

УДК 004.942, 004.67

СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ОПРЕДЕЛЕНИЯ ПРЕДОТКАЗНЫХ И АВАРИЙНЫХ СОСТОЯНИЙ АВИАДВИГАТЕЛЕЙ

© С. С. Абдуракипов, Е. Б. Бутаков

*Институт теплофизики им. С. С. Кутателадзе СО РАН,
630090, г. Новосибирск, просп. Академика Лаврентьева, 1
E-mail: s.s.abdurakipov@gmail.com*

Проведён сравнительный анализ разработанных классических моделей машинного обучения на основе линейных моделей и деревьев решений, а также современных алгоритмов свёрточных нейронных сетей и нейросетевого автоэнкодера для решения задачи предиктивного обнаружения предотказных и аварийных состояний авиационных двигателей. В основе набора данных NASA лежат показания датчиков, отражающих жизненные циклы работы авиадвигателей. В процессе исследования рассматривается несколько постановок задач: задача бинарной и многоклассовой классификации, в рамках которой прогнозируются нормальное, предотказное и аварийное состояния авиадвигателей; задача регрессии в целях прогнозирования точного количества рабочих циклов до отказа двигателя, а также задача обучения без учителя, в которой нейросетевой автоэнкодер применяется для обнаружения аномальных циклов работы авиадвигателей. Полученные алгоритмы объединены в фреймворк, который может быть полезен при анализе широкого спектра данных предиктивного обслуживания.

Ключевые слова: машинное обучение, прогнозное обслуживание, предотказные и аварийные состояния двигателей.

DOI: 10.15372/AUT20200605

Введение. В рамках концепции «Индустрия 4.0» компании активно внедряют новые технологии в производство и превращают свои заводы в умные предприятия. Интернет вещей сделал возможным преобразование физических сигналов, полученных от машин, программируемых контроллеров, MES-, CMMS- и ERP-систем, в цифровые сигналы с помощью датчиков. Развитие современных вычислительных технологий и алгоритмов, таких как градиентный бустинг, свёрточные, рекуррентные нейронные сети, уже сейчас позволяет принимать конкретные производственные, инженерные и технические решения, направленные на устранение проблем или оптимизацию процесса [1]. Алгоритмы машинного обучения показывают свою эффективность в решении широкого спектра задач регрессии и классификации [2–4].

Одной из таких технологий является предиктивное обслуживание (predictive maintenance). В отличие от традиционных типов технического обслуживания, например реактивного обслуживания, предполагающего ремонт или замену оборудования по выходу из строя (reactive maintenance), и профилактического планового обслуживания (preventive maintenance), которое осуществляется через определённые промежутки времени, предиктивное обслуживание подразумевает мониторинг состояния оборудования и принятие решений на основе методов анализа данных, полученных от датчиков и других внешних источников, в реальном времени [1]. Заблаговременное прогнозирование поломок позволяет сэкономить значительные средства, которые в противном случае были бы потрачены на дорогостоящий ремонт, а также создать безопасную среду для работы и повысить качество производимой продукции [5].

Для прогнозирования состояния техники используется ряд алгоритмов машинного и глубокого обучения. Каждый имеет свои преимущества и недостатки, и выбор правильной математической модели зависит от фокуса конкретной задачи, требований бизнеса и остаётся на усмотрение эксперта в области data science. Не менее важным является процесс обработки входящих сигналов и генерации новых признаков.

Классические алгоритмы машинного обучения нашли широкое применение в сфере предиктивного обслуживания. В качестве примера можно привести работу [6], рассматривающую классификацию различных дефектов подшипников. От датчиков поступают акустические и вибрационные сигналы, характеризующиеся высоким уровнем шума и высокой нелинейностью, и для анализа временных рядов успешно применяется метод энтропии перестановок, разработанный в [7], после чего используется метод опорных векторов (SVM) для классификации дефектов.

В [8] описан опыт успешного применения SVM для задачи регрессии, а также рассматривается возможность прогнозирования продолжительности жизненных циклов (Principal component analysis, RUL) для авиационных двигателей на данных без регулярной структуры, содержащих большое количество шума. Для обработки сигналов были применены метод главных компонент (Remaining useful life, PCA) и улучшенная мера сходства между системами, для которых производится оценка RUL, и системами, которые доступны в обучающей выборке, что означает: предсказания для систем из тестовой выборки делаются на основе наиболее похожих систем в обучающей выборке. Данный подход в сочетании с методом опорных векторов позволил точно определить длительность рабочего цикла каждого двигателя из тестовой выборки.

В исследовании [9] на основе данных о поломках и последующем ремонте транспортных средств показано, что в рамках многоклассовой классификации поломок может одновременно решаться несколько подзадач. В работе делается попытка предсказания кода ремонта и временного горизонта прогнозирования, который служит индикатором необходимости ремонта. Используемые модели: рекуррентные нейросети (Long short-term memory networks, LSTM), смешанная гауссовская модель (Gaussian mixture model, GMM), ансамбль деревьев решений (Random forest, RF) — показали достаточно скромные результаты при предсказании горизонта прогнозирования, что может быть связано с внезапными поломками оборудования при отсутствии заметных индикаторов износа, предшествующего поломкам. Для некоторых моделей (например, LSTM) проблема может заключаться в нехватке данных.

Ручная подготовка данных, которую подразумевают классические алгоритмы машинного обучения, не всегда является эффективным подходом при работе с большими данными ввиду необходимости автоматизации процесса [10]. Эта проблема успешно решается средствами глубокого обучения. Одним из преимуществ архитектуры свёрточных нейронных сетей (Convolutional neural networks, CNN) является способность автоматически выделять признаки (веса фильтров) во время обучения с помощью метода обратного распространения ошибки (backpropagation). Данное утверждение подкрепляется реальными практическими примерами. В [11] использование CNN и эмпирической модовой декомпозиции позволило выделить 91 полезный признак из шумных данных. Последующее применение алгоритмов SVM и функции активации softmax на сгенерированных данных дало возможность получить точность прогнозирования свыше 99 % в задаче многоклассовой классификации (на 52 класса). В качестве альтернативы данному методу была также предложена модель 1D CNN, использующая в качестве ядер (фильтров) и карт признаков одномерные массивы вместо 2D-матриц [12]. Одномерная свёрточная сеть хорошо подходит для мониторинга в реальном времени за счёт низкой вычислительной сложности. В работе [12] также делается акцент, что для обнаружения аномального поведения оборудования нет необходимости в применении глубоких и сложных конфигураций CNN.

Специалисты, решающие задачи предиктивного обслуживания на конкретных промышленных данных, часто попадают в ситуации, когда разметка (накопленная статистика) отказов оборудования отсутствует. В таких случаях актуальными оказываются методы обучения без учителя: методы кластеризации, обнаружения аномалий, понижения размерности и т. д. Один из методов, заслуживающий внимания при работе с неразмеченными данными, — многослойный шумоподавляющий автоэнкодер [13]. Этот алгоритм глубокого обучения автоматически выделяет важные признаки из массива сырых данных с помощью метода деструкции данных и выстраивает значимые зависимости между входными данными и состоянием оборудования, что положительно сказывается на диагностике состояния техники. В работе [14] также предлагается использовать гибридный подход, объединяющий в себе методы обучения без учителя и обучения с частичным привлечением учителя, когда инженеры присваивают метки неопознанным классам.

Цель данной работы — оценить эффективность алгоритмов машинного и глубокого обучения применительно к реальной задаче прогнозирования технического состояния авиационных двигателей. В ходе исследования проводится сравнительный анализ ряда классических моделей машинного обучения, CNN и нейросетевого автоэнкодера. Для оценки результатов работы алгоритмов используются различные метрики качества. Проблема рассматривается в разных постановках: решаются задачи бинарной и многоклассовой классификации, прогнозирующие нормальное или аномальное функционирование двигателя, задача регрессии, в рамках которой определяется точное количество рабочих циклов до отказа двигателя, а также задача обучения без учителя, в которой нейросетевой автоэнкодер осуществляет поиск аномалий на данных без разметки.

1. Описание данных. Для проведения исследования были выбраны доступные наборы данных NASA: PHM08 [15], Bearing Data Set [16], Milling Data Set [17] и др. В представленной работе подробно рассмотрены результаты для набора данных PHM08 [15]. Данные содержат показания 21 датчика (s_1, s_2, \dots, s_{21}), которые производят наблюдения за промышленными авиационными двигателями (повышенной проходимости с тягой до 90000 фунтов) в течение определённого периода времени — с момента начала их работы в нормальном режиме до момента критической отметки, по достижении которой рекомендуется немедленно прекратить эксплуатацию двигателя и провести комплексное обслуживание, а также информацию о трёх настройках двигателей ($setting_1, setting_2, setting_3$), характеризующих режимы работы. Показания датчиков содержат информацию о температуре и давлении на входе в турбину; температуре и давлении на выходе из компрессора низкого и высокого давления; температуре и давлении на выходе из турбины низкого и высокого давления; частотах и скоростях вращения турбины; давлении в байпасном канале; коэффициенте давления двигателя; расходе топлива; коэффициенте стехиометрии топлива в камере сгорания и др. [15].

Длительность мониторинга определяется количеством рабочих циклов для каждого из 218 двигателей. Среднее время работы одного двигателя составляет 209 циклов. На рис. 1 в виде диаграммы показано время работы 20 двигателей, выбранных случайным образом из набора данных.

В качестве целевой переменной для алгоритмов регрессии в исходном датасете выбран вектор значений TTF (time to failure), указывающих точное время работы, оставшееся до выхода двигателя из строя, на каждом цикле. Дополнительно для задач бинарной и многоклассовой классификации были сгенерированы векторы целевых переменных. Для бинарной классификации последние 30 рабочих циклов двигателя до критической отметки означают аварийное состояние (класс 1). Для многоклассовой классификации период аномальной работы также составляет 30 рабочих циклов, предшествующих критической отметке, но первые 15 из них фиксируют данные о работе двигателей в предаварийном

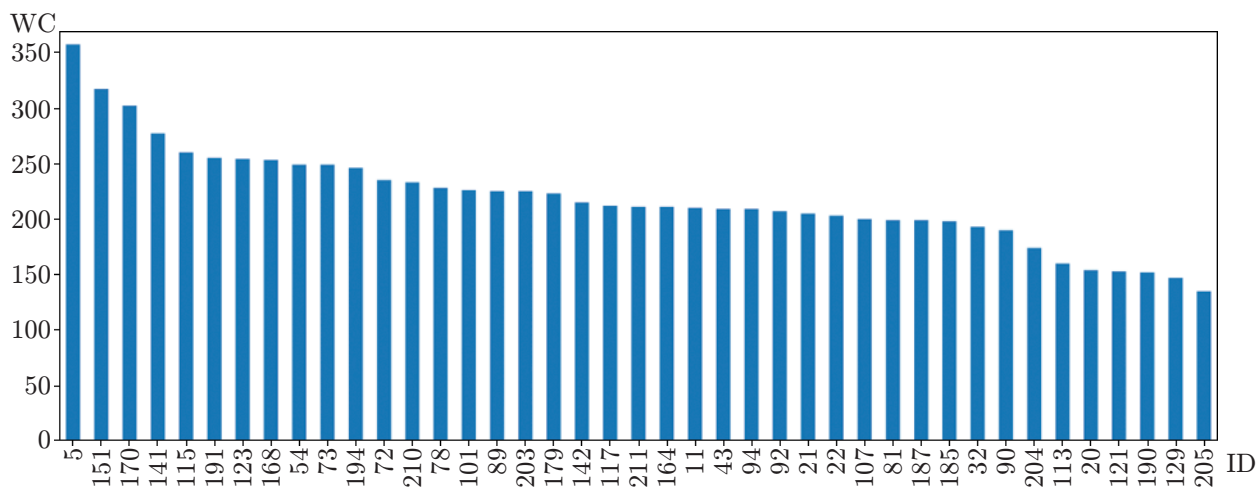


Рис. 1. Время функционирования различных двигателей, измеряемое в рабочих циклах (ID — уникальный идентификатор двигателя, WC — номер рабочего цикла)

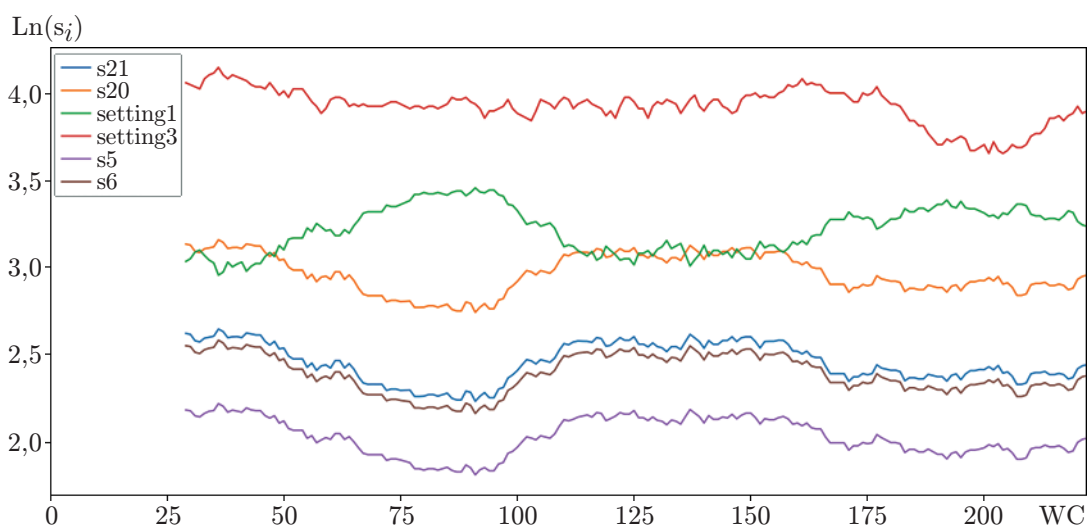


Рис. 2. Визуализация временных рядов признаков $\{s_{21}, s_{20}, \text{setting1}, \text{setting3}, s_5, s_6\}$ двигателя с ID = 1 (s_i — показания температуры в компрессоре и турбине низкого и высокого давления, setting — настроечные параметры, характеризующие режимы работы двигателей, WC — номер рабочего цикла, $\text{Ln}(s_i)$ — значения признаков s_i после применения логарифмирования)

состоянии (класс 1), последние 15 (класс 2) — в аварийном состоянии. На рис. 2 для примера визуализированы временные ряды некоторых признаков на протяжении всего периода функционирования двигателя с идентификатором 1. Для сглаживания выбрано среднее значение со скользящим окном 30.

Работа с данными несколько упрощается за счёт того, что набор не содержит нулевых значений, а все признаки являются количественными. Поскольку все двигатели согласно описанию набора данных РНМ08 являются однотипными, тестовая выборка была получена из исходных данных путём случайного выбора по уникальному идентификатору 40 двигателей для оценки работы CNN и 58 двигателей для остальных алгоритмов, рассматриваемых в данной работе.

2. Описание методов моделирования.

2.1. Трансформация временных рядов в изображения и свёрточные нейронные сети.

Первый метод применительно к нашему исследованию представляет собой многослойную свёрточную нейронную сеть, которая обрабатывает поступающие на вход $2D$ -изображения и решает задачу многоклассовой классификации.

Чтобы получить как можно больше обучающих данных, для каждого двигателя путём перемещения по значениям каждого признака с шагом 1 было сгенерировано множество последовательностей длиной 50. Число последовательностей каждого двигателя зависит от количества циклов его работы. Данная операция была выполнена для каждого из исходных 24 признаков. Затем каждый временной ряд был преобразован в изображение размером 50×50 путём вычисления евклидова расстояния между точками. Таким образом, каждое наблюдение в наборе данных теперь представлено 24 изображениями размером 50×50 . В качестве примера набор сгенерированных изображений некоторых признаков для первого наблюдения в наборе данных приведён на рис. 3.

Архитектура многослойной CNN [18], использованной в данной работе, строится путём чередования свёрточных (convolution layers) и субдискретизирующих слоёв (subsampling layers или pooling layers). На вход первому свёрточному слою из 32 нейронов для каждого образца подаются 24 изображения размером 50×50 , которые обрабатываются ядрами свёртки, представляющими собой матрицы весов размером 3×3 , по фрагментам (суммируя результаты матричного произведения для каждого фрагмента). Результат каждой свёртки попадает на нелинейную функцию активации, в данном случае — ReLU (rectified linear unit). Функция активации встроена в слой свёртки.

Получившиеся в результате карты признаков подаются ещё на один свёрточный слой из 32 нейронов с ядрами свёртки размером 3×3 и функцией активации ReLU. После этого карты признаков подаются на слой пулинга (или субдискретизации), в котором группы пикселей размером 2×2 уплотняются до одного пикселя. Существует несколько разновидностей операции субдискретизации, например в данной CNN используется пулинг на основе функции максимума, при которой из четырёх пикселей выбирается один с наибольшим значением, тем самым уменьшая пространственный объём изображения (понижение размерности).

После этого блока из двух свёрточных слоёв и одного слоя пулинга применяется метод регуляризации под названием дропаут (dropout) [18], который служит для предотвращения переобучения. Суть метода заключается в том, что в слое нейронной сети в процессе обучения случайным образом деактивируется определённая доля нейронов (в данном случае 0,2).

Следующей ступенью в архитектуре используемой CNN выступает комбинация из двух свёрточных слоёв, каждый из которых состоит из 64 нейронов, с ядрами свёртки размером 3×3 и функциями активации ReLU, за которыми следует слой пулинга с окном 2×2 и функцией максимума, после чего применяется дропаут со значением 0,2 деактивированных нейронов от общего числа нейронов в предыдущем слое. Получившиеся в результате

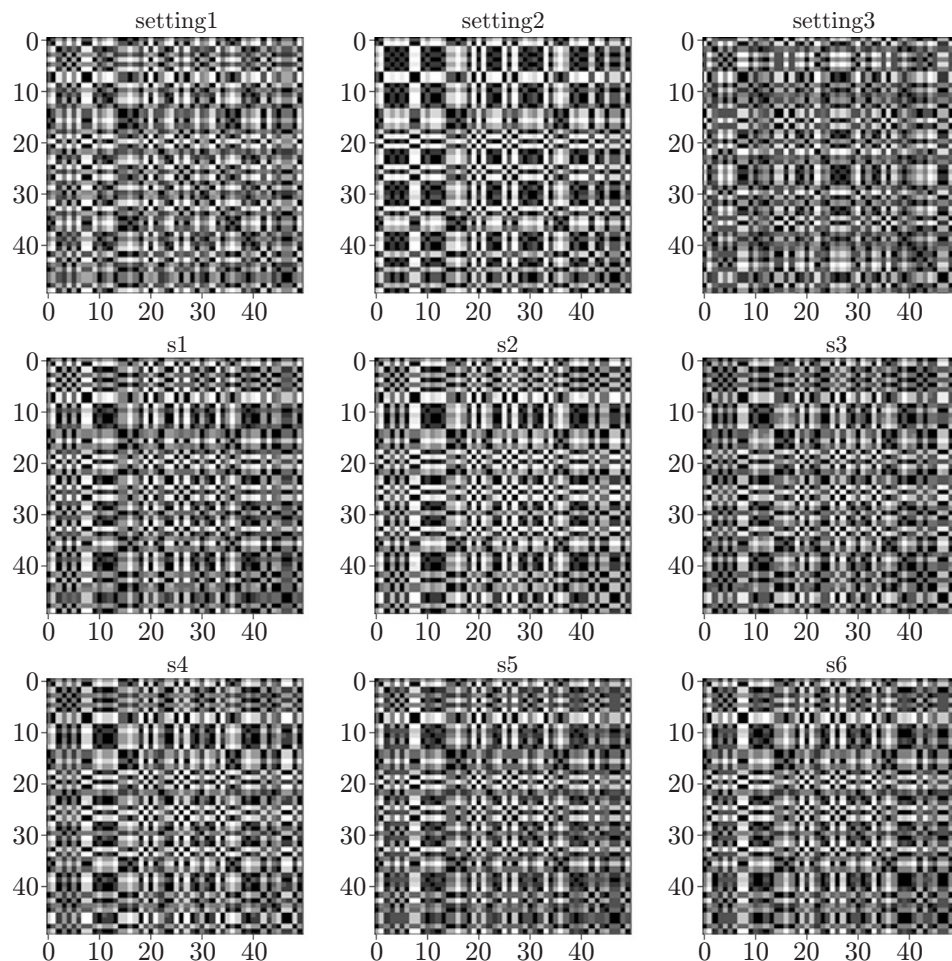


Рис. 3. Пример трансформации части признаков в изображения для первого наблюдения в наборе данных

всех преобразований изображения теперь представляют собой абстрактные карты признаков, которые подаются на слой выравнивания, преобразующий изображения из двумерных массивов в одномерный. Слой извлекает строки пикселей из всех изображений и выстраивает их в один ряд. Затем вектор передаётся на обычную полносвязную нейронную сеть. Она состоит из входящего слоя из 256 нейронов, функции активации ReLU и последующей регуляризации в виде дропаута со значением 0,2. Затем следует выходной слой, состоящий из трёх нейронов, на котором применяется функция активации softmax для определения вероятности принадлежности образца к одному из трёх классов.

Нейронная сеть обучается с помощью метода обратного распространения ошибки, в ходе которого на каждом слое оптимизируются веса ядер свёртки и свободная переменная. В качестве оптимизатора был выбран стохастический градиентный спуск с шагом 0,001 и моментом 0,9. Для параметра «размер батча» выбрано значение 200. Параметр указывает, какое количество образцов необходимо обрабатывать за одну итерацию градиентного спуска. Сеть обучается на протяжении 10 эпох, после чего её можно применять к тестовой выборке. Для оценки качества классификатора использовалась матрица ошибок.

2.2. Алгоритмы регрессии.

Для всех исходных признаков $\{s1, \dots, s21\}$ были сгенерированы дополнительные признаки в виде среднего значения со скользящим окном 5 $\{av1, \dots, av21\}$ и стандартного

отклонения со скользящим окном 5 {sd1, ..., sd21}. В общую группу признаков были также включены признаки `work_cycle`, показывающие номер рабочего цикла двигателя, и `setting1`, `setting2` и `setting3`, отвечающие за настройки режимов работы двигателей.

Для задачи регрессии, т. е. определения точного количества рабочих циклов до выхода двигателя из строя, предлагается использовать ряд подходящих для этой цели моделей и далее сравнить результаты всех алгоритмов по определённым метрикам (см. далее). Анализ устройства и принципов работы каждого алгоритма выходит за рамки данной работы, поэтому ограничимся их перечислением и указанием основных параметров каждого: линейная регрессия; LASSO-регрессия ($\alpha = 0,001$); Ridge-регрессия ($\alpha = 0,01$); полиномиальная регрессия (степень 3); дерево регрессии (максимальная глубина 7); случайный лес деревьев регрессии (количество деревьев 600, максимальная глубина 6); LightGBM (количество листьев 256, шаг обучения 0,1, доля признаков 0,6, доля экземпляров для бэггинга 0,7); XGBoost Regressor (количество деревьев 800, максимальная глубина 6, шаг обучения 0,05, доля признаков 0,6, доля экземпляров 0,6); CatBoost Regressor (глубина 8, шаг обучения 0,1).

Для линейных моделей (линейная регрессия, LASSO-регрессия, Ridge-регрессия, полиномиальная регрессия) важно провести отбор признаков, чтобы избежать проблемы мультиколлинеарности и возможного снижения качества предсказаний алгоритма. При ручном отборе признаков сначала была рассчитана корреляция целевой переменной TTF (time to failure) со всеми исходными признаками на основе коэффициента Пирсона. В результате на вход линейным моделям подаются 16 признаков {`work_cycle`, `av14`, `av11`, `av4`, `av15`, `av3`, `av17`, `s14`, `s11`, `sd1`, `s4`, `av9`, `s15`, `sd4`, `sd5`, `sd2`}, для которых также предварительно была выполнена стандартизация. Сокращения «av» и «sd» означают скользящее среднее и стандартное отклонение наблюдаемого параметра. В модели учитывались следующие переменные: температура и давление на выходе из турбины высокого давления, скорость вращения сердечника и её корректировка, температура и давление на выходе компрессора высокого давления, коэффициент давления двигателя [15].

В рамках эксперимента также проводился отбор признаков с помощью случайного леса деревьев регрессии (количество деревьев 200, максимальная глубина 6), однако результаты оказались хуже, чем в случае использования вручную отобранных признаков для каждого из упомянутых выше линейных алгоритмов. Для остальных моделей имеет смысл применять полный набор из исходных и сгенерированных признаков, предварительно выполнив стандартизацию данных.

В представленной работе для оценки результатов предсказаний алгоритмов в задаче регрессии использовались метрики: квадратный корень из среднеквадратичной ошибки (RMSE), средняя абсолютная ошибка (MAE), доля дисперсии зависимой переменной, воспроизведённой моделью, в виде коэффициента детерминации (R^2).

2.3. Классификаторы.

Для решения задачи бинарной классификации задействованы следующие алгоритмы, для каждого из которых методом решётчатого поиска были подобраны наилучшие параметры (основные указаны в скобках): логистическая регрессия ($C = 10$); дерево классификации (максимальная глубина 8); случайный лес (200 деревьев, максимальная глубина 8); метод опорных векторов ($C = 0,1$); метод ближайших соседей (число соседей 13); наивный байесовский классификатор; классификатор XGBoost (200 деревьев, максимальная глубина 7, шаг обучения 0,1); классификатор CatBoost (250 деревьев, максимальная глубина 8); классификатор AdaBoost (250 деревьев, шаг обучения 0,1).

Кроме того, в качестве дополнительных вариантов решений приводятся ансамбли классификаторов на основе «жёсткого» и «мягкого» голосования (в обоих случаях применяется комбинация: логистическая регрессия, метод опорных векторов, классификатор

XGBoost), а также стекинг моделей. При стекинге прогнозы базовых моделей используются в качестве признаков для основной модели.

Для обработки данных алгоритмами был использован полный набор из исходных и сгенерированных признаков и выполнена стандартизация данных. Поскольку набор данных является несбалансированным по числу экземпляров в каждом классе (доля аномального класса 1 составляет около 17 % от общего количества экземпляров), для оценки качества целесообразно использовать такие метрики, как: точность (precision), которая рассчитывается по формуле $TP/(TP + FP)$, где TP — доля истинно положительных предсказаний, а FP — доля ложноположительных предсказаний; полнота (recall), которая рассчитывается по формуле $TP/(TP + FN)$, где TP — доля истинно положительных предсказаний, а FN — доля ложноотрицательных предсказаний; ROC-кривая, которая показывает соотношение между долями истинно положительных предсказаний TPR и ложноположительных предсказаний FPR .

2.4. Автоэнкодер.

Рассматривается метод детекции аномалий с помощью автоэнкодера в рамках задачи обучения без учителя, когда у исследователя есть только неразмеченные данные. Архитектура этого типа нейронных сетей устроена следующим образом. На вход автоэнкодера подаются 16 признаков, процесс отбора которых был описан для линейных моделей: $\{\text{work_cycle}, \text{av14}, \text{av11}, \text{av4}, \text{av15}, \text{av3}, \text{av17}, \text{s14}, \text{s11}, \text{sd1}, \text{s4}, \text{av9}, \text{s15}, \text{sd4}, \text{sd5}, \text{sd2}\}$. Данные обрабатываются полносвязным слоем из 16 нейронов и гиперболическим тангенсом (\tanh) в качестве функции активации. В этом же слое применяется L1-регуляризация. Следующий слой — скрытый полносвязный слой h из 8 нейронов и функции активации ReLU. Эта часть архитектуры автоэнкодера называется кодировщиком. Его задача — снизить размерность поступающих на вход данных, используя для этого некую функцию кодирования $h = f(x)$. После этого декодировщик пытается реконструировать исходные данные из закодированных данных с помощью функции реконструкции $r = g(h)$. В нашем случае используется декодировщик со скрытым слоем из 8 нейронов и функции активации \tanh , за которым следует ещё один полносвязный слой из 16 нейронов и функции активации \tanh , после чего на выходе получают реконструированные данные. Важно отметить, что реконструированные данные не должны в точности повторять исходные, а лишь являться их приблизительной копией. В процессе обучения автоэнкодер способен сам приоритизировать определённые аспекты данных, узнавая таким образом их полезные свойства [15]. Процесс обучения сводится к минимизации функции ошибки $L(x, g(f(x)))$, в данном случае логарифмической среднеквадратичной ошибки; в качестве оптимизатора выбран алгоритм адаптивной инерции Adam [18]. Данные подаются на вход модели пакетами по 100 экземпляров, обучение осуществляется на протяжении 150 эпох. Для оценки качества используются ROC-кривая и матрица ошибок.

3. Результаты моделирования.

3.1. Свёрточная нейронная сеть.

Для задачи многоклассовой классификации с помощью свёрточной нейронной сети в тестовую выборку было включено 40 двигателей; размер обучающей выборки составил 178 двигателей. На рис. 4 представлены результаты работы алгоритма, описанного в разд. 2, в виде матрицы ошибок, а также визуализированы кривые AUC ROC для двух случаев бинарной классификации. Результаты показывают, что модель достаточно хорошо различает класс 0 (нормальное состояние) и класс 2 (аварийное состояние), т. е. справляется с базовой задачей предупреждения о скором выходе оборудования из строя, однако ошибается с предсказаниями класса 1, относя данные образцы то к классу 0, то к классу 2. При этом следует принять во внимание тот факт, что размер обучающей выборки невелик.

Несмотря на указанные ошибки подход с трансформацией временных рядов в изображениях с последующим применением свёрточной нейронной сети является вполне рабочим

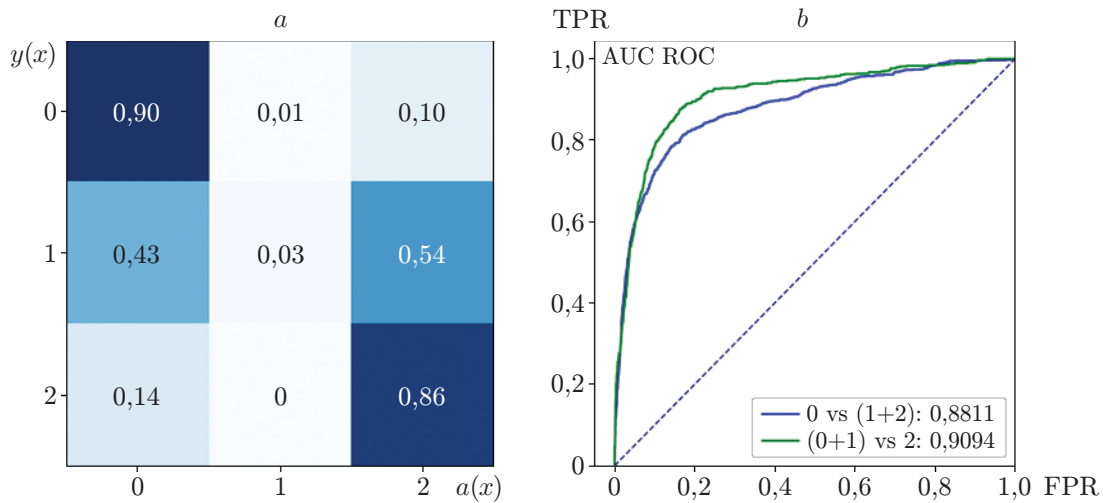


Рис. 4. Модель многоклассовой классификации отказов авиадвигателей на основе свёрточной нейросети: a — матрица ошибок для результатов прогноза модели, полученных с помощью CNN, где $y(x)$ — истинный класс, $a(x)$ — прогнозируемый класс; b — кривые AUC ROC для случаев бинарной классификации (вероятность класса 0 в сравнении с суммарной вероятностью классов 1 и 2, суммарная вероятность классов 0 и 1 в сравнении с вероятностью класса 2), где FPR — доля ложноположительных ответов, TPR — доля истинно положительных ответов

для задачи прогнозирования продолжительности жизни оборудования. Для улучшения результатов можно использовать большее количество данных, на которых будет обучаться модель, а также обратиться ко мнению экспертов в соответствующей научно-технической области, которые смогут более точно сделать разметку для целевой переменной, основываясь на своём опыте.

3.2. Модели регрессии.

Для задачи оценки точного количества оставшихся жизненных циклов оборудования были применены алгоритмы регрессии с соответствующими наборами признаков, описанные в разд. 2. Оценка дисперсии остатков регрессии при применении линейных алгоритмов выявила выраженный тренд в распределении (рис. 5).

Это говорит о том, что нелинейные модели должны показать себя лучше на имеющихся данных, что подтверждается эмпирическим путём. Линейные алгоритмы уступили в качестве некоторым нелинейным моделям по метрикам RMSE (Root mean squared error), MAE (Mean absolute error) и R^2 . Наилучший результат по всем метрикам у полиномиальной регрессии при достижении показателей 34,3 RMSE, 25,4 MAE и 0,76 R^2 . Стабильно высокое и примерно одинаковое качество показали алгоритмы CatBoost и XGBoost, заняв вторую и третью позиции по эффективности соответственно. Результаты работы трёх лучших алгоритмов регрессии отражены на рис. 5. Подробные результаты алгоритмов регрессии занесены в табл. 1.

3.3. Модели бинарной классификации.

Хорошие результаты продемонстрировали почти все алгоритмы в задаче классификации состояния двигателей как нормального или предотказного. У большинства из них очень высокий показатель ROC AUC (от 0,95). Логистическая регрессия и метод опорных векторов показали наибольшую эффективность, достигнув отметки в 0,99 по метрике ROC AUC. Высокая скорость вычислений в сочетании с впечатляющими метриками полноты и точности (0,86 и 0,885 для логистической регрессии и 0,85 и 0,89 для метода опорных

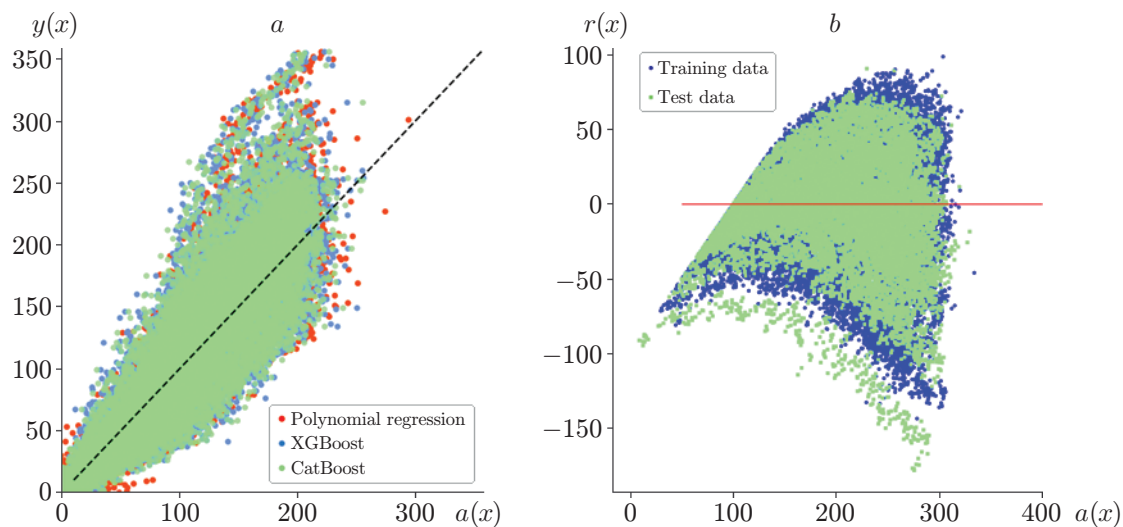


Рис. 5. Модели регрессии числа циклов до отказа авиадвигателей на основе градиентного бустинга: a — зависимость истинных и прогнозируемых значений для трёх лучших алгоритмов регрессии на тестовой выборке; b — распределение остатков линейной регрессии на тестовой и тренировочной выборках. Обозначения: $y(x)$ — истинные значения, $r(x)$ — остатки, $a(x)$ — предсказанные значения

Таблица 1
Результаты валидации алгоритмов регрессии

Алгоритмы	RMSE	MAE	R^2
LR	37,788597	28,942036	0,712939
Ridge	37,817864	28,953007	0,712494
Lasso	37,788643	28,942029	0,712938
PR	34,341934	25,417140	0,762916
DT	39,241147	29,293113	0,690446
RF	38,613698	28,795159	0,700266
LGBM	38,025324	28,479610	0,709331
XGBoost	35,359583	25,629082	0,748657
CatBoost	35,033778	25,484919	0,753267

Примечание. LR — линейная регрессия, Ridge — Ridge-регрессия, Lasso — LASSO-регрессия, PR — полиномиальная регрессия, DT — дерево решений, RF — случайный лес, LGMB — LightGBM.

векторов) делает их наилучшим выбором для описанной задачи. Стабильно высокие результаты показали алгоритмы XGBoost (ROC AUC 0,988, полнота 0,846 и точность 0,86) и CatBoost (ROC AUC 0,987, полнота 0,84 и точность 0,86), однако потребовали больше времени для вычислений, чем предыдущие два алгоритма. Лучшим алгоритмом по метрике точности оказался случайный лес деревьев (0,909), однако он существенно уступил другим моделям по остальным показателям. На рис. 6 визуализированы кривые ROC AUC для трёх алгоритмов с самыми высокими показателями по этой метрике: логистическая регрессия, метод опорных векторов и классификатор XGBoost.

Кроме того, в рамках данной задачи был применён стекинг из базовых моделей логистической регрессии и метода опорных векторов, на результатах которых был обучен классификатор XGBoost. Хотя ансамбль уступил многим алгоритмам по ROC AUC (0,93),

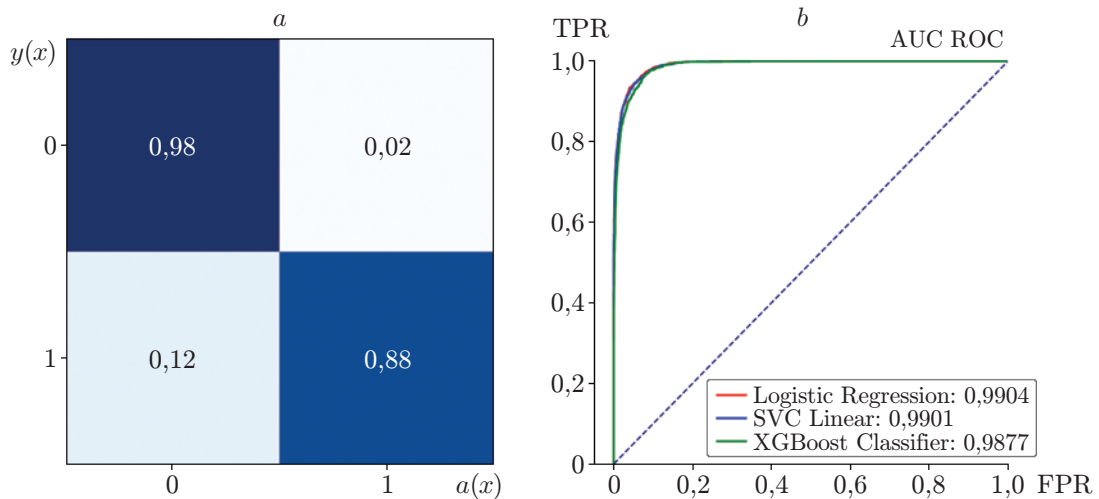


Рис. 6. Модели бинарной классификации отказов авиадвигателей: a — матрица ошибок стекинга логистической регрессии, метода опорных векторов и XGBoost; b — график ROC трёх алгоритмов с самыми высокими значениями по соответствующей метрике AUC ROC. Обозначения: $a(x)$ — предсказанные значения, $y(x)$ — истинные значения, FPR — доля ложноположительных ответов, TPR — доля истинно положительных ответов

Таблица 2

Результаты валидации алгоритмов бинарной классификации

Алгоритмы	ROC AUC	Recall	Precision	Accuracy	F1
LR	0,990442	0,860957	0,885077	0,964597	0,872850
DT	0,956919	0,725250	0,767510	0,930214	0,745782
RF	0,986218	0,756952	0,908545	0,954942	0,825850
SVM	0,990130	0,851502	0,890634	0,964283	0,870628
KNN	0,943703	0,527809	0,728319	0,905566	0,612061
GNB	0,869824	0,709121	0,464312	0,843473	0,561180
XGBoost	0,987684	0,845940	0,861268	0,959023	0,853535
CatBoost	0,987261	0,840934	0,864000	0,958866	0,852311
AdaBoost	0,975139	0,758065	0,841878	0,945757	0,797776
Stacking	0,930146	0,878198	0,879176	0,965774	0,878687
VC	0,918321	0,854283	0,888375	0,964283	0,870995

Примечание. LR — логистическая регрессия, DT — дерево решений, RF — случайный лес, SVM — метод опорных векторов, KNN — метод ближайших соседей, GNB — наивный байесовский классификатор, Stacking — стекинг моделей логистической регрессии, метода опорных векторов и XGBoost, VC — ансамбль моделей логистической регрессии, метода опорных векторов и XGBoost на основе «жёсткого» голосования.

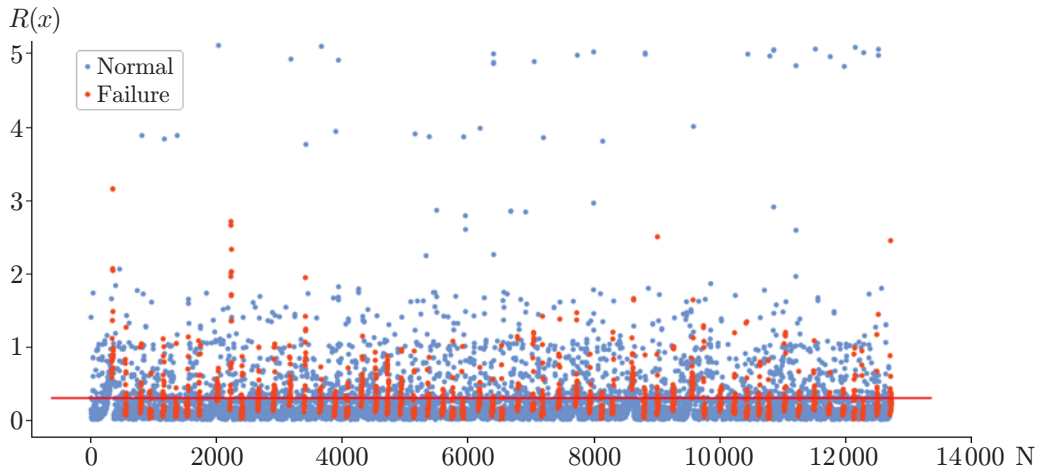


Рис. 7. График ошибки реконструкции для тестовой выборки с порогом классификации (в качестве приемлемого порога выбрана отметка 0,3)

он достиг наилучшего соотношения полноты и точности (0,878 и 0,879 соответственно). Матрица ошибок ансамбля моделей представлена на рис. 6.

Основываясь на полученных результатах, можно сделать выводы, что на этом наборе данных бинарные классификаторы показали наилучшие результаты. Детальные показатели для всех использованных алгоритмов занесены в табл. 2.

3.4. Автоэнкодер. Нейронная сеть обучается на подготовленной выборке, которая содержит данные только нормальной работы оборудования, и может определить закономерности взаимодействия между разными переменными, чтобы впоследствии попытаться восстановить их как можно точнее. В качестве тестовой выборки на вход автоэнкодера подаются как нормально функционирующие, так и аномальные примеры. При попытке реконструкции данных, которые отклоняются от нормы, ошибка реконструкции будет выше, чем в случае с нормальными экземплярами. Отнесение образцов к классу 0 или 1 базируется на заданном значении порога ошибки реконструкции, который можно определить путём визуализации распределений ошибки реконструкции каждого класса. На рис. 7 представлен график ошибки реконструкции для тестовой выборки с порогом классификации 0,3.

Состояния, ошибка реконструкции которых ниже этого значения, были признаны нормально функционирующими, другие отнесены к классу аномалий. Автоэнкодер продемонстрировал достаточно низкие результаты. Значение AUC ROC составило 0,72, матрица ошибок показывает, что модель допускает достаточно много ошибок, часто смешивая экземпляры двух классов (рис. 8). Возможная причина большого количества ошибок заключается в малом объёме выборки и существенном уровне шума в данных, из-за чего трудно провести границу, чётко разделяющую примеры двух классов.

Полученные алгоритмы объединены в фреймворк PMTool (Predictive Maintenance Toolbox), который может быть полезен при анализе широкого спектра данных предиктивного обслуживания. Фреймворк расширяет возможности стандартных библиотек машинного глубокого обучения, таких как Scikit-Learn, XGBoost, LightGBM и Keras и обеспечивает преднастроенные пайплайны (последовательность процедур) для препроцессинга временных рядов датчиков, генерации дополнительных признаков на основе статистических функций, выбора оптимальной модели классификации, регрессии и детекции аномалий. Фреймворк использует преимущества библиотек Matplotlib и Seaborn для удобной визуализации и анализа результатов моделирования.

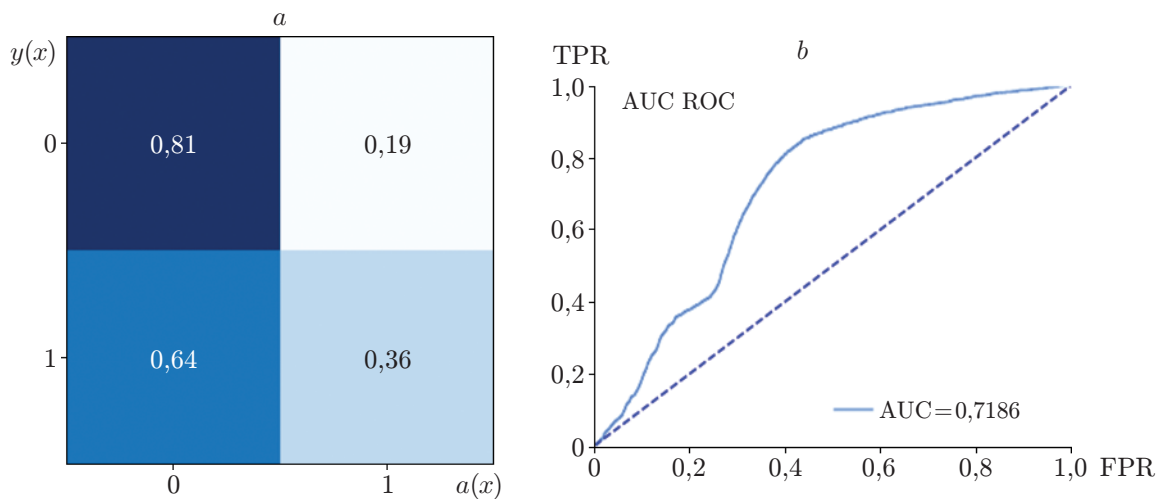


Рис. 8. Модель бинарной классификации отказов авиадвигателей на основе нейросетевого автоэнкодера: a — матрица ошибок; b — кривая ROC AUC. Обозначения: $y(x)$ — истинный класс, $a(x)$ — прогнозируемый класс, FPR — доля ложноположительных ответов, TPR — доля истинно положительных ответов

Заключение. Проведён сравнительный анализ алгоритмов машинного и глубокого обучения применительно к нескольким наборам данных NASA. Рассмотрены результаты для датасета PHM08, который содержит показания датчиков, отражающих жизненные циклы 218 авиационных двигателей. Исследование продемонстрировало разную степень эффективности моделей в решении конкретных задач предиктивного обслуживания. Для многоклассовой классификации была применена свёрточная нейронная сеть, которая основывалась на процедуре преобразования временных рядов в изображения. Показано, что CNN хорошо различает классы 0 (нормальное состояние) и 2 (аварийное состояние), достигая значений точности и полноты 0,90 и 0,86 соответственно, однако практически не показывает значимых результатов при прогнозировании класса 1 (предаварийное состояние). В задаче бинарной классификации алгоритмы логистической регрессии и SVM продемонстрировали высокие показатели по всем метрикам (ROC AUC свыше 0,99 у обоих алгоритмов, полнота 0,86 и 0,85, точность 0,885 и 0,89 у логистической регрессии и SVM соответственно). Они же являются самыми оптимальными с точки зрения времени работы. В этом отношении им проигрывает стекинг из логистической регрессии, SVM и классификатора XGBoost, однако ансамбль моделей достигает наилучших результатов в плане соотношения полноты и точности, показывая 0,878 и 0,879 соответственно. В рамках исследования также оценена эффективность алгоритмов регрессии для определения точного жизненного цикла двигателя. В силу того, что между признаками и целевой переменной наблюдается слабая линейная зависимость, в задаче регрессии наиболее эффективна полиномиальная регрессия ($R^2 = 0,76$, MAE = 25,4), однако она требует предварительного отбора признаков. На втором месте по метрикам R^2 и MAE находится алгоритм CatBoost (0,75 и 25,5 соответственно). Кроме того, в работе был задействован автоэнкодер для определения аномального состояния двигателей на данных без разметки, когда нейросеть обучается только на выборке с нормально функционирующим оборудованием. Модель достигла отметки 0,72 по метрике ROC AUC и допустила большое количество ошибок при классификации, что отражено в матрице ошибок.

На основе сравнительного анализа алгоритмов показано, что свёрточная нейронная сеть обеспечивает лучшие результаты при детекции аварийных состояний, в то время как метод опорных векторов демонстрирует высокие показатели точности и устойчивости мо-

дели и является наиболее эффективным как с вычислительной точки зрения, так и с точки зрения интерпретации результатов моделирования. Результаты исследования могут быть полезны для выбора правильной стратегии работы с данными и подходящего алгоритма машинного или глубокого обучения в различных задачах предиктивного обслуживания.

Благодарности. Авторы выражают благодарность Д. Волобуеву за помощь в обработке данных.

Финансирование. Работа выполнена при поддержке Российского научного фонда (грант № 19-79-30075).

СПИСОК ЛИТЕРАТУРЫ

1. **Mobley R. K.** An Introduction to Predictive Maintenance. Woburn: Elsevier, 2002. 438 p.
2. **Абдуракипов С. С., Токарев М. П., Первунин К. С., Дулин В. М.** Моделирование характеристик тонального шума методами машинного обучения при обтекании гидропрофиля // *Автометрия*. 2019. **55**, № 2. С. 123–131. DOI: 10.15372/AUT20190213.
3. **Абдуракипов С. С., Гобызов О. А., Токарев М. П., Дулин В. М.** Мониторинг режимов горения на основе регистрации изображений пламени и машинного обучения // *Автометрия*. 2018. **54**, № 5. С. 108–115. DOI: 10.15372/AUT20180514.
4. **Abdurakipov S., Butakov E., Burdukov A. et al.** Application of artificial neural network to modeling of burning completeness of mechanoactive coal // *Combust. Explosion Shock Waves*. 2019. **55**, N 6. P. 1–6.
5. **Lacey S.** The role of vibration monitoring in predictive maintenance // *Asset Management & Maintenance Journ.* 2011. **24**, N 1. P. 42.
6. **Vakharia V., Gupta V., Kankar P.** Ball bearing fault diagnosis using supervised and unsupervised machine learning methods // *Intern. Journ. Acoustics and Vibration*. 2015. **20**, N 4. P. 244–250.
7. **Bandt C., Pompe B.** Permutation entropy: A natural complexity measure for time series // *Phys. Rev. Lett.* 2002. **88**, N 17. 174102.
8. **Chen Z., Cao S., Mao Z.** Remaining useful life estimation of aircraft engines using a modified similarity and supporting vector machine (SVM) approach // *Energies*. 2018. **11**, N 1. P. 28.
9. **Prytz R.** Machine learning methods for vehicle predictive maintenance using off-board and on-board data: PhD diss. Halmstad: Halmstad University Press, 2014. 65 p.
10. **Nguyen K., Medjaher K.** A new dynamic predictive maintenance framework using deep learning for failure prognostics // *Reliability Eng. & Syst. Safety*. 2019. **188**. P. 251–262.
11. **Xie Y., Zhang T.** Fault diagnosis for rotating machinery based on convolutional neural network and empirical mode decomposition // *Shock and Vibration*. 2017. **19**. P. 1–12.
12. **Ince T., Kiranyaz S., Eren L. et al.** Real-time motor fault detection by 1-D convolutional neural networks // *IEEE Trans. Industrial Electron.* 2016. **63**, N 11. P. 7067–7075.
13. **Lu C., Wang Z., Qin W., Ma J.** Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification // *Signal Process.* 2017. **130**. P. 377–388.
14. **Cho S., May G., Tourkogiorgis I. et al.** A hybrid machine learning approach for predictive maintenance in smart factories of the future // *Proc. of the IFIP Intern. Conference on Advances in Production Management Systems*. Seoul, Korea, Aug. 26-30, 2018. Pt II. P. 311–317.
15. **Saxena A., Goebel K.** PHM08 challenge data set. In NASA Ames Prognostics Data Repository. NASA Ames Research Center, 2008.
16. **Lee J., Qiu H., Yu G., Lin J.** Bearing Data Set. In NASA Ames Prognostics Data Repository. NASA Ames Research Center, 2007.

17. **Agogino A., Goebel K.** Milling Data Set. In NASA Ames Prognostics Data Repository. NASA Ames Research Center, 2007.
18. **Goodfellow I., Bengio Y., Courville A.** Deep learning. MIT Press, 2016.

Поступила в редакцию 31.01.2020

После доработки 20.05.2020

Принята к публикации 20.07.2020
