Computationally and Memory-Efficient Robust Predictive Analytics Using Big Data

1st Daniel Menges
Department of Engineering Cybernetics
Norwegian University of Science and Technology
Trondheim, Norway
daniel.menges@ntnu.no

2nd Adil Rasheed
Department of Engineering Cybernetics
Norwegian University of Science and Technology
Trondheim, Norway
adil.rasheed@ntnu.no

***Abstract*—In the current data-intensive era, big data has become a significant asset for Artificial Intelligence (AI), serving as a foundation for developing data-driven models and providing insight into various unknown fields. This study navigates through the challenges of data uncertainties, storage limitations, and predictive data-driven modeling using big data. We utilize Robust Principal Component Analysis (RPCA) for effective noise reduction and outlier elimination, and Optimal Sensor Placement (OSP) for efficient data compression and storage. The proposed OSP technique enables data compression without substantial information loss while simultaneously reducing storage needs. While RPCA offers an enhanced alternative to traditional Principal Component Analysis (PCA) for high-dimensional data management, the scope of this work extends its utilization, focusing on robust, data-driven modeling applicable to huge data sets in real-time. For that purpose, Long Short-Term Memory (LSTM) networks, a type of recurrent neural network, are applied to model and predict data based on a low-dimensional subset obtained from OSP, leading to a crucial acceleration of the training phase. LSTMs are feasible for capturing long-term dependencies in time series data, making them particularly suited for predicting the future states of physical systems on historical data. All the presented algorithms are not only theorized but also simulated and validated using real thermal imaging data mapping a ship's engine.**

***Index Terms*—Big Data, Robust PCA, Optimal Sensor Placement, LSTM, Thermal Imaging, Ship Engine**

I. Introduction

In the context of Artificial Intelligence (AI), data have taken center stage, influencing decision-making processes in many domains, from healthcare [1] to econometrics [2], manufacturing [3], and more. However, while big data offers incredible potential, it is essential to understand its strengths and inherent flaws, especially since data can be erroneous due to various factors such as sensor uncertainties and transmission errors. Therefore, data can sometimes be misinterpreted if not used appropriately, particularly when the underlying data are flawed or inaccurate [4]. The ability to effectively handle, analyze, and interpret these growing volumes of data is essential. Therefore, the development and deployment of robust data analysis techniques is of critical importance.

Among various available data analysis tools, Principal Component Analysis (PCA) [5] has gained significant attention due to its ability to reduce the dimensionality of data sets while retaining most of the underlying information [6]. However, traditional PCA is highly susceptible to outliers and corruptions in the data, which can substantially impact its performance and the accuracy of subsequent analyses. Consequently, there is a need for more robust techniques that can handle such irregularities. Robust Principal Component Analysis (RPCA), an advanced variant of PCA, offers more reliable results by robustly separating low-rank and sparse components in the data, even in the presence of outliers and corruptions [7]. The concept of RPCA for decomposing a data matrix in a low-rank and a sparse component is accurately described in [8]. The decomposed components use a convex program called Principal Component Pursuit. The method, which can recover the principal components even when data entries are corrupted or missing, has applications in video surveillance for object detection, in cluttered backgrounds and face recognition for removing shadows, in specularities, and more. A detailed comparison of PCA and RPCA is given in [9], showcasing the benefits and robustness of RPCA.

In parallel, considering the growing need for big data, one of the key challenges that emerges is the efficient storage and transmission of these enormous volumes of data. A novel approach to this problem is the concept of Optimal Sensor Placement (OSP) [10]. OSP involves strategic positioning of sensors to capture the most relevant data, significantly reducing redundancy and facilitating efficient data storage and transmission. In essence, OSP aims to obtain a compressed version of the data without a significant loss of information.

Through a comprehensive examination of RPCA and OSP, this study aims to explore the synergies among these methodologies and their collective impact on improving the accuracy and efficiency of big data modeling and analysis.

Furthermore, we extend this work by integrating a data-driven modeling approach for real-time predictions using Long Short-Term Memory (LSTM) networks, which was first proposed by [11]. The specialized design of LSTMs, with its gate mechanisms, allows them to learn long-term dependencies in

This work is part of SFI AutoShip, an 8-year research-based innovation center. In addition, this research project is integrated into the PERSEUS doctoral program. We want to thank our partners, including the Research Council of Norway, under project number 309230, and the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement number 101034240. Furthermore, we thank Idletechs AS for providing us with the thermal camera data.

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00072

the data [12]. Artificial Neural Networks (ANNs) have gained considerable traction in various forecasting domains due to their adaptability, nonlinearity, and the capability to map arbitrary functions. However, they require a lot of computational time for training [13]. Therefore, we create LSTM models based on the few selected data points obtained from the OSP algorithm. This technique significantly accelerates the training phase, making the proposed methodology adaptable to a wide range of applications. Once these few data points (measurements) are predicted using LSTMs, we reconstruct the full data dimension via the concept of OSP, allowing predictions of future states in full dimension with remarkable accuracy. The integration of RPCA, OSP, and LSTM offers a novel approach to big data modeling, promising both robustness and scalability in various real-world scenarios.

In this study, we applied the algorithms on a dataset from a thermal camera mapping a ship's engine. The thermal images provided insight into the temperature profiles and fluctuations, offering a unique perspective on the engine's operational behavior and performance. Condition monitoring is crucial to maintain safe maritime operations [14] and can provide insight into the reliability of a vessel's engine and its components. By identifying anomalies early on, it is possible to predict the lifespan of these components and prevent significant breakdowns. As pointed out in [15], the maritime sector rarely employs predictive maintenance. Instead, most maintenance activities on ships tend to be preventive. This frequently leads to higher costs as replaced components might have had a longer usable life endurance.

In summary, this study addresses three core challenges:

- The robust treatment of uncertainties such as outliers and corruptions in data due to the use of affordable, nonintrusive thermal camera measurements.
- The requirement for memory-efficient storage techniques due to the vast data generated.
- The capability of proactive maintenance in real-time through predictive data-driven modeling.

II. Theory

This section provides a detailed overview of the statistical techniques used in this study. We introduce the concept of Principal Component Analysis (PCA) and its robust counterpart, Robust Principal Component Analysis (RPCA), for data cleaning. Furthermore, the section covers the idea of Optimal Sensor Placement (OSP) used for effective data compression and storage management.

A. Principal Component Analysis

Principal Component Analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of linearly uncorrelated variables, termed principal components. This procedure allows for identifying the directions (principal components) where the data vary the most. There are two main approaches to compute the PCA. The eigenvector approach and the Singular Value Decomposition (SVD) approach. The general concepts are described in detail in [16]. The SVD approach is often chosen since it is numerically more robust.

Singular Value Decomposition Approach: PCA is closely related to SVD, a factorization of a real or complex matrix. For any real matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, with $m \geq n$, there exists a factorization of the form

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{1}$$

where $\mathbf{U} \in \mathbb{R}^{m \times m}$, $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$, and $\mathbf{V} \in \mathbb{R}^{n \times n}$. The columns of $\mathbf{U}$ are orthonormal eigenvectors of $\mathbf{A}\mathbf{A}^T$, and the columns of $\mathbf{V}$ are orthonormal eigenvectors of $\mathbf{A}^T\mathbf{A}$. The diagonal elements of $\mathbf{\Sigma}$ are the square roots of the eigenvalues of $\mathbf{A}^T\mathbf{A}$ (or equivalently, $\mathbf{A}\mathbf{A}^T$), and are called the singular values of $\mathbf{A}$. To see this, we first consider the matrix $\mathbf{A}^T\mathbf{A}$, which is a symmetric matrix. By the spectral theorem, we can factorize it as

$$\mathbf{A}^T\mathbf{A} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T. \tag{2}$$

Similarly, we can factorize $\mathbf{A}\mathbf{A}^T$ as

$$\mathbf{A}\mathbf{A}^T = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T. \tag{3}$$

Using these two identities, it can be shown that

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{4}$$

which is the Singular Value Decomposition of $\mathbf{A}$.

Consider a data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, where each row is an observation and each column is a variable. We assume that the data have been centered, i.e. the column means have been subtracted off.

1) *Perform a lower-ranked SVD:* Compute the SVD of $\mathbf{X}$ by $\mathbf{X} = \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^T + \mathbf{E}$. Here, $\mathbf{U}_r \in \mathbb{R}^{m \times r}$ and $\mathbf{V}_r^\top \in \mathbb{R}^{r \times n}$ are orthogonal matrices containing left and right singular vectors and r is the number of principal components, respectively. The matrix $\mathbf{\Sigma}_r \in \mathbb{R}^{r \times r}$ contains the r largest singular values in decreasing order on the diagonal. Furthermore, the matrix $\mathbf{E}$ contains the unmodeled residuals due to the dimensionality reduction.
2) *Principal components:* Finally, the principal components of $\mathbf{X}$ are given by $\mathbf{X}\mathbf{V}_r \approx \mathbf{U}_r\mathbf{\Sigma}_r$. The i-th column of $\mathbf{X}\mathbf{V}_r$ is the projection of the data onto the i-th principal direction (i.e., the i-th eigenvector).

This process shows how PCA can be derived from the SVD of a data matrix. However, traditional PCA is highly sensitive to outliers and data corruptions.

B. Robust Principal Component Analysis

The most significant advantage of RPCA over standard PCA is its resilience to outliers. Traditional PCA is sensitive to outliers because it tries to find a lower-dimensional representation that best explains the variance in the data. If outliers are present, PCA may be heavily influenced by them, leading to a representation that does not accurately capture most of the underlying structure of the data. RPCA, on the other hand, explicitly models these outliers, resulting in a more accurate and robust representation of the primary data structure.

In certain contexts, RPCA can better recover the true underlying low-rank structure of data compared to PCA, especially when the data are grossly corrupted or when a significant amount of data is missing.

RPCA works by decomposing the data matrix into a low-rank matrix and a sparse matrix. The low-rank matrix captures the principal components, and the sparse matrix captures outliers or anomalies. This separation can be very useful in many applications, such as image and video processing, where the low-rank component can correspond to the background and the sparse component can correspond to moving objects. The general idea is to decompose the data matrix $\mathbf{X}$ into two components expressed by

$$\mathbf{X} = \mathbf{L} + \mathbf{S}. \tag{5}$$

Here, the matrix $\mathbf{L}$ describes the low-rank matrix that captures the main structure of the data, while the matrix $\mathbf{S}$ is sparse and captures outliers and corruptions. Therefore, the goal is to find $\mathbf{L}$ and $\mathbf{S}$ which satisfy

$$\begin{aligned} &\underset{\mathbf{L},\mathbf{S}}{\text{minimize}} && \operatorname{rank}(\mathbf{L}) + \|\mathbf{S}\|_0, \\ &\text{subject to} && \mathbf{L} + \mathbf{S} = \mathbf{X}, \end{aligned} \tag{6}$$

where $\|\mathbf{S}\|_0$ describes the zero norm of $\mathbf{S}$, and $\operatorname{rank}(\mathbf{L})$ specifies the rank of $\mathbf{L}$. However, due to the nonconvex nature of both the rank($\mathbf{L}$) and the $\|\mathbf{S}\|_0$, this optimization problem becomes intractable [9]. To overcome this issue, convex relaxation [17] provides an approach to approximate convexity for nonconvex problems. Convex relaxation allows transforming (6) into

$$\begin{aligned} &\underset{\mathbf{L},\mathbf{S}}{\text{minimize}} && \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1, \\ &\text{subject to} && \mathbf{L} + \mathbf{S} = \mathbf{X}, \end{aligned} \tag{7}$$

where $\|\cdot\|_1$ is the L_1 norm given by the sum of the absolute values of the matrix entries, $\|\cdot\|_*$ is the nuclear norm given by the sum of singular values, and λ is a hyperparameter. While minimization of $\|\mathbf{S}\|_1$ leads to an approximation of minimizing $\|\mathbf{S}\|_0$, minimization of $\|\mathbf{L}\|_*$ leads to an approximation of the lowest possible $\operatorname{rank}(\mathbf{L})$. The problem described in (7) is convex and known as Principal Component Pursuit (PCP). To solve this convex problem, the Augmented Lagrange Multiplier (ALM) algorithm is suggested [18]. The augmented lagrange multiplier can be formulated as

$$\mathcal{L}(\mathbf{L},\mathbf{S},\mathbf{\Lambda}) = \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 + \langle\mathbf{\Lambda}, \mathbf{X}-\mathbf{L}-\mathbf{S}\rangle + \tfrac{\mu}{2}\|\mathbf{X}-\mathbf{L}-\mathbf{S}\|_F^2, \tag{8}$$

where $\mathbf{\Lambda}$ is the matrix of Lagrange multipliers, μ is a hyperparameter, $\langle\cdot\rangle$ denotes the inner product, and $\|\cdot\|_F$ is the Frobenius norm, also known as the Euclidean norm, which is a measure of the magnitude or length of a matrix. Subsequently, we minimize $\mathcal{L}$ to solve for $\mathbf{L}_k$ and $\mathbf{S}_k$ at timestep k, where the matrix of Lagrange multipliers is updated by

$$\mathbf{\Lambda}_{k+1} = \mathbf{\Lambda}_k + \mu(\mathbf{X} - \mathbf{L}_k - \mathbf{S}_k). \tag{9}$$

As a result, RPCA decomposes a data matrix $\mathbf{X}$ into a low-rank component $\mathbf{L}$ and a sparse component $\mathbf{S}$.

C. Optimal Sensor Placement

Optimal Sensor Placement (OSP) is a method to identify the most insightful locations within a system for the positioning of sensors. This approach can maximize the measurements' entropy while minimizing the amount of sensors required. Here, entropy describes the abundance of information within a system.

Let $\boldsymbol{x} \in \mathbb{R}^n$ be a single data point in time, which can be approximated as

$$\boldsymbol{x} \approx \mathbf{\Psi}_r \boldsymbol{a}, \tag{10}$$

where $\boldsymbol{a} \in \mathbb{R}^r$ contain the coefficients that vary over time while the columns of $\mathbf{\Psi}_r$ are the modes of the lower-ranked Proper Orthogonal Decomposition (POD). POD is very similar to PCA. However, POD modes are not scaled by the singular value matrix $\mathbf{\Sigma}$, such as the principal components of PCA. Therefore, $\mathbf{\Psi}_r = \mathbf{U}_r$. If we assume that the measurements can be expressed by

$$\boldsymbol{y} = \mathbf{C}\boldsymbol{x}, \tag{11}$$

with $\mathbf{C} \in \mathbb{R}^{s\times n}$ being a sparse measurement matrix and s the number of sensors, the measurements can be approximated by

$$\boldsymbol{y} \approx \mathbf{C}\mathbf{\Psi}_r\boldsymbol{a}. \tag{12}$$

If we denote $\mathbf{\Theta} = \mathbf{C}\mathbf{\Psi}_r$, the estimated coefficients can be represented by

$$\hat{\boldsymbol{a}} = \mathbf{\Theta}^{\dagger}\boldsymbol{y}. \tag{13}$$

Hence, we can derive an estimate of $\boldsymbol{x}$ yielding

$$\hat{\boldsymbol{x}} = \mathbf{\Psi}_r\hat{\boldsymbol{a}} = \mathbf{\Psi}_r(\mathbf{C}\mathbf{\Psi}_r)^{\dagger}\boldsymbol{y}. \tag{14}$$

As $\mathbf{\Psi}_r$ can be determined using the lower-ranked SVD, the only unknown entity is the sparse measurement matrix $\mathbf{C}$. As described by [10], optimal sensor placement can be achieved by applying QR factorization with column pivoting to the POD modes $\mathbf{\Psi}_r$. In this relation, it is important to note that the number of sensors s must satisfy $s \geq r$.

III. Methodology

This section describes a potential big data workflow for data cleaning, data compression, and computationally efficient data-driven modeling. The core of the proposed framework is built by RPCA, OSP, and LSTMs.

A. Data Cleaning

We use RPCA for data cleaning, introduced in Section II-B. The tuning parameters described in (8) were chosen as $\lambda = 0.006$ and $\mu = 10^{-5}$. After obtaining the decomposition of the data matrix $\mathbf{X}$ into $\mathbf{L}$ (low-rank matrix) and $\mathbf{S}$ (sparse matrix), a cleaned version of the data can be reconstructed. The low-rank matrix $\mathbf{L}$ represents the underlying physics, while the sparse matrix $\mathbf{S}$ contains anomalies and perturbations. As a result, the matrix $\mathbf{L}$ represents a cleaned version of the data matrix $\mathbf{X}$.

B. Data Compression

To compress the data while simultaneously retaining the essential information about the underlying system, we apply OSP described in Section II-C to the cleaned data matrix $\mathbf{L}$ obtained from RPCA. The fundamental principle behind OSP is to maximize the fidelity of the data while minimizing the number of sensors or data points. By placing sensors in locations that capture the most variance or information in the data, we can represent the original high-dimensional data $\mathbf{X}$ with a significantly smaller set of measurements $\mathbf{Y}$, where $\mathbf{Y}$ contains $\boldsymbol{y}$ stacked over a specific historical window. This smaller set of measurements is represented by the sparse measurement matrix $\mathbf{C}$. The selected measurements or sensors produce a compressed version of the original data. By reducing the number of required sensors, OSP can lead to significant cost savings in scenarios where sensor deployment is expensive.

C. LSTM-based Modeling of Sparse Measurements

In the field of data-driven modeling, the power of neural networks, particularly LSTM networks, has been proven in many applications. LSTMs are designed to remember patterns over long sequences, making them suitable for modeling time-series data. However, LSTMs may not be computationally suitable for large datasets. Therefore, we apply LSTM to the lower-dimensional subset $\mathbf{Y}$ obtained from OSP. The combination of LSTMs and OSP can drastically reduce the computational costs required to train LSTM networks. When we use LSTMs to model these sparse measurements selected by OSP, we aim to capture the underlying temporal dynamics. Once trained, these networks can be used to predict the sparse data points. By subsequently applying the reconstruction algorithm given by (14), we can transform these sparse predictions into full-sized sensor space, mapping the original data dimensions. Note that if the data is sampled with an inconsistent frequency, an initial interpolation of the data can lead to more accurate models.

D. Big Data Workflow

The previously described approaches can interact to combine their strengths into an optimized big data workflow. In Fig. 1, we demonstrate a potential framework, employable to various applications for data preprocessing, compression, and modeling. The workflow has the following structure:

1) **Data Cleaning:** RPCA generates a cleaned version $\mathbf{L}$ of the data matrix $\mathbf{X}$. Since $\mathbf{L}$ contains the information of interest (e.g., the underlying dynamics of the system), $\mathbf{L}$ can be propagated to subsequent processing and analysis methods.
2) **Data Compression:** The OSP algorithm enables drastic compression of the cleaned data matrix $\mathbf{L}$. Computing the POD modes $\boldsymbol{\Psi}_r$ of $\mathbf{L}$ and finding the sparse measurement matrix $\mathbf{C}$, a small subset $\mathbf{Y}$ can be sufficient for data storage. The subset $\mathbf{Y}$ can be forwarded for continuative analysis and modeling. Note that $\boldsymbol{\Psi}_r$ and $\mathbf{C}$ must also be stored to extend the subset $\boldsymbol{y}$ to its original dimension $\hat{\boldsymbol{x}}$ (see (14)).
3) **Data-Driven Modeling:** In this step, data-driven models of the propagated subset $\mathbf{Y}$ are built using an LSTM-based neural network. The built data-driven models of the subset can thus be used to predict future states. After predicting the future subset, predictions of the original data dimension $\hat{\mathbf{X}}_{\mathbf{pred}}$ can be computed using $\boldsymbol{\Psi}_r$ and $\mathbf{C}$ from the OSP algorithm.

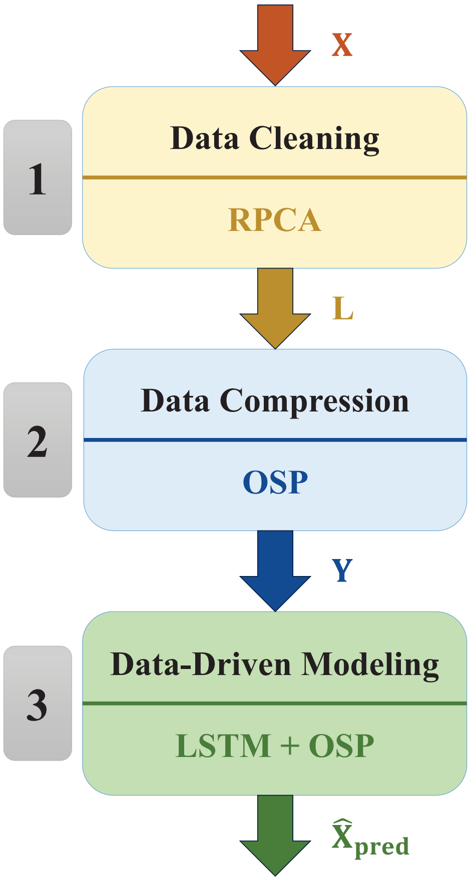


Fig. 1. Concept of cleaning, compressing, and modeling the data.

IV. Simulation Setup

This study uses data acquired from a thermal camera mapping a ship's engine. The data were provided by Idletechs AS. Since the data were uncorrupted and free from outliers, we simulated synthetic perturbations affecting the data specified below. In addition, we describe the LSTM neural network setup that we chose for this study.

A. Data

The dataset under examination is derived from thermal camera imagery, which captures the engine of a ship, specifically that of a ferry. The primary intent behind acquiring these images was to observe the thermal behavior of the engine during various operational states, including taking off, steady-state driving, and docking.

The data acquisition spanned a total of four consecutive days. On each day, the engine was continuously monitored for a duration of approximately six hours, resulting in a cumulative observation period of 24 hours over the four days. However, the sampling frequency of the recordings was not consistent. The average time between consecutive samples is approximately 0.5 seconds. A snapshot of a thermal image mapping the ship's engine is depicted in Fig. 2.

Each image sourced from the thermal camera comprises 19,200 pixels, with dimensions set at 120x160 pixels. Each pixel captures the thermal radiations from the engine, which can potentially offer insights into the ship's engine's thermal performance and any anomalous patterns or hotspots that may arise during its operation.

B. Perturbations

To evaluate the methods under various conditions, we performed four simulation scenarios comprising outliers, corruptions, noise, and a combination of them.

Scenario 1: The data is perturbed by Gaussian noise, where the noise was generated with a mean of 0 and a standard deviation of 4, ensuring that the noise values are concentrated mainly within the range of [-4, 4]. These parameters were chosen to mimic the range of intense noise present in the measurement processes.

Scenario 2: The data is perturbed by outliers. These outliers were introduced by randomly selecting 100 data points (pixels) and replacing their original values with randomly generated values in the range of [30, 40] and [-40, -30]. This range was chosen to ensure that the magnitude of the outliers was significantly different from that of the actual variables to simulate severe measurement anomalies.

Scenario 3: The data is perturbed by corruptions. These corruptions were simulated by adding uniformly distributed random noise to 10% of the dataset over the interval [-15, 30]. This interval was selected to ensure a substantial magnitude for the corruptions, to pretend a distortion, and to provide a stringent test for the robustness of the PCA, RPCA, and OSP algorithms.

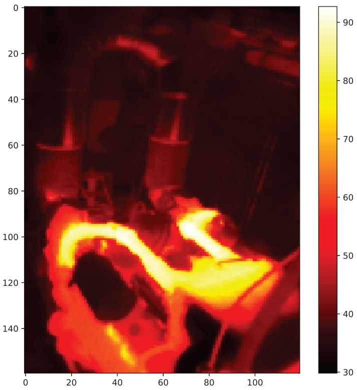


Fig. 2. Unperturbed thermal camera image mapping a ship's engine.

Scenario 4: The data is perturbed by a combination of the previously mentioned scenarios 1, 2, and 3, leading to a superposition of all scenarios.

C. LSTM network architecture

To train the LSTM network, various parameterizations were tested. Finally, the parameters shown in Table I were chosen. The network was trained using the Adam optimizer, where the Root-Mean-Squared Error (RMSE) was set as a metric to evaluate the model's performance during training. For the predictions, we trained the network with a window size of 50 historical samples, and the considered forecast time was chosen to be 100 time steps. The network structure consists of an input layer, an LSTM layer, a dense feedforward layer, and an output layer. Since deep neural networks with numerous parameters often overfit, we inserted a dropout layer. Dropout is a technique to address overfitting in which, during training, random units and their connections are omitted [19].

TABLE I
LSTM NEURAL NETWORK TRAINING PARAMETERS

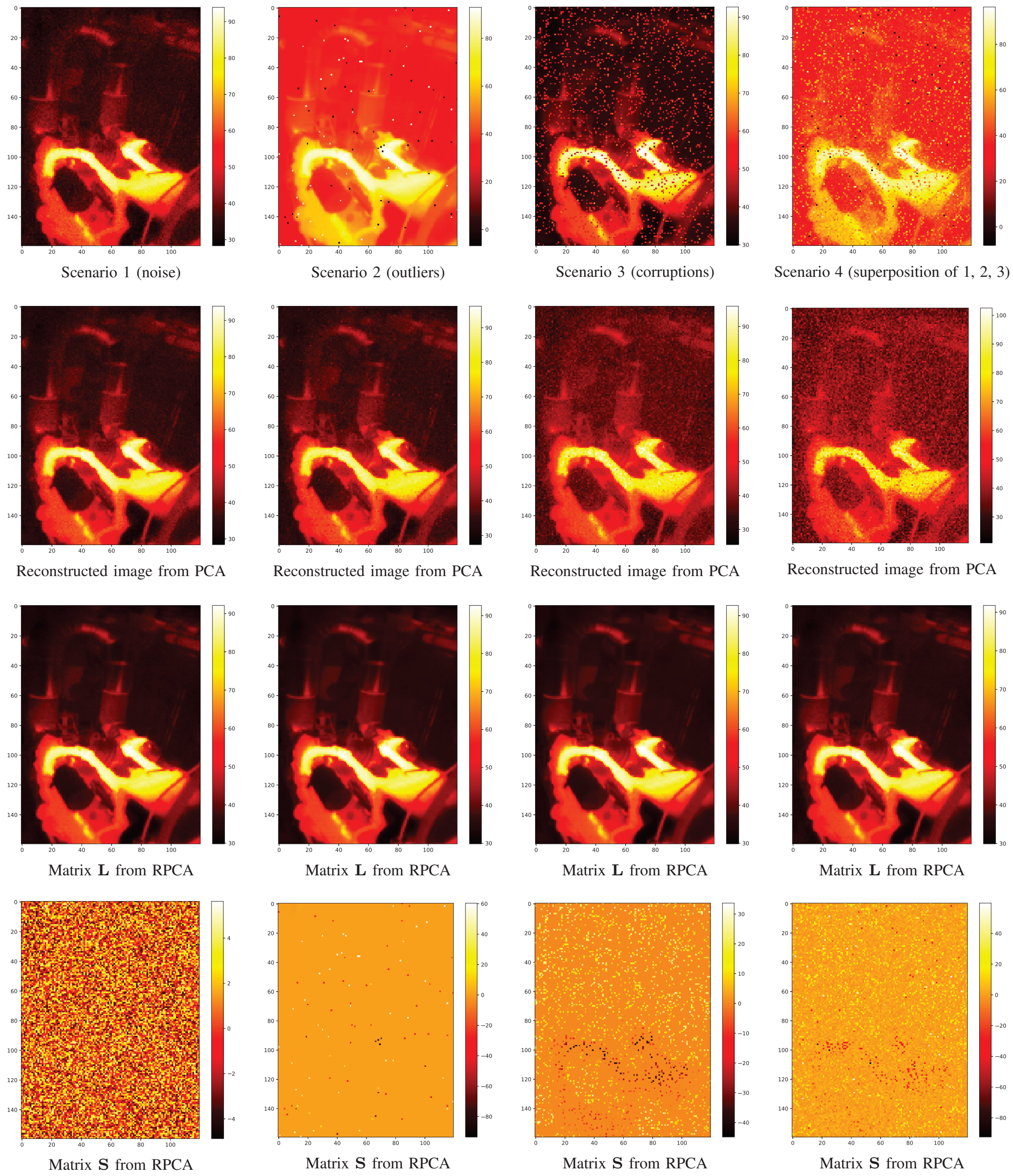
Parameter	Value
Input layer shape	50×10
LSTM layer size (neurons)	128
Dropout ratio	0.2
Feedforward layer size (neurons)	128
Output layer size (neurons)	10
Learning rate	10^{-4}
Epochs	100

V. Results and Discussion

In the following, the results of the individual approaches regarding data cleaning, data compression, and data-driven modeling are discussed.

A. Data Cleaning

The data cleaning phase is demonstrated in Fig. 3. Presented are the four various scenarios described in Section IV-B. Note that the unperturbed image of Fig. 2 reflects the ground truth. The results of RPCA are compared with those of PCA. It is shown how RPCA decomposes the thermal image data into the matrices $\mathbf{L}$ and $\mathbf{S}$. The matrix $\mathbf{L}$ clearly depicts the unperturbed image, while the matrix $\mathbf{S}$ captures the sparse components of the data, which mainly contain all unwanted fragments and anomalies. In contrast, the image reconstructions of traditional PCA are especially susceptible to intensive corruptions and outliers. Therefore, the capability of RPCA to decompose data into a low-rank matrix $\mathbf{L}$ and a sparse matrix $\mathbf{S}$ can improve the accuracy of many AI applications utilizing big data.


Scenario 1 (noise)

Scenario 2 (outliers)

Scenario 3 (corruptions)

Scenario 4 (superposition of 1, 2, 3)

Reconstructed image from PCA

Reconstructed image from PCA

Reconstructed image from PCA

Reconstructed image from PCA

Matrix $\mathbf{L}$ from RPCA

Matrix $\mathbf{L}$ from RPCA

Matrix $\mathbf{L}$ from RPCA

Matrix $\mathbf{L}$ from RPCA

Matrix $\mathbf{S}$ from RPCA

Matrix $\mathbf{S}$ from RPCA

Matrix $\mathbf{S}$ from RPCA

Matrix $\mathbf{S}$ from RPCA

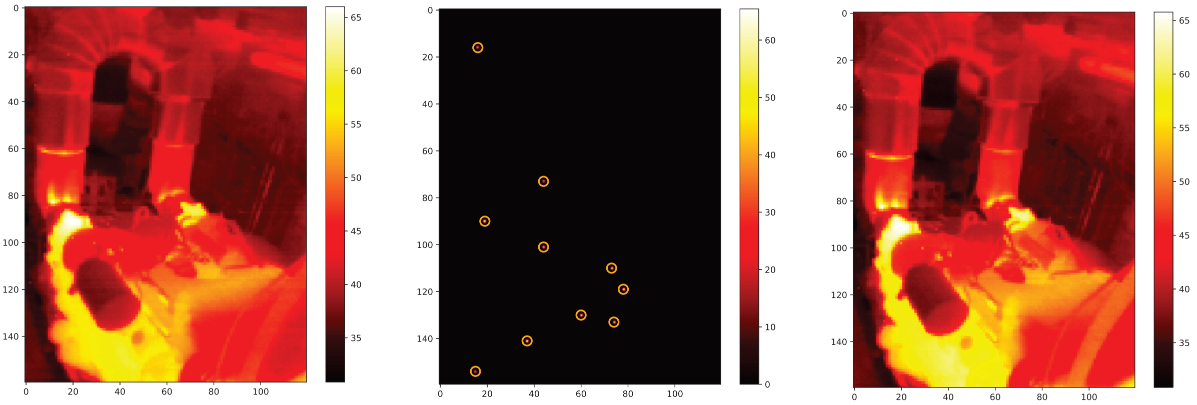
Fig. 3. Results of RPCA and PCA applied on thermal camera data under various conditions. The different scenarios are described in Section IV-B.

B. Data Compression

OSP applied to the thermal image data can drastically reduce the data dimension. In this study, we used only 10 of the original 19,200 pixel measurements. As illustrated in Fig. 4, it is evident that the original thermal images could be accurately reconstructed using a substantially reduced set of pixel measurements.

From a data compression standpoint, the ability to reconstruct comprehensive thermal images using limited pixel measurements underscores the energy and memory efficiency of OSP. This reduced representation not only implies a significant reduction in data size but also means that the essential features and characteristics of the thermal images are captured with minimal loss of information. Consequently, this data compression approach enables faster processing times, reduced memory requirements, and lower energy usage in real-time applications or scenarios with bandwidth constraints.

Memory savings can be considered as follows. Assuming a data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, where m is the spatial dimension and



Original image Measurements Reconstructed image

Fig. 4. Optimal sensor positions and their capability to reconstruct the original images.

n expresses the time dimension, while the lower-dimensional measurement matrix $\mathbf{Y} \in \mathbb{R}^{r \times n}$ is spanned by a low dimension r, then the ratio of saving memory is given by

$$\alpha = \frac{m}{r}. \tag{15}$$

In this case study, the ratio of saving memory yields

$$\alpha = \frac{19200}{10} = 1920. \tag{16}$$

This implies that, under the consideration of equal memory, we can store 1920 times more thermal images.

C. Predictive Data-Driven Modeling

The LSTM network was trained using a sparse subspace $\mathbf{Y}$, obtained by OSP. Since this study dealt with data containing inconsistent time samples, we interpolated the data before building data-driven models via LSTM networks. To show the influence of previously interpolating the data, we depict the RMSE of the model predictions concerning the data-driven model with and without an initial interpolation in Fig. 5. The RMSE is related to the reconstructed images of the original image size (19,200 pixels) using the model predictions from the few OSP measurements (10 pixels). Furthermore, a comparison of the computational time for a training phase is demonstrated in Fig. 6, showing the tremendous efficiency improvement of the proposed approach. For comparison, the network structure and training parameters of Table I were used. The computational efficiency underscores the practicality of the method, especially when considering real-time applications. Once trained, the model's ability to make predictions is instantaneous, allowing real-time forecasts to be made in milliseconds. In addition, depending on the application and the parameters chosen for training (e.g., number of epochs), the proposed approach can enable online training in real-time.

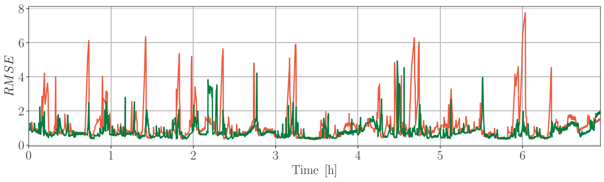


Fig. 5. RMSE of the LSTM predictions after reconstructing the entire images with the few pixel measurements from OSP. Compared are the predictions for 100 timesteps, where the LSTM model is trained on the original OSP pixel data (—) and the interpolated OSP pixel data (—).

VI. Conclusion

In conclusion, the application of Robust Principal Component Analysis (RPCA) on thermal image data significantly enhances the quality of the data, allowing for more insightful subsequent analyses. Given its robustness and versatility, this method can be extended to various data applications, broadening its relevance and potential impact across diverse domains. Furthermore, the use of Optimal Sensor Placement (OSP) offers a promising approach for those looking to maximize the efficiency of their data storage and compression strategies, especially in environments where storage space and data

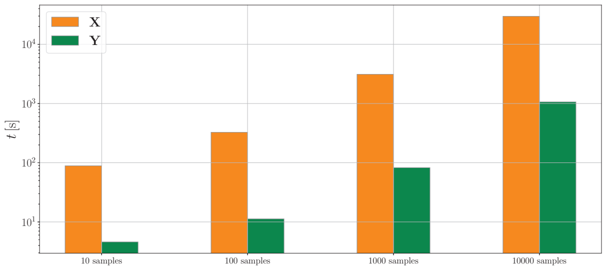


Fig. 6. Comparison of the computational time for a training phase of the LSTM network regarding full image data $\mathbf{X}$ and the compressed image data $\mathbf{Y}$. The training time is compared to various data sizes. Note that the time is scaled logarithmically. The number of epochs was set relatively high (100). Therefore, changing training parameters can allow online training with a duration of less than a second.

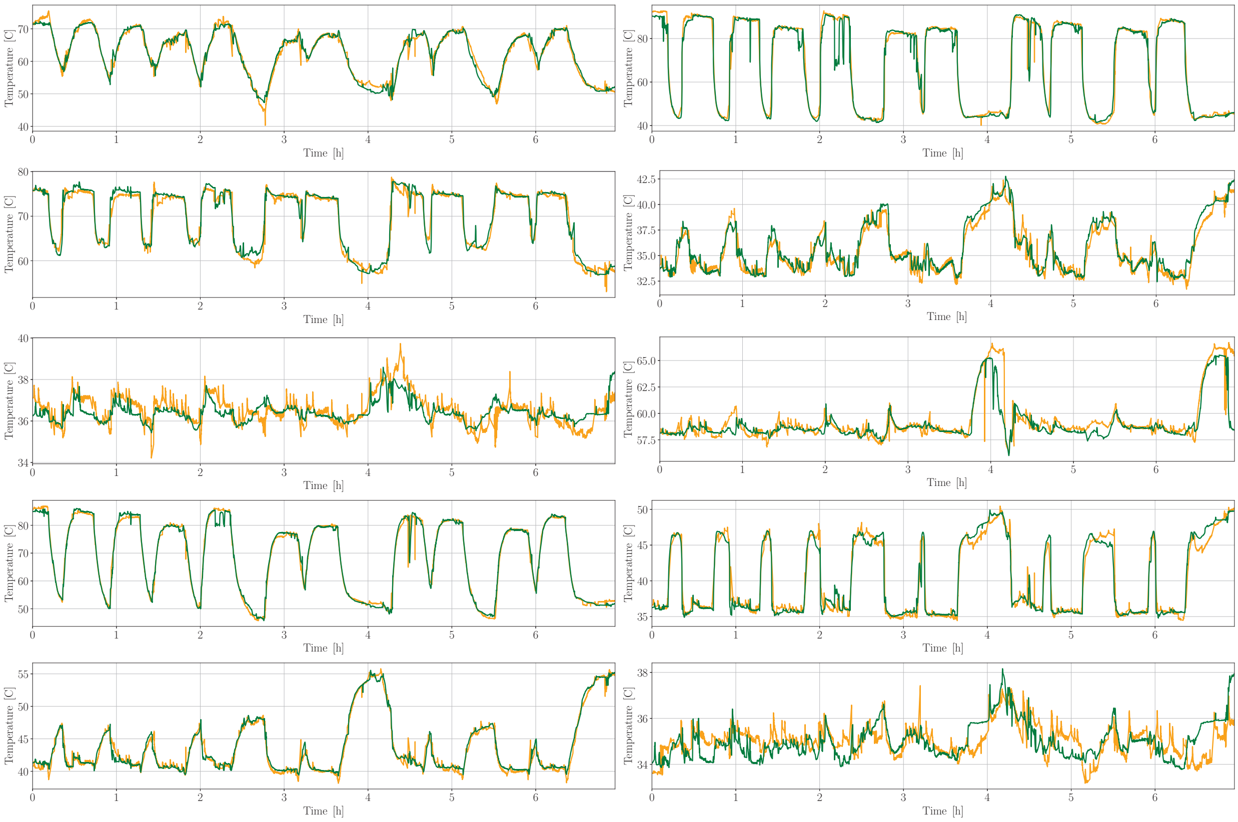


Fig. 7. Pixel predictions of the 10 optimal sensor positions (—) compared to the ground truth (—).

transmission capabilities are limited. Applying LSTMs to a lower-dimensional space obtained by OSP can improve computational efficiency and can enhance the accuracy of time-series predictions. The interaction of the presented approaches optimizes both data processing and subsequent analyses, which can improve data quality, computational efficiency, and memory efficiency while enabling real-time predictive capabilities.

References

[1] W. Raghupathi, V. Raghupathi, Big data analytics in healthcare: promise and potential, *Health Information Science and Systems*, 2:3 (2014).

[2] H. R. Varian, Big Data: New Tricks for Econometrics, *Journal of Economic Perspectives*, 28:3–28 (2014).

[3] K. Nagorny, P. Lima-Monteiro, J. Barata, A. W. Colombo, Big Data Analysis in Smart Manufacturing: A Review, *International Journal of Communications, Network and System Sciences*, 10:31–58 (2017).

[4] K. Cukier, V. Mayer-Schönberger, The Rise of Big Data: How It's Changing the Way We Think about the World, in M. Pitici (ed.), *The Best Writing on Mathematics 2014*, Princeton University Press, 2014, pp. 20–32.

[5] I. T. Jolliffe, *Principal component analysis*, 2nd edition, Springer series in statistics, Springer, New York, 2002.

[6] H. Abdi, L. J. Williams, Principal component analysis: Principal component analysis, *Wiley Interdisciplinary Reviews: Computational Statistics*, 2:433–459 (2010).

[7] M. Hubert, P. J. Rousseeuw, K. Vanden Branden, ROBPCA: A New Approach to Robust Principal Component Analysis, *Technometrics*, 47:64–79 (2005).

[8] E. J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis?, *Journal of the ACM*, 58:1–37 (2011).

[9] I. Scherl, B. Strom, J. K. Shang, O. Williams, B. L. Polagye, S. L. Brunton, Robust Principal Component Analysis for Modal Decomposition of Corrupt Fluid Flows (2019), Publisher: arXiv Version Number: 2.

[10] K. Manohar, B. W. Brunton, J. N. Kutz, S. L. Brunton, Data-Driven Sparse Sensor Placement for Reconstruction: Demonstrating the Benefits of Exploiting Known Patterns, *IEEE Control Systems*, 38:63–86 (2018).

[11] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, *Neural Computation*, 9:1735–1780 (1997).

[12] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Gated Feedback Recurrent Neural Networks (2015), Publisher: arXiv Version Number: 4.

[13] G. Zhang, B. Eddy Patuwo, M. Y. Hu, Forecasting with artificial neural networks:, *International Journal of Forecasting*, 14:35–62 (1998).

[14] A. R. Mohanty, *Machinery Condition Monitoring: Principles and Practices*, 0th edition, CRC Press, 2014.

[15] C. Gkerekos, I. Lazakis, G. Theotokatos, Ship machinery condition monitoring using performance data through supervised learning, 2017.

[16] J. Shlens, A Tutorial on Principal Component Analysis (2014), Publisher: arXiv Version Number: 1.

[17] T. Zhang, Analysis of multi-stage convex relaxation for sparse regularization, *Journal of Machine Learning Research*, 11:1081–1107 (2010).

[18] Z. Lin, M. Chen, Y. Ma, The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices (2010), Publisher: arXiv Version Number: 3.

[19] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research*, 15:1929–1958 (2014).