This publication is licensed under a CC BY-NC 4.0

Research Article
Received: March 17, 2025 *Accepted: April 6, 2025* *Published: April 29, 2025*
ISSN 2658-5553

Machine learning methods for building reduced-order models

Avdonyushkin, Dmitrij Viktorovich[1*]
Matveeva, Anastasia Igorevna[1]
Kravchinskij, Sergej Andreevich[1]
Karchevskaia Anna Stanislavovna[1]
Novokshenov, Aleksei Dmitrievich[1]
Borovkov, Aleksei Ivanovich[1]
Sherbakov, Sergei Sergeevich[2,3]
Klimkovich, Nikita Mikhailovich[2,3]
Podgayskaya Daria Aleksandrovna[2,3]

[1] Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russian Federation; matveeva_ai@spbstu.ru (M.A.I.); avdonyushkin_dv@spbstu.ru (A.D.V.); kravchinskij_sa@spbstu.ru (K.S.A.); karchevskaya_as@spbstu.ru (K.A.S.); novoksh_ad@spbstu.ru (N.A.D.); borovkov@spbstu.ru (B.A.I);
[2] Belarusian State University, Minsk, Belarus; sherbakovss@mail.ru (S.S.S), nikita.klimkovitch@yandex.ru (K.N.M.), dapodgayskaya@gmail.com (P.D.A.)
[3] Joint Institute of Mechanical Engineering of the National Academy of Sciences of Belarus, Minsk, Belarus;
Correspondence:* email avdonyushkin_dv@spbstu.ru; contact phone +79633155591

Keywords:

Machine learning; Surrogate model; Reduced ordered model; Gaussian process regression; SVD; Neural network; Levenberg-Marquardt algorithm

Abstract:

The object of research the development of surrogate and reduced-order models for engineering systems based on machine-learning techniques. The study focuses on replacing time-consuming high-fidelity numerical simulations with computationally efficient models that preserve the accuracy of prediction. The approach is verified using a technical system for which reduced models are built from hydrodynamic simulation data. **Method.** The proposed methodology combines neural networks trained with the Levenberg–Marquardt algorithm and Gaussian process regression with the Matérn kernel. Singular value decomposition is employed to form reduced-order representations of the system. The Levenberg–Marquardt algorithm demonstrated faster convergence and higher stability compared to conventional gradient descent, while Gaussian process regression ensured accurate interpolation of nonlinear dependencies. **Results**. The integration of singular value decomposition with Gaussian process regression enables rapid reconstruction of the system state vector within seconds while maintaining adequate model fidelity. The developed surrogate models provide reliable approximation of high-fidelity simulation results and significantly reduce computational time. The obtained results confirm the effectiveness of the proposed approach for accelerating engineering analysis and creating digital-twin-based predictive models.

1 Introduction

Развитие цифровых технологий в инженерии сопровождается активным внедрением концепции цифровых двойников, которые объединяют физические и виртуальные представления

This publication is licensed under a CC BY-NC 4.0

изделий. Это открывает новые возможности для проектирования, эксплуатации и совершенствования технических систем, делая процесс их жизненного цикла более управляемым и эффективным. Концепция цифровых двойников применяется в различных предметных областях, включая промышленное производство, управление качеством, авиационную и судостроительную отрасли [1], [2]. В промышленной инженерии цифровые двойники используются для анализа эксплуатационных режимов и оптимизации технологических процессов [3], а в морских и транспортных системах — для моделирования и прогнозирования поведения сложных инженерных объектов [4]. Разработка цифрового двойника включает разработку семейства высокоадекватных математических и компьютерных моделей, которые должны подробно описывать свойства, состояния, поведение и изменения реального объекта на протяжении всего его жизненного цикла, включая разработку, изготовление и эксплуатацию [5].

Однако создание и использование таких моделей требуют высоких вычислительных ресурсов, поскольку численные расчеты, выполняемые в многодисциплинарных пакетных решателях, могут быть весьма трудоемкими. На этапе эксплуатации изделия это ограничение становится критичным, поскольку для поддержки цифрового двойника требуется получение результатов моделирования вблизи реального времени. Поэтому сегодня многие исследования направлены на повышение скорости расчетов при сохранении их точности. Для этого используются модели пониженного порядка [6]–[8] и суррогатные модели [9], которые интегрируются в системную модель цифрового двойника, и позволяет на основе исторических данных выявлять сложные зависимости между входными и выходными параметрами.

Благодаря способности обучения с учителем делать предсказания в реальном времени, технологии искусственного интеллекта и машинного обучения нашли широкое применение в концепции цифровых двойников [10], [11]. Суррогатные модели, представляющие собой приближенные модели реальных систем, играют ключевую роль в этих приложениях. Суррогатные моделирование активно используется для предсказания отдельных интересующих величин, а также характера их изменения во времени.

Одним из примеров применения суррогатных моделей является прогнозирование эксплуатационных характеристик судов с помощью искусственных нейронных сетей. Такие модели позволяют воспроизводить взаимосвязь между скоростью, нагрузкой и расходом топлива, обеспечивая точный прогноз параметров движения и оптимизацию режимов эксплуатации [12], [13].

В другом исследовании [14] нейронная сеть применяется для предсказания перемещений в композитных трубах под воздействием нагрузок с разной скоростью нагружения, что позволяет повысить надежность и долговечность таких труб в различных условиях эксплуатации. Стоит отдельно отметить, что суррогатные модели на основе нейронных сетей позволяют значительно ускорить расчёты и лучше воспроизводить сложные нелинейные зависимости, однако остаются чувствительными к объёму и качеству обучающих данных [15].

В инженерной диагностике существенную роль играют модели-классификаторы: систематические обзоры показывают устойчивый рост их применения для задач обнаружения и распознавания неисправностей в механических системах и производстве. В рамках этих задач используются алгоритмы машинного обучения различной сложности, включая как классические методы, так и нейросетевые архитектуры [16].

В машиностроении и энергетике модели-классификаторы применяются для анализа сигналов вибрации и изображений в задачах обнаружения дефектов и классификации неисправностей, обеспечивая высокую точность распознавания даже при сложных эксплуатационных условиях. Они активно используются для обнаружения повреждений в ветряных электрогенераторах [17,18], что помогает предотвратить аварии и улучшить эксплуатационные характеристики. Модели классификации применяются для прогнозирования неисправностей оборудования автоклавов [19], что позволяет сократить время простоя и затраты на техническое обслуживание.

Ещё одно активно развивающееся направление в цифровых двойниках связано с использованием моделей пониженного порядка (ROM), основанных на сокращении размерности исходных физических моделей без существенной потери точности и предназначенных для ускорения расчётов при моделировании сложных многопараметрических систем [20], [21]. Такой подход особенно востребован в задачах гидродинамики и теплоэнергетики, где проведение многовариантных численных экспериментов требует значительных вычислительных ресурсов.

This publication is licensed under a CC BY-NC 4.0

Модели пониженного порядка применяются при создании цифровых двойников ядерных реакторов [22], угольных [23] и газовых котлов [24], обеспечивая ускоренный анализ сложных нестационарных процессов и возможность оперативного прогнозирования. В последние годы активно развиваются гибридные подходы, сочетающие методы ROM и машинного обучения, что позволяет улучшать аппроксимацию нелинейных эффектов и адаптивно обновлять модели при изменении параметров системы [25].

В настоящем исследовании предлагается комплексный подход к построению суррогатных моделей и моделей пониженного порядка, направленный на повышение эффективности инженерных расчётов при сохранении их точности. Рассматриваются нейронные сети, обучаемые методом Левенберга–Марквардта, и регрессия гауссовских процессов с ядром Матерна, а также реализована редуцированная модель на основе сингулярного разложения (SVD-ROM).

Целью настоящего исследования стало создание интегрированной модели, объединяющей подходы машинного обучения и редуцирования размерности, способной воспроизводить результаты численного моделирования и обеспечивать быстрое восстановление состояния системы в режиме, близком к реальному времени. Для достижения этой цели были поставлены следующие задачи:

- реализовать и обучить отдельные суррогатные модели на основе нейронных сетей и регрессии гауссовских процессов;
- разработать модель пониженного порядка на основе сингулярного разложения (SVD-ROM);
- провести сравнение точности, устойчивости и вычислительной производительности отдельных моделей на едином наборе CFD-данных;
- построить комбинированную модель, объединяющую SVD-ROM и регрессию гауссовских процессов, и оценить её точность при восстановлении состояния системы;
- проанализировать потенциал интегрированного подхода для использования в цифровых двойниках и в инженерных расчётах, требующих высокой скорости отклика.

2 Materials and Methods

2.1 Surrogate models

Суррогатная модель не описывает объект напрямую, а аппроксимирует некоторую сложную модель этого объекта на основе обучающего набора данных. Данные для обучения получают путем запуска расчетов полной модели в ограниченном пространстве входных параметров. Набор пар входных и выходных параметров образует обучающую выборку.

Для создания суррогатных моделей широко используются различные статистические методы и методы машинного обучения, такие как нейронные сети, случайные леса, метод опорных векторов, регрессия гауссовских процессов. Эти методы позволяют эффективно моделировать сложные зависимости между входными и выходными данными, когда точная математическая формула этих зависимостей неизвестна. Методы машинного обучения подразделяются на группы: обучение с учителем, обучение без учителя и обучение с подкреплением. Обучение без учителя использует алгоритмы для обнаружения скрытых закономерностей в неразмеченных данных. В обучении с подкреплением модель учится взаимодействовать с окружающей средой для максимизации награды методом проб и ошибок. В данной работе рассматриваются методы обучения с учителем, которые требуют размеченные данные, а именно нейронные сети и регрессия гауссовских процессов.

При заданных размеченных данных, состоящих из пар входа и выхода $(x_1, y_1), \ldots, (x_n, y_n)$ задача сводится к поиску аппроксимирующей функции, способной преобразовывать пространство параметров $\mathbf{x}$ в $\mathbf{y}$. Искомая функция описывается выражением (1).

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{y}, \tag{1}$$

где $\mathbf{w}$ — набор параметров, который контролирует поведение модели.

This publication is licensed under a CC BY-NC 4.0

2.2 Neural networks for surrogate models

Суррогатная модель не описывает объект напрямую, а аппроксимирует некоторую сложную модель этого объекта на основе обучающего набора данных. Данные для обучения получают путем запуска расчетов полной модели в ограниченном пространстве входных параметров. Набор пар входных и выходных параметров образует обучающую выборку.

Цель обучения с учителем состоит в поиске оптимальных параметров $\mathbf{w}$, таким образом, чтобы минимизировать ошибку предсказания на обучающих данных. Математическая формулировка задачи выглядит следующим образом:

$$\mathbf{w} := \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - f\left(x_i, \mathbf{w}\right) \right)^2. \tag{2}$$

Таким образом, задача обучения с учителем сводится к оптимизации: нахождению таких параметров модели $\mathbf{w}$, которые минимизируют функцию потерь на тренировочном наборе данных. Это позволяет модели наиболее точно предсказывать результаты на новых данных, основываясь на знаниях, полученных из обучающего набора.

В нейронных сетях параметры $\mathbf{w}$ называются весовыми коэффициентами и являются искомыми параметрами при минимизации функции потерь. Наиболее эффективными алгоритмами оптимизации для обучения нейронных сетей являются метод наискорейшего спуска, адаптивный градиентный спуск, метод Ньютона, метод Гаусса-Ньютона и метод Левенберга-Маркадта. Исследования показывают, что ньютоновские методы обычно превосходят метод наискорейшего спуска по эффективности. Однако это преимущество уменьшается по мере увеличения размеров нейронной сети. Из-за ограниченного объема данных, полученных в результате расчетов полной физической модели, глубокие нейронные сети могут привести к переобучению. Поэтому использование глубоких сетей в рассматриваемых инженерных задачах нецелесообразно. Мы сосредотачиваемся на нейронных сетях небольшого размера, для которых метод Левенберга-Маркадта оказывается наиболее эффективным, обеспечивая надежные результаты и избегая излишней сложности. Далее рассмотрим этот метод подробнее.

Алгоритм Левенберга-Марквардта. Алгоритм Левенберга-Марквардта является итерационным методом, разработанным Кенни Левенбергом и Дональдом Марквардтом для численного решения задачи минимизации нелинейных функций. Параметр алгоритма Левенберга-Марквардта может принимать значения, превращая его в метод градиентного спуска или метод Гаусса-Ньютона. В случаях, когда текущее решение значительно отличается от локального минимума, алгоритм действует как градиентный спуск, а при приближении к локальному минимуму — как метод Гаусса-Ньютона. Благодаря этому, метод обладает как высокой скоростью сходимости, так и стабильностью. Однако, один из недостатков заключается в необходимости вычислять обратную матрицу Гессе на каждом шаге, что требуется для корректировки весов в нейронных сетях. В контексте нейронных сетей данный алгоритм эффективен при работе с выборками маленького и среднего размера.

Пусть задан вектор экспериментальных данных $\mathbf{d}$ и вектор ожидаемых значений суррогатной модели $\mathbf{y}$. Представим целевую функцию в виде:

$$E(\mathbf{w}) = \frac{1}{2} \sum_{s=1}^{m} \left[e_s(\mathbf{w}) \right]^2, \tag{3}$$

где невязка $e_s = y_s\left(\mathbf{w}\right) - d_s$, $\mathbf{w}$ — параметры суррогатной модели.

Разложим в ряд Тейлора функцию ошибки $E(w)$ в окрестности точки w_k до второго порядка малости:

$$E\left(w_k + \Delta w\right) \approx E\left(w_k\right) + \nabla E\left(w_k\right)^T \Delta w + \frac{1}{2} \Delta w^T \nabla^2 E\left(w_k\right) \Delta w_k. \tag{4}$$

Градиент функции ошибки записывается следующим образом:

This publication is licensed under a CC BY-NC 4.0

$$\nabla \mathrm{E}(\mathbf{w}) = \sum_{s=1}^{m} e_s(\mathbf{w}) \frac{\partial e_s(\mathbf{w})}{\partial \mathbf{w}}. \tag{5}$$

Градиент также может быть представлен следующим образом в матричном виде:

$$\nabla \mathrm{E}(\mathbf{w}) = \mathbf{J}^{\mathbf{T}}(\mathbf{w})\mathbf{e}(\mathbf{w}), \tag{6}$$

где Якобиан:

$$\mathbf{J}(\mathbf{w}) = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \frac{\partial e_1}{\partial w_2} & \cdots & \frac{\partial e_1}{\partial w_p} \\ \frac{\partial e_2}{\partial w_1} & \frac{\partial e_2}{\partial w_2} & \cdots & \frac{\partial e_2}{\partial w_p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_m}{\partial w_1} & \frac{\partial e_m}{\partial w_2} & \cdots & \frac{\partial e_m}{\partial w_p} \end{bmatrix}, \tag{7}$$

вектор невязок:

$$\mathbf{e}(\mathbf{w}) = \begin{pmatrix} e_1(\mathbf{w}) \\ e_2(\mathbf{w}) \\ \vdots \\ e_m(\mathbf{w}) \end{pmatrix}. \tag{8}$$

Распишем вторую производную функции ошибки:

$$\nabla^2 E(\mathbf{w}) = \sum_{s=1}^{m} \left[\left(\frac{\partial e_s(\mathbf{w})}{\partial \mathbf{w}} \right)^T \cdot \frac{\partial e_s(\mathbf{w})}{\partial \mathbf{w}} + e_s(\mathbf{w}) \frac{\partial^2 e_s(\mathbf{w})}{\partial \mathbf{w}^2} \right]. \tag{9}$$

В матричной форме имеет следующий вид:

$$\nabla^2 E(\mathbf{w}) = \mathbf{J}^T(\mathbf{w})\mathbf{J}(\mathbf{w}) + \mathbf{S}(\mathbf{w}). \tag{10}$$

Получим следующее уравнение путем дифференцирования по w разложения $E(w)$ в ряд Тейлора:

$$\nabla E(\mathbf{w}) + \nabla^2 E(\mathbf{w}) \delta w_k = 0. \tag{11}$$

Введем обозначения градиента $\mathbf{G}(\mathbf{w}) = \nabla E(\mathbf{w})$ и матрицы Гессиана $\mathbf{H}(\mathbf{w}) = \nabla^2 E(\mathbf{w})$:

$$\mathbf{G}(\mathbf{w}) + \mathbf{H}(\mathbf{w})\Delta \mathbf{w} = 0, \tag{12}$$

$$\Delta \mathbf{w} = -\mathbf{H}^{-1}(\mathbf{w})\mathbf{G}(\mathbf{w}). \tag{13}$$

Сравнив матричную форму с покомпонентной, можно заметить, что $\mathbf{S}(\mathbf{w})$ содержит вторые производные относительно $\mathbf{w}$. В алгоритме Ньютона-Гаусса $\mathbf{S}(\mathbf{w})$ пренебрежимо мало, поэтому матрица Гессиана:

$$\mathbf{H}(\mathbf{w}) \approx \mathbf{J}^T(\mathbf{w})\mathbf{J}(\mathbf{w}). \tag{14}$$

Таким образом, корректировка параметров суррогатной модели на $k+1$ итерации для Ньютона-Гаусса, осуществляет по формуле:

This publication is licensed under a CC BY-NC 4.0

$$w^{k+1} = w^k - [\mathbf{J}^T \mathbf{J}(\mathbf{w})]^{-1} \mathbf{e}(\mathbf{w}). \quad (15)$$

Однако, стоит отметить, что алгоритм столкнется с проблемой сходимости, если матрица $\mathbf{J}^T(\mathbf{w})\mathbf{J}(\mathbf{w})$ будет невырожденной. Сущность алгоритма Левенберга-Марквардта состоит в аппроксимации $\mathbf{S}(\mathbf{w})$ с помощью $\mu\mathbf{I}$, где μ — параметр Левенберга-Марквардта, $\mathbf{I}$ — единичная матрица.

$$\mathbf{H}(\mathbf{w}) \approx \mathbf{J}^T(\mathbf{w})\mathbf{J}(\mathbf{w}) + \mu\mathbf{I}. \quad (16)$$

В таком случае, матрица Гессиана будет вырожденной, так как элементы главной диагонали будут положительные. Таким образом, корректировка параметров суррогатной модели на k+1 итерации, осуществляет по формуле:

$$w^{k+1} = w^k - [\mathbf{J}^T \mathbf{J}(\mathbf{w}) + \mu\mathbf{I}]^{-1} \mathbf{e}(\mathbf{w}). \quad (17)$$

Стратегия изменения диагональных элементов называется демпфированием, поэтому μ называют коэффициентом демпфирования. Когда параметр μ равен 0, алгоритм вырождается в метод Ньютона-Гаусса, когда μ велико, алгоритм становится градиентным спуском. Таким образом, μ уменьшается после каждого успешного шага (уменьшение целевой функции) и увеличивается только тогда, когда предварительный шаг приведет к увеличению целевой функции. Таким образом, целевая функция всегда снижается на каждой итерации алгоритма для обеспечения быстрой сходимости.

Для прерывания алгоритма Левенберга-Марквардта обычно применяются следующие условия:

- Предел итераций. Если количество итераций достигает заранее установленного максимального значения.
- Малое изменение функции. Если изменение значения целевой функции между итерациями становится меньше заранее заданного порога.
- Малое изменение параметров. Если изменения параметром модели между итерациями становятся меньше определенного порога.
- Малое значение градиента. Если норма градиента становится меньше заданного порога, это может указывать на то, что метод достиг точки, где градиент близок к нулю.

2.3 Gaussian process regression

Гауссовский процесс - набор случайных величин, каждое конечное подмножество которого подчиняется нормальному распределению. Гауссовский процесс $f(x)$ полностью описывается ковариационной функцией и средним.

$$m(x) = M[f(x)], \quad (18)$$

$$k(x,x') = M\left[\left(f(x) - m(x)\right)\left(f(x') - m(x')\right)\right], \quad (19)$$

где M - оператор математического ожидания.

Часто математическое ожидание принимается за ноль и тогда процесс определяется только ковариационной функцией $k(x,x')$, так же называемым ядром. Пусть $\mathbf{f}_*$ — значения процесса в точках $\mathbf{X}_*$, а $\mathbf{f}$ — значения процесса в точках $\mathbf{X}$ тогда они распределены по следующему нормальному закону:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} \mathbf{K}(\mathbf{X},\mathbf{X}) & \mathbf{K}(\mathbf{X},\mathbf{X}_*) \\ \mathbf{K}(\mathbf{X}_*,\mathbf{X}) & \mathbf{K}(\mathbf{X}_*,\mathbf{X}_*) \end{bmatrix}\right), \quad (20)$$

где $\mathbf{K}(\mathbf{X},\mathbf{X}), \mathbf{K}(\mathbf{X}_*,\mathbf{X}), \mathbf{K}(\mathbf{X},\mathbf{X}_*), \mathbf{K}(\mathbf{X}_*,\mathbf{X}_*)$ — матрицы значений ковариационной функции.

Если допустить, что $\mathbf{f}$ — известная(обучающая) выборка, то распределение $\mathbf{f}_*$ примет вид:

This publication is licensed under a CC BY-NC 4.0

$$\mathbf{f}_* \mid \mathbf{f} \sim \left(\mathbf{K}(\mathbf{X}_*,\mathbf{X})\mathbf{K}(\mathbf{X},\mathbf{X})^{-1}\mathbf{f}, \mathbf{K}(\mathbf{X}_*,\mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*,\mathbf{X})\mathbf{K}(\mathbf{X},\mathbf{X})^{-1}\mathbf{K}(\mathbf{X},\mathbf{X}_*)\right), \tag{21}$$

Таким образом, взяв за аппроксимацию значений f_* математическое ожидание полученного распределения, получаем:

$$\tilde{\mathbf{f}}_* = \mathbf{K}(\mathbf{X}_*,\mathbf{X})\mathbf{K}(\mathbf{X},\mathbf{X})^{-1}\mathbf{f}. \tag{22}$$

Тогда оценка в конкретной точке будет линейной комбинацией ковариаций точки с точками из обучающей выборки:

$$\tilde{f}(x_*) = \sum_{i=1} a_i k(x_i, x_*), \tag{23}$$

$$a = \mathbf{K}(\mathbf{X},\mathbf{X})^{-1}\mathbf{f}. \tag{24}$$

Функция $k(x_i, x_*)$ оценивает зависимость значения функции в точке x_i от значения в точке x_*. Обычно для таких целей выбирается функция, зависящая от евклидовой нормы $|x_i - x_*|$.

Несмотря на то, что при построении предиктивной модели не нужно оперировать параметрами ядра, эти параметры влияют на качество модели. Для подбора гиперпараметров модели используют градиентный алгоритм оптимизации с целевой функцией в виде предельного логарифмического правдоподобия [26].

Важно отметить следующие свойства регрессии гауссовских процессов:

- Регрессия гауссовских процессов способна строить предсказания зашумлённых объектов;
- Регрессия гауссовских процессов выдает доверительные интервалы для значений;
- Из-за описанных выше свойств регрессия гауссовских процессов может использоваться не только для интерполяции, но и для предсказания временных рядов.

2.4 Reduced order model (ROM)

Модель пониженного порядка (ROM — Reduced Order Model) представляет собой упрощенную версию высокоточной статической или динамической модели. Она сохраняет ключевые поведенческие характеристики и основные эффекты исходной модели, что позволяет значительно сократить время, необходимое для решения задачи. В настоящей статье рассмотрена модель на базе сингулярного разложения (SVD-ROM). SVD или singular value decomposition — это способ разложения прямоугольной матрицы на три составляющих, каждая из которых обладает определенным смыслом. Разложение выглядит следующим образом:

$$\mathbf{X}_{m\times n} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \tag{25}$$

Само SVD применяется к матрице, составленной из векторов состояния модели при различных параметрах:

$$\mathbf{X} = \begin{bmatrix} \vdots & \vdots & & \vdots \\ x_1 & x_2 & \cdots & x_k \\ \vdots & \vdots & & \vdots \end{bmatrix}, \tag{26}$$

где x_k соответствует вектору параметров μ_k.

$\mathbf{U}$ — матрица сингулярных векторов (сингулярных мод). Столбцы матрицы — это моды, которые показывают состояния, характерные для данной системы. Моды расположены в порядке убывания эффективной модальной массы.

$$\mathbf{U} = \begin{bmatrix} u_{11} & \vdots & & \vdots \\ \vdots & u_2 & \cdots & u_n \\ u_{1m} & \vdots & & \vdots \end{bmatrix}, \tag{27}$$

This publication is licensed under a CC BY-NC 4.0

$\mathbf{\Sigma}$ — диагональная матрица сингулярных чисел. Сингулярные числа положительны и вещественны, они являются корнями из эффективных модальных масс соответствующих сингулярных векторов. Руководствуясь определенными критериями и исследуя матрицу сингулярных чисел, можно отбросить из рассмотрения моды, эффективные модальные массы которых слишком малы в сравнении с остальными. В случае редуцирования матрицы $\mathbf{\Sigma}$, уравнение (25) можно представить в виде $\mathbf{X}_{m\times n} \approx \mathbf{U}_{m\times r}\mathbf{\Sigma}_{r\times r}(\mathbf{V}_{n\times r})^T$.

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots \\ 0 & 0 & \cdots & \sigma_n \end{bmatrix}, \tag{28}$$

$\mathbf{V}$ — матрица размерами $n \times n$ Компонента матрицы v_{ij} показывает коэффициент при j-ой моде, помноженной на j-ю эффективную массу, в i-ой точке пространства параметров при разложении вектора состояния в ряд по сингулярным векторам. Чтобы найти значение вектора состояния в точке пространства параметров, необходимо построить интерполяционную поверхность. Входом для построения поверхности является набор точек в пространстве параметров, продемонстрированных в матрице $\mathbf{X}$, а выход — это набор векторов v_i, соответствующих точкам пространства параметров $\{\mu_i\}$.

Сингулярное разложение во многом напоминает спектральное разложение матрицы. Однако, при помощи матрицы V сингулярного разложения, можно получить взаимосвязь между точкой в пространстве параметров и соответствующим ей вектору состояния. При помощи SVD можно достаточно точно моделировать гармонические и периодические процессы, поэтому сингулярное разложение наиболее эффективно в работе с CFD задачами.

3 Results and Discussion

Примером индустриальной задачи, для решения которой востребовано суррогатной моделирование, является задача по обтеканию препятствия в концентрическом зазоре. Геометрия задачи приведена на (Рис. 1).

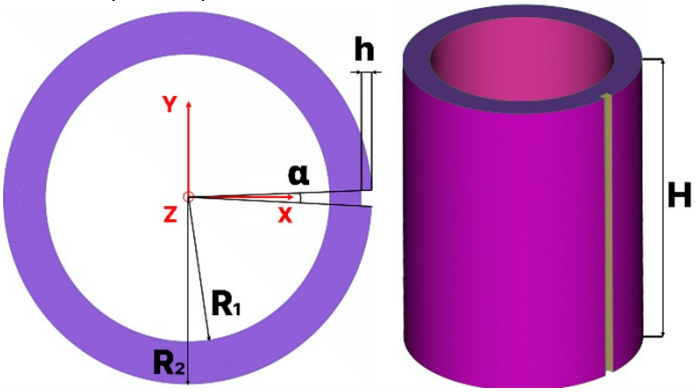


Рис. 1 – Геометрия задачи
Fig. 1 – Model geometry

В данной задаче входными параметрами являются давление и угловая скорость, а выходными — проекции силы f_x, f_y и момент m_z. Обучающий набор данных был получен в результате гидродинамического моделирования с использованием программного обеспечения Ansys Fluent, где набор входных параметров был сгенерирован с помощью метода латинского гиперкуба. Этот метод позволяет разбить пространство входных параметров таким образом, чтобы получить репрезентативную выборку. Входные данные распределены в диапазонах $P \in [20,800]$ Па, $w \in [20,2000]$ рад/с (Рис. 2).

This publication is licensed under a CC BY-NC 4.0

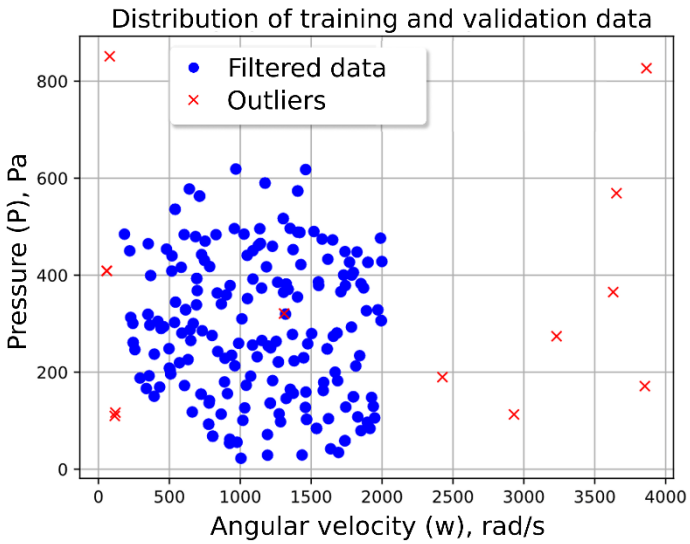


Рис. 2 – Распределение обучающих и валидационных данных. Зависимость давления P (Па) от угловой скорости w (рад/с)
Fig. 2 – Distribution of training and validation data. Dependence of pressure P (Pa) on angular velocity w (rad/s)

3.1 Neural networks

Прежде чем приступить к разработке архитектуры нейронной сети, необходимо провести предварительную обработку данных. Входные данные были стандартизированы, а выходные — нормализованы. Эти процедуры помогают обеспечить более стабильное обучение нейронной сети. После нормализации выходные данные были приведены к диапазону от 0 до 1, но основная их часть сосредоточилась в интервале от 0 до 0.2. Применение десятичного логарифма к выходным данным помогло равномернее распределить их по пространству значений.

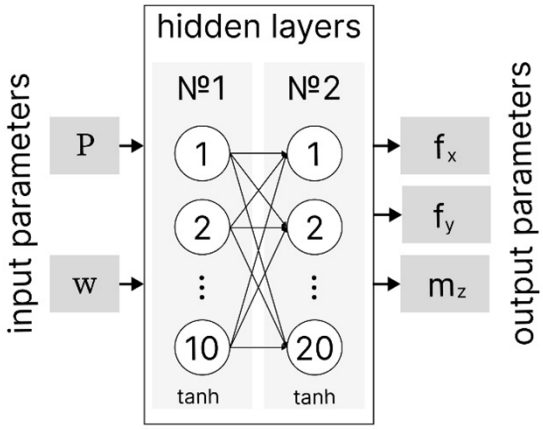


Рис. 3 – Структура нейронной сети
Fig. 3 – Neural network structure

Всего для обучения было использовано 195 экспериментов, которые были разделены на тренировочную и валидационную выборки в соотношении 9:1. Архитектура нейронной сети изображена на (Рис. 3), она включает в себя два скрытых слоя, содержащих 10 и 20 нейронов соответственно. Функцией активации для обоих слоев является гиперболический тангенс, который позволяет модели учитывать нелинейные зависимости в данных. Для оценки качества модели используется среднеквадратичная ошибка (MSE):

$$MSE = \sum_{i=1}^{N} \frac{1}{N} (y_i - d_i)^2, \tag{29}$$

где y — предсказанное значение, d — ожидаемое значение, N — количество экспериментов.

This publication is licensed under a CC BY-NC 4.0

Для обучения нейронной сети был использован метод Левенберга-Марквардта. Процесс обучения занял 150 эпох, что позволило сети достичь хорошей сходимости и точности. Время, затраченное на обучение, составило 2.12 секунды, что указывает на эффективность использования метода.

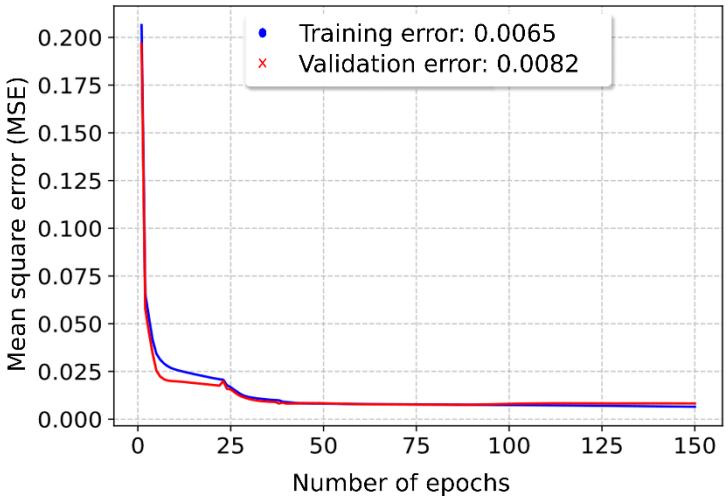


Рис. 4 – Зависимость среднеквадратичной ошибки от количества эпох тренировочной и валидационной выборок
Fig. 4 – Mean square error (MSE) dependence on the number of epochs

Для визуализации процесса обучения был построен график зависимости среднеквадратичной ошибки (MSE) от количества эпох для тренировочных и валидационных данных. Этот график показывает, как изменяется ошибка сети на тренировочном и валидационном наборах данных по мере увеличения числа эпох.

Обычно по графику можно наблюдать снижение ошибки на тренировочных данных, что свидетельствует об успешном обучении модели. Также важно следить за поведением ошибки на валидационных данных, чтобы предотвратить переобучение модели. Если ошибка на валидационных данных начинает возрастать, это может указывать на то, что сеть начинает запоминать данные из тренировочной выборки, а не обучаться обобщению.

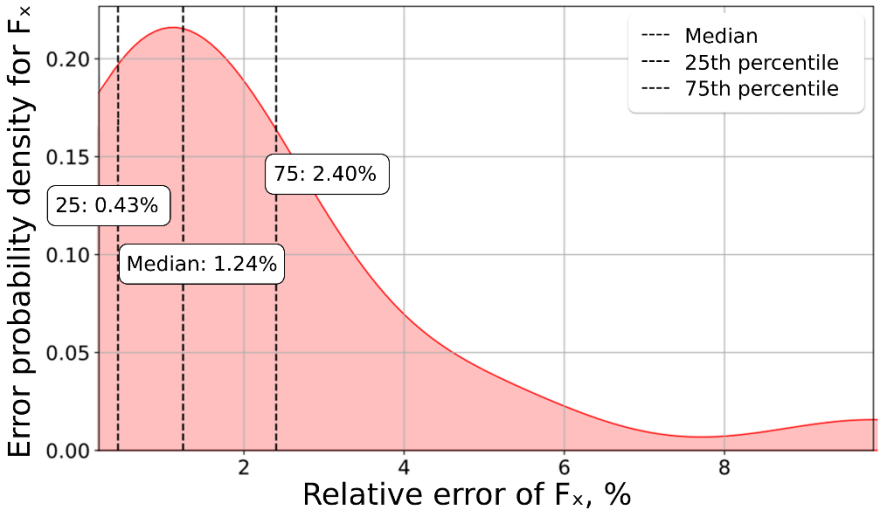


Рис. 5 – Распределение плотности относительной погрешности f_x
Fig. 5 – Error probability density for f_x

This publication is licensed under a CC BY-NC 4.0

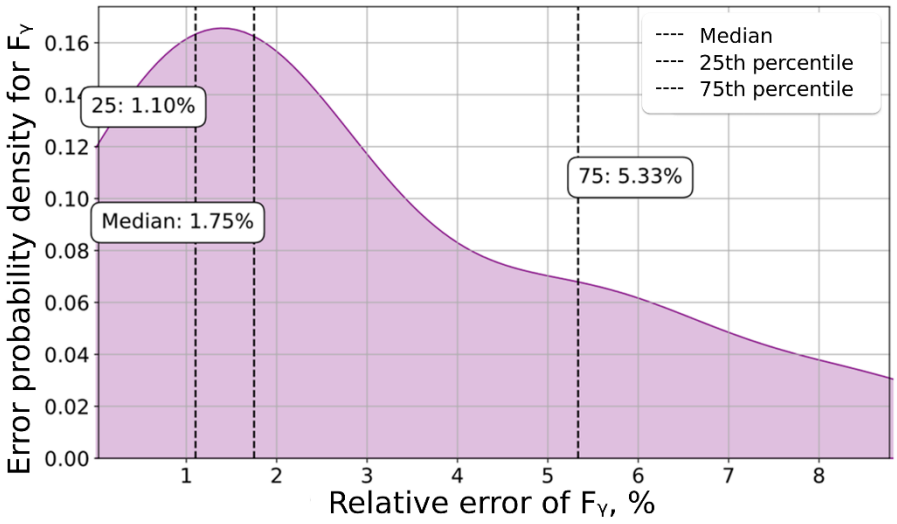


Рис. 6 – Распределение плотности относительной погрешности f_y
Fig. 6 – Error probability density for f_y

В качестве результатов представлены графики плотности распределения относительной погрешности для трех выходных параметров — проекций силы f_x, f_y и момента m_z. Для каждой выходной переменной плотность распределения показывает частоту значений относительной погрешности.

На графиках отмечены 25-ый, 75-ый квантиль и медианное значение. Эти статистические показатели помогают оценить точность предсказаний модели. Из графиков можно сделать вывод, что для всех выходных значений медианная относительная погрешность не превысила 1.75%.

75-й квантиль указывает на погрешность, которую не превышает 75% предсказаний. Согласно этим показателям, для 75% всех предсказаний относительная погрешность составила менее 2.4% для f_x, менее 5.33% для f_y и менее 1.71% для m_z.

Эти результаты свидетельствуют о том, что нейронная сеть демонстрирует хорошую точность и стабильность в предсказаниях, с большей частью ошибок, находящихся в пределах приемлемых границ. Высокие значения точности, указанные через медиану и квартили, подтверждают, что модель в основном успешно справляется с задачей предсказания выходных параметров, хотя есть пространство для улучшений в точности предсказания f_y.

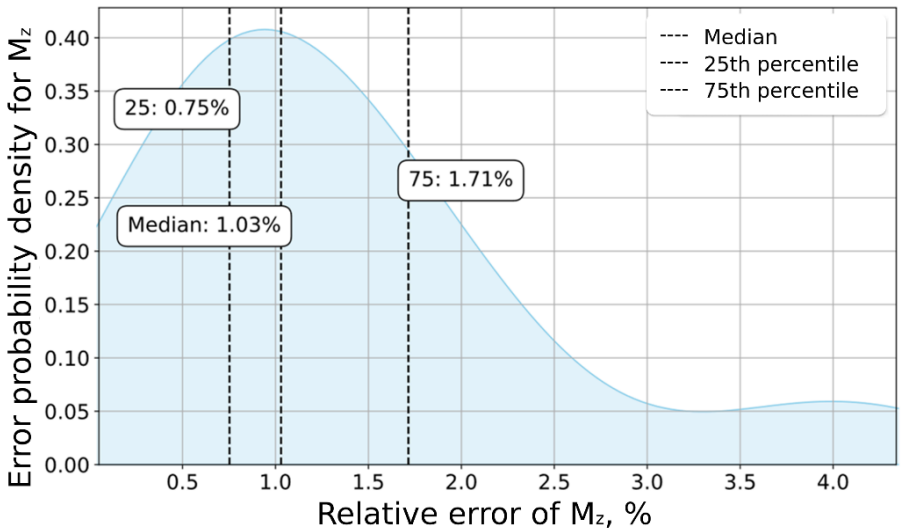


Рис. 7 – Распределение плотности относительной погрешности m_z
Fig. 7 – Error probability density for m_z

This publication is licensed under a CC BY-NC 4.0

3.2 Gaussian process regression

Прежде чем приступить к разработке архитектуры нейронной сети, необходимо провести предварительную обработку данных. Входные данные были стандартизированы, а выходные — нормализованы. Эти процедуры помогают обеспечить более стабильное. Для обучения гауссовского процесса используем библиотеку scikit-learn и модуль GaussianProcessRegressor. Основным компонентом гауссовского процесса является ковариационная функция (ядро). Оно используется для определения структуры зависимости между данными. В данном случае используется функция Matern, которая задаётся следующим образом:

$$k(x_i, x_j) = \frac{1}{\Gamma(v)2^{v-1}}\left(\frac{\sqrt{2v}}{l}d(x_i, x_j)\right)^v K_v\left(\frac{\sqrt{2v}}{l}d(x_i, x_j)\right), \quad (30)$$

где $d = |x - x'|$ — расстояние между двумя точками в пространстве признаков, l — длина масштаба, регулирующая степень зависимости между точками, v — параметр, определяющий гладкость функции, K_v — модифицированная функция Бесселя второго рода, $\Gamma(v)$ — гамма-функция.

В данной модели задано значение параметра гладкости $v = 1.5$, что обеспечивает умеренную степень гладкости. При этом более высокие значения v приводят к ещё более гладким предсказаниям. Уровень гладкости, получаемый при этом значении, находится между гладкостью экспоненциального и RBF ядер.

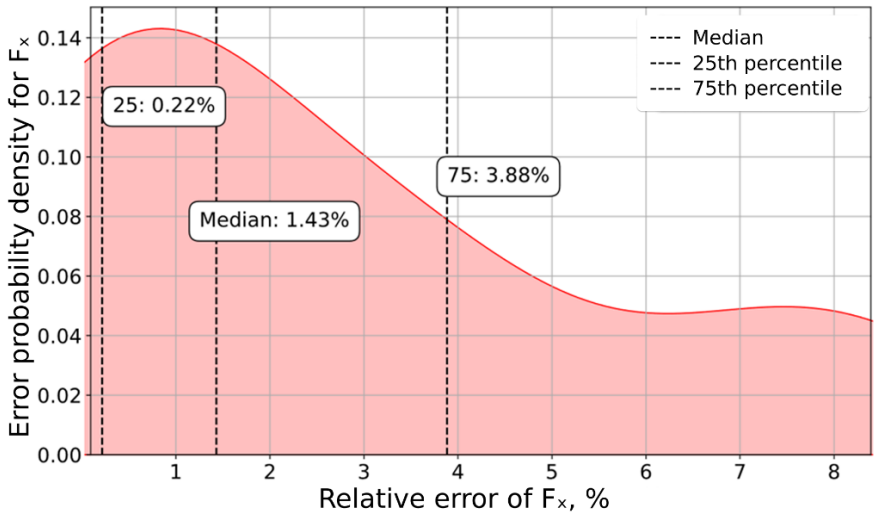
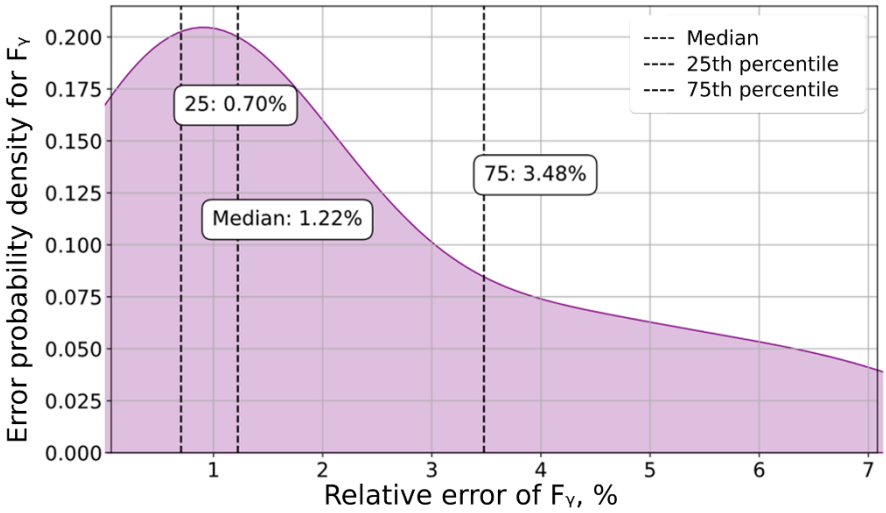


Рис. 8 – Распределение плотности относительной погрешности f_x
Fig. 8 – Error probability density for f_x



Avdonyushkin, D.; Matveeva, A.; Kravchinskij, S.; Karchevskaia, A.; Novokshenov, A.; Borovkov, A.; Sherbakov, S.; Klimkovich, N.; Podgayskaya, D.
Machine learning methods for building reduced-order models;
2025; AlfaBuild; **34** Article No 3405. doi: 10.57728/ALF.34.5

This publication is licensed under a CC BY-NC 4.0

Рис. 9 – Распределение плотности относительной погрешности f_y
Fig. 9 – Error probability density for f_y

Модель включает гиперпараметры, такие как длина масштаба l и параметры шума, которые оптимизируются с использованием метода максимального правдоподобия. Процесс обучения занял 3.5 секунды. После оптимизации гиперпараметры были установлены на следующие значения: длина масштаба $l = 1.0$ и уровень шума $1 \cdot 10^{-4}$, при условии, что данные считаются чистыми от ошибок. Оптимизатор выполнил 1000 перезапусков, что позволило ему исследовать 1000 различных начальных точек и избежать локальных минимумов. По завершении обучения модель готова к применению для предсказаний на валидационной выборке. На (Рис. 8–Рис. 10) представлены графики плотности распределения относительной погрешности для трех выходных параметров.

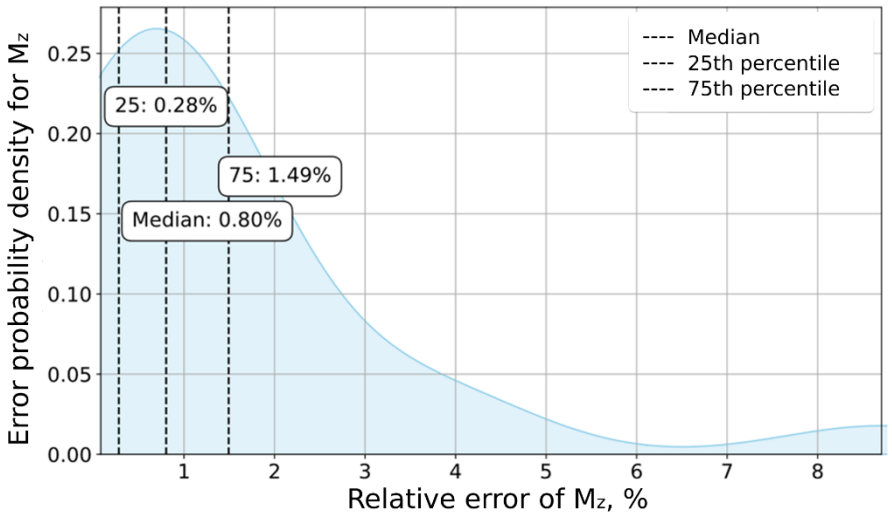


Рис. 10 – Распределение плотности относительной погрешности m_z
Fig. 10 – Relative error density distribution m_z

Можно сделать вывод, что для всех выходных значений медианное значение не превысило 1.43%. 75-ый квантиль указывает, что относительная погрешность у 75% точек от общего количества валидационных пар точек составила менее 3.88% для f_x, менее 3.48% для f_y, менее 1.49% для m_z. Данные результаты свидетельствуют о том, что гауссовский процесс демонстрирует хорошую точность и стабильность в предсказаниях.

3.3 Results comparison

На (Рис. 11–Рис. 13) представлены столбчатые диаграммы, сравнивающие предсказанные значения, полученные с помощью нейронной сети и регрессии гауссовских процессов, с истинными значениями для валидационной выборки.

Над каждым столбцом указана относительная погрешность для соответствующего эксперимента. Эти диаграммы позволяют визуально оценить точность и качество предсказаний обеих моделей, а также сравнить их производительность по отношению к истинным значениям.

This publication is licensed under a CC BY-NC 4.0

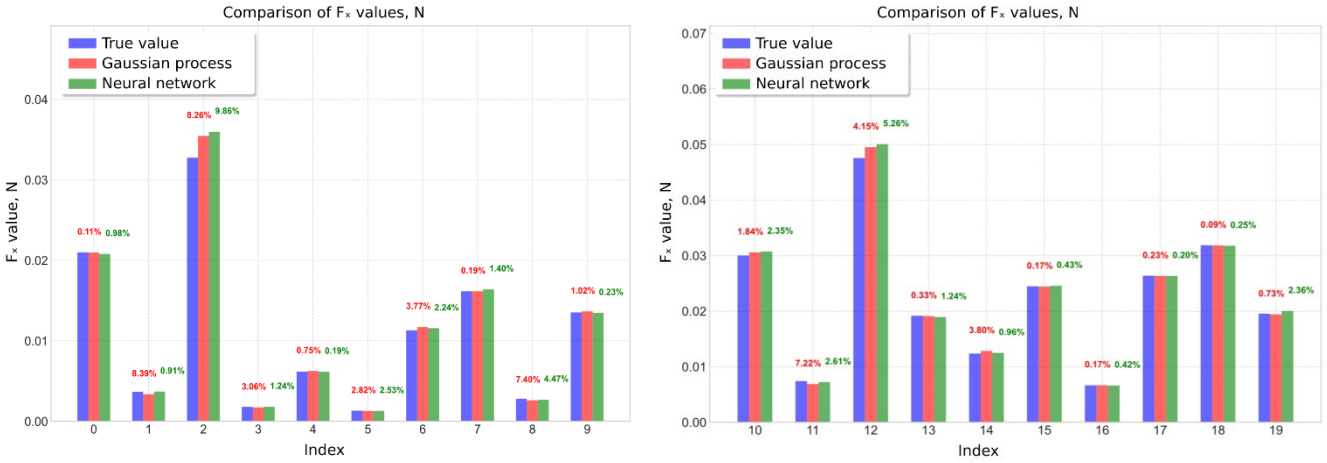


Рис. 11 – Сравнение значений f_x
Fig. 11 – Comparison of f_x values

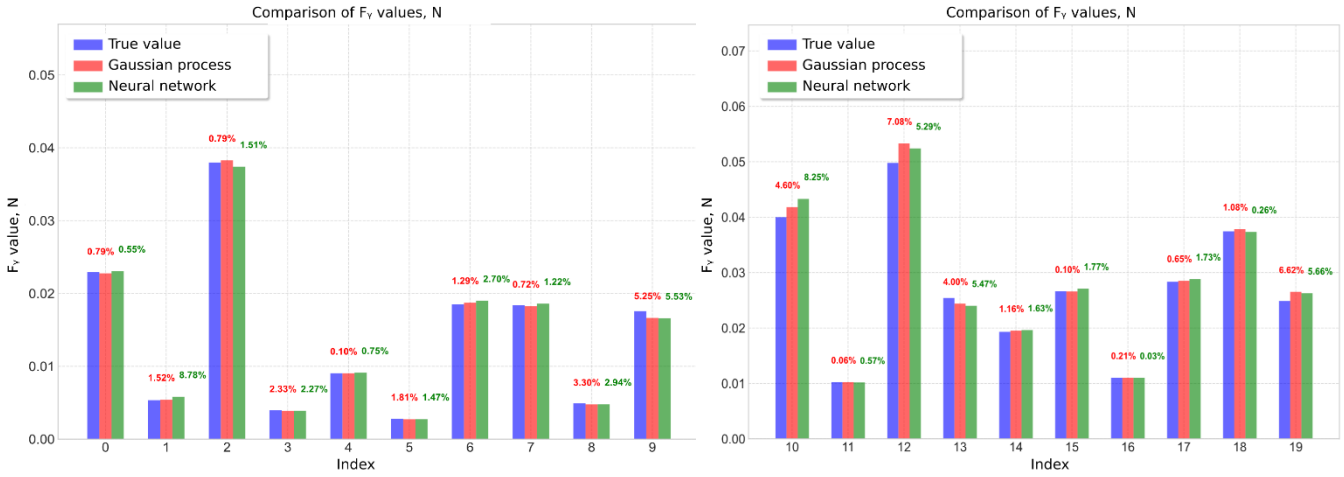


Рис. 12 – Сравнение значений f_y
Fig. 12 – Comparison of f_y values

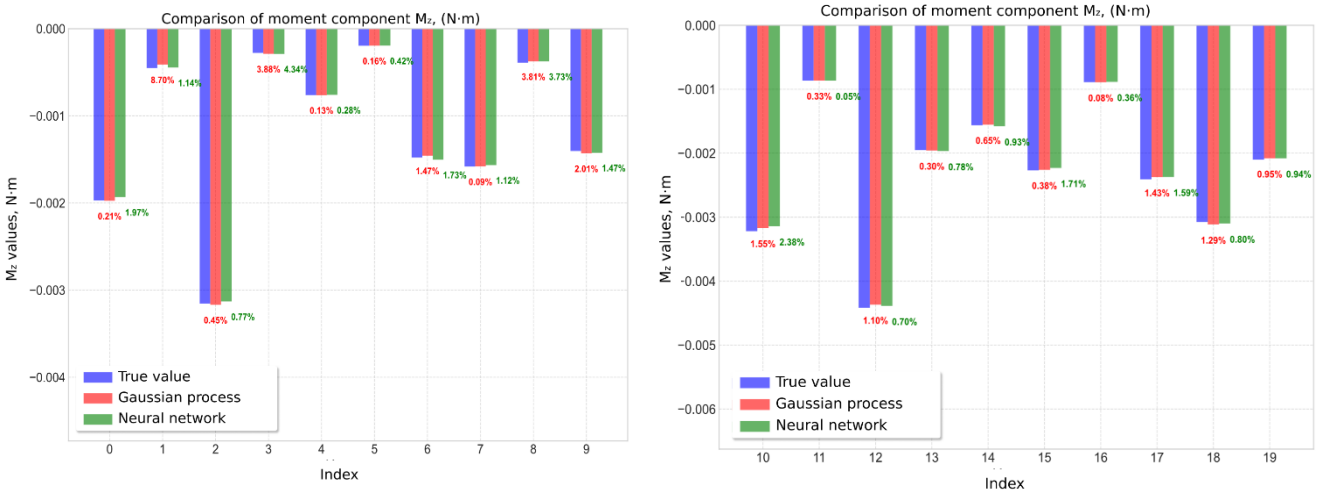


Рис. 13 – Сравнение значений m_z
Fig. 13 – Comparison of m_z values

На диаграммах заметно, что большие погрешности, как для регрессии гауссовского процесса, так и для нейронной сети, возникают на границах выборки. В большинстве экспериментов обе модели демонстрируют либо одинаково высокие относительные погрешности (от 4% до 8%), либо одинаково низкие (менее 4%). Однако, нейронная сеть проявляет большую стабильность в предсказаниях: в некоторых экспериментах относительные погрешности для

This publication is licensed under a CC BY-NC 4.0

гауссовской регрессии значительно выше, чем для нейронной сети. Это указывает на то, что нейронная сеть лучше справляется с задачей предсказания в этих условиях, демонстрируя меньшую вариативность погрешностей и более надёжные результаты.

3.4 Reduced order model (ROM)

В качестве примера индустриальной задачи возьмем ту же физическую модель, что и в примере для суррогатного моделирования. В данной задаче входными параметрами являются давление и угловая скорость, а выходной параметр — модуль вектора скорости. Перед применением процедуры расчета SVD, входные параметры подверглись стандартизации, а выходные нелинейно привелись к нормальному распределению. Для построения интерполяционной поверхности использовалась регрессия гауссовского процесса. Всего для построения модели было использовано 78 экспериментов. Для определения погрешностей была построена модель, не включающая последний эксперимент, а результаты сравнивались с результатами, полученными при помощи полной модели.

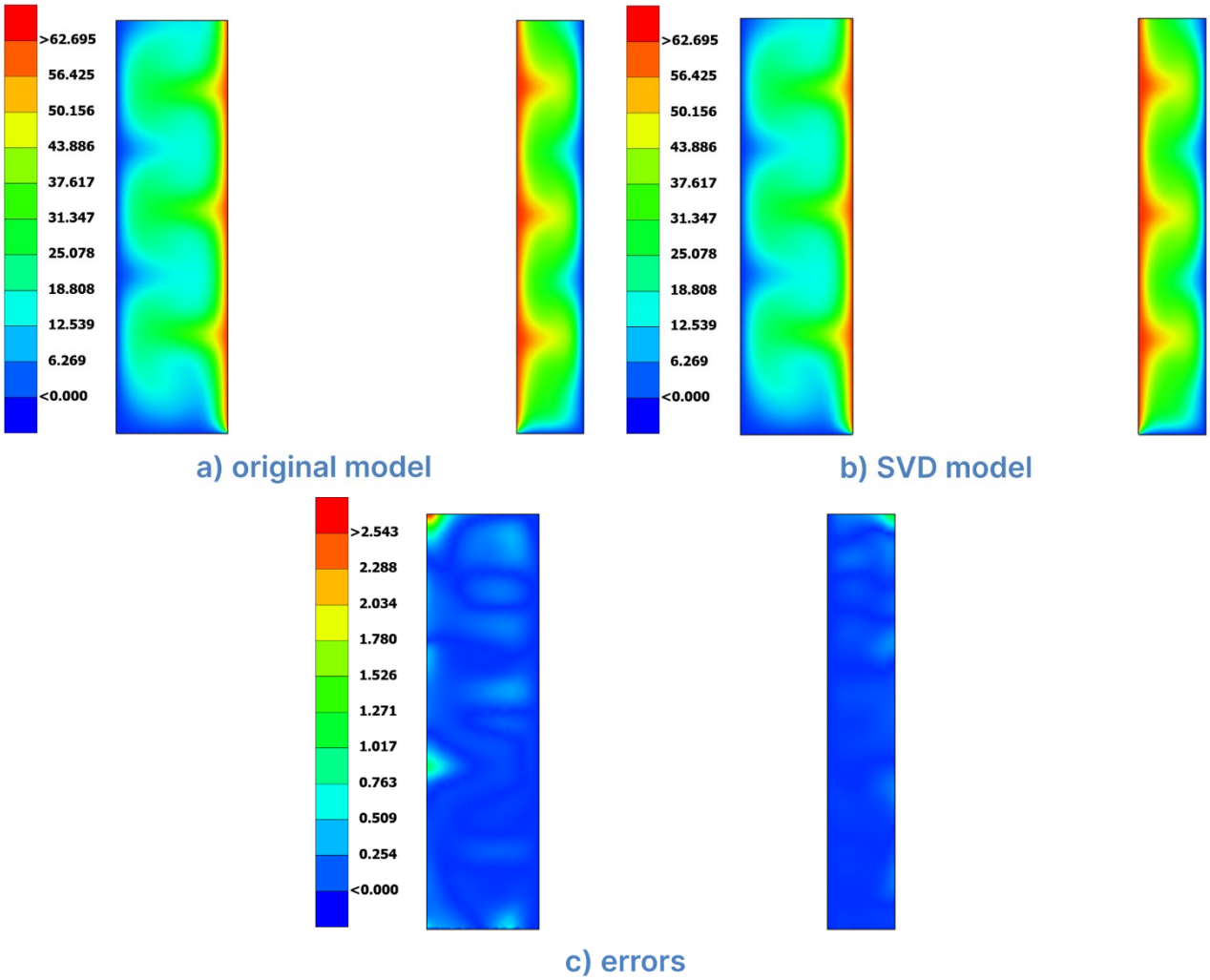


Рис. 14 – Сравнение результатов использования модели пониженного порядка с исходной моделью: a) результаты исходной модели, b) результаты модели, полученной с помощью SVD, c) распределение ошибки в пространстве
Fig. 14 – Comparison of the results of using the reduced-order model with the original model: a) results of the original model, b) results of the model obtained using SVD, c) error distribution in space

Средняя погрешность составила менее 1%, а максимальная погрешность составила около 2,6% и наблюдалась в зоне, где значения были близки к нулю. Для инженеров больший интерес представляют локальные максимумы, в которых SVD продемонстрировала близкие к нулю ошибки.

This publication is licensed under a CC BY-NC 4.0

4 Conclusions

Таким образом, в данной работе были получены следующие основные научные результаты:

1. Разработан комплексный подход к построению моделей пониженного порядка и суррогатных моделей для инженерных задач с высокими вычислительными затратами, объединяющий нейронные сети, регрессию гауссовских процессов и метод сингулярного разложения (SVD-ROM).
2. Выполнено сравнение нейронной сети, обучаемой методом Левенберга–Марквардта, и регрессии гауссовских процессов с ядром Матерна. Показано, что обе модели обеспечивают сопоставимую точность предсказаний, при этом нейронная сеть демонстрирует более устойчивое поведение в отдельных режимах, а гауссовская регрессия — более гладкую интерполяцию и возможность оценки доверительных интервалов.
3. Реализована модель пониженного порядка на основе сингулярного разложения, обеспечивающая эффективное снижение размерности данных при сохранении физических характеристик системы. Средняя погрешность модели составила менее 1%, максимальная — 2.6%.
4. Показано, что применение SVD-ROM позволяет существенно ускорить расчёты и является особенно эффективным для задач вычислительной гидродинамики, где метод обеспечивает выделение доминирующих гармонических и периодических компонент.

Для будущих исследований важно улучшить планирование экспериментов, добавляя входные параметры с учётом физики задачи. В контексте модели, основанной на сингулярном разложении, такой алгоритм должен предлагать вариант, соответствующий каждой из мод, и на основе определенного критерия качества варианта и эффективных модальных масс мод, выбирать одну точку в пространстве параметров. Поскольку на границах выборки погрешности могут быть выше, требуется внедрение рекомендательного алгоритма для суррогатного моделирования. Такой алгоритм должен оценивать возможные риски и информировать инженера, работающего с цифровым двойником, о вероятности точности предсказаний суррогатной модели и модели пониженного порядка.

5 Fundings

Работа выполнена при поддержке Санкт-Петербургского государственного автономного учреждения «Фонд поддержки научной, научно-технической, инновационной деятельности», проект № 23-РБ-09-27 от 15.12.2023, а также при поддержке Белорусского республиканского фонда фундаментальных исследований (проект № Т24СПбГ-003).

References

1 Kukushkin, K., Ryabov, Y. and Borovkov, A. (2022) Digital Twins: A Systematic Literature Review Based on Data Analysis and Topic Modeling. *Data*, MDPI, **7**. https://doi.org/10.3390/data7120173

2 Yao, J.F., Yang, Y., Wang, X.C. and Zhang, X.P. (2023, December 1) Systematic Review of Digital Twin Technology and Applications. Visual Computing for Industry, Biomedicine, and Art, Springer. https://doi.org/10.1186/s42492-023-00137-4

3 Rodionov, N. and Tatarnikova, L. (2021) Digital Twin Technology as a Modern Approach to Quality Management. E3S Web of Conferences, EDP Sciences. https://doi.org/10.1051/e3sconf/202128404013

4 Madusanka, N.S., Fan, Y., Yang, S. and Xiang, X. (2023, May 1) Digital Twin in the Maritime Domain: A Review and Emerging Trends. Journal of Marine Science and Engineering, MDPI. https://doi.org/10.3390/jmse11051021

5 Iliuţă, M.E., Moisescu, M.A., Pop, E., Ionita, A.D., Caramihai, S.I. and Mitulescu, T.C. (2024) Digital Twin—A Review of the Evolution from Concept to Technology and Its Analytical Perspectives on Applications in Various Fields. *Applied Sciences (Switzerland)*, Multidisciplinary Digital Publishing Institute (MDPI), **14**. https://doi.org/10.3390/app14135454

This publication is licensed under a CC BY-NC 4.0

6 Hartmann, D., Herz, M. and Wever, U. (2018) Model Order Reduction a Key Technology for Digital Twins. In: Keiper, W., Milde, A. and Volkwein, S., Eds., Reduced-Order Modeling (ROM) for Simulation and Optimization: Powerful Algorithms as Key Enablers for Scientific Computing, Springer International Publishing, Cham, 167–179. https://doi.org/10.1007/978-3-319-75319-5_8

7 Kantaros, A., Ganetsos, T., Pallis, E. and Papoutsidakis, M. (2025, August 1) From Mathematical Modeling and Simulation to Digital Twins: Bridging Theory and Digital Realities in Industry and Emerging Technologies. Applied Sciences (Switzerland), Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/app15169213

8 He, Y., Li, Y., Li, X., Yuan, Y., Yang, F. and Lu, Z. (2025) A Reduced-Order Algorithm for a Digital Twin Model of Ultra-High-Voltage Valve-Side Bushing Considering Spatio-Temporal Non-Uniformity. *Energies*, Multidisciplinary Digital Publishing Institute (MDPI), **18**. https://doi.org/10.3390/en18061481

9 Bárkányi, Á., Chován, T., Németh, S. and Abonyi, J. (2021, March 1) Modelling for Digital Twins—Potential Role of Surrogate Models. Processes, MDPI AG. https://doi.org/10.3390/pr9030476

10 Borovkov, A.I., Vafaeva, K.M., Vatin, N.I. and Ponyaeva, I. (2024) Synergistic Integration of Digital Twins and Neural Networks for Advancing Optimization in the Construction Industry: A Comprehensive Review. *Construction Materials and Products*, Belgorod V G Shukhov State Technology University, **7**. https://doi.org/10.58224/2618-7183-2024-7-4-7

11 Tao, F., Qi, Q., Liu, A. and Kusiak, A. (2018) Data-Driven Smart Manufacturing. *Journal of Manufacturing Systems*, Elsevier B.V., **48**, 157–169. https://doi.org/10.1016/j.jmsy.2018.01.006

12 Moreira, L., Vettor, R. and Soares, C.G. (2021) Neural Network Approach for Predicting Ship Speed and Fuel Consumption. *Journal of Marine Science and Engineering*, MDPI AG, **9**, 1–14. https://doi.org/10.3390/jmse9020119

13 Nielsen, R.E., Papageorgiou, D., Nalpantidis, L., Jensen, B.T. and Blanke, M. (2022) Machine Learning Enhancement of Manoeuvring Prediction for Ship Digital Twin Using Full-Scale Recordings. *Ocean Engineering*, Elsevier Ltd, **257**. https://doi.org/10.1016/j.oceaneng.2022.111579

14 Ghandourah, E., Khatir, S., Banoqitah, E.M., Alhawsawi, A.M., Benaissa, B. and Wahab, M.A. (2023) Enhanced ANN Predictive Model for Composite Pipes Subjected to Low-Velocity Impact Loads. *Buildings*, MDPI, **13**. https://doi.org/10.3390/buildings13040973

15 Chen, X., Chen, R., Wan, Q., Xu, R. and Liu, J. (2021) An Improved Data-Free Surrogate Model for Solving Partial Differential Equations Using Deep Neural Networks. *Scientific Reports*, Nature Research, **11**. https://doi.org/10.1038/s41598-021-99037-x

16 Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N. and Nandi, A.K. (2020, April 1) Applications of Machine Learning to Machine Fault Diagnosis: A Review and Roadmap. Mechanical Systems and Signal Processing, Academic Press. https://doi.org/10.1016/j.ymssp.2019.106587

17 Ritto, T. and Rochinha, F. (2020) Digital Twin, Physics-Based Model, and Machine Learning Applied to Damage Detection in Structures. https://doi.org/10.1016/j.ymssp.2021.107614

18 Xiao, C., Liu, Z., Zhang, T. and Zhang, X. (2021) Deep Learning Method for Fault Detection of Wind Turbine Converter. *Applied Sciences (Switzerland)*, MDPI AG, **11**, 1–22. https://doi.org/10.3390/app11031280

19 Wang, Y., Tao, F., Zuo, Y., Zhang, M. and Qi, Q. (2024) Digital Twin Enhanced Fault Diagnosis Reasoning for Autoclave. *Journal of Intelligent Manufacturing*, Springer, **35**, 2913–2928. https://doi.org/10.1007/s10845-023-02174-5

20 Angshu, K.T.Z., Tasnim, A., Ullah Khan, M.H., Rashid Molla, M.H.O., Udoy, M.I.H., Chowdhury, M. and Choudhuri, A.R. (2023) Review on Reduced Order Modeling and Its Application in the Digital Twinning Industry. AIAA SciTech Forum and Exposition, 2023, American Institute of Aeronautics and Astronautics Inc, AIAA. https://doi.org/10.2514/6.2023-1206

21 Zhu, X. and Ji, Y. (2023) A Reduced Order Model Based on Adaptive Proper Orthogonal Decomposition Incorporated with Modal Coefficient Learning for Digital Twin in Process Industry. *Journal of Manufacturing Processes*, Elsevier Ltd, **102**, 780–794. https://doi.org/10.1016/j.jmapro.2023.07.061

22 Gong, H., Cheng, S., Chen, Z. and Li, Q. Data-Enabled Physics-Informed Machine Learning for Reduced-Order Modeling Digital Twin: Application to Nuclear Reactor Physics. https://doi.org/10.1080/00295639.2021.2014752

This publication is licensed under a CC BY-NC 4.0

23 Lee, W., Jang, K., Han, W. and Huh, K.Y. (2021) Model Order Reduction by Proper Orthogonal Decomposition for a 500 MWe Tangentially Fired Pulverized Coal Boiler. *Case Studies in Thermal Engineering*, Elsevier Ltd, **28**. https://doi.org/10.1016/j.csite.2021.101414

24 Park, J., Lee, W. and Huh, K.Y. (2022) Model Order Reduction by Radial Basis Function Network for Sparse Reconstruction of an Industrial Natural Gas Boiler. *Case Studies in Thermal Engineering*, Elsevier Ltd, **37**. https://doi.org/10.1016/j.csite.2022.102288

25 Tannous, M., Ghnatios, C., Fonn, E., Kvamsdal, T. and Chinesta, F. (2025) Machine Learning (ML) Based Reduced Order Modelling (ROM) for Linear and Non-Linear Solid and Structural Mechanics. *Advanced Modeling and Simulation in Engineering Sciences*, Springer Science and Business Media Deutschland GmbH, **12**. https://doi.org/10.1186/s40323-025-00299-1

26 Rasmussen, C.E. (2004) Gaussian Processes in Machine Learning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, **3176**, 63–71. https://doi.org/10.1007/978-3-540-28650-9_4