ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

УДК 004.855.5

МАШИНА ЭКСТРЕМАЛЬНОГО ОБУЧЕНИЯ В ЗАДАЧАХ ПРЕДСКАЗАНИЯ ОСТАТОЧНОГО СРОКА ПОЛЕЗНОЙ СЛУЖБЫ ДИСКОВЫХ НАКОПИТЕЛЕЙ

Л. А. Демидова, д.т.н., профессор кафедры корпоративных информационных систем РТУ МИРЭА, Москва, Россия;

orcid.org/0000-0003-4516-3746, e-mail: demidova.liliya@gmail.com

И. А. Фурсов, аспирант кафедры корпоративных информационных систем РТУ МИРЭА, Москва, Россия;

orcid.org/0000-0001-8583-4351, e-mail: ilya.fursow@gmail.com

Рассматривается задача предсказания остаточного срока полезной службы дисковых накопителей моделью машинного обучения. Целью работы является создание нейросетевой модели на основе машины экстремального обучения ELM (Extreme Learning Machine), а также улучшение значений метрик качества модели посредством генерации новых признаков-предикторов для получения результатов, сопоставимых со значениями метрик качества моделей на основе других нейронных сетей прямого распространения. Показано, что рекуррентные нейронные сети типа SimpleRNN (Simple Recurrent Neural Network), а также их улучшенные версии в виде нейронной сети с долгой краткосрочной памятью LSTM (Long-Short Term Memory) и управляемого рекуррентного блока GRU (Gated Recurrent Unit) показывают хорошую обобщающую способность, однако время их обучения может быть большим, в то время как машине экстремального обучения требуется намного меньше времени на обучение.

Ключевые слова: остаточный срок полезной службы, дисковый накопитель, нейронная сеть, временной ряд, машинное обучение, ELM, SimpleRNN, LSTM, GRU.

DOI:10.21667/1995-4565-2023-83-22-35

Введение

Каждая серверная система не обходится без дисковых накопителей. В настоящий момент почти все данные обрабатываются и хранятся на жестких и твердотельных накопителях разных серверов и центров обработки данных. В течение срока эксплуатации накопителей нежелательно, чтобы вышел из строя хотя бы один из них, так как на нём могут храниться данные пользователей, потеря которых обычно может очень дорого стоить для компаний.

В каждый современный дисковый накопитель встроена система самоконтроля, анализа и отчетности (Self-Monitoring, Analysis and Reporting Technology, SMART), которая представляет собой набор SMART-датчиков, каждый из которых следит за определенным параметром накопителя, например температурой, числом «битых» секторов и т.д. [9, 10].

Существуют специальные системы мониторинга, которые собирают данные показателей со всех накопительных устройств. Это очень большой объём информации, который может превышать терабайты. Так, например, компания BackBlaze выложила отчет за 2020 год, где привела статистику по почти 163 тысячам дисковых накопителей, 1302 из которых вышли из строя. При этом среднее время жизненного цикла диска составило 25 месяцев [1].

Человеку затруднительно самостоятельно проанализировать такой объём данных. Подходы, позволяющие получить остаточный срок полезной службы дискового накопителя (Remaining Useful Life, RUL), условно можно разделить на два вида. Первый предполагает использование нескольких показателей датчиков в единой формуле, результатом которой является примерная оценка RUL. Такой подход использует, например, Samsung [2]. Во втором, который считается более прогрессивным, используются алгоритмы машинного обучения [3, 11, 13]. Его преимущество заключается в том, что при расчете модель может учитывать степень воздействия каждого датчика на итоговое значение RUL, а также скрытые закономерности в данных, которые могут заключаться в косвенном влиянии показателя одного датчика на другой [4, 14].

Самыми эффективными по точности принимаемых решений моделями нейронных сетей в задаче предсказания остаточного срока службы тех или иных устройств являются рекуррентные сети. Эффективность таких моделей достигается за счёт использования встроенного механизма памяти, благодаря которому модель способна обучаться, опираясь на свой предыдущий опыт. Однако если объем набора данных, используемого для обучения модели, будет очень большим, а архитектура такой модели будет многослойной, то процесс обучения может быть очень долгим. ELM сеть отличается от рекуррентной сети своей простотой и другим подходом к механизму обучения, заключающимся в вычислении обратной матрицы Мура – Пенроуза, что в итоге обеспечивает более высокую скорость обучения при чуть менее хороших значениях метрик качества, рассчитываемых при обучении [5].

В предлагаемой статье рассматривается решение задачи прогнозирования остаточного срока полезной службы дисковых накопителей данных. При решении этой задачи применяются рекуррентные нейронные сети типа SimpleRNN и их модификации, в частности LSTM и GRU [3, 6]. При этом каждая модель обучается на двух разных наборах данных: в первом наборе в качестве признаков-предикторов используются значения показателей SMART-датчиков, сопоставленные с временными рядами (BP), а во втором наборе используемые признаки-предикторы сгенерированы на основе значений показателей SMART-датчиков.

Теоретическая часть

В задаче обучения с учителем обычно даны набор наблюдений X и набор значений целевого признака Y.

Набор наблюдений X имеет вид $\{\mathbf{x}_1,...,\mathbf{x}_G\}$, где G – число наблюдений, при этом отдельно взятое наблюдение $\mathbf{x}_j (j = \overline{1,G})$ имеет вид $\mathbf{x}_j = \{x_{j,1},...,x_{j,K}\}$, где $x_{j,K} - j$ -е значение для k-го признака-предиктора $(j = \overline{1,G}; k = \overline{1,K}); K$ – число признаков-предикторов.

Набор значений целевого признака Y имеет вид $\{y_1, ..., y_G\}$, где $y_j - j$ -е значение целевого признака $(j = \overline{1,G})$; G – число наблюдений. Каждому значению целевого признака y_j соответствует наблюдение \mathbf{x}_j , значения в котором определяются на основе *j*-х значений всех признаков-предикторов.

Рекуррентные нейронные сети способны передавать поступающую информацию в следующую итерацию во время обучения [3]. Благодаря этому учитываются предыдущие входные данные, что является расширением возможностей сетей прямого распространения..

Каждая скрытая *j*-я ячейка представляет собой вектор $\mathbf{h}_{j} = \{h_{j,1}, \dots, h_{j,G}\}$, где $h_{j,i} - i$ -е значение для *j*-й ячейки ($i = \overline{1,L}$; $j = \overline{1,G}$),); L – число скрытых ячеек. При этом \mathbf{h}_{j+1} вычисляется как:

$$\mathbf{h}_{j+1} = \tanh(\mathbf{W}\mathbf{h}_j + \mathbf{U}\mathbf{x}_{j+1} + \mathbf{b}), \ j = \overline{\mathbf{0}, \mathbf{G} - \mathbf{1}},\tag{1}$$

где \mathbf{h}_{j+1} – текущая скрытая ячейка; \mathbf{h}_j – предыдущая скрытая ячейка; \mathbf{x}_j , – входной вектор; W и U – матрицы весов соединений; b – вектор смещения. Ячейку \mathbf{h}_0 обычно инициируют нулями

Выходной вектор y_i при скрытой ячейке \mathbf{h}_i вычисляется как

$$\mathbf{y}_{i} = \mathbf{V}\mathbf{h}_{i} + \mathbf{c},\tag{2}$$

где V – вектор весов при скрытой ячейке \mathbf{h}_i ; \mathbf{c} – вектор смещения.

Исходя из формул (1) и (2), каждая скрытая ячейка передает результаты своих вычислений другой скрытой ячейке, за счет чего учитываются предыдущие входные данные. Для простоты архитектуру такой сети можно представить в виде развертки, показанной на рисунке 1.





Каждый слой развертки в *j*-й момент времени включает в себя входной вектор \mathbf{x}_j , и выходной вектор \mathbf{y}_j размером 1xK и Gx1 соответственно, где G – число наблюдений (число значений признака-предиктора); K – число признаков-предикторов. Между входным и выходным векторами находится скрытая ячейка \mathbf{h}_j U, W – матрицы весов размером KxL и LxL соответственно, где L – число скрытых ячеек; используемые для вычисления текущей скрытой ячейки \mathbf{h}_j , a V – вектор размером Gx1, используемый для прогнозирования компонент выходного вектора \mathbf{y}_{j+1} в (*j* + 1)-й момент времени. Матрица W, от которой зависит скрытая ячейка \mathbf{h}_{i+1} , передается на следующий слой развертки.

Принципиальная разница между обычной рекуррентной сетью – SimpleRNN – и рекуррентными сетями с памятью – LSTM и GRU заключается в структуре их ячеек. В ячейке LSTM сети используются три вентиля: «исключающий», «входной» и «выходной». В основе каждого вентиля заложена сигмоидальная функция [2], которая определяет, какую долю информации пропустить дальше к следующему вентилю. Сигмоидальная функция возвращает число из диапазона [0, 1], при этом «0» означает «полностью отбросить информацию», а «1» – «полностью сохранить информацию». «Исключающий» вентиль определяет, какую информацию можно исключить из состояния ячейки, «входной» вентиль решает, какую инфорцию в текущем состоянии ячейки нужно обновить, а «выходной» вентиль решает, какую информацию следует получить в выходном сигнале ячейки. Ячейка GRU имеет два вентиля: первый вентиль определяет долю информации, которую нужно пропустить дальше по развёртке, второй вентиль определяет, какую информацию нужно отбросить [12, 15].

Для того чтобы организовать процесс обучения нейронной сети прямого распространения, обычно используется техника градиентного спуска, которая вычисляет градиент функции ошибки и изменяет веса модели с целью достижения глобального минимума ошибки.

В ELM сети вместо градиентного спуска используется инверсия Мура – Пенроуза [5, 19], благодаря которой вся архитектура нейронной сети состоит из одного скрытого слоя. Из-за этого скорость обучения увеличивается.

Выходной вектор ELM сети вычисляется как

$$t_j = \sum_{b=1}^{L} \beta_b f(\boldsymbol{\omega}_b * \mathbf{x}_j + \mathbf{b}_b), \ j = \overline{1, G},$$
(3)

где \mathbf{x}_{j} , – входной вектор, компоненты которого формируются на основе *j*-го наблюдения (на основе значений признаков-предикторов в *j*-м наблюдении); t_{j} – предсказанное значение целевого признака для *j*-го наблюдения; L – число скрытых нейронов; G – число наблюдений (число значений по каждому признаку-предиктору или по целевому признаку); β_{b} – *b*-я компонента вектора весов между скрытым и выходным слоями, компонентами которого являются t_{j} ; $\mathbf{\omega}_{b}$ – вектор весов между входным и скрытым слоями; f – функция активации; \mathbf{b}_{b} – вектор смещений.

Формулу (3) можно записать в виде:

$$\mathbf{T} = \mathbf{H}\boldsymbol{\beta},\tag{4}$$

где

$$\mathbf{T} = \begin{bmatrix} t_1 \\ \vdots \\ t_G \end{bmatrix}_{G \times 1}, \ \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_L \end{bmatrix}_{L \times 1},$$
$$\mathbf{H} = \begin{bmatrix} f(\boldsymbol{\omega}_1 * \mathbf{x}_1 + \mathbf{b}_1) & \cdots & f(\boldsymbol{\omega}_L * \mathbf{x}_1 + \mathbf{b}_L) \\ \vdots & \ddots & \vdots \\ f(\boldsymbol{\omega}_1 * \mathbf{x}_G + \mathbf{b}_1) & \cdots & f(\boldsymbol{\omega}_L * \mathbf{x}_G + \mathbf{b}_L) \end{bmatrix}_{G \times L},$$

где Н – выходная матрица скрытого слоя нейронной сети.

В случае, когда набор значений целевого признака Y является матрицей размерностью *GxM*, β и T будут иметь размерности *LxM* и *GxM* соответственно, где *M* – размерность каждой компоненты набора значений целевого признака Y.

В общем виде процесс обучения выглядит так:

- случайным образом определить веса $\boldsymbol{\omega}_{b}$ и смещения $\mathbf{b}_{b}(b=1,L)$;
- вычислить выход скрытого слоя Н;
- вычислить выходную матрицу $\beta = H^{\dagger}Y$, где H^{\dagger} обратная матрица Мура-Пенроуза;
- использовать β для предсказания новых целевых переменных $T = H\beta$.

Архитектуру машины экстремального обучения с учётом формул (3) и (4) можно представить в виде схемы, изображенной на рисунке 2.



Рисунок 2 – Архитектура ELM сети Figure 2 – Architecture of ELM network

Разные дисковые накопители могут иметь по каждому признаку-предиктору существенного различающиеся диапазоны изменения показаний датчиков, что может привести к искаженному восприятию данных моделью. В связи с этим было принято решение о применении стандартизации к значениям показаний датчиков по каждому *k*-му признаку-предиктору $(k = \overline{1, K}; K - число признаков-предикторов):$

$$vs_k^j = \frac{v_k^j - m_k}{\sigma_k},\tag{5}$$

где v_k^j – исходное *j*-е значение по *k*-му признаку-предиктору; vs_k^j – стандартизированное *j*-е значение по *k*-му признаку-предиктору; m_k – среднее арифметическое по всем значениям *k*-го признака-предиктора; σ_k – стандартное отклонение по всем значениям *k*-го признака-предиктора; $k = \overline{1, K}$.

При реализации процедуры отбора признаков-предикторов были рассмотрены следующие методы:

 метод ВҮ (метод Беньямини – Йекутили, Benjamini-Yekutieli method) [18], основанный на отборе уровней значимости коэффициентов ранговой корреляции Кендалла между признаком-предиктором и целевым признаком [20];

– метод MI (Mutual Information method), основанный на расчете взаимной информации между признаком-предиктором и целевым признаком [17];

– метод SKL (Symmetric Kullback-Leibler divergence method), основанный на расчете симметричного расстояния Кульбака – Ляйблера между признаком-предиктором и целевым признаком [17].

Реализация этих трёх методов предполагает использование вектора целевой переменной **Y** и множества векторов признаков-предикторов $X = \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_k\}$, где \mathbf{s}_k – вектор *k*-го признака-предиктора; $k = \overline{1, K}$; K – число признаков-предикторов.

При использовании метода ВҮ необходимо предварительно вычислить уровень значимости p_val_k каждого *k*-го признака-предиктора ($k = \overline{1, K}$), представленного вектором \mathbf{s}_k , на основе коэффициента ранговой корреляции Кендалла τ_k [20], который рассчитывается следующим образом.

Из набора данных отбирается по две пары наблюдений (s_{ak}, y_a) , (s_{bk}, y_b) , где $k = \overline{1, K}$; $a = \overline{1, G}$; $b = \overline{1, G}$; K – число признаков-предикторов; G – число значений k-го признакапредиктора. Отобранные пары являются конкордантными, если выполняется следующее условие:

 $(s_{ak} > s_{bk} \text{ И } y_a > y_b)$ ИЛИ $(s_{ak} < s_{bk} \text{ И } y_a < y_b)$.

В противном случае пары считаются дисконкордантными.

Для каждого k-го признака-предиктора (k = 1, K) вычисляются число конкордантных пар *nconcr_k* и число дисконкордантных пар *ndisconcr_k*.

Коэффициент ранговой корреляции Кендалла τ_k для *k*-го признака-предиктора определяется как

$$\tau_k = \frac{2^* (nconcr_k - ndisconcr_k)}{G(G-1)},\tag{6}$$

где τ_k — коэффициент ранговой корреляции Кендалла для *k*-го признака-предиктора; *nconcr_k* — число конкордантных пар для *k*-го признака-предиктора; *ndisconcr_k* — число дисконкордантных пар для *k*-го признака-предиктора; *G* — число значений *k*-го признакапредиктора. В результате можно сформировать вектор **p_val**, компонентами которого являются числа p_val_k , равные значениям коэффициента ранговой корреляции Кендалла $\tau_k(k=\overline{1,K})$ из формулы (6).

Вектор **p_val** используется в методе Беньямини – Йекутили при реализации процедуры отбора признаков-предикторов, которая может быть представлена следующей последовательностью шагов.

1. Компоненты p_val_k (k = 1, K) вектора **p_val** сортируются по возрастанию.

2. Каждой компоненте p_val_k $(k = \overline{1, K})$ вектора **p_var** ставится в соответствие нулевая гипотеза Z_k $(k = \overline{1, K})$.

3. Вычисляется такой максимальный индекс о, для которого выполняется неравенство:

$$p_val_o \leq \frac{o}{K} * \frac{q}{c},$$

где *p_val_o* – *o*-я компонента вектора **p_val**; *K* – число признаков-предикторов; *q* – ожидаемая доля ложных отклонений, которая рассматривается как константа, равная 0,05; *c* – константа, значение которой рассчитывается по формуле:

$$c = \sum_{k=1}^{K} \frac{1}{k} \; .$$

4. Отклоняются все гипотезы с индексами $k \le 0$ и, как следствие, отбрасываются все признаки-предикторы, которым соответствуют p_val_k с такими индексами k.

Методы *MI* и *SKL* основаны на расчете энтропии признаков-предикторов исследуемого набора данных. Идея использования энтропии и связанных с ней понятий из теории информации является естественной, поскольку задачу понижения размерности можно сформулировать в терминах выбора наиболее значимых, то есть информативных, признаков-предикторов. Традиционно понятие энтропии используется для определения количества информации, заложенной в цифровом сигнале [16], поэтому, вычисляя энтропию отдельно для каждого признака-предиктора, можно определить, какие признаки-предикторы являются наиболее информативными с учётом их значения энтропии.

В контексте решаемой задачи предсказания остаточного срока полезной службы дисковых накопителей вектор целевого признака **у** формируется на основе значений остаточного срока службы RUL: $\mathbf{y} = (y_1, y_2, ..., y_G)$, где G – число значений по каждому признакупредиктору; вектор k-го признака-предиктора \mathbf{s}_k ($k = \overline{1, K}$) формируется на основе значений показаний k-го датчика ($k = \overline{1, K}$): $\mathbf{s}_k = (s_{1,k}, s_{2,k}, ..., s_{G,k})$.

Взаимная информация *MI* характеризует количество информации, содержащейся в одной случайной величине относительно другой, и определяется как [17]:

$$MI(\mathbf{y}, \mathbf{s}_{k}) = \sum_{g=1}^{G} \sum_{i=1}^{K} p_{g,i}(\mathbf{y}, \mathbf{s}_{k}) \ln\left(\frac{p_{g,i}(\mathbf{y}, \mathbf{s}_{k})}{p_{i}(\mathbf{y})p_{i}(\mathbf{s}_{k})}\right),$$
(7)

где G – число значений по каждому признаку-предиктору; K – число признаков-предикторов; **у** – вектор целевого признака; \mathbf{s}_k – вектор k-го признака-предиктора; $p_{g,i}(\mathbf{y}, \mathbf{s}_k)$ – значение на пересечении g-й строки и i-го столбца матрицы, содержащей значения функции совместной плотности распределения вероятности для векторов **у** и \mathbf{s}_k ; $p_i(\mathbf{y}) - i$ -я компонента вектора, содержащего значения функции плотности распределения вероятности для вектора целевого признака **у**; $p_i(\mathbf{s}_k) - i$ -я компонента вектора, содержащего значения функции плотности распределения вероятности для вектора k-го признака-предиктора \mathbf{s}_k . Взаимная информация *MI* – это неотрицательная величина, которая принимает значение, равное нулю, для независимых случайных величин. Следовательно, отбор признаков-предикторов можно реализовать, максимизируя взаимную информацию *MI* для признаков-предикторов и целевого признака:

$$\max_{k=1, K} (MI(\mathbf{y}, \mathbf{s}_k)), \tag{8}$$

где у – вектор целевого признака; \mathbf{s}_k – вектор k-го признака-предиктора; $k = \overline{1, K}$; K – число признаков-предикторов.

Расстояние Кульбака – Ляйблера (KL) используется в теории информации и математической статистике для определения близости функций распределения различных случайных величин. При решении задачи понижения размерности на основе расстояния Кульбака – Ляйблера в качестве критерия отбора признаков-предикторов следует использовать различие функций плотности распределения вероятности признака-предиктора \mathbf{s}_k ($k = \overline{1, K}$) и целевого признака \mathbf{y} . Значение расстояния KL характеризует близость распределений значений признака-предиктора \mathbf{s}_k ($k = \overline{1, K}$) относительно распределения целевого признака \mathbf{y} . Чем больше значение расстояния KL, тем меньше похожи эти распределения, и наоборот, чем значение расстояния KL меньше, тем больше сходство между распределениями, что говорит о неинформативности рассматриваемого k-го признака-предиктора \mathbf{s}_k ($k = \overline{1, K}$).

Расстояние КL определяется как

$$KL(\mathbf{y}, \mathbf{s}_k) = \sum_{t=1}^{K} p_t(\mathbf{s}_k) \ln\left(\frac{p_t(\mathbf{s}_k)}{p_t(\mathbf{y})}\right),\tag{9}$$

где $p_t(\mathbf{s}_k) - t$ -я компонента вектора, содержащего значения функции плотности распределения вероятности для вектора *k*-го признака-предиктора \mathbf{s}_k ; $p_t(\mathbf{y}) - t$ -я компонента вектора, содержащего значения функции плотности распределения вероятности для вектора целевого признака \mathbf{y} ; K – число признаков-предикторов.

Так как расстояние *KL* несимметрично, то есть $KL(\mathbf{y}, \mathbf{s}_k) \neq KL(\mathbf{s}_k, \mathbf{y})$, то на практике обычно используется симметричный вариант для расстояния Кульбака – Ляйблера (*SKL*):

$$SKL(\mathbf{s}_k, \mathbf{y}) = KL(\mathbf{s}_k, \mathbf{y}) + KL(\mathbf{y}, \mathbf{s}_k).$$
(10)

Экспериментальные исследования

При выполнении экспериментов был использован открытый набор данных компании BackBlaze за 2013 год, содержащий данные о 35 тысячах дисковых накопителей за 8 месяцев [7]. При этом были рассмотрены показания только пяти репрезентативных SMARTдатчиков, перечень которых приведён в таблице 1.

Таблица	1 – Описание репрезентативных	SMART-датчиков
Table 1 –	Description of representative SMA	ART-sensors

Датчик	Описание				
smart_1_raw	Частота ошибок при чтении				
smart_5_raw	Число операций переназначения секторов				
smart_9_raw	Число часов, проведенных во включенном состоянии				
smart_194_raw	Температура				
smart_197_raw	Текущее число нестабильных секторов				

Показания SMART-датчиков каждого дискового накопителя использовались в совокупности в виде многомерного ВР. На рисунке 3 приведен пример такого ВР для одного из дисковых накопителей.



Рисунок 3 – Показания SMART-датчиков диска 5XW01PPF Figure 3 – Readings of SMART sensors of 5XW01PPF disk

При проведении экспериментов были разработаны модели прогнозирования на основе SimpleRNN, GRU, LSTM, а также ELM. Структура всех рекуррентных нейронных сетей была идентичной: она показана на примере LSTM сети на рисунке 4. В каждой модели на основе рекуррентной нейронной сети присутствуют 3 слоя, содержащие 100, 50 и 25 ячеек (units), начиная с первого слоя к последнему. При этом после каждого такого слоя расположен Dropout-слой с коэффициентом 0,2, означающим, что 20 % случайно выбранных ячеек в слое будут отключены (информация от них не будет в дальнейшем учитываться). Благодаря использованию Dropout-слоев уменьшается вероятность переобучения, которая свойственна нейронным сетям.

В качестве выходного используется полносвязный слой – Dense-слой с функцией активации ReLU (Rectified Linear Unit) вида:

$$f(z) = \begin{cases} 0, \ z < 0 \\ z, \ z \ge 0 \end{cases},$$
(11)

где z – элемент вектора на входе Dense-слоя.





Остаточный срок полезной службы дискового накопителя RUL играет роль целевой переменной и не содержится в наборе данных, но легко может быть вычислен на основе числа записей для конкретного дискового накопителя, так как одно показание датчика дискового накопителя соответствует одному дню.

В ходе исследований были выполнены 2 эксперимента по разработке моделей прогнозирования. В первом эксперименте был задействован набор данных из 5 исходных временных рядов, а во втором – набор данных из сгенерированных признаков-предикторов, сформированных на основе набора данных из 5 исходных временных рядов (таблица 1). При этом предварительно было сгенерировано 50 признаков-предикторов, фиксирующих, в частности, минимальные, максимальные значения исходных временных рядов, число максимальных и минимальных пиков исходных временных рядов, наиболее часто повторяющиеся значения исходных временных рядов, а также некоторые другие статистические характеристики. С целью выбора наиболее информативных и некоррелирующих между собой признаковпредикторов из 50 сгенерированных была выполнена процедура отбора.

Условием отбора признака-предиктора при применении методов *MI* и *SKL* являлось попадание значений мер близости (7) и (10) соответственно в диапазон между 3 и 4 квартилями. Результаты отбора признаков-предикторов представлены в таблице 2, и приняты следующие обозначения. На пересечении строки, соответствующей ВР для некоторого датчика, и столбца, соответствующего некоторому сгенерированному на основе этого ВР признакупредиктору, стоит символ b, m, s, сопоставленный соответственно с методом *BY*, *MI*, *SKL*, если признак-предиктор выбран; в противном случае символ, сопоставленный с методом, отсутствует (в ячейке стоит прочерк).

Анализ данных в таблице 2 показывает, что метод *BY* выбрал 45 признаков-предикторов, а методы – *MI* и *SKL* по 15.

	Сгенерированные признаки-предикторы									
Датчики	sum values	median	mean	length	standard deviation	variance	root mean square	max	absolute max	min
smart_1_raw	bm	b	b	-	b	bs	b	b	b	b
smart_5_raw	b	bm	bm	-	bms	bms	bm	bm	bm	bm
smart_9_raw	b	b	b	-	bms	bms	bs	b	b	b
smart_194_raw	b	bs	bs	-	bms	bms	bs	bs	bs	bs
smart_197_raw	b	b	bm	-	bm	bms	bm	b	b	b

Таблица 2 – Признаки-предикторы, выбранные тремя методами Table 2 – The features selected by three methods

В каждом эксперименте набор данных преобразовывался таким образом, чтобы получить массив Z, состоящий из подмассивов Z_unit_q , где q – порядковый номер подмассива. При этом выбиралось определенное число дней window_days, которое использовалось в качестве скользящего окна при сканировании многомерного временного ряда с шагом step =1. Это окно перемещалось по всей истории каждого дискового накопителя и позволяло составить набор матриц размерностью window_days, K, где K – число признаков-предикторов. Итоговое число подмассивов для конкретного дискового накопителя рассчитывалось как:

$$total_slices = \frac{disk_days - window_days}{step} + 1,$$
(12)

где disk _ days – общее число дней наблюдений для конкретного дискового накопителя.

На рисунке 5 представлен схематичный пример разделения набора данных при window days = 4 и step = 1.



Рисунок 5 – Схематичное разделение набора данных при *window_days* = 4 и *step* = 1 Figure 5 – Schematic split of dataset at *window_days* = 4 and *step* = 1

В качестве функции потерь была выбрана метрика MSE (Mean Squared Error)

$$MSE = \frac{1}{G} \sum_{j=1}^{G} (y_j - \hat{y}_j)^2, \qquad (13)$$

а для наглядной интерпретации результатов была использована метрика RMSE (Root Mean Squared Error)

$$RMSE = \sqrt{\frac{1}{G} \sum_{j=1}^{G} (y_j - \hat{y}_j)^2},$$
(14)

где G – число предсказанных значений параметра RUL; y_j – истинное значение *j*-й компоненты вектора целевого признака; \hat{y}_j – прогнозируемое значение *j*-й компоненты вектора целевого признака.

Метрики (13) и (14) минимизируются в процессе обучения.

Разработка моделей прогнозирования выполнялась в среде в Google Colab на языке Python 3.10.

Время разработки моделей на основе нейронных сетей типа SimpleRNN, LSTM, GRU и ELM с использованием исходного набора данных (т.е. набора без генерации признаковпредикторов) и наборов данных со сгенерированными и отобранными признаками-предикторами представлено в таблице 3.

Таблица 3 – Сравнение времени обучения моделей в минутах Table 3 – Comparison of models learning rate in minutes

	Время разработки модели, мин.					
Молець	Без генерации признаков- предикторов	Генерирование + отбор признаков-				
модель		предикторов				
		BY	MI	SKL		
SimpleRNN	53,33	39,91	13,33	13,46		
LSTM	15,08	11,18	4,02	4,10		
GRU	13,33	9,56	3,43	3,51		
ELM	1,12	0,6	0,1	0,1		

На основе данных таблицы 3 можно сделать вывод о том, что модель на основе ELM сети обучилась ощутимо быстрее, чем другие модели, а при отборе признаков-предикторов энтропийными методами удалось ощутимо уменьшить время обучения по сравнению с временем обучения при отборе признаков-предикторов методом *BY*. Это связано с тем, что метод *BY* выбрал 45 признаков-предикторов, а методы *MI* и *SKL* – по 15 признаков-предикторов.

В таблице 4 представлены значения метрик качества моделей. На наборах данных со сгенерированными и отобранными признаками-предикторами, качество моделей, оцениваемое с применением метрики *RMSE*, значение которой является погрешностью в предсказанном остаточном числе дней работы дискового накопителя, иногда отличается в несколько раз при использовании методов *BY*, *SKL* и *MI*, при этом значения метрик *MSE* и *RMSE* у метода *MI* оказались хуже, чем в случае работы с методом *SKL*, а именно – на 2 % и 1 % соответственно больше, поэтому для краткости приведены значения метрик качества только для методов *SKL* и *BY*. Кроме того, на основе таблицы 4 можно сделать вывод о том, что при отборе признаков-предикторов из числа сгенерированных энтропийными методами удалось получить такую же точность прогнозирования, как и при отборе признаков-предикторов методом *BY*.

	Метрики							
	Без гене признаков-п	ерации редикторов	Генерирон признаков (мет	зание + отбор -предикторов год <i>BY</i>)	Генерирование + отбор признаков-предикторов (метод SKL)			
Модели	MSE	RMSE	MSE	RMSE	MSE	RMSE		
SimpleRNN	484,38	22,03	312,26	17,66	214,91	14,66		
LSTM	60,05	7,74	10,09	3,17	9,06	3,01		
GRU	44,45	6,66	6,12	2,47	5,42	2,33		
ELM	87,23	9,34	6,27	2,5	6,2	2,49		

Таблица 4 – Сравнение метрик качества моделей Table 4 – Comparison of model quality metrics

На основе таблиц 3 и 4 можно сделать следующие выводы:

– несмотря на то, что значения метрик качества у модели на основе ELM сети немного хуже относительно моделей на основе LSTM и GRU сетей, время обучения на порядок меньше;

– модель на основе SimpleRNN сети нецелесообразно использовать в рассматриваемой задаче из-за существенно большего времени обучения и не очень хороших значений метрик качества.

Заключение

В статье рассмотрено решение задачи предсказания остаточного срока полезной службы дисковых накопителей с применением модели на основе ELM сети, а также моделей на основе рекуррентных нейронных сетей, для разработанных моделей приведены сравнительные таблицы по времени и качеству обучения.

Следует отметить, что в среднем модель на основе ELM сети всегда обучается хуже (с точки зрения значений метрик качества), чем другие модели, однако, при должной обработке данных, используемых для обучения, может приблизиться по качеству обучения к моделям на основе LSTM и GRU сетей так, что различие будет несущественным.

Библиографический список

1. Данные по дисковым накопителям за 2020 год. https://www.backblaze.com/blog/backblaze-hard-drive-stats-for-2020.

2. Li Qiang, Li Hui, Zhang Kai. A Survey of SSD Lifecycle Prediction. IEEE 10th International Conference on Software Engineering and Service Science. 2019. pp. 195-198.

3. Xu C., Wang G., Liu X., Guo D., Liu T.-Y. Health status assessment and failure prediction for hard drives with recurrent neural networks. IEEE Trans. Comput. № 65(11). 2016. pp. 3502-3508.

4. Anantharaman P., Qiao M., Jadav D. Large scale predictive analytics for hard disk remaining useful life estimation. 2018 IEEE International Congress on Big Data. 2018. pp. 251-254. 5. Huang Guang-Bin, Zhu Qin-Yu, Siew Chee. Extreme learning machine: A new learning scheme of feedforward neural networks. IEEE International Conference on Neural Networks - Conference Proceedings. № 2(2). 2004. pp. 985-990.

6. Lu S., Luo B., Patel T., Yao Y., Tiwari D., Shi W. Making disk failure predictions SMARTer. 18th USENIX Conference on File and Storage Technologies (FAST 20). 2020. pp. 151-167.

7. BackBlaze набор данных. https://www.backblaze.com/b2/hard-drive-test-data.html

8. Serneels S., De Nolf E., van Espen P. J. Spatial sign preprocessing: A simple way to impart moderate robustness to multivariate estimators. Journal of Chemical Information and Modeling, no. 46(3). 2006, pp. 1402-1409.

9. **Bagul Y. G.** Assessment of current health and remaining useful life of hard disk drives. Master thesis, Northeastern University, Boston, MA. 2009.

10. Aussel N., Jaulin S., Gandon G., Petetin Y., Fazli E., Chabridon S. Predictive models of hard drive failures based on operational data. 16th IEEE International Conference on Machine Learning and Applications (ICMLA). 2017. pp. 619-625.

11. Basak S., Sengupta S., Dubey A. Mechanisms for integrated feature normalization and remaining useful life estimation using LSTMs applied to hard-disks. 2019 IEEE International Conference on Smart Computing (SMARTCOMP). 2019, pp. 208-216.

12. Wang Q., Zheng S., Farahat A., Serita S., Gupta, C. Remaining useful life estimation using functional data analysis. ArXiv preprtint on Machine Learning. 2019.

13. Демидова Л. А., Ивкина М. С., Марчев Д. В. Применение средств машинного обучения в задаче классификации отказов в работе сложных технических систем. Материалы 1-й Международной конференции по системам управления, математическому моделированию, автоматизации и энергоэффективности. 2020. С. 540-545.

14. Андрианова Е. Г., Головин С. А., Жуков С. В., Леско С. А., Чукалина Е. Р. Обзор современных моделей и методов анализа временных рядов динамики процессов в социальных, экономических и социально-технических системах. Российский технологический журнал. 2020. 8(4). С. 7-45.

15. Демидова Л. А., Марчев Д. В. Применение рекуррентных нейронных сетей в задаче классификации отказов сложных технических систем в рамках проактивного обслуживания. Вестник Рязанского государственного радиотехнического университета. 2019. С. 135-148.

16. Cover T. M., Thomas J. A. Elements of information theory. John Wiley and Sons Ltd. New-York. 1991. 561 p.

17. Дубнов Ю. А. Об энтропийных критериях отбора признаков в задачах анализа данных. ИТиВС. 2018. № 2. С. 60-69.

18. Yoav Benjamini, Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. The Annals of Statistics, Ann. Statist. 29(4), 2001. pp. 1165-1188.

19. Демидова Л. А., Горчаков А. В. Применение биоинспирированных алгоритмов глобальной оптимизации для повышения точности прогнозов компактных машин экстремального обучения. Russ. Technol. J. 2022. С. 59-74.

20. Kendall M. G. The Treatment of Ties in Ranking Problems. Biometrika, vol. 33, no. 3. 1945. pp. 239-251.

UDC 004.855.5

ELM NEURAL NETWORKS IN THE PROBLEMS OF PREDICTING RESIDUAL USEFUL LIFE OF DISK DRIVES

L. A. Demidova, Dr. Sc. (Tech.), full professor, professor of corporate information systems department, Institute for Information Technologies, MIREA – Russian Technological University, Moscow, Russia; orcid.org/0000-0003-4516-3746, e-mail: demidova.liliya@gmail.com

I. A. Fursov, post-graduate student of corporate information systems department, Institute for Information Technologies, MIREA – Russian Technological University, Moscow, Russia;

orcid.org/0000-0001-8583-4351, e-mail: ilya.furow@gmail.com

The problem of predicting the remaining useful life of disk drives by a machine learning model using Extreme Learning Machine (ELM) tools is considered. **The aim is to** create a neural network model based on ELM, as well as to adapt it by including new generated features to obtain results comparable to those of other feed-forward neural networks. The SimpleRNN recurrent neural network that has already become standard as well as its improved versions in the form of neural network with long short-term memory (Long-Short Term Memory, LSTM) and controlled recurrent unit (Gated Recurrent Unit, GRU) show good generalization ability, however, speed learning can be long, while the extreme learning machine does spend much less time on this process. This is especially evident in the problems where there is a lot of data, while ELM generalizes almost as well as its recurrent counterparts.

Key words: residual useful life, disk storage, neural network, time series, machine learning, SimpleRNN, LSTM, GRU, ELM.

DOI:10.21667/1995-4565-2023-83-22-35

References

1. Backblaze Hard Drive Stats for 2020 // https://www.backblaze.com/blog/backblaze-hard-drive-stats-for-2020.

2. Li Qiang, Li Hui, Zhang Kai. A Survey of SSD Lifecycle Prediction. *IEEE 10th International Con*ference on Software Engineering and Service Science. 2019. pp. 195-198.

3. Xu, C., Wang, G., Liu, X., Guo, D., Liu, T.-Y. Health status assessment and failure prediction for hard drives with recurrent neural networks. *IEEE Trans. Comput.* 2016, no. 65(11), pp. 3502-3508.

4. Anantharaman P., Qiao M., Jadav D. Large scale predictive analytics for hard disk remaining useful life estimation. 2018 IEEE International Congress on Big Data. 2018, pp. 251-254.

5. Huang, Guang-Bin & Zhu, Qin-Yu & Siew, Chee. Extreme learning machine: A new learning scheme of feedforward neural networks. *IEEE International Conference on Neural Networks – Conference Proceedings*. 2004, no. 2(2), pp. 985-990.

6. Lu, S., Luo, B., Patel, T., Yao, Y., Tiwari, D., Shi, W. Making disk failure predictions SMARTer. 18th USENIX Conference on File and Storage Technologies (FAST 20). 2020, pp. 151-167.

7. BackBlaze dataset. https://www.backblaze.com/b2/hard-drive-test-data.html

8. Serneels, S., De Nolf E., and van Espen, P. J. Spatial sign preprocessing: A simple way to impart moderate robustness to multivariate estimators. *Journal of Chemical Information and Modeling*. 2006, no. 46(3), pp. 1402-1409.

9. **Bagul, Y. G.** Assessment of current health and remaining useful life of hard disk drives. Master thesis. Northeastern University, Boston, MA. 2009.

10. Aussel, N., Jaulin, S., Gandon, G., Petetin, Y., Fazli, E., Chabridon, S. Predictive models of hard drive failures based on operational data. *16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2017, pp. 619-625.

11. Basak, S., Sengupta, S., and Dubey, A. Mechanisms for integrated feature normalization and remaining useful life estimation using LSTMs applied to hard-disks. 2019 IEEE International Conference on Smart Computing (SMARTCOMP). 2019, pp. 208-216. 12. Wang, Q., Zheng S., Farahat A., Serita S., Gupta C. Remaining useful life estimation using functional data analysis. ArXiv preprtint on Machine Learning. 2019.

13. **Demidova L. A., Ivkina M. S., Marchev, D. V.** Primenenie sredstv mashinnogo obuchenija v zadache klassifikacii otkazov v rabote slozhnyh tehnicheskih system. *Materialy 1-j Mezhdunarodnoj konferencii po sistemam upravlenija, matematicheskomu modelirovaniju, avtomatizacii i jenergojeffektivnosti.* 2020, pp. 540-545. (in Russian).

14. Andrianova E. G., Golovin S. A., Zhukov S. V., Lesko S. A., Chukalina E. R. Obzor sovremennyh modelej i metodov analiza vremennyh rjadov dinamiki processov v social'nyh, jekonomicheskih i social'no-tehnicheskih sistemah. *Rossijskij tehnologicheskij zhurnal*. 2020. 8(4), pp. 7-45. (in Russian).

15. **Demidova L. A., Marchev D. V.** Primenenie rekurrentnyh nejronnyh setej v zadache klas-sifikacii otkazov slozhnyh tehnicheskih sistem v ramkah proaktivnogo obsluzhivanija. *Vestnik Rjazanskogo gosudar-stvennogo radiotehnicheskogo universiteta*. 2019, pp. 135-148. (in Russian).

16. T. M. Cover, J. A. Thomas. *Elements of information theory*. John Wiley and Sons Ltd., New-York. 1991, pp. 561.

17. **Dubnov Ju. A.** Ob jentropijnyh kriterijah otbora priznakov v zadachah analiza dannyh. *ITiVS*, no. 2. 2018, pp. 60-69. (in Russian).

18. Yoav Benjamini, Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics, Ann. Statist.* 29(4), 2001, pp. 1165-1188.

19. **Demidova L. A., Gorchakov A. V.** Primenenie Bioinspirirovannyh Algoritmov Global'noj Optimizacii Dlya Poveysheniya Tochnosti Prognozov Kompaktnyh Mashin Ekstremal'nogo Obucheniyaya. *Russ. Technol. J.* 2022, pp. 59-74. (in Russian).

20. Kendall M. G. The Treatment of Ties in Ranking Problems. *Biometrika*. 1945, vol. 33, no. 3, pp. 239-251.