

**ПРИМЕНЕНИЕ ЭВОЛЮЦИОННЫХ И ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ ПРИ ФОРМИРОВАНИИ АРХИТЕКТУРЫ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ ДЛЯ ПРОГНОЗИРОВАНИЯ СОСТОЯНИЯ ТЕХНИЧЕСКОГО ОБЪЕКТА**

© 2024 Ю.Е. Кувайскова, А.А. Немькин

Ульяновский государственный технический университет, г. Ульяновск, Россия

Статья поступила в редакцию 06.06.2024

Для сокращения времени на ликвидацию критических ситуаций в работе технического объекта необходимо своевременное реагирование на нарушения процесса его функционирования. Отсюда возникает задача прогнозирования состояния объекта и возможных нарушений в процессе его работы по результатам исследования набора контролируемых параметров объекта. В этом исследовании поставленная задача решается на основе нейросетевых моделей. Однако при построении модели нейронной сети необходимо тщательно подбирать архитектуру модели для обеспечения наилучшей точности прогнозирования состояния объектов. В данной работе для автоматического проектирования моделей нейронных сетей предложена новая методика, которая заключается в последовательном использовании трех эволюционных алгоритмов: алгоритма декартова генетического программирования (CGP) для первичной инициализации, многокритериального эволюционного алгоритма NSGA-II для настройки архитектур нейронных сетей и эволюционного алгоритма CMA-ES для уточнения параметров архитектур. Для оценки качества прогнозирования по моделям нейронных сетей используется среднеабсолютная ошибка (MAE) и коэффициент детерминации ( $R^2$ ). Для исключения возможности подгонки модели под оптимальные прогнозные характеристики используется кросс-валидация. Она позволяет получать несмещенные оценки метрик качества. Для реализации этих методов и моделей на языке программирования Python при использовании библиотек tensorflow и keras была написана специальная программа. В качестве объектов исследования выступили турбореактивный двигатель, литий-ионный аккумулятор и подшипники. Для сравнения эффективности предлагаемой методики была использована библиотека AutoKeras. Исследование показало, что использование предлагаемого подхода значительно улучшает метрики качества моделей нейронных сетей для всех технических объектов по сравнению с моделями, найденными с помощью библиотеки AutoKeras: значение функции ошибки MAE для всех наборов данных при прогнозировании уменьшается в среднем в 4 раза, а значение коэффициента детерминации увеличивается в 1,9 раза. Данный подход может быть применен специалистами для прогнозирования технического состояния объектов в разных отраслях техники, особенно в авиации.

*Ключевые слова:* технический объект, прогнозирование, нейронные сети, генетические и эволюционные алгоритмы.

DOI: 10.37313/1990-5378-2024-26-4(3)-383-394

EDN: MHQLAG

*Исследование выполнено за счет гранта Российского научного фонда № 24-29-00432, <https://rscf.ru/project/24-29-00432/>.*

**ВВЕДЕНИЕ**

Почти все области человеческой деятельности связаны с применением технических объектов, и для обеспечения их надежной работы требуется точное прогнозирование состояния этих объектов [1-3].

Рассмотрим технический объект, состояние которого характеризуется набором параметров функционирования  $X_1, X_2, \dots, X_p$ . Пусть известны

*Кувайскова Юлия Евгеньевна, кандидат технических наук, доцент, заведующий кафедрой прикладной математики и информатики. E-mail: u.kuvaiskova@mail.ru*  
*Немькин Александр Андреевич, магистрант кафедры прикладной математики и информатики. E-mail: nemykin.alexander@yandex.ru*

значения этих параметров в различные моменты времени и множество возможных состояний объекта  $Y$  при соответствующих значениях контролируемых параметров объекта  $Y \in [0; \infty)$  - число успешных рабочих циклов при эксплуатации технического объекта, причем  $Y = 0$  означает неисправность объекта.

Требуется создать математическую модель, которая будет описывать связь между диагностическими параметрами объекта  $X_1, X_2, \dots, X_p$  и его состоянием  $Y$ , а также обеспечит высокую точность прогнозирования технического состояния объекта  $Y$  в будущем.

Анализ последних публикаций [4-7] показывает, что модели нейронных сетей являются наиболее эффективными методами для реше-

ния поставленной задачи. Однако для глубокого обучения требуется значительное количество данных. Наиболее часто используемыми моделями являются многослойный перцептрон (MLP) или полносвязная нейронная сеть (Dense), сети долгой краткосрочной памяти (LSTM), сверточные нейронные сети (CNN), рекуррентные нейронные сети (RNN) и другие.

В статье [4] описано применение модели MLP при прогнозировании оставшегося срока службы оборудования на производственных линиях. Исследования показали, что модели нейронных сетей могут применяться для анализа и обработки информации, собранной с оборудования в ходе его использования. Работа [5] посвящена применению сетей RNN для определения сложных временных и мгновенных связей между показаниями сенсоров для более точного оценивания оставшегося срока службы оборудования. Авторы работы [6] для предсказания неисправностей при эксплуатации турбовентиляторных двигателей использовали двухслойные сети LSTM. При этом они применили ансамбль моделей, что позволило уменьшить вероятность переобучения моделей и повысить их обобщающую способность. В работе [7] представлена гибридная модель CNN-LSTM, которая превосходит аналоги при прогнозировании деградации турбовентиляторного двигателя.

Все эти модели имеют свои плюсы и минусы, поэтому важно разработать и определить эффективную архитектуру нейросетевой модели, способную наиболее точно предсказывать состояние конкретного технического объекта. Традиционные методы подбора архитектуры, такие как ручная настройка или случайный поиск, часто ограничены своей способностью обеспечить оптимальность и эффективность нейросетевой модели.

Для решения этой проблемы в работе предлагается применение эволюционных и генетических алгоритмов. Эти методы позволяют автоматизировать процесс поиска оптимальной архитектуры нейронной сети путем систематического изменения структуры сети и оценки их производительности на основе заранее заданных критериев. Эволюционные алгоритмы, такие как генетическое программирование или эволюционные стратегии, могут эффективно искать оптимальные комбинации слоев, функций активации и связей между нейронами в сети.

Так статья [8] посвящена использованию метода NeuroEvolution of Augmenting Topologies (NEAT) для создания и эволюции нейронных сетей с целью оптимизации их архитектурной сложности и адаптивности к различным задачам машинного обучения. Для совместной эволюции нескольких нейронных сетей с целью создания и оптимизации сложных архитектур, способных эффективно решать задачи, требу-

ющие комбинирования различных подходов, в работе [9] предложено применение метода Cooperative NeuroEvolution (CNE). Исследование, представленное в работе [10], предлагает метод Evolution Strategies (ES) для оптимизации параметров глубоких нейронных сетей с целью повышения их производительности и точности на задачах классификации и регрессии. Также для эволюции структур и параметров нейронных сетей с целью создания оптимальных моделей, способных автоматически адаптироваться к новым данным и условиям, применяются генетические алгоритмы (GA) [11], генетическое программирование (GP) [12] и метод дифференциальной эволюции (DE) [13].

Эти подходы показывают высокую эффективность, но часто используют ограниченные наборы предопределенных архитектур. В настоящей работе для более гибкой адаптации архитектур нейронных сетей под конкретные условия задачи предлагается новый подход, основанный на последовательном применении эволюционных и генетических алгоритмов: алгоритма декартова генетического программирования (CGP) [14] для начальной инициализации, генетического алгоритма мультикритериальной оптимизации Non-Dominated Sorting Genetic Algorithm II (NSGA-II) для дальнейшей настройки архитектур [15] и эволюционного алгоритма Covariance matrix adaptation evolution strategy (CMA-ES) [16] с целью уточнения параметров архитектур нейронных сетей.

Цель данного исследования заключается в разработке новой методики обучения нейронных сетей на основе применения эволюционных и генетических алгоритмов для оптимизации архитектуры нейросетевых моделей, используемых при прогнозировании состояния технических объектов.

### МЕТОДИКА ФОРМИРОВАНИЯ АРХИТЕКТУР НЕЙРОННЫХ СЕТЕЙ НА ОСНОВЕ ЭВОЛЮЦИОННЫХ И ГЕНЕТИЧЕСКИХ МЕТОДОВ

Методика формирования архитектур нейронных сетей на основе применения эволюционных и генетических методов содержит четыре этапа.

*Этап 1. Формирование графа архитектуры сети*

На этом этапе создается граф архитектуры нейронной сети  $G$ , который можно представить следующим образом:

$$G = (V, E), \quad (1)$$

где  $V$  – множество вершин (узлов), каждый из которых представляет один слой сети, а  $E$  – множество ребер, которые определяют соединения между узлами.

Входной слой нейронной сети состоит из  $p$  нейронов, то есть определяется по набору па-

раметров функционирования объекта  $X_1, X_2, \dots, X_p$ . Также выбирается выходной слой  $Y$ , то есть набор из  $m$  нейронов – один или несколько откликов системы на входные параметры.

Каждый  $i$ -й слой сети является функцией от входных данных, подаваемых из предыдущего ( $i-1$ )-го слоя. Однако на данном этапе входной и выходной слою не связаны, так как между ними нет промежуточных (скрытых) слоев. Задача состоит в том, чтобы найти слои, которые обеспечивают наилучшие значения метрик качества для конкретной конфигурации искусственной нейронной сети. Первый скрытый слой рассматривается как начальная популяция, а его модификации приводят к созданию новых морфизмов сети.

Граф архитектуры сети будет в дальнейшем описывать структуру построенной сети, включая количество слоев, типы слоев (например, LSTM, CNN, MLP) и их взаимосвязи.

*Этап 2. Генерация начальной популяции графов*

Для создания начальной популяции графов используется алгоритм декартова генетического программирования. При этом архитектуры сетей кодируются последовательностью чисел, определяющих структуру графа. Эти числа определяют возможные конфигурации графов, формируя начальную популяцию различных архитектур сети для дальнейшей оптимизации:

$$P_0 = \{G_1, \dots, G_n\}, (2)$$

где  $G_i = (V_i, E_i)$  – возможная конфигурация графа.

*Этап 3. Грубая настройка архитектуры с использованием метода NSGA-II*

Грубая настройка архитектур осуществляется с помощью многокритериального эволюционного алгоритма NSGA-II, который оптимизирует несколько критериев одновременно, таких как точность и сложность модели. Для настройки архитектур сетей в многокритериальной оптимизации рассматриваются функции приспособленности по нескольким критериям, формируя многомерное пространство целей. Основной задачей является нахождение «оптимального по Парето» решения, когда ни одно решение не может считаться лучшим, чем другое, без улучшения хотя бы одной цели за счет другой [15].

Цель алгоритма NSGA-II при поиске архитектуры нейронной сети – минимизировать функции потерь и одновременно максимизировать качество модели. При этом NSGA-II ранжирует популяцию по уровню доминирования и оптимизирует структуру графа.

Преимуществами применения метода NSGA-II являются быстрая недоминирующая сортировка, то есть алгоритм разделяет популяцию на различные фронты по доминированию, при этом каждый фронт содержит решения, которые не доминируются никакими другими в этом

фронте, и каждый следующий фронт доминируется хотя бы одним решением из предыдущих фронтов, а также для поддержания разнообразия популяции метод NSGA-II использует метрику «crowding distance» (расстояние уплотнения), которая измеряет плотность решений вокруг данного решения, позволяя предпочитать решения с меньшей локальной конкуренцией.

*Этап 4. Уточнение параметров с использованием алгоритма CMA-ES*

Для уточнения параметров архитектур применяется эволюционный алгоритм CMA-ES, который эффективно адаптирует стратегию поиска в многомерных непрерывных пространствах [16]. Основной идеей метода CMA-ES является использование кэшированной информации для перерасчета ковариационной матрицы, которая направляет выборку новых решений. Этот метод обновляет параметры среднего вектора и ковариационной матрицы на основе производительности текущих решений, что позволяет находить оптимальные параметры для каждой архитектуры.

Пусть  $N(m, C)$  – многомерное нормальное распределение с математическим ожиданием  $m$  и ковариационной матрицей  $C$ . В каждом поколении решения-кандидаты выбираются на основе этого распределения, и с учетом их производительности обновляются параметры. Новое среднее вычисляется как взвешенное среднее лучших решений текущего поколения:

$$m^{(g+1)} = \sum_{i=1}^{\mu} \omega_i x_{i:\lambda}^{(g+1)}, (3)$$

где  $\omega_i$  – весовые коэффициенты для рекомбинации, удовлетворяющие условиям  $\sum_{i=1}^{\mu} \omega_i = 1$  и

$\omega_i > 0 \forall i$ ,  $x_{i:\lambda}^{(g+1)}$  –  $i$ -я лучшая точка из  $\lambda$  точек текущей популяции,  $\lambda$  – размер текущей популяции,  $\mu$  – количество выбранных родительских особей или количество точек поиска.

Ковариационная матрица адаптируется таким образом, чтобы увеличить вероятность выбора решений в направлениях, в которых прошлые успешные решения имели большие различия. Формула обновления ковариационной матрицы имеет вид:

$$C^{(g+1)} = (1 - c_{cov})^{g+1} C^{(0)} + c_{cov} \sum_{i=0}^g (1 - c_{cov})^{g-i} \frac{1}{\sigma_i^2} C_{\mu}^{(i+1)}, (4)$$

где  $C^{(g+1)}$  – ковариационная матрица на итерации  $(g + 1)$ ,  $c_{cov}$  – скорость обучения для обновления ковариационной матрицы ( $0 < c_{cov} \leq 1$ ), выбор которой имеет решающее значение, так как малые значения приводят к медленному обучению, слишком большие значения приводят к неудаче, поскольку ковариационная матрица вы-

рождается, приближение первого порядка для хорошего выбора  $c_{cov} \approx \frac{\mu_{eff}}{n^2}$ ,  $\mu_{eff} = \left( \sum_{i=1}^{\mu} \omega_i^2 \right)^{-1}$ ,

$n$  – размерность пространства поиска,  $\sigma_i$  – масштабный параметр на  $i$ -м шаге,  $C^{(0)}$  – начальная ковариационная матрица,  $C_{\mu}^{(i+1)}$  – оценка ковариационной матрицы для поколения  $(i + 1)$ , вычисляемая по формуле:

$$C_{\mu}^{(g+1)} = \sum_{i=1}^{\mu} \omega_i (x_{i:\lambda}^{(g+1)} - m^{(g)})(x_{i:\lambda}^{(g+1)} - m^{(g)})^T. \quad (5)$$

Когда пространство поиска архитектур нейросетевых моделей на основе применения эволюционных и генетических алгоритмов окончательно определено и границы настроены, запускается байесовская оптимизация, которая позволяет найти оптимальную архитектуру модели на основе максимизации функции вероятности успеха, и настроить ее гиперпараметры на основе минимизации функции потерь на тестовом наборе данных.

В качестве функции потерь будем использовать функцию ошибок MAE, которую можно вычислить по формуле:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (6)$$

где  $n$  – объем выборки,  $\hat{y}_i$  – прогнозируемое по модели  $i$ -е значение переменной  $Y$ ,  $y_i$  – наблюдаемое  $i$ -е значение переменной  $Y$ .

Метрика MAE позволяет определять степень несоответствия прогноза реальным данным. Этот показатель менее подвержен влиянию выбросов и требует меньше вычислительных ресурсов по сравнению с другими критериями качества.

Дополнительно для определения качества моделей нейронных сетей будем использовать коэффициент детерминации, который показывает, насколько хорошо модель описывает данные, и вычисляется по формуле:

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (7)$$

где  $\bar{y}$  – среднее значение переменной  $Y$ .

Для предотвращения переобучения при построении моделей нейросетей применим метод кросс-валидации. При перекрестной проверке обучающая выборка делится на  $N$  блоков. Один блок используется для контроля качества модели, а остальные  $(N - 1)$  объединяются в одну выборку для обучения модели. Таким образом, последовательно перебираются все возможные варианты. Решением задачи является набор параметров модели, который обеспечивает наилучшие значения метрик качества.

Для улучшения производительности моделей и предотвращения переобучения применим дополнительные методы: Dropout, Early stopping и Learning rate scheduling. Метод Dropout является техникой регуляризации, которая случайным образом отключает нейроны во время обучения, что помогает избежать переобучения. Подход Early stopping будет использоваться для остановки обучения, если ошибка на валидационной выборке перестает уменьшаться, что также поможет предотвратить переобучение. Learning rate scheduling предполагает динамическое изменение скорости обучения в процессе тренировки, что улучшает сходимость модели и позволяет более точно находить оптимальные параметры. Эти методы вместе с основными моделями нейронных сетей обеспечат более точные прогнозы и стабильность в процессе обучения.

Для реализации разработанной методики формирования архитектур нейронных на языке программирования Python была написана программа создания и обучения нейронных сетей, где использовалось следующее программное обеспечение: IDE PyCharm Community Edition, библиотеки tensorflow, keras, pandas, matplotlib. Основной принцип работы программы состоит в автоматическом поиске оптимальной по соответствующим критериям качества архитектуры модели нейронной сети на основе описанного подхода для конкретного технического объекта, а затем использование выбранной модели для прогнозирования состояния данного объекта.

## ЧИСЛЕННОЕ ИССЛЕДОВАНИЕ

Для исследования эффективности методики формирования архитектур нейронных сетей на основе применения эволюционных и генетических алгоритмов рассмотрим три технических объекта.

Первым объектом исследования является турбореактивный двигатель, техническое состояние которого характеризуется набором из 21 параметра функционирования (температура на входе вентилятора, давление в перепускном канале, физическая скорость вентилятора, коэффициент расхода топлива, скорость ядра, соотношения давления в двигателе и другие), значения которых поступают с датчиков [17]. Набор данных о состоянии турбореактивного двигателя включает 16000 обучающих и 4000 тестовых выборок.

Вторым объектом является литий-ионный аккумулятор. Данные о состоянии аккумулятора созданы в NASA Ames Prognostics Center of Excellence [18]. Исследование проводилось по трем разным операционным профилям: зарядка, разрядка и электрохимическая импедансная

спектроскопия (EIS) при различных температурах. Повторяющиеся циклы зарядки и разрядки привели к ускоренному старению батарей. Эксперименты были завершены, когда батареи достигли критериев окончания срока службы (EOL): 30% снижения номинальной емкости (с 2 Ач до 1,4 Ач). Набор данных включает в себя 35000 наблюдений [19].

Третий набор данных описывает состояние четырех подшипников, которые установлены на валу [20]. Скорость вращения поддерживалась постоянной на уровне 2000 оборотов в минуту при помощи асинхронного двигателя, соединенного с валом ремнями. Радиальная нагрузка в 6000 фунтов (2722 кг) передавалась на вал и подшипники через пружинный механизм. Все подшипники снабжались принудительной смазкой. Набор данных включает в себя 20480 наблюдений.

Для сравнения качества построенных моделей на основе предлагаемого подхода поиска архитектур нейронных сетей будем использовать библиотеку AutoKeras [21].

AutoKeras - это система автоматизированного машинного обучения, основанная на глубоком обучении и оптимизации гиперпараметров. Она использует эффективные методы оптимизации архитектуры моделей, такие как эвристический поиск и поиск по гиперпараметрам для автоматического определения оптимальных структур глубоких нейронных сетей под конкретные задачи.

Для построения нейросетевых моделей разобьем исходные выборки данных в соотношении 80% - для обучения модели, 10% - для валидации и 10% - для тестирования. Такое разделение поможет оценить обобщающую способность модели на новых данных.

Найдем наилучшую архитектуру нейросетевой модели для турбореактивного двигателя с помощью библиотеки AutoKeras. Полученную модель представим на рис. 1.

Нейронная сеть, представленная на рис. 1, была обучена с использованием алгоритма оптимизации Adam. Скорость обучения была задана динамически, размер батча - 32. Обучение модели проводилось на протяжении 50 эпох.

Для полученной модели на рис. 2а представим графики зависимости функции ошибок (MAE) от количества эпох обучения (Epochs) при обучении (сплошная линия) и валидации (штриховая линия), которые иллюстрируют, что примерно после 20 эпох обучения значения функции ошибок стабилизируются, что указывает на завершение процесса обучения модели.

На рис. 2б для данной модели представим графики зависимости значения коэффициента детерминации  $R^2$  от эпох обучения для обучающей выборки (сплошная линия) и валидационной выборки (штриховая линия). Из рис. 2б видно, что коэффициент  $R^2$  постепенно увеличивается по мере увеличения числа эпох, достигая стабилизации примерно после 30 эпох обучения. Это свидетельствует о том, что модель

```
Model: "sequential"
-----
```

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 21)	231
batch_normalization (Batch Normalization)	(None, 21)	84
dense_1 (Dense)	(None, 32)	704
dropout (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 1024)	33792
dropout_1 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 512)	524800
dense_4 (Dense)	(None, 64)	32832
dense_5 (Dense)	(None, 1)	65

```
=====
```

Рис. 1. Архитектура нейросетевой модели для турбореактивного двигателя, найденная с помощью библиотеки AutoKeras

постепенно улучшает свои предсказательные способности как на обучающей, так и на валидационной выборках. При этом коэффициент  $R^2$  на валидационной выборке начинает колебаться после 30 эпох, что может указывать на переобучение модели.

Теперь найдем наилучшую архитектуру нейронной сети для данных функционирования турбореактивного двигателя с помощью предлагаемой методики. Получим модель, представленную на рис. 3.

Из рис. 3 следует, что архитектура нейронной сети значительно усложнилась, так как были добавлены слои LSTM и Convolutional (свертка). Сочетание LSTM, сверток и пулинга делает архитектуру очень гибкой для разных типов данных и задач, позволяя извлекать как временные, так и пространственные зависимости в данных.

Обучим модель с использованием алгоритма оптимизации Adam, скорость обучения задается динамически, размер батча - 64. Обучение модели проведем на протяжении 50 эпох.

На рис. 4а приведем динамику функции ошибок MAE в зависимости от эпох обучения сплошной линией на обучающем наборе данных, штриховой - на валидационном. Графики на рис. 4а демонстрируют, что модель быстро обучается в первые 10-15 эпох, но затем ее способность к обобщению на новых данных перестает улучшаться. Разница между MAE на обучающей и валидационной выборках после 20-й эпохи может свидетельствовать о начале переобучения.

На рис. 4б представим зависимость коэффициента детерминации  $R^2$  от эпох обучения на обучающем наборе данных сплошной линией, на валидационном - штриховой. Из графиков на рис. 4б видно, что значение  $R^2$  значительно увеличивается в первые 10-15 эпох, что свидетельствует о том, что модель быстро улучшает свои

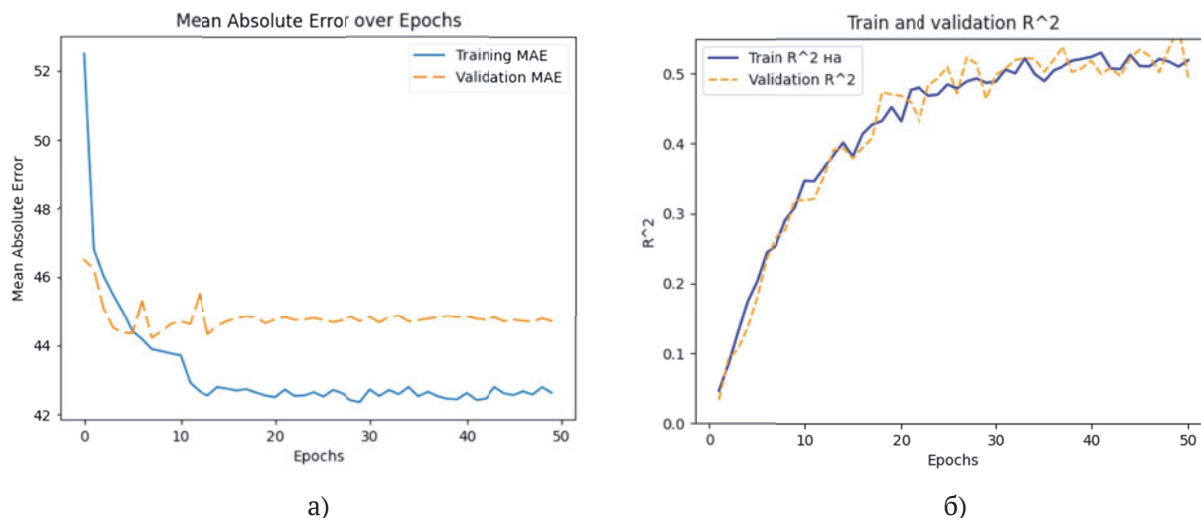
предсказательные способности. Затем значение  $R^2$  начинает стабилизироваться и окончательно принимает значение 0,96, что указывает на достижение моделью высокого уровня точности.

Аналогично с помощью библиотеки AutoKeras и разработанной методики подберем архитектуры моделей для двух других технических объектов: литий-ионного аккумулятора и подшипников.

На рис. 5 представим графики зависимости функции ошибок MAE от эпох обучения на обучающем (сплошная линия) и валидационном (штриховая линия) наборе данных для моделей литий-ионного аккумулятора, найденных на основе библиотеки AutoKeras (график а) и предлагаемого подхода (график б). По представленным графикам видно, что ошибка MAE начинает стабилизироваться уже после 10 эпох обучения, причем для модели AutoKeras ошибка принимает значение 24, а для модели, полученной по основе предлагаемой методики - 9,5.

Аналогично на рис. 6 представим графики зависимости ошибки MAE от количества эпох обучения для обучающего (сплошная линия) и валидационного (штриховая линия) наборов данных для моделей, которые описывают работу подшипников и были созданы с использованием библиотеки AutoKeras (график а) и предложенного метода (график б). Из графиков видно, что ошибка MAE стабилизируется после 15-20 эпох обучения, и для модели на основе AutoKeras она составляет 58, в то время как для модели, созданной с использованием предложенного подхода, ошибка равна 9,3.

Аналогично рассчитаем значения коэффициента детерминации для моделей, найденных с помощью библиотеки AutoKeras и предлагаемого подхода, на валидационной выборке для всех рассматриваемых объектов, все значения вычисленных метрик качества моделей представим в табл. 1.



**Рис. 2.** Зависимость функции ошибок MAE (а) и коэффициента  $R^2$  (б) от эпох обучения для нейросетевой модели турбореактивного двигателя, найденной с помощью библиотеки AutoKeras

```

Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
lstm (LSTM)                  (None, None, 64)         19200
dropout (Dropout)           (None, None, 64)         0
batch_normalization (BatchN (None, None, 64)         256
ormalization)
lstm_1 (LSTM)                (None, None, 64)         33024
dropout_1 (Dropout)         (None, None, 64)         0
batch_normalization_1 (Bate (None, None, 64)         256
hNormalization)
conv1d (Conv1D)              (None, None, 32)         6176
max_pooling1d (MaxPooling1D (None, None, 32)         0
)
dropout_2 (Dropout)         (None, None, 32)         0
batch_normalization_2 (Bate (None, None, 32)         128
hNormalization)
dense (Dense)                (None, None, 1)          33
-----

```

Рис. 3. Архитектура нейросетевой модели для турбореактивного двигателя, найденная с помощью предлагаемого подхода

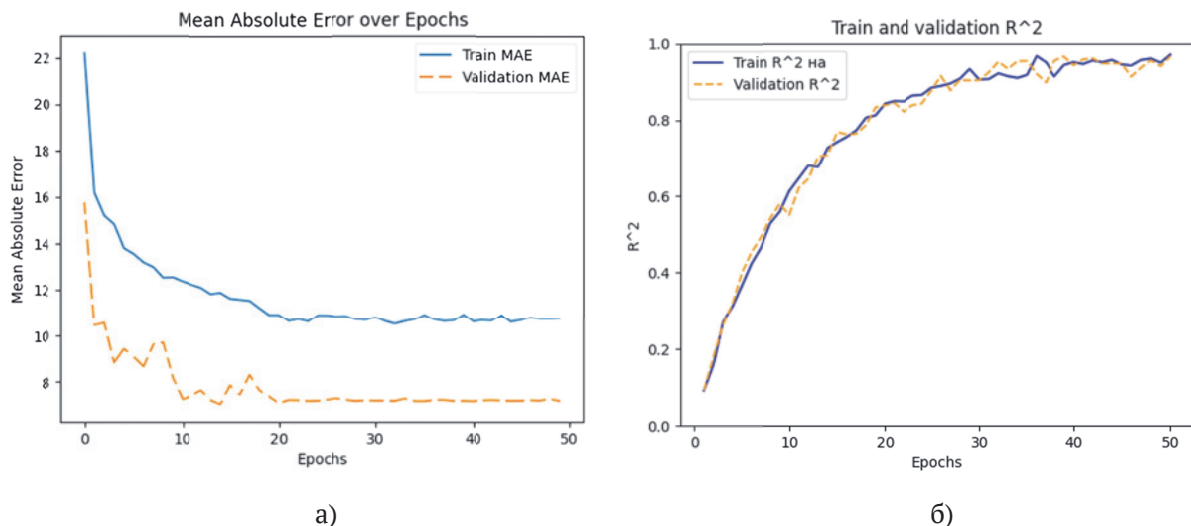


Рис. 4. Зависимость функции ошибок MAE (а) и коэффициента  $R^2$  (б) от эпох обучения для нейросетевой модели двигателя, найденной с помощью предлагаемого подхода

На основании данных из табл. 1 можно сделать вывод, что использование новой методики формирования архитектур нейронных сетей на основе применения эволюционных и генетических методов значительно улучшает метрики качества моделей нейронных сетей для всех рассмотренных технических объектов по срав-

нению с моделями, найденными с помощью библиотеки AutoKeras. Для данных турбореактивного двигателя значение функции ошибки MAE на валидационной выборке уменьшается в 6,25 раза, для данных литий-ионного аккумулятора - в 2,53 раза, для набора данных подшипников - в 6,24 раза. При этом значение коэффициента

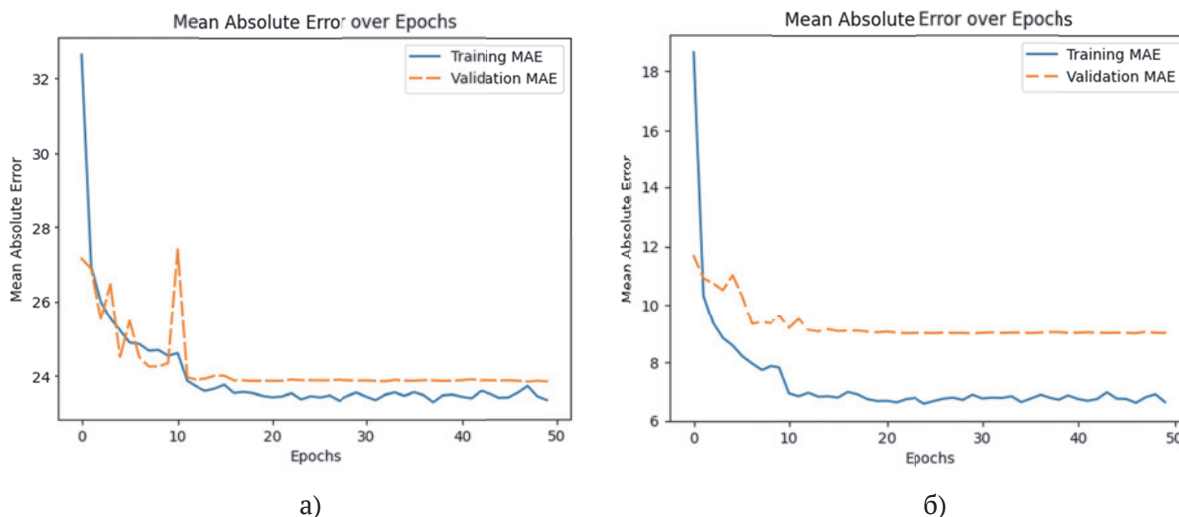


Рис. 5. Зависимость функции ошибок MAE от эпох обучения для нейросетевой модели литий-ионного аккумулятора, найденной с помощью: а) библиотеки AutoKeras, б) предлагаемого подхода

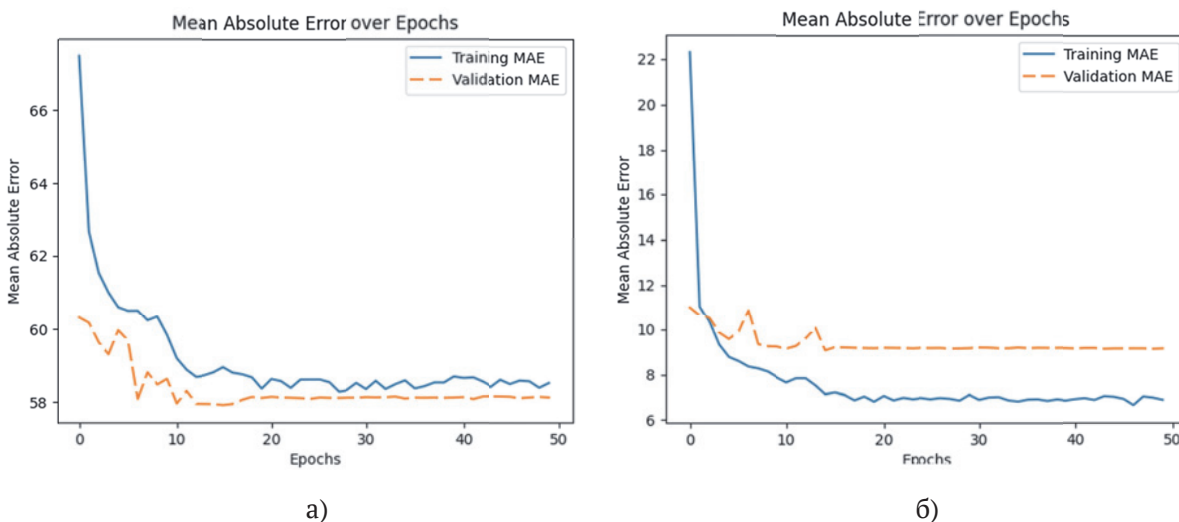


Рис. 6. Зависимость функции ошибок MAE от эпох обучения для нейросетевой модели подшипников, найденной с помощью: а) библиотеки AutoKeras, б) предлагаемого подхода

Таблица 1. Метрики качества нейросетевых моделей на валидационной выборке

Технический объект	Метод	MAE	R <sup>2</sup>
Турбореактивный двигатель	AutoKeras	45	0,52
	Предлагаемый подход	7,2	0,96
Литий-ионный аккумулятор	AutoKeras	24	0,70
	Предлагаемый подход	9,5	0,93
Подшипники	AutoKeras	58	0,43
	Предлагаемый подход	9,3	0,91

детерминации для всех наборов данных увеличивается в среднем в 1,76 раза.

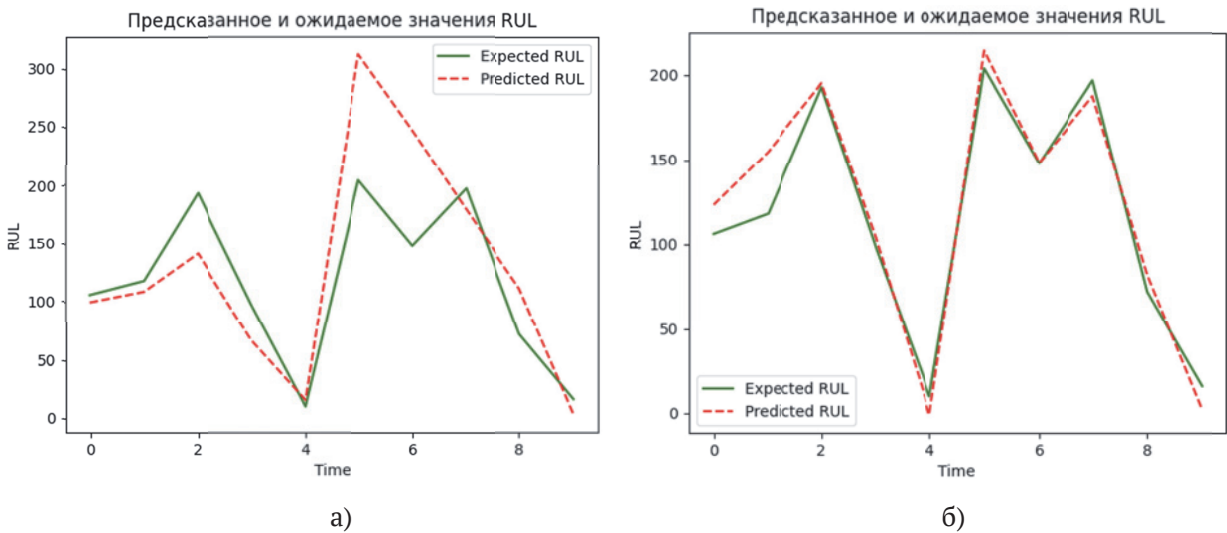
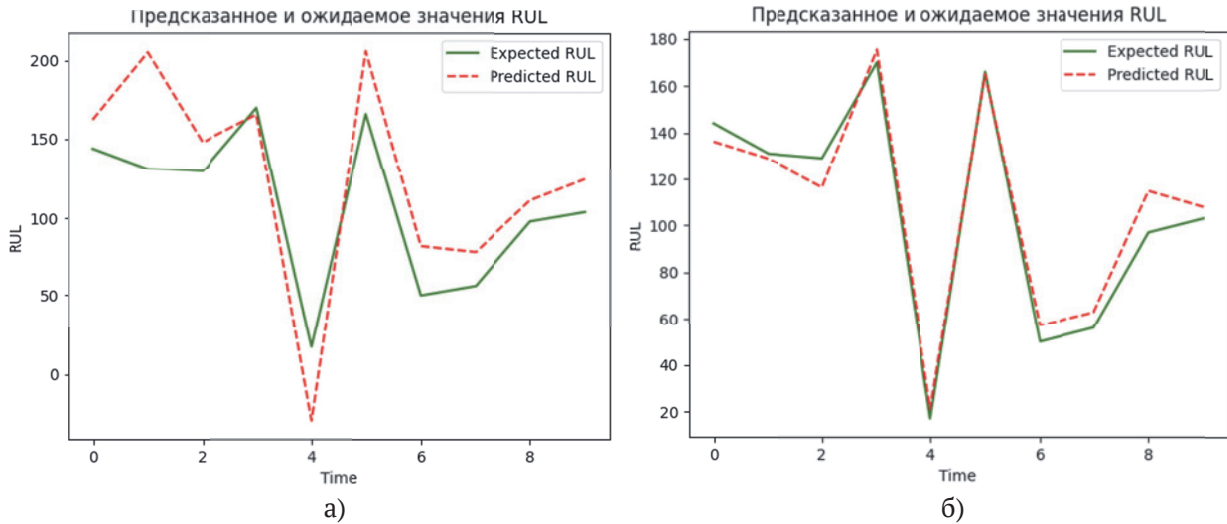
Проведем прогнозирование оставшегося срока полезного использования турбореактивного двигателя, литий-ионного аккумулятора и подшипников по полученным моделям на тестовой выборке.

На Рис. 7 показаны графики наблюдаемых (сплошная линия) и прогнозируемых (штриховая линия) значений состояния турбореактивного двигателя Y по моделям, полученным на

основе библиотеки AutoKeras (график а) и предлагаемого подхода (график б), где по оси абсцисс отложено время (Time), по оси ординат – значения Y, то есть количество успешных циклов эксплуатации технического объекта (RUL).

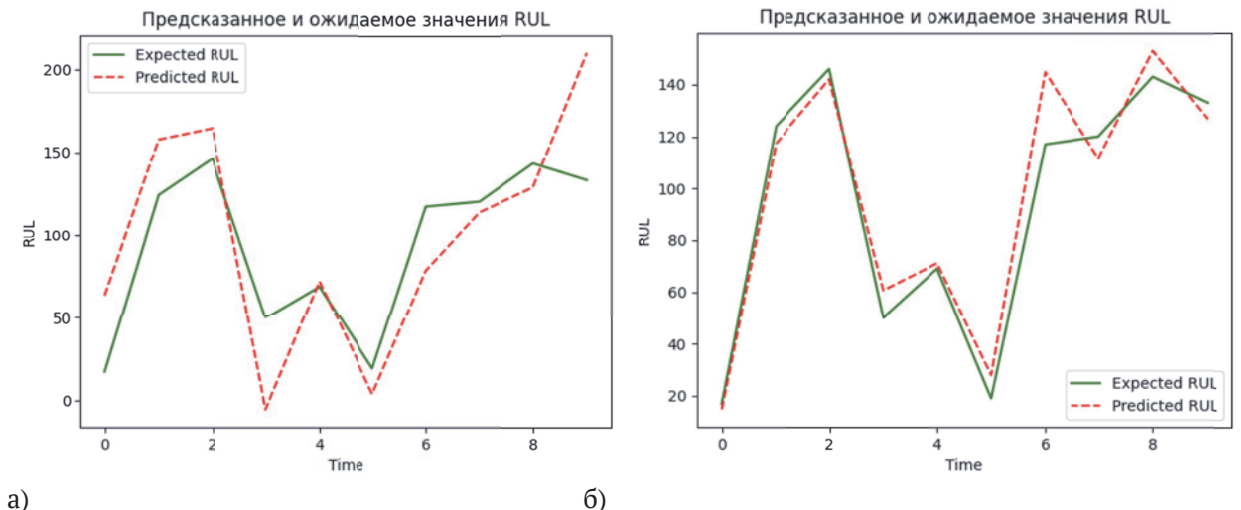
Составим прогноз для параметра RUL литий-ионного аккумулятора, результаты представим на Рис. 8а для модели AutoKeras и Рис. 8б для модели, полученной с использованием новой методики поиска архитектуры нейронной сети.





Аналогично, проведем прогнозирование оставшегося срока работы подшипников. На Рис. 9а представим график прогнозируемых и

наблюдаемых значений для модели AutoKeras, на Рис. 9б - для сети, полученной с использованием предлагаемого подхода.



**Таблица 2.** Метрики качества нейросетевых моделей при прогнозировании

Технический объект	Метод	MAE	R <sup>2</sup>
Турбореактивный двигатель	AutoKeras	38,09	0,57
	Предлагаемый подход	10,11	0,93
Литий-ионный аккумулятор	AutoKeras	54,14	0,44
	Предлагаемый подход	11,58	0,90
Подшипники	AutoKeras	64,16	0,41
	Предлагаемый подход	16,77	0,85

Из Рис. 5 - Рис. 7 видно, что значения  $Y$ , спрогнозированные моделью (штриховая линия) наименее отклоняются от наблюдаемых значений (сплошная линия) для моделей, полученных на основе новой методики поиска архитектур нейронных сетей, по сравнению с моделями на основе AutoKeras, для всех рассмотренных наборов данных.

Рассчитаем значения функции ошибок MAE и коэффициента детерминации  $R^2$  для всех нейросетевых моделей рассматриваемых объектов на тестовой выборке, которая использовалась при прогнозировании. Полученные значения метрик качества запишем в табл. 2.

Результаты, представленные в табл. 2, показывают, что применение новой методики формирования архитектуры нейронных сетей с использованием эволюционных и генетических алгоритмов при прогнозировании состояния всех рассматриваемых технических объектов позволяет в среднем снизить значение функции ошибки MAE в 4 раза, а значение коэффициента детерминации увеличить в 1,9 раза, по сравнению с моделями, построенными с помощью библиотеки AutoKeras.

Исходя из представленных результатов, можно сделать вывод, что использование новой методики поиска архитектур нейронных сетей позволит максимально точно выявить оставшийся срок службы технического объекта и, соответственно, своевременно отреагировать на нарушения в его работе.

### ЗАКЛЮЧЕНИЕ

Для прогнозирования состояния технического объекта и возможных нарушений в процессе его работы по результатам исследования набора контролируемых параметров объекта в работе предложено использовать нейросетевые модели, а для автоматического проектирования архитектур моделей разработана новая методика на основе применения эволюционных и генетических алгоритмов: алгоритма декартова генетического программирования для первичной инициализации, генетического алгоритма с мультикритериальной оптимизацией NSGA-II для дальнейшей настройки архитектур и эволюционного алгоритма CMA-ES для уточнения па-

раметров архитектур нейронных сетей. Использование данной методики позволит более гибко настраивать и адаптировать архитектуры нейронных сетей под конкретные условия задачи.

Для реализации предлагаемого подхода на языке программирования Python была написана специальная программа поиска архитектур и обучения моделей нейронных сетей.

В исследовании были рассмотрены три технических объекта: турбореактивный двигатель, литий-ионный аккумулятор и подшипники. Для оценки эффективности предложенного подхода использовалась библиотека AutoKeras. Было обнаружено, что применение этого подхода значительно улучшает качество моделей нейронных сетей для всех рассматриваемых технических объектов по сравнению с моделями, построенными с помощью AutoKeras. В частности, при прогнозировании значения функции ошибки MAE для всех наборов данных снизились в среднем в 4 раза, а коэффициент детерминации увеличился в 1,9 раза.

Этот подход может быть использован специалистами для прогнозирования технического состояния объектов в разных отраслях техники, особенно в авиации.

### СПИСОК ЛИТЕРАТУРЫ

1. Клячкин, В.Н. Прогнозирование и диагностика стабильности функционирования технических объектов / В.Н. Клячкин, В.Р. Крашенинников, Ю.Е. Кувайскова. – М.: Русайнс, 2020. – 200 с.
2. Науменко, А.П. Введение в техническую диагностику и неразрушающий контроль : учеб. пособие / А.П. Науменко. – Омск: ОмГТУ, 2019. – 152 с.
3. Kuvayiskova Y., Klyachkin V., Krashenninikov V. Recognition and Forecasting of a Technical Object State based on Its Operation Indicators Monitoring Results // International Multi-Conference on Industrial Engineering and Modern Technologies, FarEastCon 2020. 2020. P. 9271119.
4. Kang Z., Catal C., Tekinerdogan B. Remaining Useful Life (RUL) Prediction of Equipment in Production Lines Using Artificial Neural Networks // Sensors. 2021. V. 21. P. 932.
5. Gugulothu N., Vishnu TV, Malhotra P., Vig L., Agarwal P., Shroff G. Predicting Remaining Useful Life using Time Series Embeddings based on Recurrent Neural Networks // International Journal of Prognostics and Health Management. 2017. V. 9. P. 1-10.

6. *Srinivasan A., Andresen J.C., Holst A.* Ensemble Neural Networks for Remaining Useful Life (RUL) Prediction // *Proceedings of the Asia Pacific Conference of the PHM Society.* 2023. V. 4 No. 1. P. 1-6.
7. *Сай, В.К.* Метод прогнозирования остаточного ресурса на основе обработки многообъектных сложных систем / В.К. Сай, М.В. Щербаков // *Прикаспийский журнал: управление и высокие технологии.* – 2019. – № 1(45). – С. 33-44.
8. *Stanley K. O., Miikkulainen R.* Evolving Neural Networks through Augmenting Topologies // *Evolutionary Computation.* 2002. V. 10. No. 2. P. 99-127.
9. *Garcia-Pedrajas N., Hervás-Martínez C., Muñoz-Perez J.* COVNET: A Cooperative Coevolutionary Model for Evolving Artificial Neural Networks // *IEEE Transactions on Neural Networks.* 2003. V. 14. No. 3. P. 575-593.
10. *Wierstra D., Schaul T., Glasmachers T., Sun Y., Peters J., Schmidhuber J.* Natural Evolution Strategies // *Journal of Machine Learning Research.* 2014. V. 15. P. 949-980.
11. *Whitley D., Starkweather T., Bogart C.* Genetic algorithms and neural networks: optimizing connections and connectivity // *Parallel Computing.* 1990. V. 14. No. 3. P. 347-361.
12. *NourAshrafoddin N., Vahdat A. R., Ebadzadeh M. M.* Automatic Design of Modular Neural Networks Using Genetic Programming // *Artificial Neural Networks - ICANN 2007: 17th International Conference, Proceedings, Part I* 17. 2007. P. 788-798.
13. *Storn R., Price K.* Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces // *Journal of Global Optimization.* 1997. V. 11. P. 341-359.
14. *Miller J.F.* Cartesian Genetic Programming. Heidelberg: Springer, 2011. 346 p.
15. *Chatterjee S., Sarkar S., Dey N., Ashour A. S., Sen S.* Hybrid non-dominated sorting genetic algorithm: II-neural network approach // *Advancements in Applied Metaheuristic Computing.* 2018. P. 264-286.
16. *Hansen N.* The CMA Evolution Strategy: A Comparing Review // *StudFuzz.* 2006.V. 192. P. 75-102.
17. NASA's Open Data Portal, C-MAPSS Aircraft Engine Simulator Data. URL: [https://data.nasa.gov/dataset/C-MAPSS-Aircraft-Engine-Simulator-Data/xautbemq/about\\_data](https://data.nasa.gov/dataset/C-MAPSS-Aircraft-Engine-Simulator-Data/xautbemq/about_data) (дата обращения: 30.05.2024).
18. NASA Dashlink, Li-ion Battery Aging Datasets. URL: <https://c3.ndc.nasa.gov/dashlink/resources/133/> (Дата обращения: 30.05.2024).
19. *Zhang C., He Y., Yuan L., Xiang S., Wang J.* Prognostics of Lithium-Ion Batteries Based on Wavelet Denoising and DE-RVM // *Computational Intelligence and Neuroscience.* 2015. V. 2015. P. 1-8.
20. NASA Intelligent Systems Division. Discovery and Systems Health - PCoE Data Set Repository. URL: <https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/> (дата обращения: 30.05.2024).
21. *Jin H., Song Q., Hu X.* Auto-keras: An efficient neural architecture search system // *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining.* 2019. P. 1946-1956.

## APPLICATION OF EVOLUTIONARY AND GENETIC ALGORITHMS IN FORMING THE ARCHITECTURE OF NEURAL NETWORK MODELS FOR FORECASTING THE STATE OF A TECHNICAL OBJECT

© 2024 Yu.E. Kuvayskova, A.A. Nemykin

Ulyanovsk State Technical University, Ulyanovsk, Russia

In order to reduce the time for eliminating critical situations in the operation of a technical facility, it is necessary to respond in a timely manner to disturbances in its functioning. This raises the problem of predicting the state of an object and possible disturbances in its operation based on the results of studying a set of controlled parameters of the object. In this study, the problem is solved based on neural network models. However, when constructing a neural network model, it is necessary to carefully select the model architecture to ensure the best accuracy of predicting the state of objects. In this paper, a new technique is proposed for the automatic design of neural network models, which consists of the sequential use of three evolutionary algorithms: the Cartesian genetic programming (CGP) algorithm for primary initialization, the multi-criteria evolutionary algorithm NSGA-II for tuning neural network architectures, and the evolutionary algorithm CMA-ES for refining the architecture parameters. To assess the quality of forecasting using neural network models, the mean absolute error (MAE) and the determination coefficient (R<sup>2</sup>) are used. Cross-validation is used to exclude the possibility of fitting the model to the optimal forecast characteristics. It allows obtaining unbiased estimates of quality metrics. To implement these methods and models in the Python programming language using the tensorflow and keras libraries, a special program was written. The objects of the study were a turbojet engine, a lithium-ion battery, and bearings. The AutoKeras library was used to compare the effectiveness of the proposed method. The study showed that the use of the proposed approach significantly improves the quality metrics of neural network models for all technical objects compared to models found using the AutoKeras library: the value of the MAE error function for all data sets during forecasting decreases by an average of 4 times, and the value of the determination coefficient increases by 1.9 times. This approach can be used by specialists to predict the technical condition of objects in various fields of technology, especially in aviation.

*Keywords:* technical object, forecasting, neural networks, genetic and evolutionary algorithms.

DOI: 10.37313/1990-5378-2024-26-4(3)-383-394

EDN: MHQLAG

REFERENCES

1. *Klyachkin, V.N.* Prognozirovanie i diagnostika stabil'nosti funkcionirovaniya tekhnicheskikh ob'ektov / V.N. Klyachkin, V.R. Krashennnikov, YU.E. Kuvajskova. – M.: Rusajns, 2020. – 200 s.
2. *Naumenko, A.P.* Vvedenie v tekhnicheskuyu diagnostiku i nerazrushayushchij kontrol' : ucheb. posobie / A.P. Naumenko. – Omsk: OmGTU, 2019. – 152 s.
3. *Kuvayskova Y., Klyachkin V., Krashennnikov V.* Recognition and Forecasting of a Technical Object State based on Its Operation Indicators Monitoring Results // International Multi-Conference on Industrial Engineering and Modern Technologies, FarEastCon 2020. 2020. P. 9271119.
4. *Kang Z., Catal C., Tekinerdogan B.* Remaining Useful Life (RUL) Prediction of Equipment in Production Lines Using Artificial Neural Networks // Sensors. 2021. V. 21. P. 932.
5. *Gugulothu N., Vishnu TV, Malhotra P., Vig L., Agarwal P., Shroff G.* Predicting Remaining Useful Life using Time Series Embeddings based on Recurrent Neural Networks // International Journal of Prognostics and Health Management. 2017. V. 9. P. 1-10.
6. *Srinivasan A., Andresen J.C., Holst A.* Ensemble Neural Networks for Remaining Useful Life (RUL) Prediction // Proceedings of the Asia Pacific Conference of the PHM Society. 2023. V. 4 No. 1. P. 1-6.
7. *Saj, V.K.* Metod prognozirovaniya ostatochnogo resursa na osnove obrabotki mnogoob'ektnyh slozhnyh sistem / V.K. Saj, M.V. SHCHerbakov // Prikaspijskij zhurnal: upravlenie i vysokie tekhnologii. – 2019. – № 1(45). – S. 33-44.
8. *Stanley K. O., Miikkulainen R.* Evolving Neural Networks through Augmenting Topologies // Evolutionary Computation. 2002. V. 10. No. 2. P. 99-127.
9. *Garcia-Pedrajas N., Hervas-Martinez C., Muñoz-Perez J.* COVNET: A Cooperative Coevolutionary Model for Evolving Artificial Neural Networks // IEEE Transactions on Neural Networks. 2003. V. 14. No. 3. P. 575-593.
10. *Wierstra D., Schaul T., Glasmachers T., Sun Y., Peters J., Schmidhuber J.* Natural Evolution Strategies // Journal of Machine Learning Research. 2014. V. 15. P. 949-980.
11. *Whitley D., Starkweather T., Bogart C.* Genetic algorithms and neural networks: optimizing connections and connectivity // Parallel Computing. 1990. V. 14. No. 3. P. 347-361.
12. *NourAshrafoddin N., Vahdat A. R., Ebadzadeh M. M.* Automatic Design of Modular Neural Networks Using Genetic Programming // Artificial Neural Networks - ICANN 2007: 17th International Conference, Proceedings, Part I 17. 2007. P. 788-798.
13. *Storn R., Price K.* Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces // Journal of Global Optimization. 1997. V. 11. P. 341-359.
14. *Miller J.F.* Cartesian Genetic Programming. Heidelberg: Springer, 2011. 346 p.
15. *Chatterjee S., Sarkar S., Dey N., Ashour A. S., Sen S.* Hybrid non-dominated sorting genetic algorithm: II-neural network approach // Advancements in Applied Metaheuristic Computing. 2018. P. 264-286.
16. *Hansen N.* The CMA Evolution Strategy: A Comparing Review // StudFuzz. 2006.V. 192. P. 75-102.
17. NASA's Open Data Portal, C-MAPSS Aircraft Engine Simulator Data. URL: [https://data.nasa.gov/dataset/C-MAPSS-Aircraft-Engine-Simulator-Data/xaut-bemq/about\\_data](https://data.nasa.gov/dataset/C-MAPSS-Aircraft-Engine-Simulator-Data/xaut-bemq/about_data) (data obrashcheniya: 30.05.2024).
18. NASA Dashlink, Li-ion Battery Aging Datasets. URL: <https://c3.ndc.nasa.gov/dashlink/resources/133/> (Data obrashcheniya: 30.05.2024).
19. *Zhang C., He Y., Yuan L., Xiang S., Wang J.* Prognostics of Lithium-Ion Batteries Based on Wavelet Denoising and DE-RVM // Computational Intelligence and Neuroscience. 2015. V. 2015. P. 1-8.
20. NASA Intelligent Systems Division. Discovery and Systems Health - PCoE Data Set Repository. URL: <https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/> (data obrashcheniya: 30.05.2024).
21. *Jin H., Song Q., Hu X.* Auto-keras: An efficient neural architecture search system // Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019. P. 1946-1956.

*Yuliya Kuvayskova, Candidate of Technical Sciences, Associate Professor, Head of the Department of Applied Mathematics and Informatics. E-mail: u.kuvayskova@mail.ru*  
*Nemykin Alexander Andreevich, Master's Student at the Department of Applied Mathematics and Informatics. E-mail: nemykin.alexander@yandex.ru*