A hybrid virtual sensing approach for approximating non-linear dynamic system behavior using LSTM networks

L. Heindel, P. Hantschke, M. Kästner

"Technische Universität Dresden, Institute of Solid Mechanics, 01062 Dresden, Germany
"Dresden Center for Fatigue and Reliability (DCFR), 01062 Dresden, Germany

Abstract

Modern Internet of Things solutions are used in a variety of different areas, ranging from connected vehicles and healthcare to industrial applications. They rely on a large amount of interconnected sensors, which can lead to both technical and economical challenges. Virtual sensing techniques aim to reduce the number of physical sensors in a system by using data from available measurements to estimate additional unknown quantities of interest. Successful model-based solutions include Kalman filters or the combination of finite element models and modal analysis, while many data-driven methods rely on machine learning algorithms.

The presented hybrid virtual sensing approach combines Long Short-Term Memory networks with frequency response function models in order to estimate the behavior of non-linear dynamic systems with multiple input and output channels. Network training and prediction make use of short signal subsequences, which are later recombined by applying a windowing technique. The frequency response function model acts as a baseline estimate which perfectly captures linear dynamic systems and is augmented by the non-linear Long Short-Term Memory network following two different hybrid modeling strategies.

The approach is tested using a non-linear experimental dataset, which results from measurements of a three-component servo-hydraulic fatigue test bench. A variety of metrics in time and frequency domains, as well as fatigue strength under variable amplitudes are used to evaluate the approximation quality of the proposed method. In addition to virtual sensing, the algorithm is also applied to a forward prediction task. Synthetic data are used in a separate study to estimate the prediction quality on datasets of different size.

Keywords: virtual sensing, LSTM, non-linear dynamic systems, hybrid modeling, forward prediction

1. Introduction

Sensors are fundamental to all measuring systems in engineering, ranging from small scale experimental setups in research to large scale production surveillance. Depending on the application, difficulties arise if important locations in a system are inaccessible, impractical to instrument or if the costs of the respective sensors are too high. Since modern Internet of Things (IoT) approaches heavily rely on low cost hardware, it is also highly desirable to extract as much information as possible from a given sensor setup. The aim of virtual sensing (VS) is to approximate unmeasured physical quantities in a system using existing sensor information. This can be especially beneficial in applications where sensors are very expensive or their application is difficult. Existing frameworks for VS can mainly be divided into two categories, namely analytical and data-driven approaches. While the former rely on a model of the dynamic system which provides a physically grounded relationship of input and output quantities, the latter generate approximations using available measurement data.

A variety of analytical approaches for VS in linear dynamic systems exist in the field of structural monitoring. Here, modal analysis can be used in conjunction with a finite element (FE) model, since the structural
response of linear systems can be expressed as a sum of modal contributions. This approach was successfully used to estimate stress histories in fatigue sensitive joints from acceleration measurements in Hjelm et al. [1], to provide fatigue monitoring at inaccessible locations of offshore wind turbines [2] and to predict the full-field response of a system by modeling either the entire structure [3] or only parts of interest [4]. Different VS approaches based on Kalman Filters are demonstrated in [5] and [6] which require complete or partial information about the dynamic system matrices. In many situations, an FE solution might not be available due to insufficient information about the systems geometry and material characteristics. Furthermore, the model of a linear system can be insufficient for a wide range of applications which feature non-linear system characteristics, e.g., non-linear stiffness, kinematics or friction. In these cases, data-driven approaches can be used to implicitly approximate the system using data from sensor measurements.

Data-driven VS approaches can be further subdivided into white and black box models, where an interpretation of the model parameters is available in the former and difficult or impossible in the latter case. Frequency response function (FRF) based approaches may provide white box models for discrete systems. While they act as black box models in the continuum case, they perfectly approximate the relationship between measurements in linear systems, see Natke [7]. Data-driven VS can also provide more accurate estimates of noisy sensor data as demonstrated in Kullaa [3, 4]. In recent years, various Machine Learning algorithms have been used to create data-driven black box models for virtual sensing. In chemical process monitoring, Long Short-Term Memory (LSTM) networks were applied in [8, 9], while [10] compares recurrent neural networks and LSTM networks and features an approach for transfer learning. Regarding VS in structural mechanics, Rouss et al. [11] employs a non-linear autoregressive exogenous model (NARX) for response estimation in a non-linear dynamic system while [12] applies LSTM networks to model linear time varying systems.

The aim of this paper is to provide a black box virtual sensing framework based on LSTM networks which is suitable for multiaxial fatigue applications in non-linear mechanical systems. Special attention must therefore be paid to the error metrics of the VS approximations necessary for fatigue analyses, e.g., multiaxial rainflow counting [13]. Since LSTM networks rely on a dataset of example inputs and outputs to approximate functions, they can be combined with existing approaches like FRF models, by incorporating their predictions into this training dataset. Similar approaches are already used for lifetime predictions in [14]. As a result, this framework is very suited for hybrid modeling. Since both FRF and LSTM models provide estimates of sensor data using measurements, they can also be employed to predict system responses from system actuation data, which is referred to as forward prediction (FP) in fatigue analysis.

This paper is structured as follows: In Section 2, important fundamentals regarding frequency response function models and LSTM networks will be reviewed. Signal prediction as a combination of short subsequences is described and different hybrid modeling approaches are proposed. In Section 3, a variety of error metrics are introduced and the approach is tested on experimental data originating from a fatigue test bench. The paper concludes with a discussion of the results and an outlook in Section 4.

2. Model setup

System responses are of interest in many cases where no system descriptions are available. Unknown parameters, complex physical correlations or even time and cost consumption lead to a demand for generally applicable algorithms based on measurement data. Since 1976, FRF-Models are used for test rig descriptions as shown in Cryer et al. [15]. Including further developments they are state of the art, up to now [16]. In this paper, vectors and matrices are indicated using single and double underscores respectively, e.g., \( \mathbf{v} \) and \( \mathbf{M} \). Whenever a scalar operator is applied to a vector, it represents an element-wise application of that operator.

2.1. Frequency response function model

A dynamic system with multiple inputs as well as multiple outputs (MIMO) is characterized by a set of input and output channels. Using \( x_k(t) \) to denote a physical quantity measured for the \( k \)-th input channel
at time $t$ and $y_l(t)$ to identify the corresponding measurement for the $l$-th output channel, their relation in the time domain is given by

$$y(t) = g(t) \ast x(t)$$  \hspace{1cm} (1)

with $g(t)$ being the weight function matrix. Unfortunately, measuring the components of the weight function matrix directly, by applying an impulse excitation with sufficient energy content at the input channels, would severely damage the system in many practical applications. Moreover, the calculation of the convolution operation ($\ast$) using Duhamel’s integral is very time consuming. It is therefore common to compute the frequency response function matrix $H(j\omega)$ in the frequency domain

$$Y(j\omega) = H(j\omega)X(j\omega),$$  \hspace{1cm} (2)

where $X(j\omega) = \mathcal{F}(x(t))$ and $Y(j\omega) = \mathcal{F}(y(t))$ are the Fourier transforms of the input and output channel vectors. The elements of $H(j\omega)$ can be estimated as

$$H_{kl}(j\omega_n) = \frac{\overline{S}_{kl}(j\omega_n)}{\overline{S}_{kk}(j\omega_n)}$$  \hspace{1cm} (3)

where the effective power spectral density (PSD) $\overline{S}_{kk}$ and effective cross power spectral density $\overline{S}_{kl}$ can be computed as

$$\overline{S}_{kk}(j\omega_n) = \frac{1}{T \cdot M} \sum_{m=1}^{M} X_{m,k}(j\omega_n)X_{m,k}(j\omega_n)$$  \hspace{1cm} (4)

$$\overline{S}_{kl}(j\omega_n) = \frac{1}{T \cdot M} \sum_{m=1}^{M} X_{m,k}(j\omega_n)Y_{m,l}(j\omega_n)$$  \hspace{1cm} (5)

by averaging over the $M$ signal windows of size $T$ for each discrete $\omega_n$ obtained from the Fast Fourier Transform (FFT). This procedure allows for a system characterization using a white-pink-noise excitation. For further details, the reader is referred to [7], [17] and [16].

After parameterization, the frequency response function matrix can be used as a predictive model which computes output channel signals perfectly from known input channel information if the dynamic system is linear and time invariant. In the single input, single output (SISO) case, this method can also be extended to non-linear systems by a system linearization, see [16], or the usage of multiple models at different support points of a piecewise linearization. In the MIMO case, however, the total number of support points increases exponentially as the non-linear characteristic of each channel combination can be different, making the parameterization and application of this approach very challenging.

### 2.2. LSTM prediction and windowing

The Long Short-Term Memory network, introduced in Hochreiter and Schmidhuber [18, 19], is a type of artificial neural network used for sequential data processing. Its gated structure was designed to overcome the vanishing gradient problem of previous recurrent neural networks. As a result, these networks can be trained more efficiently and are generally better at memorizing relationships over long time periods. This section only provides a short overview of the LSTM network algorithm. For more detailed information, the reader is referred to the aforementioned works.

An LSTM network is composed of one or multiple memory blocks. Each memory block contains a number of memory cells, where $c_i(t_i)$ denotes the vector of inner cell states at time step $t_i$ with $i = 1 \ldots L$. Figure 1 provides an overview of the data processing steps that occur in a single memory block.

At the beginning of each prediction, the cell states are initialized to zero. The LSTM model processes the input channel values $x(t_i)$ in sequence at discrete points in time, starting at $x(t_1)$ and incrementing $i$ until the terminal input $x(t_L)$ of the sequence with length $L$ is reached. The block input $x(t_i)$ is concatenated
Figure 1: LSTM networks are composed of memory blocks, which process information using the inner cell state $c_t$. This inner state can be updated, forgotten or used to create the block output $h_t$ depending on the block input at the current time step $x_{ti}$ and the previous output $h_{ti-1}$. All operations related to the inner state are carried out by respective gates $G$ and the network $N$, whose parameters are learned from a dataset during training.

with the output of the previous timestep $h_{ti-1}$ to provide input data for the gate networks $G_{store}$, $G_{forget}$, $G_{out}$ and the input network $N_{in}$. Each gate network uses the sigmoid activation function $\sigma(x) = 1/(1+e^{-x})$ and provides an output

$$g_{store} (ti) = \sigma( W_{x_{store}}x(t_i) + W_{h_{store}}h(t_{i-1}) + b_{store})$$

$$g_{out} (ti) = \sigma( W_{x_{out}}x(t_i) + W_{h_{out}}h(t_{i-1}) + b_{out})$$

$$g_{forget} (ti) = \sigma( W_{x_{forget}}x(t_i) + W_{h_{forget}}h(t_{i-1}) + b_{forget})$$

of the respective gate in the range $(0,1)$, while the input network

$$a_{in} (ti) = tanh( W_{x_{in}}x(t_i) + W_{h_{in}}h(t_{i-1}) + b_{in})$$

uses the tanh activation function. The parameters of each network are given by the weight matrices $W$ and the bias vector $b$, which are initialized randomly. The memory cell state of the current time step

$$c(t_i) = c(t_{i-1}) \odot g_{forget} (ti) + a_{in} (ti) \odot g_{store} (ti)$$

as well as the block output

$$h(t_i) = tanh(c(t_i)) \odot g_{out} (ti)$$

are now computed using the Hadamard product $\odot$ for element-wise multiplication. Afterwards, this process is repeated for the next time step. In architectures with multiple LSTM blocks, the output vector $h_t$ of one block is used as the input $x$ of the next block. After the final block, a single fully connected layer

$$y^*(t_i) = W_{FC}h(t_i) + b_{FC}$$

with weights $W_{FC}$, bias $b_{FC}$ and no activation function is used to generate the network prediction $y^*$. 

In this work, the LSTM network is implemented using the Python libraries Tensorflow [20] and Keras [21]. As a preprocessing step, all training input and output data is collectively standardized to a mean of zero.
and a standard deviation of one in each channel. During training, the input sequence is used to generate a prediction

\[ y^* (t_i) = \text{LSTM} (x (t_i)) , \]  

which is then compared to the true output channel values using the mean squared error loss function

\[ E_{\text{MSE}} = \frac{1}{L} \sum_{i=1}^{L} (y (t_i) - y^* (t_i))^2 . \]

The optimization algorithm RMSProp \cite{Huang2015} updates the weights and biases after each mini-batch of training data in order to minimize the loss function. The rate of change of these updates is determined by the learning rate hyper-parameter \( \lambda \). Training proceeds for a given number of training dataset repetitions called epochs and the assignment of data samples to mini-batches is randomized after each epoch.

In order to ensure an efficient training process, all processed sequences are required to have the same length. The application of LSTM networks to measurement data from sensors is therefore carried out on short subsequences with a fixed length \( L \). The extraction of subsequences from the dataset is rather straightforward. Starting at the first time sample \( t_i = t_1 \) of each measurement data file, data in the range \( t \in [t_i, t_i + L - 1] \) is extracted and the starting position of the next subsequence is given by

\[ t_i \leftarrow t_i + o \cdot L, \]  

where the overlap factor \( o \) determines the number of shared time samples between neighboring subsequences. Both training and prediction of the LSTM network are carried out on the subsequence level. In order to achieve a continuous prediction \( y_{\text{comb}}^* \) for a complete measurement file, the \( n_{\text{sub}} \) individual subsequence predictions \( y_{\text{sub}}^* \) are combined as a weighted sum

\[ y_{\text{comb}}^* = \sum_{i=1}^{n_{\text{sub}}} y_{\text{sub}}^* \odot w_{\text{sub}} \]  

using window functions \( w_{\text{sub}} \), which are non-zero only in the range of the corresponding subsequence. The window functions are also weighted to ensure the partition of unity property

\[ \sum_{i=1}^{n_{\text{sub}}} w_{\text{sub}} = 1 \]

is fulfilled at every time step. Figure 2 illustrates the subsequence combination process for a single output channel.

In this paper, the Welch window function

\[ w_{\text{sub}} (t) = 1 - \left( \frac{2t - L}{L} \right)^2 \]  

is used to interpolate between individual predictions. It is additionally raised to the power of 10 to provide smooth transitions. As a result of this weighting scheme, the first and last 15% of the data in each subsequence have a very low impact on the combined prediction. A special treatment is used to extend this approach to the beginning of the measurement process. Here, an additional subsequence is utilized, which spans from \( t = -L/2 \) to \( t = L/2 \). For all negative points in time, the unmeasured input channel data is assumed to be equal to the corresponding measurements at the starting time \( t_1 \). A prediction is generated for this subsequence and included into the weighting process. Finally, all negative points in time are discarded from the resulting combined prediction.

5
Figure 2: Individual subsequence predictions (top) are combined as a weighted sum using corresponding window functions (center). As a consequence, predictions for long sequences (bottom) resulting from measurement data can be realized.
2.3. Hybrid modeling strategies

The FRF model excels at predicting linear system behavior, while the LSTM network can be trained to approximate arbitrary non-linear time-dependent functions. The aim of a hybrid model is to combine both methods in order to exploit their respective strength while minimizing their downsides. In this paper, two different approaches to hybrid modeling are compared.

In the first approach, denoted by hybrid 1, the FRF model is used to generate a prediction from the input channel values \( x \). This prediction is then subtracted from the true output channel values \( y \) in the dataset

\[
e (t) = y(t) - \text{FRF}(x(t))
\]  

(19)

in order to obtain the model error \( e \) of the linear FRF prediction. This error data can now be used as output channel values during the training process of an LSTM network. Since the trained network now provides an estimate

\[
e^*(t) = \text{LSTM}(x(t))
\]  

(20)

for the FRF model error, the sum

\[
y^*_1(t) = \text{FRF}(x(t)) + \text{LSTM}(x(t))
\]  

(21)

yields the hybrid prediction of the model hybrid 1.

The second approach, denoted by hybrid 2, is designed to provide the LSTM network with more information. Again, the FRF model is used to create a prediction, which is now concatenated to the original input data during LSTM training. This way, the training dataset already contains the FRF prediction as a baseline solution. The hybrid prediction of hybrid 2

\[
y^*_2(t) = \text{LSTM}(x(t), \text{FRF}(x(t)))
\]  

(22)

can now simply be a copy of the FRF prediction or a modification by the LSTM, depending on the input channel data \( x \).

3. Experiments

VS and FP achieve substantial benefits in experimental fatigue tests under multiaxial variable amplitude loading. Predictive maintenance approaches also require an accurate sensor signal prediction from easily accessible measurements, but lead to larger scattering of the resulting data caused by uncertain excitations. Therefore, a real live measurement setup of a fatigue test is chosen in this work to generate a reproducible and large dataset for further research.

3.1. Experimental setup and data

In order to validate the proposed approach, experimental data is collected from the three-component servo-hydraulic fatigue test bench for suspension hydro-mounts, depicted in Figure 3. This system features a variety of non-linearities. The hydro-mounts are filled with oil to provide highly non-linear damping, while the pendulum kinematics of the setup introduce non-linear interactions of the excitations in different spatial directions. The most influential non-linearity originates from the system stiffness. In order to capture the static force-displacement-relationship, each channel is loaded and measured individually, while the two remaining channels are load controlled at a force of zero. The resulting static characteristics, depicted in Figure 4, show a friction induced hysteresis with significant non-linearities near the upper and lower reversal points. Apart from the hysteresis, the stiffness is nearly constant in the central regions of parameter space. The FRF model is parameterized using only noise data from this region in order to provide the best linear approximation to the overall system behavior.

The measured dataset contains a large collection of system responses that result from different excitations, which are sampled with a frequency of 1 kHz. It can be subdivided into 1 h 53 min of uncorrelated noise
Figure 3: A three-component servo-hydraulic test bench (a) for the fatigue assessment of suspension hydro-mounts (b) is used to generate an experimental dataset. It is equipped with 3 inertia compensated force and 3 displacement sensors.

Figure 4: The static characteristics of each spatial direction form a friction induced hysteresis. While the stiffness is almost constant in large regions of the parameter space, depicted in green color, it changes significantly in the vicinity of the reversal points as indicated by the yellow and orange areas. Only data from regions of constant stiffness is used in the parameterization of the FRF model.
Figure 5: The experimental dataset contains a large amount of measured noise time series (a), which are used to train LSTM networks and parameterize FRF models. For validation and testing, mainly service load data files (b) are used to emulate fatigue testing conditions.
signals, 2 h 37 min of fatigue service loads whose input signals are rescaled from three independent service load shapes, 20 min of sinusoidal excitations and 20 min of sweep. Figure 5 shows a time series example for noise and service load data, respectively. In order to ensure that the system characteristics are constant over the duration of all measurements, the entire testing program was repeated once, resulting in identical responses according to the sensor accuracy. A subsequently performed data postprocessing only included a low pass FFT-filtering at 80 Hz. The remaining signal contains the complete controllable frequency range of the used test setup.

For the purpose of LSTM network parameterization, the dataset is split into three parts, namely training, validation and test data. Only the training dataset is directly used to update the network weights, while the validation dataset is used to determine the best choice of hyper-parameters of the LSTM architecture and subsequence windowing process. The test dataset is never used during parameterization in order to enable a completely independent evaluation of the predictive performance for each model. From a fatigue analysis point of view, it is very desirable to parameterize predictive models using noise signals exclusively, as they contain a lot of information on the system, while causing significantly less damage to the test specimen if compared to service loads. For this reason, the majority of the noise data is used for training while only a small portion is left to assess the noise prediction quality during validation and testing. The service load data on the other hand is only used during validation and testing. Here, two of the three basic service load signal shapes are assigned for validation and the remaining one is reserved for testing to ensure that both datasets are completely independent. The composition of the dataset is specified in more detail in Table 1.

### 3.2. Error metrics

In order to evaluate the prediction quality of the models, a variety of error metrics are taken into account. The Root Mean Square (RMS) error

\[
\text{RMS}(y^*, y) = \sqrt{\frac{\sum_{t=1}^{T} (y(t) - y^*(t))^2}{\sum_{t=1}^{T} y(t)^2}}
\]  

provides a general measure of how well the signal shape of a particular channel prediction \(y^*\) corresponds to the target \(y\), averaged over all \(T\) time steps of the series. Similarly, the power spectral density RMS error

\[
\text{RMS}_{\text{PSD}}(y^*, y) = \text{RMS}(S_{kk}(y^*), S_{kk}(y))
\]  

uses the PSD \(S_{kk}\) introduced in Subsection 2.1 to evaluate the prediction quality of a signal channel in the frequency domain, averaged over the corresponding frequencies \(\omega_n\).

Both of these error metrics do not provide information about the approximation quality of the signals’ global extrema, which are of high importance for accurate fatigue damage predictions. For this reason, fictitious fatigue damage calculations according to the nominal stress concept \[23\] are performed for both prediction and target signal. Here, a fictitious Wöhler curve

\[
N = K \cdot S_a^{-k}
\]  

relates between the load amplitude \(S_a\) and the number of load cycles before component failure denoted by \(N\), with \(k = 5\) and \(K = 10^7\). In addition, the 4-point Rainflow counting algorithm [24] and the elementary

| Data type      | Runtime (total) | Proportional usage | Test |
|----------------|-----------------|--------------------|------|
| Noise          | 1 h 53 min      | 85%                | 6%   | 9%   |
| Service load   | 2 h 37 min      | 0%                 | 72%  | 28%  |
| Sinus          | 20 min          | 0%                 | 80%  | 20%  |
| Sweep          | 20 min          | 0%                 | 80%  | 20%  |
Figure 6: The proposed hybrid model can be applied to different signal estimation problems. In a virtual sensing task (blue), one or more output sensors are estimated from the remaining measurements. In forward prediction (orange), the output sensor data is estimated based on the drive signal, which controls the system excitation. The assignment of input and output data during model parameterization changes depending on the use case.

Palmgren-Miner rule [25] are used to calculate a fictitious accumulated fatigue damage $d$ for the prediction and target signal, respectively. The metric of the damage ratio

$$\text{damage}(y^*, y) = \frac{d(y^*)}{d(y)}$$

therefore informs about the relative error between predicted and target fatigue damage, introduced by the virtual sensing model. The Multi-Rain fatigue damage generalizes the fatigue damage accumulation to multiple spatial directions, see Beste et al. [13]. This necessity arises as a result of multiaxial stress states following from multiaxial component loading. For a given direction $\psi$, specified by the components $\psi_x$, $\psi_y$ and $\psi_z$ of its unit direction vector satisfying

$$\psi_x^2 + \psi_y^2 + \psi_z^2 = 1,$$

the fictitious damage in this direction can be computed using a weighted sum

$$d_\psi(s) = d(\psi_x s_x + \psi_y s_y + \psi_z s_z)$$

of the original signal channels $s_x$, $s_y$ and $s_z$ for arbitrary signals $s$. The Multi-Rain damage ratio

$$\text{damage}_{\text{MR}}(y^*, y) = \frac{\max (d_\psi(y^*))}{\max (d_\psi(y))}$$

follows by comparing the maximum damages of prediction and target, where 500 uniformly distributed spatial directions $\psi$ are considered in each case.

3.3. Virtual sensing evaluation

In the virtual sensing experiment, the three displacement sensor measurements are used to predict the three channels of force data as shown in Figure 6. Using this dataset, a comparison is drawn between the FRF model described in Subsection 2.1, a pure LSTM network and both hybrid models introduced in Subsection 2.3. As noted in Subsection 3.1, the FRF model is parameterized using only data from regions where the system stiffness is nearly constant, while the complete training dataset was used for the LSTM and hybrid models. The network hyper-parameters were chosen after conducting multiple large scale automated parameter studies on a high performance computing cluster. Due to the dataset size, a global optimization in this hyper-parameter space is not feasible. To account for the random initialization process of network parameters, each model architecture was trained three times using different starting initializations. This parameter identification process results in varying learning rates and training epoch numbers of the different approaches. It does not limit the comparability of the respective methods, since, in each category, the model with the best prediction quality of the validation dataset was selected.

For all model types, good results were achieved by setting the subsequence length $L$ to 256 and the overlap factor $o$ to 0.5. The chosen pure LSTM model features a single memory block with 29 memory cells and
was trained for 501 epochs using a learning rate $\lambda$ of 0.0002. Both hybrid models use one memory block with 39 cells, where hybrid 1 was trained for 402 epochs with $\lambda = 0.0001$ and hybrid 2 was trained for 501 epochs with $\lambda = 0.0002$. The models are compared by evaluating a variety of metrics using the test dataset, visualized in Figure 7. Regarding the RMS and PSD RMS errors, the hybrid models outperform both the FRF model and the pure LSTM network. Especially in the highly non-linear offset examples, the LSTM and hybrid models approximate the system significantly better than the linear FRF model. The hybrid 2 model achieves the best prediction of fatigue-related signal properties by scoring a Multi-Rain damage ratio which is closest to one in most service load predictions of the test dataset. Exemplarily, Figure 8 provides a visual time series comparison of the models for a single force channel.

3.4. Forward prediction evaluation

Apart from its application in virtual sensing, the proposed hybrid methods can also be applied to the task of forward prediction. Here, the aim is to predict the measurements of all displacement and force sensors from the test bench drive signal, which determines the system excitation. From a physical point of view, this task is different from VS, since it relates system input quantities to output quantities as depicted in Figure 6. From a data-driven perspective, however, it simply requires a different dataset while the algorithm remains unchanged.

The general procedure of hyper-parameter identification is performed as described in Subsection 3.3. Like during VS, the best overall results were achieved by using a subsequence length $L$ of 256 and an overlap factor $o$ of 0.5. In this application, the best pure LSTM model uses a single LSTM block with 39 memory
Figure 8: The prediction quality of the presented models is visualized using short time series examples of a service load from the test dataset. Very large amplitudes (top left) benefit the most from LSTM network predictions, since the system stiffness is highly non-linear in regions of high absolute force. Average (top right) amplitudes are generally predicted with a very high accuracy by all models. Small signal offsets in the FRF model are especially noticeable for small oscillations (bottom left).
cells and was trained for 253 epochs with a learning rate $\lambda$ of 0.0002. For the chosen hybrid 1 model, two memory blocks of 25 memory cells each were trained for 75 epochs using a learning rate $\lambda$ of 0.003. The best hybrid 2 model again uses one memory block with 39 memory cells and was trained for 800 epochs with $\lambda = 0.0001$. The test dataset results are visualized separately for the prediction of displacements and forces in Figure 9, although in all cases both quantities were predicted by the same model.

Regarding only the prediction of the displacement sensors, the hybrid models reliably reproduce the low RMS results of the FRF model while significantly improving the prediction for the offset examples. Here, the hybrid nature of the approach is especially noticeable, since the pure LSTM network performs comparatively poor for most examples in terms of RMS. In contrast, the FRF model yields a higher PSD RMS error than both hybrid and pure LSTM approaches. While the pure FRF and LSTM models tend to under- and overestimate the fatigue damage, respectively, as shown by the Multi-Rain ratio, the prediction of the hybrid models is much closer to the perfect ratio of 1. Especially the hybrid 2 approach provides comparatively good results in almost all cases and consistently performs best within the offset examples.

Unfortunately, the overall trend of the displacement predictions does not translate to the forward prediction of force sensors. When compared to the FRF model predictions, the hybrid models only yield improvements in the offset examples, while the pure LSTM performs worse overall. The Multi-Rain damage results are very similar between all model configurations, where a ratio of 0.1 corresponds to an underestimation of amplitudes by 37% on average.

3.5. Dataset size dependency

The prediction quality of data-driven algorithms strongly depends on the availability of training data. Unfortunately, knowledge about this dependency is rarely available in practical applications, since the collection of large datasets can be very expensive and time consuming. In order to estimate how well LSTM network predictions scale to both larger and smaller datasets, an additional study was conducted using synthetic data.

Assuming a linear time invariant dynamic system, the relation between input and output channels is perfectly captured by an FRF model. This model can therefore be used to generate arbitrary amounts of synthetic output data from randomly generated input data in order to train an LSTM network. In this study, the FRF model of the virtual sensing experiment in Subsection 3.3 is used. For training and validation purposes, 250 data files of uncorrelated white pink noise data are generated with a length of 180 s each. The white section is limited at 20 Hz, the following pink section is characterized by a power spectral density of the form

$$\text{PSD}(\omega) \propto \omega^{-1}$$

and stops at 50 Hz. The channel mean is randomly sampled in the range $[-4 \text{kN}, 4 \text{kN}]$ and the amplitude range is $[0.5 \text{kN}, 2 \text{kN}]$.

Four different LSTM architectures, shown in Table 2, are compared in this study. The network with one block of only 10 inner states is chosen as an example with an exceptionally low degree of freedom. The architecture of the following networks with one block of 39 cells or two blocks of 23 cells are each very similar to the best performing networks of the virtual sensing and forward prediction tasks, see Subsection 3.3 and Subsection 3.4 respectively. The final network with two blocks of 39 memory cells each is considerably more complex than any previously examined architecture.

Since this study is designed to investigate the influence of the dataset size on the LSTM prediction, the model hyper-parameters are fixed to a subsequence length $L$ of 256, an overlap factor $o$ of 0.5, a learning rate $\lambda$ of 0.001 and a fixed training length of 100 epochs. The number of randomly generated noise training

| Architecture | [10] | [39] | [23,23] | [39,39] |
|--------------|-----|-----|-------|--------|
| Number of parameters | 593 | 6,828 | 6,880 | 19,152 |

14
Figure 9: In the forward prediction example, both displacement (left) and force (right) sensor data are predicted from the drive signal of the force-controlled servo-hydraulic test bench. FRF, LSTM and hybrid models are compared on all service load test data files, whose corresponding drive signals are rescaled versions of the same original signal shape. The results are sorted by the respective scaling factor. A scaling of 0.5− indicates that a negative offset was applied to the data with scaling factor 0.5, while 0.5+ symbolizes a positive offset. Both RMS and PSD RMS errors are averaged over the predicted channels.
Figure 10: The influence of the dataset size on the predictive quality of LSTM networks is studied using a large dataset of random noise excitations and synthetic FRF model responses. The comparison of different model architectures shows that very small models are not able to properly exploit the benefits of a large dataset, while the prediction quality of more complex models scales very well with the dataset size.

4. Conclusions

A novel hybrid approach for Virtual Sensing in non-linear dynamic systems was introduced based on Long Short-Term Memory networks and frequency response function models. These methods synergize very well, since the FRF model perfectly captures the behavior of linear systems and therefore provides a very good starting point for the non-linear LSTM predictions. Extraction and recombination of short subsequence signals enable the direct application of the LSTM algorithm to measurement data. Two strategies for the hybrid combination of LSTM and FRF models were suggested and compared. The effectiveness of the proposed approaches was demonstrated using a non-linear experimental dataset from a servo-hydraulic fatigue test bench. Different error metrics were employed to determine the predictive quality of the models in time and frequency domains as well as in a fatigue context.

Virtual Sensing results indicate that hybrid models significantly outperform both pure FRF and LSTM models in most situations. Extending the LSTM network inputs by the FRF prediction generally yields more accurate results than using the LSTM network to correct the FRF model error, especially regarding fatigue metrics. The approach was also applied to the Forward Prediction of sensor data, where hybrid models provided very good estimates for displacement sensors. Linear studies on a large synthetic dataset suggest that further improvements in prediction quality are likely to be achieved by increasing the dataset size.

Since low cost acceleration sensors are readily available and strain measurements are of high interest for fatigue applications, the approach should be extended to these physical quantities in further studies.
should also be tested how well the LSTM training process can be scaled to scenarios with a significantly higher number of sensors. Finally, it would be very beneficial to identify algorithms which find an optimal dataset composition for training and validation. Such algorithms could estimate the exact amount of data required to parameterize models in specific tasks and ensure that the available measurements are used as efficiently as possible.

5. 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.

6. Acknowledgments

The authors gratefully acknowledge the GWK support for funding this project by providing computing time through the Center for Information Services and HPC (ZIH) at TU Dresden.

References

[1] H. Hjelm, R. Brincker, J. Graugaard-Jensen, K. Munch, Determination of stress histories in structures by natural input modal analysis, Proceedings of IMAC-XXIII, A Conference & Exposition on Structural Dynamics.
[2] A. Ilipoulos, C. Devriendt, P. Guillaume, D. Van Hemelrijck, Continuous fatigue assessment of an offshore wind turbine using a limited number of vibration sensors, EWSHM - 7th European Workshop on Structural Health Monitoring.
[3] J. Kullaa, Bayesian virtual sensing in structural dynamics, Mechanical Systems and Signal Processing 115 (2019) 497–513. doi:https://doi.org/10.1016/j.ymssp.2018.06.010 URL https://www.sciencedirect.com/science/article/pii/S0888327018303467
[4] J. Kullaa, Virtual sensing of structural vibrations using dynamic substructuring. Mechanical Systems and Signal Processing 79 (2016) 203–222, special issue from ICEDyn 2015. doi:https://doi.org/10.1016/j.ymssp.2016.02.045 URL https://www.sciencedirect.com/science/article/pii/S0888327016000960
[5] J. Ching, J. Beck, Real-time reliability estimation for serviceability limit states in structures with uncertain dynamic excitation and incomplete output data, Probabilistic Engineering Mechanics 22 (1) (2007) 50–62. doi:https://doi.org/10.1016/j.probengmech.2006.05.006 URL https://www.sciencedirect.com/science/article/pii/S0266892006000233
[6] K. Erazo, E. M. Hernandez, A model-based observer for state and stress estimation in structural and mechanical systems: Experimental validation, Mechanical Systems and Signal Processing 43 (1) (2014) 141–152. doi:https://doi.org/10.1016/j.ymssp.2013.10.011 URL https://www.sciencedirect.com/science/article/pii/S0888327013005141
[7] H. G. Natke, Einführung in Theorie und Praxis der Zeitreihen- und Modalanalyse: Identifikation schwingungsfähiger elastomechanischer Systeme (1988).
[8] W. Ke, D. Huang, F. Yang, Y. Jiang, Soft sensor development and applications based on LSTM in deep neural networks, in: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), 2017, pp. 1–6. doi:10.1109/SSCI.2017.8289564
[9] X. Yuan, L. Li, Y. A. W. Shardt, Y. Wang, C. Yang, Deep learning with spatiotemporal attention-based LSTM for industrial soft sensor model development, IEEE Transactions on Industrial Electronics 68 (5) (2021) 4404–4414. doi:10.1109/TIE.2020.2984443 URL https://www.mdpi.com/1424-8220/21/3/823
[10] F. Curreri, L. Patanè, M. G. Xilibria, RNN- and LSTM-based soft sensors transferability for an industrial process Sensors 21 (3). doi:10.3390/s21030823 URL https://www.mdpi.com/1424-8220/21/3/823
[11] V. Roust, W. Charon, G. Citrovecce, Neural model of the dynamic behaviour of a non-linear mechanical system, Mechanical Systems and Signal Processing 23 (4) (2009) 1145–1159. doi:https://doi.org/10.1016/j.ymssp.2008.09.004 URL https://www.sciencedirect.com/science/article/pii/S0888327008002215
[12] Z. Yue, L. Liu, T. Long, Y. Ma, Virtual sensing method for monitoring vibration of continuously variable configuration structures using long short-term memory networks, Chinese Journal of Aeronautics 33 (1) (2020) 244–254. doi:https://doi.org/10.1016/j.cja.2019.09.019
URL https://www.sciencedirect.com/science/article/pii/S1000936119303425

[13] A. Beste, K. Drefler, H. Kötze, W. Krüger, B. Maier, J. Petersen, Multiaxial rainflow: A consequent continuation of professor tatsuos endo's work in: Y. Murakami (Ed.), The Rainflow Method in Fatigue, Butterworth-Heinemann, 1992, pp. 31–40. doi:https://doi.org/10.1016/0-7506-0504-5.50012-6
URL https://www.sciencedirect.com/science/article/pii/B9780750605045000126

[14] M. Hildebrandt, M. Khalil, C. Berge, V. Tresp, R. Wuchner, K.-U. Bletzinger, M. Heizmann, Remaining useful life estimation for unknown motors using a hybrid modeling approach, in: 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), Vol. 1, 2019, pp. 1327–1332. doi:10.1109/INDIN41052.2019.8972200

[15] B. W. Cryer, P. E. Nawrocki, R. A. Lund, A Road Simulation System for Heavy Duty Vehicles SAE Technical Paper 760361 (1976) 1. doi:https://doi.org/10.4271/760361
URL https://doi.org/10.1162/neco.1997.9.8.1735

[16] N. C. Hay, D. E. Roberts, Iterative control in automotive testing, Proceedings of the Institution of Mechanical Engineers, Part I. Journal of Systems and Control Engineering 221 (3) (2007) 223–233. doi:10.1243/09596518JSCE306

[17] C. Dodds, A. Plummer, Laboratory road simulation for full vehicle testing: A review, in: SIAT 2001, 2001. doi:10.4271/2001-26-0047

[18] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Computation 9 (8) (1997) 1735–1780. arXiv:https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf, doi:10.1162/neco.1997.9.8.1735
URL https://doi.org/10.1162/neco.1997.9.8.1735

[19] F. Gers, J. Schmidhuber, F. Cummins, Learning to forget: continual prediction with LSTM, IET Conference Proceedings (1999) 850–855(5).
URL https://digital-library.theiet.org/content/conferences/10.1049/cp_19991218

[20] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015).
URL https://www.tensorflow.org/

[21] F. Chollet, et al., Keras, https://keras.io (2015).

[22] G. Hinton, Lecture 6a overview of mini-batch gradient descent (2012).

[23] E. Hailbacher, Betriebfestigkeit: Verfahren und Daten zur Bauteilberechnung, VDI-Buch, Springer, 2002.
URL https://books.google.de/books?id=vvz_U6e7ycCe

[24] C. McInnes, P. Meehan, Equivalence of four-point and three-point rainflow cycle counting algorithms, International Journal of Fatigue 30 (3) (2008) 547–559. doi:https://doi.org/10.1016/j.ijfatigue.2007.03.006
URL https://www.sciencedirect.com/science/article/pii/S0142112307001119

[25] A. Palmgren, Die Lebensdauer von Kugellagern, Zeitschrift des Vereines Deutscher Ingenieure 68 (14) (1924) 339–341.