsides. Baseline performed particularly worse for anomalies in nYaw, aSteer or gLat. VAE2 was overall better than VAE1, however, in the absence of anomalies both exhibited bad results. Overall, the averaged RMSE in Table 7 produced the following ranking: VASP, Baseline, VAE3, VAE2 and VAE1.

Discussion For both aSlip and vLong the rankings regarding the overall RMSE were equal for $AS1$ and $AS2$, which underlines the superior performance of VASP.

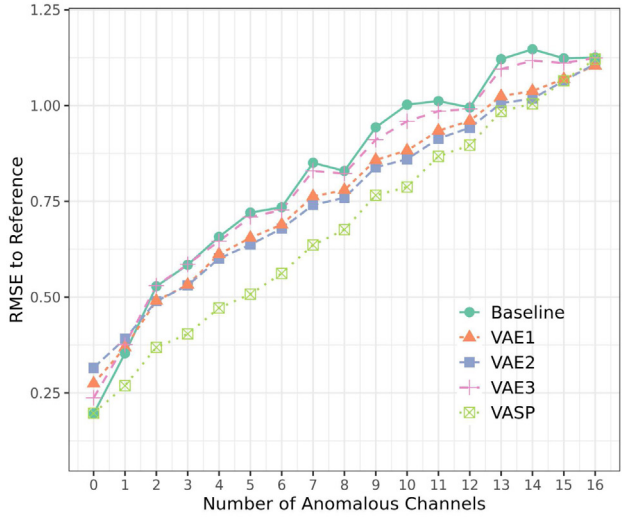
The biggest difference where VASP was worse than Baseline occurred for anomalies in nGear for the vLong prediction. These seemed to affect the VAE methods heavily and Baseline only slightly (see the dip at nGear in Fig. 20 a). nGear and vLong are strongly correlated due to their physical relation (each gear defines a range of possible vehicle speeds through the transmission ratio from the engine to the wheels). The Baseline model did not seem to use this relation, making it unaffected by anomalies in nGear. In contrast, the VAE models made use of this but could not detect that the source of anomalies were only in nGear. Therefore, regular signals were replaced by the VAE outputs, which increased the RMSE in this special case.

4.2.3. AS3

With $AS3$ we proceeded analogously the effect of multiple anomalies. Fig. 21 depicts the number of anomalous signals on the x-axis.



a) **AS3 vLong**: For 0 anomalies, Baseline is as good as VASP, then Baseline falls behind significantly. VASP is superior to the other methods until 6 anomalous channels and stays very close to the best until 16.



b) **AS3 aSlip**: Again, VASP is equal to Baseline for 0 anomalies, then it has a distinct advantage over all other methods until the number of anomalies reaches 12.

Fig. 21. AS3: RMSE comparison for an increasing number of anomalous signals. The x-axis depicts the number of anomalous signals and the y-axis the averaged prediction RMSE to the reference signal.

As described in Section 3.6, the RMSE on the y-axis is obtained by averaging over all drawn instances from the sample space.

vLong For all models the RMSE rose with increasing number of anomalous signals. VAE3 produced the highest error for all cases. Baseline showed a steep slope from 0 to 1 anomaly and performed visibly worse than VASP until 13 anomalies. VAE1 and VAE2 were very similar to each other and performed worse than VASP from 0 to 7 anomalies. For a higher number of anomalies the models performed similarly. VASP had the lowest error from 0 to 7 anomalies.

Overall, Table 6 shows that VASP performed best, followed by VAE2, VAE1, Baseline and VAE3.

aSlip Similar to vLong, the RMSE increased with increasing number of anomalous signals. Baseline and VAE3 performed similarly, both were the worst models for two anomalies or more. VAE1 and VAE2 had a lower error from 2 anomalies onward. The best model was again VASP, with a clear advantage until 13 anomalies.

Discussion For both vLong and aSlip, VASP was overall the best model. The advantage was larger for aSlip than for vLong. Especially in the case of few anomalies, it performed clearly better than the other models. Notice that this is also the most important case for the application described in this paper since multiple sensor failures are rare, not to mention the fact that in this case the data might be excluded for further analysis.

Overall, the VAE-based methods were more robust than Baseline. However, Baseline was more accurate for no or unimportant anomalies. For both investigated signals and all three anomaly simulation approaches, the simple selective procedure of VASP was able to combine both advantages which led to its superior overall performance.

5. Conclusion

Our proposed VASP framework for robust time series prediction outperformed the other investigated approaches on our data set. The selective prediction strategy worked well if anomalies were present and suffered from almost no disadvantages. What is more, the VASP framework can easily be integrated into existing models. Given a prediction model, it can be upgraded to the VASP framework by adding a VAE and anomaly detector. The prediction model does not have to be retrained and labeled anomalies are not needed, those are the main advantages.

Our approach is currently being implemented in motorsport practice. To the best of the authors' knowledge, machine learning models are rarely used in this area, because of a lack of interpretability and uncertainty in model behavior. The aim of study is a robustification of machine learning models, which is one of the key features relevant for usage by experts in the field.

One line of future research is to apply our method to different datasets from motorsport and other areas. Moreover, the influence of the properties of the prediction signal should be investigated in more detail, since the difference in prediction accuracy between the two signals from this work were significant.

CRediT authorship contribution statement

Julian von Schleinitz: Writing, Editing, Methodology, Software. **Michael Graf:** Writing - review & editing, Supervision, Conceptualization. **Wolfgang Trutschnig:** Writing - review & editing, Supervision, Conceptualization. **Andreas Schröder:** Writing - review & editing, Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Allaire, J.J., Chollet, F., 2018. Keras: R interface to 'keras'. https://CRAN.R-project.org/package=keras.

Allaire, J.J., Tang, Y., 2018. Tensorflow: R interface to 'tensorflow'. https://CRAN.R-project.org/package=tensorflow.

Altan, A., Karasu, S., Bekiros, S., 2019. Digital currency forecasting with chaotic meta-heuristic bio-inspired signal processing techniques. Chaos Solitons Fractals 126 (2), 325–336. http://dx.doi.org/10.1016/j.chaos.2019.07.011.

Altan, A., Karasu, S., Zio, E., 2021. A new hybrid model for wind speed forecasting combining long short-term memory neural network, decomposition methods and grey wolf optimizer. Appl. Soft Comput. 100 (4), 106996. http://dx.doi.org/10.1016/j.asoc.2020.106996.

An, J., Cho, S., 2015. Variational autoencoder based anomaly detection using reconstruction probability. In: Special Lecture on IE. (2015–2).

Bao, W., Yue, J., Rao, Y., 2017. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PLoS One 12 (7), e0180944. http://dx.doi.org/10.1371/journal.pone.0180944.

Borghesi, A., Bartolini, A., Lombardi, M., Milano, M., Benini, L., 2019. A semisupervised autoencoder-based approach for anomaly detection in high performance computing systems. Eng. Appl. Artif. Intell. 85, 634–644. http://dx.doi.org/10.1016/j.engappai.2019.07.008.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32. http://dx.doi.org/10.1023/A:1010933404324.

Chen, R.-Q., Shi, G.-H., Zhao, W.-L., Liang, C.-H., 2019a. A joint model for anomaly detection and trend prediction on it operation series. http://arxiv.org/pdf/1910.03818v3.

Chen, R.-Q., Shi, G.-H., Zhao, W.-L., Liang, C.-H., 2019b. Sequential VAE-LSTM for anomaly detection on time series. ArXiv Preprint arXiv:1910.03818.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734.

Chollet, F., et al., 2015. Keras.

Essien, A., Giannetti, C., 2019. A deep learning framework for univariate time series prediction using convolutional LSTM stacked autoencoders. In: 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA). IEEE, pp. 1–6. http://dx.doi.org/10.1109/INISTA.2019.8778417.

Fisher, A., Rudin, C., Dominici, F., 2018. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. J. Mach. Learn. Res. 20 (177), http://arxiv.org/pdf/1801.01489v5.

Gao, Z., Ma, C., Luo, Y., Liu, Z., 2018. IMA health state evaluation using deep feature learning with quantum neural network. Eng. Appl. Artif. Intell. 76, 119–129. http://dx.doi.org/10.1016/j.engappai.2018.08.013.

Goldstein, M., Uchida, S., 2016. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PLOS ONE 11 (4), 1–31. http://dx.doi.org/10.1371/journal.pone.0152173.

Graves, A., Schmidhuber, J., 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw.: Off. J. Int. Neural Netw. Soc. 18 (5–6), 602–610. http://dx.doi.org/10.1016/j.neunet.2005.06.042.

Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J., 2017. LSTM: A search space odyssey. IEEE Trans. Neural Netw. Learn. Syst. 28 (10), 2222–2232. http://dx.doi.org/10.1109/TNNLS.2016.2582924, http://arxiv.org/pdf/1503.04069v2.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Karasu, S., Altan, A., Bekiros, S., Ahmad, W., 2020. A new forecasting model with wrapper-based feature selection approach using multi-objective optimization technique for chaotic crude oil time series. Energy 212 (4), 118750. http://dx.doi.org/10.1016/j.energy.2020.118750.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. http://arxiv.org/pdf/1412.6980v9.

Kingma, D.P., Welling, M., 2013. Auto-encoding variational Bayes. http://arxiv.org/pdf/1312.6114v10.

Kingma, D.P., Welling, M., 2019. An introduction to variational autoencoders. FNT Mach. Learn. 12 (4), 307–392. http://dx.doi.org/10.1561/2200000056, http://arxiv.org/pdf/1906.02691v3.

Krstanovic, S., Paulheim, H., 2017. Ensembles of recurrent neural networks for robust time series forecasting. In: Bramer, M., Petridis, M. (Eds.), Artificial Intelligence XXXIV. Springer International Publishing, Cham, pp. 34–46.

Lee, S., Kwak, M., Tsui, K.-L., Kim, S.B., 2019. Process monitoring using variational autoencoder for high-dimensional nonlinear processes. Eng. Appl. Artif. Intell. 83, 13–27. http://dx.doi.org/10.1016/j.engappai.2019.04.013.

Li, Y., Zhu, Z., Kong, D., Han, H., Zhao, Y., 2019. EA-LSTM: Evolutionary attention-based LSTM for time series prediction. Knowl.-Based Syst. 181, 104785. http://dx.doi.org/10.1016/j.knosys.2019.05.028.

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G.M., 2016. LSTM-based encoder-decoder for multi-sensor anomaly detection. CoRR abs/1607.00148.

Marchi, E., Vesperini, F., Eyben, F., Squartini, S., Schuller, B., 2015a. A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks. In: Proceedings 40th IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2015, IEEE, Institute of Electrical and Electronics Engineers and IEEE Signal Processing Society and IEEE International Conference on Acoustics, Speech and Signal Processing and ICASSP, Piscataway, NJ, http://ieeexplore.ieee.org/servlet/opac?punumber=7158221.

Marchi, E., Vesperini, F., Eyben, F., Squartini, S., Schuller, B., 2015b. A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1996–2000. http://dx.doi.org/10.1109/ICASSP.2015.7178320.

Park, D., Hoshi, Y., Kemp, C.C., 2017. A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder. http://arxiv.org/pdf/1711.00614v1.

Principi, E., Vesperini, F., Squartini, S., Piazza, F., 2017. Acoustic novelty detection with adversarial autoencoders. In: 2017 International Joint Conference on Neural Networks (IJCNN). pp. 3324–3330. http://dx.doi.org/10.1109/IJCNN.2017.7966273.

R Development Core Team, 2008. R: A language and environment for statistical computing. http://www.R-project.org.

Saito, T., Rehmsmeier, M., 2017. Precrec: fast and accurate precision-recall and ROC curve calculations in r. Bioinformatics (Oxford, England) 33 (1), 145–147. http://dx.doi.org/10.1093/bioinformatics/btw570.

Sakurada, M., Yairi, T., 2014. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis 2014. pp. 4–11. http://dx.doi.org/10.1145/2689746.2689747.

Schleinitz, J.v., Wörle, L., Graf, M., Schröder, A., Trutschnig, W., 2019. Analysis of race car drivers' pedal interactions by means of supervised learning. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC). pp. 4152–4157. http://dx.doi.org/10.1109/ITSC.2019.8917120.

Shen, C., Qi, Y., Wang, J., Cai, G., Zhu, Z., 2018. An automatic and robust features learning method for rotating machinery fault diagnosis based on contractive autoencoder. Eng. Appl. Artif. Intell. 76, 170–184. http://dx.doi.org/10.1016/j.engappai.2018.09.010.

van Houdt, G., Mosquera, C., Nápoles, G., 2020. A review on the long short-term memory model. Artif. Intell. Rev. 53 (8), 5929–5955. http://dx.doi.org/10.1007/s10462-020-09838-1.

Wielgosz, M., Mertik, M., Skoczeń, A., de Matteis, E., 2018. The model of an anomaly detector for HiLumi LHC magnets based on recurrent neural networks and adaptive quantization. Eng. Appl. Artif. Intell. 74, 166–185. http://dx.doi.org/10.1016/j.engappai.2018.06.012.

Williams, R.J., Zipser, D., 1989. A learning algorithm for continually running fully recurrent neural networks. Neural Comput. 1 (2), 270–280.

Wörle, L., Graf, M., Eichberger, A., 2018. Objective metrics for control inputs of racecar drivers. In: Fisita F2018/F2018-VDY-050.

Wörle, L., Schleinitz, J.v., Graf, M., Eichberger, A., 2019. Driver detection from objective criteria describing the driving style of race car drivers. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC). pp. 1198–1203. http://dx.doi.org/10.1109/ITSC.2019.8917325.

Zhang, C., Chen, Y., 2019. Time series anomaly detection with variational autoencoders. http://arxiv.org/pdf/1907.01702v1.