УДК 621.355:681.5:004.032.26

В.Г. Букреев, Ф.Н. Хоанг

Прогнозирование степени заряженности литий-ионного аккумулятора на основе нейросети

Рассматривается задача прогнозирования степени заряженности (State of Charge, SOC) литий-ионных аккумуляторов (Lithium-Ion Battery, LIB) систем накопления электроэнергии для технологического оборудования и подвижных объектов. Предлагается подход на основе нейронных сетей, преодолевающий ограничения традиционных математических моделей, требующих предварительного знания параметров аккумулятора. Исследуются три типа архитектур: многослойные персептроны (Multi-Layer Perceptron, MLP), рекуррентные нейронные сети (Recurrent Neural Network, RNN) и управляемые рекуррентные блоки (Gated Recurrent Unit, GRU) – для анализа их эффективности прогнозирования SOC. Сформулирована методика обучения этих нейросетевых моделей на экспериментальных данных, включающих временные ряды напряжения, тока и температуры при динамических режимах разряда. Подтверждены различия в точности и вычислительных затратах: нейросеть типа MLP обучается быстрее, но не учитывает временные зависимости; нейросети типа RNN и GRU показывают меньшую ошибку на тестовой выборке, но требуют больше вычислительных затрат. Практическая применимость результатов проверена при оптимизации алгоритмов оценки SOC на вычислительных экспериментах.

Ключевые слова: литий-ионной аккумулятор, степень заряженности, нейросеть, архитектуры нейросетей: MLP, RNN, GRU.

DOI: 10.21293/1818-0442-2025-28-4-155-161

Литий-ионные аккумуляторы стали неотъемлемой частью современных технологий хранения электроэнергии благодаря высокой плотности энергии и длительному сроку службы. Их широкое применение охватывает электротранспорт, портативную электронику и системы резервного электропитания. Одним из ключевых параметров, характеризующих текущее состояние LIB, является степень заряженности, выраженная в процентах от полной ёмкости аккумулятора. Этот показатель отражает, какую долю от общей ёмкости аккумулятор способен ещё отдать, и играет решающую роль в управлении накопителями электроэнергии, обеспечении безопасности и продлении срока службы батареи.

Значение SOC(t) в текущий момент времени определяется по следующей формуле:

$$\mathrm{SOC}(t) = \frac{C_{\mathrm{т}}(t)}{C_{\mathrm{н}}} \times 100\%, \qquad (1)$$

где $C_{\mathrm{т}}(t)$ и $C_{\mathrm{н}}$ – текущая остаточная и номинальная ёмкость аккумулятора (Ач).

Точная оценка SOC аккумулятора остается сложной задачей, напрямую влияющей на надежность и безопасность систем управления аккумуляторными батареями (Battery Management System, BMS). Ошибки в оценке SOC могут привести к преждевременному старению аккумулятора и даже аварийным ситуациям [1].

Существующие подходы к оценке SOC условно делятся на модельные и основанные на временных рядах.

Так, например, метод Кулона [2] использует принцип баланса разряда и описывается формулой

$$\mathrm{SOC}(t) = \mathrm{SOC}(t_0) - \frac{1}{C_{\mathrm{H}}} \int_{t_0}^{t} I(t)dt, \qquad (2)$$

где SOC(t_0) – начальная степень заряженности; $I(t)$ – ток в момент времени t (А).

Данный метод заключается в простоте реализации и низкой вычислительной сложности. К недостаткам можно отнести ошибки при измерении тока, зависимость от точного знания начального значения, а также чувствительность к деградации ёмкости аккумулятора.

Другой модельный подход: использование эквивалентных схем замещения [3] как комбинация пассивных элементов (резисторы, конденсаторы), отражающих электрические процессы в аккумуляторе. Наиболее распространённой является модель RC-цепи, для которой напряжение описывается уравнением

$$U(t) = U_{\mathrm{OCV}}(\mathrm{SOC}) - I(t) \times R_0 - U_{\mathrm{RC}}(t), \qquad (3)$$

где $U(t)$ – напряжение аккумулятора в момент времени t, (В); U_{OCV}(SOC) – напряжение холостого хода аккумулятора (Open-circuit Voltage, OCV), зависящее от SOC (В); $I(t)$ – ток в момент времени t (А); R_0 – внутреннее сопротивление аккумулятора (Ом), $U_{\mathrm{RC}}(t)$ – напряжение на RC-цепи в момент времени t (В).

Преимущества этого метода – возможность описания переходных процессов и адаптация к различным режимам работы аккумулятора.

Несмотря на широкое распространение, традиционные методы [4–6] обладают рядом ограничений, связанных с точностью, необходимостью калибровки, чувствительностью к условиям эксплуатации и невозможностью оперативной адаптации. В последние годы методы, управляемые данными, особенно нейронные сети, привлекли внимание исследователей благодаря способности адаптироваться к изменяющимся условиям эксплуатации без необходимости детального физического моделирования аккумуляторных батарей. Например, многослойные персептроны [7] применяются для предсказания SOC, но ограничены способностью учитывать временные зависимости. Рекуррентные нейронные сети [8] учиты-

вают временные зависимости, но сталкиваются с проблемой исчезающего градиента, что ухудшает точность долгосрочных предсказаний. Улучшенная архитектура, нейронная сеть GRU [9], решает проблему исчезающего градиента, позволяя более эффективно моделировать нелинейную динамику аккумулятора.

В данной статье проводится сравнительный анализ трех типов нейронных сетей – MLP, RNN и GRU для оценки степени заряженности LIB. В работе используется методология, основанная на обучении сетей реальными данными, что позволяет учесть сложные динамические процессы внутри аккумуляторов и повысить надежность прогнозирования.

Многослойные персептроны

Многослойные перцептроны относятся к базовым типам нейросетей, используемых для решения задач регрессии и классификации. В контексте оценки степени заряженности литий-ионного аккумулятора MLP может рассматриваться как эталонная модель, позволяющая установить уровень точности, относительно которого исследуются более сложные архитектуры. Нейросеть MLP состоит из последовательно соединённых слоёв нейронов, каждый из которых выполняет линейное преобразование входных данных с последующим применением нелинейной функции активации. Отсутствие рекуррентных связей между слоями означает, что модель обрабатывает каждое наблюдение независимо.

Архитектура сети MLP может иметь вид рис. 1.

В данной работе реализована модель MLP со следующими характеристиками, представленными в табл. 1.

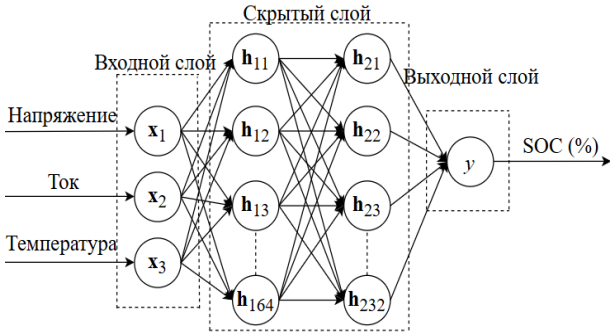

Рис. 1. Архитектура сети MLP

Таблица 1

Характеристики модели MLP

№ п/п	Слой	Кол-во нейронов
1	Входной	3
2	Полносвязный	64
3	BatchNorm	–
4	ReLU	–
5	Dropout (10%)	–
6	Полносвязный	32
7	BatchNorm	–
8	ReLU	–
9	Dropout (10%)	–
10	Выходной	1

Чтобы продемонстрировать пример работы сети MLP, мы обобщили алгоритм обучения, который был рассмотрен в работе [10].

Алгоритм обучения модели MLP

Шаг 1. Подготовка входных данных.

Перед поступлением в модель входные данные нормализуются и формируется вектор **x**.

Шаг 2. Преобразование входного вектора.

Нормализованный вектор **x** поступает на первый скрытый слой сети, где выполняется линейное преобразование с последующей пакетной нормализацией (Batch Normalization, BatchNorm) [11] функцией активации ReLU (Rectified Linear Unit) [12] и операцией Dropout [11]: пакетная нормализация уменьшает смещение распределения входов слоя и стабилизирует процесс обучения, нормализуя значения активаций внутри каждого мини-батча (небольшая группа обучающих примеров, обрабатываемых одновременно); функция активации ReLU вводит нелинейность и пропускает только положительные значения, что позволяет эффективно обучать глубокие сети и ускоряет сходимость по сравнению с классическими сигмоидальными функциями; операция Dropout представляет собой коммутатор, отключающий случайным образом часть нейронов на этапе обучения. Это снижает зависимость модели от отдельных признаков и уменьшает риск переобучения.

$$\mathbf{h}_1 = \mathrm{Droput}(\mathrm{ReLU}(\mathrm{BatchNorm}(\mathbf{W}_1 x + \mathbf{b}_1))), \quad (4)$$

где $\mathbf{W}_1$ – матрица весов первого скрытого слоя; $\mathbf{b}_1$ – вектор смещений первого скрытого слоя.

Для последующих скрытых слоёв выполняется аналогичная операция:

$$\mathbf{h}_2 = \mathrm{Dropout}(\mathrm{ReLU}(\mathrm{BatchNorm}(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2))), \quad (5)$$

где $\mathbf{W}_2$ – матрица весов второго скрытого слоя; $\mathbf{b}_2$ – вектор смещений второго скрытого слоя.

Шаг 3. Формирование выходного слоя.

Результат последнего скрытого слоя модели передаётся на выходной слой, в котором также применяет линейное преобразование

$$\hat{y} = \mathbf{W}_{\text{вых}}\mathbf{h}_2 + \mathbf{b}_{\text{вых}}, \quad (6)$$

где $\mathbf{h}_2$ – выходной вектор скрытого слоя; $\mathbf{W}_{\text{вых}}$ – матрица весов выходного слоя; $\mathbf{b}_{\text{вых}}$ – вектор смещений выходного слоя.

Шаг 4. Вычисление квадратичной функции L потерь (Mean Squared Error, MSE).

Сравнивается предсказанное значение $\hat{y}$ с истинным значением y:

$$L = \mathrm{MSE}(y, y) = \frac{1}{n}\sum_{i=1}^{n}(y - y)^2, \quad (7)$$

где n – количество данных (размер выборки).

Шаг 5. Вычисление обратного распространения ошибки.

Градиенты функции потерь по отношению ко всем параметрам модели вычисляются с использованием метода обратного распространения ошибки.

Шаг 6. Обновление весов.

При обучении параметры модели, такие как веса и смещения, обновляются.

$$\mathbf{W} = \mathbf{W} - \alpha\frac{\partial L}{\partial \mathbf{W}}, \quad (8)$$

где α – коэффициент скорости обучения; $\partial L / \partial \mathbf{W}$ – градиент функции потерь относительно весов.

Шаг 7. Организация цикла.

Шаги 2–6 повторяются в процессе обучения на протяжении нескольких эпох. Обучение продолжается до момента выполнения одного из условий остановки: превышения установленного количества эпох или достижения стабилизации функции потерь.

Рекуррентные нейронные сети

Нейросети типа RNN используются в задачах обработки временных последовательностей, поскольку они способны сохранять состояние от одной итерации к другой, что делает их эффективным инструментом для прогнозирования параметров литий-ионных аккумуляторов [13]. В отличие от нейросетей прямого распространения, где входные данные обрабатываются последовательно без учета предыдущих состояний, нейросети RNN могут учитывать динамику процессов, таких как SOC(t) аккумулятора, а также деградацию его характеристик [14].

В базовой архитектуре RNN входные данные поступают в рекуррентный нейронный слой, где они объединяются со значениями предыдущего шага. Вычислительный результат передается на следующий слой [9]. Как показано на рис. 2, рекуррентный блок имеет петлю обратной связи.

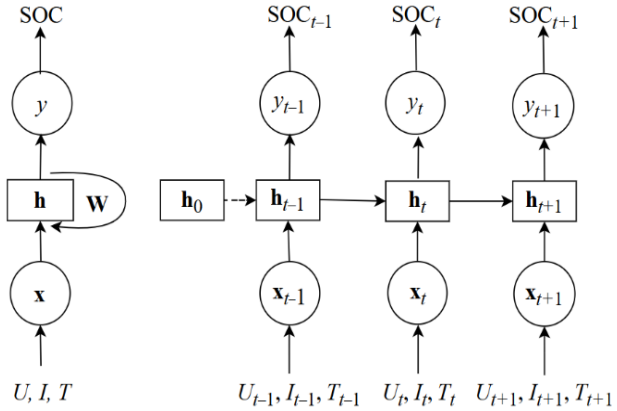

Рис. 2. Рекуррентный блок

В данном исследовании представлена реализация нейросетевой RNN-модели с заданными параметрами, показаны в табл. 2.

Таблица 2

Характеристики модели RNN

№ п/п	Слой	Кол-во нейронов
1	Входной	3
2	RNN	16
3	Полносвязный	16
4	Tanh	–
5	Выходной	1

Алгоритм обучения модели на нейросети RNN

Шаг 1. Подготовка входных данных.

Входные данные нормализуются для работы с RNN. Данные формируются в виде последовательностей длиной 20 временных интервалов, каждая из которых содержит значения напряжения, тока и температуры аккумулятора.

Шаг 2. Инициализация скрытого состояния.

Перед началом обработки последовательности скрытое состояние $\mathbf{h}_0$ инициализируется начальным значением (часто нулевым вектором).

$$\mathbf{h}_0 = 0. \qquad (9)$$

Шаг 3. Вычисление нового скрытого состояния.

На каждом интервале времени t нейросеть получает входные данные $\mathbf{x}_t$, предыдущие данные скрытого состояния $\mathbf{h}_{t-1}$ и использует их для вычисления нового скрытого состояния $\mathbf{h}_t$ с помощью функции активации tanh [15]:

$$\mathbf{h}_t = \tanh(\mathbf{W}_x\mathbf{x}_t + \mathbf{W}_h\mathbf{h}_{t-1} + \mathbf{b}), \qquad (10)$$

где $\mathbf{W}_x$ – матрица параметров для входных данных; $\mathbf{W}_h$ – матрица параметров для данных скрытого состояния; $\mathbf{b}$ – вектор параметров смещения.

Функция активации tanh (Hyperbolic tangent) преобразует входные значения в диапазон от –1 до 1 и обладает свойством сглаживания, что делает её удобной для обработки последовательных данных, содержащих шумы и помехи [12].

Шаг 4. Вычисление выходных данных.

После вычисления нового скрытого состояния $\mathbf{h}_t$ определяются значения выхода y_t в момент времени t [15]:

$$y_t = \mathbf{W}_0\mathbf{h}_t + \mathbf{b}_0, \qquad (11)$$

где $\mathbf{W}_0$ – матрица параметров выходных данных; $\mathbf{b}_0$ – вектор параметров смещения.

Шаг 5. Организация цикла.

Шаги 3 и 4 повторяются до тех пор, пока нейросеть RNN не пройдет всю последовательность входных данных.

Управляемые рекуррентные блоки

Наиболее мощная архитектура RNN – это управляемые рекуррентные блоки. В ее структуре предыдущее скрытое состояние и текущий вход не оказывают прямого воздействия на формирование текущего скрытого состояния, как это происходит в классической RNN. Вместо этого механизм обновления контролирует, какую часть информации из прошлого следует сохранить, а механизм сброса определяет объем данных, которые необходимо игнорировать.

На рис. 3 представлена общая структура GRU [9], где функция активации является линейной комбинацией предыдущего состояния и состояния-кандидата; σ – слой с сигмоидальной функцией активации; $(1-)$ – поэлементное вычитание из 1 значений вектора.

Используемая нейронная сеть GRU доказала свою эффективность для задачи оценки состояния литий-ионных аккумуляторов в работе [16]. В данной статье представлена модель GRU с заданными характеристиками, которые сведены в табл. 3.

Алгоритм обучения модели на нейросети GRU

Шаг 1. Подготовка входных данных.

Входные данные нормализуются и формируются в виде последовательностей длиной 20 временных интервалов.

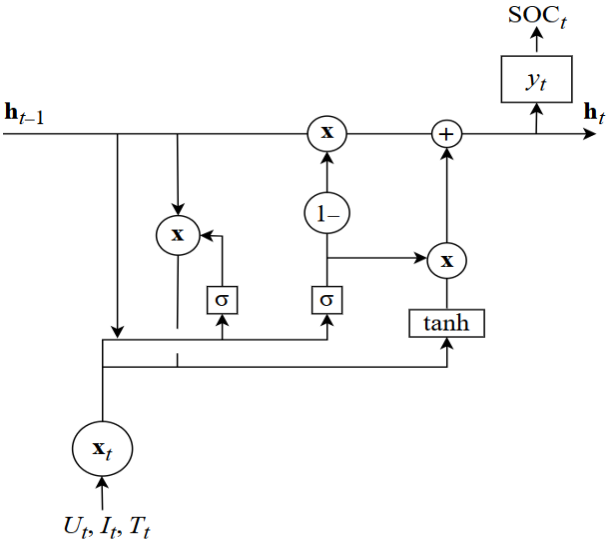

Рис. 3. Общая структура GRU

Таблица 3

Характеристики модели GRU

№ п/п	Слой	Кол-во нейронов
1	Входной	3
2	GRU	16
3	Полносвязный	16
4	Tanh	–
5	Выходной	1

Шаг 2. Инициализация скрытого состояния.

Перед началом обработки последовательности данных скрытое состояние $\mathbf{h}_0$ инициализируется начальным значением (часто нулевым вектором).

Шаг 3. Вычисление новых состояний.

На каждом интервале времени t нейросеть GRU получает входные данные и данные предыдущего скрытого состояния и использует их для вычисления нового скрытого состояния $\mathbf{h}_t$ по следующим формулам [17]:

$$\mathbf{z}_t = \text{sigmoid}(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1} + \mathbf{b}_z), \quad (12)$$

$$\mathbf{r}_t = \text{sigmoid}(\mathbf{W}_r\mathbf{x}_t + \mathbf{U}_r\mathbf{h}_{t-1} + \mathbf{b}_r), \quad (13)$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_h\mathbf{x}_t + \mathbf{U}_h(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h), \quad (14)$$

$$\mathbf{h}_t = (1-\mathbf{z}_t)\odot\mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t, \quad (15)$$

где $\mathbf{x}_t$ – входные данные на интервале времени t; $\mathbf{h}_{t-1}$ – данные предыдущего скрытого состояния; $\mathbf{z}_t$ – вектор вентиля обновления; $\mathbf{r}_t$ – вектор вентиля сброса; $\mathbf{W}$, $\mathbf{U}$ – матрицы параметров; $\mathbf{b}$ – вектор параметров смещения; $\tilde{\boldsymbol{h}}_t$ – кандидат на скрытое состояние; $\odot$ – символ операции поэлементного произведения.

Шаг 4. Вычисление выходных данных.

После определения нового скрытого состояния $\mathbf{h}_t$, GRU использует его для вычисления выхода $\mathbf{y}_t$ в момент времени t согласно уравнению (11).

Шаг 5. Организация цикла.

Шаги 3 и 4 повторяются до тех пор, пока GRU полностью не обработает всю последовательность входных данных.

Архитектура GRU обладает улучшенной способностью к моделированию временных зависимостей по сравнению с классической RNN благодаря встроенным механизмам управления – обновляющему и сбрасывающему вентилям (см. рис. 3).

Вычислительные эксперименты на реальных данных

В статье используется набор данных из репозитория Центра передовой инженерии жизненного цикла университета Мэриленда (Center for Advanced Life Cycle Engineering, CALCE) [18]. Характеристики аккумуляторов сведены в табл. 4.

Обучение нейронной сети можно сделать более эффективным посредством правильной нормализации данных. Поэтому перед обучением входные данные нормализуются [19].

$$x_{\text{норм}} = \frac{x-\mu}{\sigma}, \quad (16)$$

где $x_{\text{норм}}$ – нормализованные данные; x – исходные данные; μ – среднее значение в обучающем наборе данных; σ – стандартное отклонение в обучающем наборе данных.

После нормализации данные будут иметь среднее значение, равное 0, и стандартное отклонение, равное 1.

В процессе обучения модели используются различные подходы к подаче входных данных. В эксперименте входными параметрами являются напряжение, ток и температура LIB, а выходным значением – степень заряженности.

В рамках данной работы фактические значения SOC рассчитывались исключительно на основе экспериментальной разрядной ёмкости аккумулятора. Для каждого момента времени SOC вычисляется по формуле (1). Используемые в работе фактические значения SOC являются расчётными, но основанными на реальных экспериментальных измерениях.

В процессе оценки производительности трёх моделей – MLP, RNN и GRU – мы основывались на таких критериях, как средняя абсолютная ошибка (Mean Absolute Error, MAE), время обучения и время тестирования (табл. 5).

Таблица 4

Технические характеристики аккумуляторов

Тип	A123-18650
Катод	LiFePO4
Анод	Графит
Номинальная емкость	1,1 А.ч
Номинальное напряжение	3,3 В
Напряжение отключения заряда	3,6 В
Напряжение отключения разряда	2,0 В

Таблица 5

Результаты оценки SOC с использованием нейронных сетей

Тип сети	MAE, %	Время обучения, с	Время тестирования, с
MLP	2,2	160,6	0,6
RNN	1,5	286,9	2,1
GRU	1,1	296,1	2,1

Как видно из табл. 5, модель GRU демонстрирует наилучшую точность, что указывает на мини-

мальное среднее отклонение (1,1%) предсказаний от истинных значений SOC. Модель RNN следует за ней с MAE = 1,5%, показывая близкую производительность, в то время как MLP отстает со средней погрешностью 2,5%. С другой стороны, сеть MLP требует наименьшего времени обучения и тестирования – 160,6 с и 0,6 с соответственно. Сеть GRU требует наибольшего времени обучения – 296,1 с, а сеть RNN – 286,9 с. Время тестирования обеих сетей также больше – 2,1 с.

На рис. 4 и 5 показаны сравнения результатов прогнозов моделей MLP, RNN и GRU на тестовой выборке, позволяющие увидеть различия в точности прогнозирования.

При выборе архитектуры нейронной сети необходимо учитывать соотношение между точностью и затратами на вычисления. Нейросеть MLP показывает хорошую обобщающую способность при минимальных затратах времени. Нейросети RNN и GRU лучше справляются с задачами, где важен временной контекст, но их устойчивость к переобучению и вычислительная эффективность требуют дополнительной настройки. Следует также отметить, что эффективность прогнозирования во многом зависит от используемых наборов данных. Разные режимы нагрузки и условия эксплуатации могут существенно повлиять на результаты. Тем не менее проведённые эксперименты показывают, что все три архитектуры способны обеспечивать приемлемое качество прогнозирования SOC, что подтверждает их практическую применимость.

С практической точки зрения разработанные нейросетевые модели ориентированы на использование в составе промышленных вычислительных систем, где возможно внедрение программного обеспечения на языке Python. Это делает возможным их применение для мониторинга состояния аккумуляторов на объектах энергетики, транспорта.

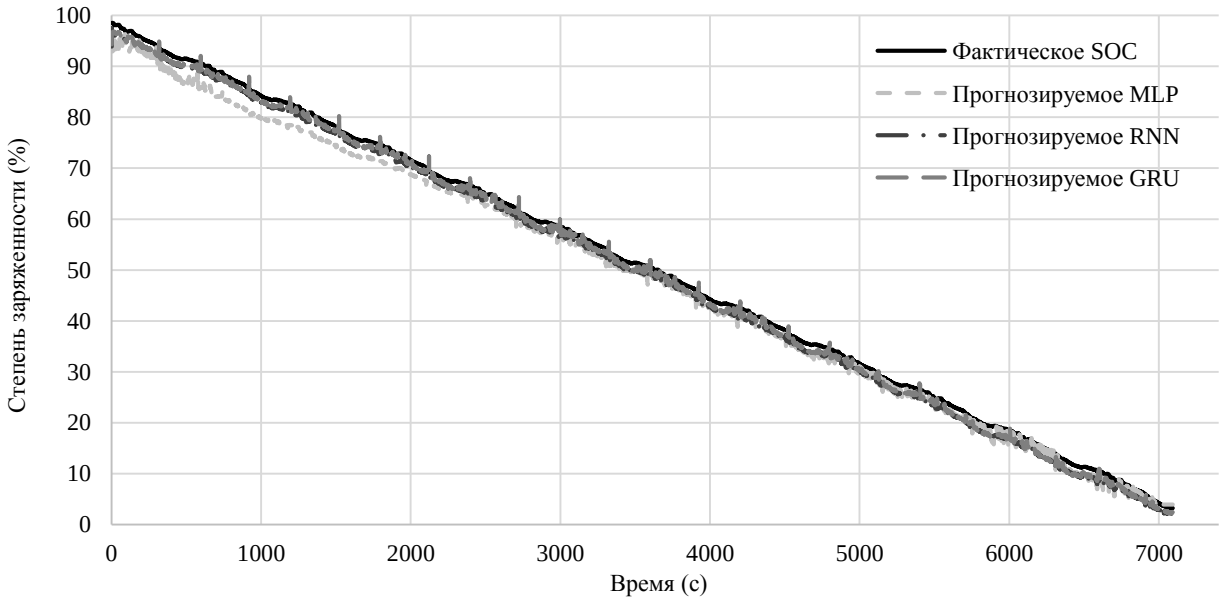

Рис. 4. Сравнение результатов прогнозов моделей MLP, RNN и GRU на тестовой выборке

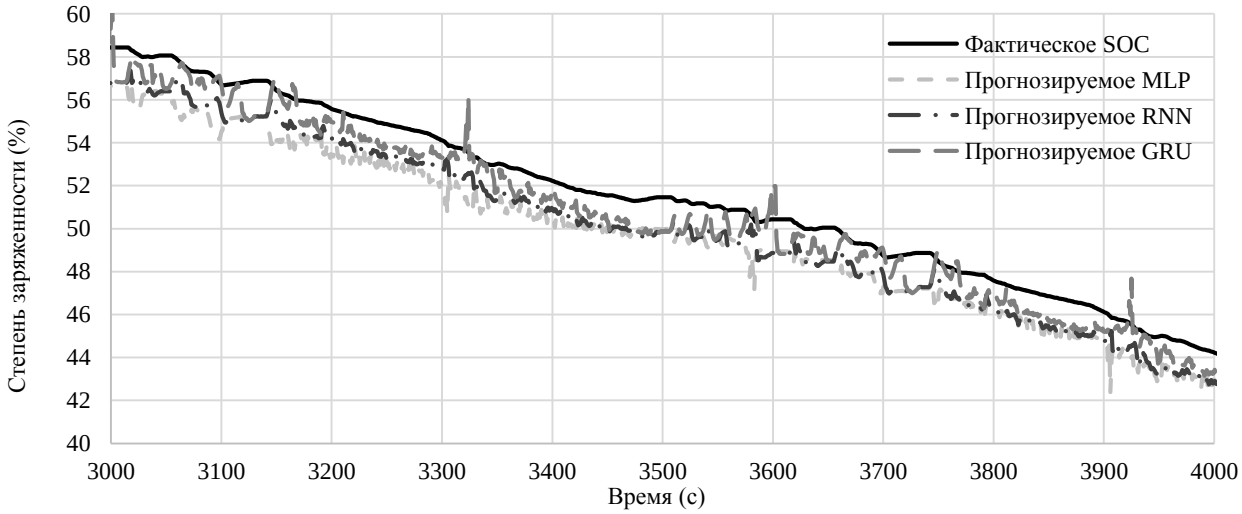

Рис. 5. Увеличенный фрагмент для детального анализа различий между моделями

Таким образом, предлагаемое решение может быть не только исследовательским инструментом, но и технологической основой для создания интеллектуальных систем контроля и управления аккумуляторными батареями в условиях реальной эксплуатации.

Заключение

В настоящем исследовании были проанализированы три архитектуры нейронных сетей – MLP, RNN и GRU – с целью оценки степени заряженности литий-ионного аккумулятора на основе временных данных тока, напряжения и температуры. Проведённое сравнение показало, что каждая из моделей имеет свои особенности: MLP обеспечивает высокую скорость обучения, тогда как RNN и GRU лучше учитывают временную структуру данных и демонстрируют более высокую точность при прогнозировании.

Полученные результаты подтверждают практическую применимость нейросетевых методов для оценки SOC в реальных энергетических системах. Обученные модели могут быть интегрированы в системы управления аккумуляторами электротранспорта, автономных робототехнических устройств, систем накопления электрической энергии, резервных источников питания и интеллектуальных систем мониторинга. Такой прогноз позволяет заблаговременно оценивать доступную энергию, оптимизировать алгоритмы заряда и разряда, уменьшать деградацию аккумулятора и повышать общую безопасность эксплуатации.

С точки зрения аппаратной реализации, такие модели могут быть развёрнуты на промышленных компьютерах или серверных системах в зависимости от требуемой точности и скорости обработки. При необходимости модели могут быть дополнительно оптимизированы (квантование, упрощение архитектуры) для выполнения в реальном времени на ресурсоограниченных контроллерах. Например, модели могут быть реализованы на процессорах типа Intel core i7-10850H, ОЗУ 32Гб.

В дальнейшем планируется расширение экспериментальной базы, исследование влияния условий эксплуатации на точность прогнозирования, а также реализация полной интеграции обученных моделей в BMS с последующим тестированием в реальных эксплуатационных условиях.

Литература

1. Tian N. On Parameter Identification of an Equivalent Circuit Model for Lithium-Ion Batteries / N. Tian, Y. Wang, J. Chen and H. Fang // 2017 Conference on Control Technology and Applications (CCTA). – Hawai'i, USA, 2017. – P. 187–192.

2. Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries / K.S. Ng, C.-S. Moo, Y.-P. Chen, Y.-C. Hsieh // Appl Energy. – 2009. – Vol. 86. – P. 1506–1511.

3. Analysis of equivalent circuit models in lithium-ion batteries / H.W. You, J.I. Bae, S.J. Cho, J.M. Lee, S.-H. Kim // AIP Advances. – 2018. – Vol. 8, No. 12. – P. 125101.

4. Plett G.L. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs. Part 1: Background // Journal of Power Sources. – 2004. – Vol. 134. – P. 252–261.

5. Lee J. Li-ion battery SOC estimation method based on the reduced order extended Kalman filtering / J. Lee, O. Nam, B.H. Cho // Journal of Power Sources. – 2007. –Vol. 174. – P. 9–15.

6. Plett G.L. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs. Part 2: Modeling and identification // Journal of Power Sources. – 2004. – Vol. 134. – P. 262–276.

7. He W. State of charge estimation for Li-ion batteries using neural network modeling and unscented Kalman filter-based error cancellation / W. He, N. Williard, C. Chen, M. Pecht // Electrical Power and Energy Systems. – 2014. – Vol. 62. – P. 783–791.

8. Казаков Ф.А. Использование нейронных сетей с временными рядами данных для анализа потоков данных / Ф.А. Казаков, А.В. Шнайдер // Современные наукоемкие технологии. – 2021. – Т. 6, № 2. – С. 260–264.

9. Феофилов С.В. Применение рекуррентных нейронных сетей в замкнутых следящих системах с нелинейными объектами / С.В. Феофилов, Д.Л. Хапкин // Известия ТулГУ. Технические науки. – 2021. – Вып. 11. – С. 72–80.

10. Хайкин С. Нейронные сети: полный курс. – 2-е изд.; пер. с англ. С. Хайкин. – М.: Вильямс, 2006. – 1104 с.

11. Garbin C. Dropout vs. Batch Normalization: An Empirical Study of Their Impact to Deep Learning / C. Garbin, X. Zhu, O. Marques // Multimedia Tools and Applications. – 2020. – Vol. 79. – P. 12777–12815.

12. Jagtap A.D. How important are activation functions in regression and classification? A survey, performance comparison, and future directions / A.D. Jagtap, G.E. Karniadakis // Journal of Machine Learning for Modeling and Computing. – 2023. – Vol. 4, No. 1. – P. 21–75.

13. Chaoui H. State of Charge and State of Health Estimation for Lithium Batteries Using Recurrent Neural Networks / H. Chaoui, C.C. Ibe-Ekeocha // IEEE Trans. Veh. Technol. – 2017. – Vol. 66. – P. 8773–8783.

14. Новиков А.А. Анализ методов прогнозирования временных рядов / А.А. Новиков, Е.О. Будзинская, О.Н. Канева // Междунар. молодежная науч.-практ. конф. с элементами научной школы. – Омск: ОмГТУ, 2020. – Т. 4, № 1. – С. 37–43.

15. Перепелкин В.Ю. Использование рекуррентных нейросетей сетей для прогнозирования временных рядов // Современная наука: актуальные проблемы теории и практики. Сер.: Естественные и технические науки. – 2023. – № 07. – С. 80–82.

16. Yang F. State-of-charge estimation of lithium-ion batteries based on gated recurrent neural network / F. Yang, W. Li, C. Li, Q. Miao // Energy. – 2019. – Vol. 175. – P. 66–75.

17. Пашшоев Б. Анализ нейросетевых моделей для прогнозирования временных рядов / Б. Пашшоев, Д.А. Петрусевич // Russian Technological Journal. – 2024. – Т. 12, No. 4. – P. 106–116.

18. Pecht M. Battery Data Set. CALCE, CALCE Battery Research Group, Maryland, MD, 2017 [Электронный ресурс]. – URL: https://web.calce.umd.edu/batteries/data.htm (дата обращения: 20.05.2025).

19. Старовойтов В.В. Нормализация данных в машинном обучении / В.В. Старовойтов, Ю.И. Голуб // Информатика. – 2021. − Т. 18, № 3. – С. 83–96.

Букреев Виктор Григорьевич
Д-р техн. наук, проф. инженерной школы энергетики (ИШЭ) Национального исследовательского Томского политехнического университета (НИ ТПУ)
Ленина пр-т, 30, г. Томск, Россия, 634050
ORCID: 0000-0001-9861-9765
Тел.: +7-913-854-21-96
Эл. почта: bukreev@tpu.ru

Хоанг Фу Нам
Аспирант ИШЭ НИ ТПУ
Ленина пр-т, 30, г. Томск, Россия, 634050
ORCID: 0009-0008-2554-609X
Тел.: +7-999-618-81-37
Эл. почта: namhp0108@gmail.com

Поступила в редакцию: 04.09.2025.
Принята к публикации: 19.01.2026.

Bukreev V.G., Hoang P.N.

Neural network-based prediction of the state of charge of Lithium-ion battery

The problem of predicting the state of charge (SOC) of lithium-ion batteries (LIB) for energy storage systems of technological equipment and mobile objects is considered. A neural network-based approach is proposed to overcome the limitations of traditional mathematical models that require prior knowledge of battery parameters. Three types of architectures are examined-multilayer perceptrons (MLP), recurrent neural networks (RNN), and gated recurrent units (GRU) – to analyze their effectiveness in SOC prediction. A methodology is developed for training these neural network models on experimental data, including time series of voltage, current and temperature under dynamic discharge conditions. Differences in accuracy and computational costs are confirmed: MLP networks train faster but fail to capture temporal dependencies, while RNN and GRU networks demonstrate lower test errors but demand higher computational resources. The practical applicability of the results is validated through the optimization of SOC estimation algorithms under real-world conditions.

Keywords: Lithium-ion battery, state of charge, neural networks, neural network architectures: MLP, RNN, GRU.
DOI: 10.21293/1818-0442-2025-28-4-155-161

References

1. Tian N., Wang Y., Chen J., Fang H. On Parameter Identification of an Equivalent Circuit Model for Lithium-Ion Batteries. *2017 Conference on Control Technology and Applications (CCTA)*, Hawai'i, USA, 2017, pp. 187–192.

2. Ng K.S., Moo C.-S., Chen Y.-P., Hsieh Y.-C. Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. *Appl Energy*, 2009, vol. 86, pp. 1506–1511.

3. You H.W., Bae J.I., Cho S.J., Lee J.M., Kim S.-H. Analysis of equivalent circuit models in lithium-ion batteries. *AIP Advances*, 2018, vol. 8, no. 12, pp. 125101.

4. Plett G.L. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 1: Background. *Journal of Power Sources*, 2004, vol. 134, pp. 252–261.

5. Lee J., Nam O., Cho B.H. Li-ion battery SOC estimation method based on the reduced order extended Kalman filtering. *Journal of Power Sources*, 2007, vol. 174, pp. 9–15.

6. Plett G.L. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 2: Modeling and identification. *Journal of Power Sources*, 2004, vol. 134, pp. 262–276.

7. He W., Williard N., Chen C., Pecht M. State of charge estimation for Li-ion batteries using neural network modeling and unscented Kalman filter-based error cancellation. *Electrical Power and Energy Systems*, 2014, vol. 62, pp. 783–791.

8. Kazakov F.A., Shnaider A.V. [Using neural networks with time series of data to analyze data flows]. *Modern High Technologies*, 2021, no. 6, pp. 260–264 (in Russ.).

9. Feofilov S.V., Khapkin D.L. [Application of recurrent neural networks in closed tracking systems with nonlinear objects]. *Izvestiya Tula State University*, 2021, vol. 11, pp. 72–80 (in Russ.).

10. Haykin S. Neironnye seti: polnyi kurs, 2-e izdanie [Neural Networks: A Comprehensive Foundation, 2nd ed], translated from English. Moscow: Williams Publishing House, 2006, 1104 p. (in Russ.).

11. Garbin C., X. Zhu, O. Marques. Dropout vs. Batch Normalization: An Empirical Study of Their Impact to Deep Learning. *Multimedia Tools and Applications*, 2020, vol. 79, pp. 12777–12815.

12. Jagtap A.D., Karniadakis G.E. How important are activation functions in regression and classification? A survey, performance comparison, and future directions. *Journal of Machine Learning for Modeling and Computing*, 2023, vol. 4, no. 1, pp. 21–75.

13. Chaoui H., Ibe-Ekeocha C.C. State of Charge and State of Health Estimation for Lithium Batteries Using Recurrent Neural Networks. *IEEE Trans. Veh. Technol*, 2017, vol. 66, pp. 8773–8783.

14. Novikov A.A., Budzinskaya E.O., Kaneva O.N. Analiz metodov prognozirovaniya vremennykh ryadov [Analysis of Time Series Forecasting Methods]. *International Youth Scientific and Practical Conference with Elements of Scientific School*, Omsk, OmSTU, 2020, vol. 4, no. 1, pp. 37–43 (in Russ.).

15. Perepelkin V. [Using recurrent neural networks for time series forecasting]. *Modern Science: actual problems of theory and practice. Series: Natural and Technical Sciences*, 2023, no. 7, pp. 80–82 (in Russ.).

16. Yang F. State-of-charge estimation of lithium-ion batteries based on gated recurrent neural network. *Energy*, 2019, vol. 175, pp. 66–75.

17. Pashshoev B., Petrusevich D.A. Neural network analysis in time series forecasting. *Russ. Technol. J,* 2024, vol 12, no. 4, pp. 106−116.

18. Pecht M. Battery Data Set. CALCE, CALCE Battery Research Group, Maryland, MD; 2017. Available at: https://web.calce.umd.edu/batteries/data.htm.

19. Starovoitov V.V., Golub Y.I. [Data normalization in machine learning]. *Informatics*, 2021, vol. 18, no. 3, pp. 83−96 (in Russ.).

Viktor G. Bukreev
Doctor of Science in Engineering, Professor,
School of Energy & Power Engineering,
National Research Tomsk Polytechnic University (NR TPU)
30, Lenin pr., Tomsk, Russia, 634050
ORCID: 0000-0001-9861-9765
Phone: +7-913-854-21-96
Email: bukreev@tpu.ru

Hoang Phu Nam
Postgraduate student, School of Energy & Power Engineering, NR TPU
30, Lenin pr., Tomsk, Russia, 634050
ORCID: 0009-0008-2554-609X
Phone: +7-999-618-81-37
Email: namhp0108@gmail.com

Received: 04.09.2025.
Accepted: 19.01.2026.