

**ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ МОЛДОВЫ
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ
ДЕПАРТАМЕНТ ИНФОРМАТИКИ**

БАБУШКИН Владимир Ваганович.

Магистерский тезис по теме:

**Разработка алгоритмов машинного обучения для
предсказания цены и нагрузки на электросети**

**Заведующий Департаментом
Информатики:** _____

**Валерий УНГУРЕАНУ,
доцент, доктор физико-
математических наук**

Научный руководитель: _____

**Георге КЭПЭЦЫНЭ,
профессор,
доктор инженер**

Автор: _____

**Владимир БАБУШКИН,
группа TPS1801**

Кишинев 2020

СОДЕРЖАНИЕ

АННОТАЦИЯ	3
СПИСОК СОКРАЩЕНИЙ.....	5
ВВЕДЕНИЕ.....	7
ВАЖНОСТЬ ИССЛЕДОВАНИЯ	7
ЗАДАЧИ И ОБЪЕМ ДИССЕРТАЦИИ.....	8
МЕТОДЫ ИССЛЕДОВАНИЯ.....	9
НОВИЗНА ИССЛЕДОВАНИЯ.....	9
КЛЮЧЕВЫЕ СЛОВА.....	10
ГЛАВА 1. ОБЗОР ЛИТЕРАТУРЫ.....	11
ГЛАВА 2. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ.....	24
ПОСТАНОВКА ЗАДАЧИ	24
ЭКСПЕРТНЫЕ АЛГОРИТМЫ	25
k -средние	25
k -медоиды.....	26
Агломеративная кластеризация.....	27
Нечеткая кластеризация С-средних	28
Самоорганизующиеся карты (<i>Self-Organizing Map (SOM)</i>)	29
АЛГОРИТМ ПРОГНОЗИРОВАНИЯ НА ОСНОВЕ ПОСЛЕДОВАТЕЛЬНОСТИ ПАТТЕРНОВ	33
СЕТИ ГЛУБОКОГО ОБУЧЕНИЯ ДЛЯ ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ.....	34
Сверточные Нейронные Сети (<i>Convolutional Neural Networks (CNNs)</i>)	39
Сети долговременной кратковременной памяти (<i>Long-Short Term Memory Networks (LSTM)</i>).....	40
ПОКАЗАТЕЛИ ЭФФЕКТИВНОСТИ	44
Средняя абсолютная процентная погрешность (<i>Mean Absolute Percentage Error (MAPE)</i>)	44
Средняя относительная погрешность (<i>Mean Relative Error (MRE)</i>).....	44
Средняя абсолютная погрешность (<i>Mean Absolute Error (MAE)</i>).....	45
Среднеквадратическая погрешность (<i>Root Mean Square Error (RMSE)</i>)	45
ПРОГНОЗИРОВАНИЕ ЭЛЕКТРИЧЕСКОЙ НАГРУЗКИ В СЕТЯХ LSTM И CNN	45
ГЛУБОКАЯ CNN ДЛЯ ПРОГНОЗИРОВАНИЯ ЦЕН НА ЭЛЕКТОЭНЕРГИЮ	47
ГЛАВА 3. РЕЗУЛЬТАТЫ	50
ПОИСК ОПТИМАЛЬНЫХ ПАРАМЕТРОВ	50
АНАЛИЗ МОДЕЛЕЙ КЛАСТЕРИЗАЦИИ	53
ОЦЕНКА ЭФФЕКТИВНОСТИ АЛГОРИТМОВ ПРОГНОЗИРОВАНИЯ НАГРУЗКИ.....	55
ПРОГНОЗИРОВАНИЕ ЦЕНЫ НА ЭЛЕКТРОЭНЕРГИЮ ИЗ ДАННЫХ О НАГРУЗКЕ.....	60
ВЫВОДЫ И РЕКОМЕНДАЦИИ	63
БИБЛИОГРАФИЯ.....	65

АННОТАЦИЯ

В настоящее время рынок электроэнергии становится все более разнообразным и дерегулированным в связи с постепенным внедрением интеллектуальных сетей. Интеллектуальная сеть анализирует информацию о спросе и предложении электроэнергии для оптимизации ее производства. Для этого требуется знание будущей электрической нагрузки, позволяющее независимым производителям (*Independent Power Producers (IPP)*) планировать выработку электроэнергии для удовлетворения пикового спроса и минимизации производственных затрат. IPP также имеют возможность внедрять динамическое ценообразование, чтобы стимулировать конечных потребителей переключать потребление электроэнергии на внепиковые часы. Иными словами, потребители, зная будущие цены на электроэнергию, могут корректировать свой профиль потребления для достижения оптимального использования электроэнергии при наименьших затратах. Другие субъекты рынка электроэнергии, например розничные торговцы и трейдеры, также заинтересованы в прогнозировании цен на электроэнергию для оптимизации своих финансовых операций. Таким образом, точное прогнозирование цены и нагрузки имеет важное значение для поддержания стабильного взаимодействия спроса и предложения на динамичном рынке электроэнергии.

Цель данной работы - исследование нелинейных методов прогнозирования временных рядов и их сравнение с современными методами глубокого обучения. Сети глубокого обучения способны выявлять скрытые временные зависимости и тенденции, а также распознавать внутренние факторы, влияющие на динамичность временных рядов электрической нагрузки и цен. Мы рассматриваем применение сетей долговременной кратковременной памяти (LSTM) и *сверточных нейронных сетей* (CNN) для прогнозирования электрической нагрузки и цены на электроэнергию в Нью-Йорке.

Мы также предлагаем новый подход глубокого обучения для прогнозирования цен на электроэнергию на основе исторических данных о ценах / нагрузке и прогнозируемого значения нагрузки на день вперед. Предлагаемая система может быть реализована в настройках интеллектуальных сетей для онлайн и оффлайн прогнозирования цен на электроэнергию для горизонтов прогнозирования различной протяженности, начиная от одного дня и заканчивая неделями и месяцами.

ADNOTARE

În prezent, piața de energie electrică devine din ce în ce mai diversă și de reglată datorită introducerii treptate a rețelelor inteligente. O rețea inteligentă analizează informațiile privind cererea și oferta de energie electrică pentru a-și optimiza producția. Acest lucru necesită cunoașterea încărcării electrice viitoare, permitând producătorilor de energie independenti (IPP) să-și planifice generarea de energie pentru a răspunde cererii maxime și a minimiza costurile de producție. IPP-urile au, de asemenea, capacitatea de a implementa prețuri dinamice pentru a încuraja utilizatorii finali să schimbe consumul de energie în orele de vârf. Cu alte cuvinte, consumatorii, cunoscând prețurile viitoare ale energiei electrice, își pot regla profilul de consum pentru a obține o utilizare optimă a energiei electrice la cel mai mic cost. Alți actori de pe piața energiei electrice, cum ar fi comercianții cu amănuntul și comercianții, sunt de asemenea interesați să prevadă prețurile energiei electrice pentru a-și optimiza tranzacțiile financiare. Prin urmare, previziunea exactă a prețurilor și încărcărilor este importantă pentru a menține o interacțiune stabilă între ofertă și cerere pe o piață dinamică a energiei electrice.

Scopul acestei lucrări este studierea metodelor neliniare pentru predicția seriilor de timp și compararea acestora cu metodele moderne de învățare profundă. Rețelele de învățare profundă sunt capabile să identifice dependențele și tendințele de timp ascunse, precum și să recunoască factorii interni care afectează dinamica seriei de timp a sarcinii electrice și a prețurilor. Avem în vedere utilizarea rețelelor de memorie pe termen scurt (LSTM) și a *rețelelor neuronale convoluționale* (CNN) pentru a prezice prețurile de încărcare electrică și electricitate în New York.

Oferim, de asemenea, o nouă abordare de învățare profundă pentru prognozarea prețurilor la energia electrică bazată pe datele istorice ale prețurilor / încărcării și a valorii preconizate a încărcării pentru ziua următoare. Sistemul propus poate fi implementat în setările rețelelor inteligente pentru prognozarea online și offline a prețurilor la energia electrică pentru prognozarea orizonturilor de diferite lungimi, de la o zi la săptămâni și luni.

СПИСОК СОКРАЩЕНИЙ

IPP (*Independent Power Producers*) – независимые производители электроэнергии
LSTM (*Long-Short Term Memory*) – сеть долговременной краткосрочной памяти
CNN (*Convolutional Neural Network*) – сверточная нейронная сеть
PSF (*Pattern Sequence-based Forecasting algorithm*) – алгоритм прогнозирования последовательности паттернов
MAE (*Mean Absolute Error*) – средняя абсолютная погрешность
MAPE (*Mean Absolute Percentage Error*) – средняя абсолютная процентная погрешность
MRE (*Mean Relative Error*) – средняя относительная погрешность
RMSE (*Root Mean Square Error*) – среднеквадратичная погрешность
API (*Application Programming Interface*) – интерфейс программирования приложений
CNTK (*Microsoft Cognitive Toolkit*) – когнитивный инструментарий Майкрософт
IQR (*Interquartile Range*) – межквартильный диапазон
TSF (*Time Series Forecasting*) – прогнозирование временных рядов
SOM (*Self- Organizing Maps*) – самоорганизующиеся карты
VSTLF (*Very Short-Term Load Forecasting*) – прогнозирование очень краткосрочных нагрузок
STLF (*Short-Term Load Forecasting*) – краткосрочное прогнозирование нагрузки
MTLF and LTLF (*Medium-Term and Long-Term Load Forecasting*) – среднесрочное и долгосрочное прогнозирование нагрузки
OLS (*Ordinary Least Squares*) – обычные наименьшие квадраты
MA (*Moving Average*) – скользящее среднее значение
ARIMA (*Autoregressive Integrated Moving Average*) – интегрированная модель авторегрессии скользящего среднего
LBF (*Label-Based Forecasting*) – алгоритм прогнозирования на основе меток
EV (*Electric Vehicle*) – электромобиль
ANN (*Artificial Neural Networks*) – искусственные нейронные сети
SVR (*Support Vector Regression*) – опорная векторная регрессия
DBN (*Deep Learning Believe Network*) – сеть доверия глубокого обучения
RBM (*Restricted Boltzmann Machine*) – ограниченная машина Больцмана
FCRBM (*Factored Conditional Restricted Boltzmann Machine*) – факторизованное условно-ограничение машины Больцмана

GARCH (*Generalized Autoregressive Conditional Heteroskedasticity*) – обобщенная авторегрессионная условная гетероскедастичность

RBFNN (*Radial Basis Function Neural Network*) – нейронная сеть радиальных базисных функций

BMU (*Best Matching Unit*) – блок наилучшего соответствия

ES (*Exact Sequence*) – точная последовательность

ReLU (*Rectified Linear Unit*) – усеченное линейное преобразование

NYISO (*New York Independent System Operator*) – независимый системный оператор Нью-Йорка

LBMP (*Location Based Marginal Pricing*) – установленная маржа для данной локации

GAN (*Generative Adversarial Networks*) – генеративные состязательные нейросети

ВВЕДЕНИЕ

ВАЖНОСТЬ ИССЛЕДОВАНИЯ

Точное прогнозирование спроса на электрическую энергию имеет решающее значение для производства электроэнергии, поскольку она относится к тем товарам, которые не могут быть складированы и должны потребляться сразу же после ее производства. Важность точного прогнозирования спроса на электроэнергию позволяет не только планировать нагрузку, но и манипулировать ценой электроэнергии для достижения желаемых уровней потребления и максимизации прибыли. Кроме того, знание ожидаемого спроса позволяет более эффективно использовать природные ресурсы для промышленного производства электроэнергии [1].

В рамках традиционного рынка электроэнергии ее потребление было крепко связано с обычной кривой пикового спроса и, таким образом, отличалось высокой предсказуемостью. Производство электроэнергии в те времена было более ресурсоемким и дорогостоящим процессом, поскольку производители игнорировали непосредственную обратную связь для расчета мгновенного потребления. Вместо этого рассчитывалось среднее потребление на более продолжительные периоды. Внедрение интеллектуальных сетей на рынке электроэнергии привело к более эффективному управлению нагрузкой, позволив поставщику устанавливать цену в соответствии с профилем потребления, что сделало производство электроэнергии более рентабельным и экологически чистым. Это в свою очередь облегчило корректировку часовых схем потребления конечных потребителей для минимизации затрат, тем самым снижая электрическую нагрузку в сети в часы пик. Внедрение интеллектуальных сетей также привело к увеличению динамики рынка электроэнергии – конечные потребители стали зависеть от величины цены в определенное время суток. Знание будущих цен важно для трейдеров, ритейлеров и других субъектов рынка, поскольку позволяет им оптимизировать финансовые стратегии. Таким образом, внедрение алгоритмов прогнозирования цен имеет существенное значение в условиях deregулированного рынка.

Электрическая нагрузка зависит от других внешних факторов, таких как сезонные закономерности, возрастающие с падением температуры. Традиционные статистические, линейные и нелинейные модели в большинстве случаев не справляются с высокой волатильностью данных временных рядов цен на электроэнергию.

Искусственные нейронные сети используются для изучения сложных нелинейных зависимостей. Разработанные не так давно рекуррентные нейронные сети, в частности сеть долговременной краткосрочной памяти (Long-Short Term Memory (LSTM)) и ее разновидности, способны извлекать долгосрочные временные зависимости, становясь тем самым более подходящим инструментом для прогнозирования цен на электроэнергию и нагрузки. Несмотря на растущую популярность глубокого обучения, применение его методов для прогнозирования временных рядов все еще находится в зачаточном состоянии и поэтому требует всестороннего и тщательного анализа.

ЗАДАЧИ И ОБЪЕМ ДИССЕРТАЦИИ

Целью данной работы является изучение применения методов глубокого обучения к прогнозированию электрической нагрузки и цен, а также сравнение эффективности разработанных моделей с традиционными методами машинного обучения, в частности с алгоритмами кластеризации для прогнозирования временных рядов. Для достижения этой цели мы предварительно обработали данные об электрической нагрузке и ценах, предоставленные независимым системным оператором Нью-Йорка [2] для городской сети Нью-Йорка (N. Y. C.). Мы фокусируемся на применении традиционных моделей кластеризации на основе алгоритма прогнозирования последовательности паттернов (Pattern Sequence-based Forecasting (PSF)) и двух моделей глубокого обучения для получения прогнозов на один год. Точность предсказаний каждой модели оценивалась в терминах средней абсолютной погрешностью (Mean Absolute Error (MAE)), средней абсолютной процентной погрешностью (Mean Absolute Percentage Error (MAPE)), средней относительной погрешностью (Mean Relative Error (MRE)) и среднеквадратичной погрешностью (Root Mean Square Error (RMSE)). Детальный анализ продемонстрировал более высокую эффективность подхода на основе долговременной краткосрочной памяти (Long-Short Term Memory (LSTM)) для прогнозирования нагрузки по сравнению с другими традиционными моделями кластеризации на основе PSF, рассмотренными в данной работе. Мы также разработали новый подход на основе сверточной нейронной сети (Convolutional Neural Network (CNN)) для прогнозирования цен на электроэнергию на день вперед, который использует в качестве входных данных исторические данные о ценах и нагрузке наряду с прогнозируемой нагрузкой для формирования прогноза цен

на данный интересующий день. Предлагаемая сеть может быть использована как для оперативного, так и для автономного прогнозирования цен на электроэнергию.

МЕТОДЫ ИССЛЕДОВАНИЯ

Набор данных:

Данные об электрической нагрузке и ценах, представленные независимым оператором Нью-Йоркской системы [2] для района Нью-Йорк (N. Y. C.).

Инструменты глубокого обучения:

Keras(*Keras*) [3] — высокоуровневая нейронная сеть (Application Programming Interface (API)), написанная на языке программирования Python и способная работать на *Tensorflow*, *CNTK* или *Theano*. Мы используем эту структуру для разработки последовательных моделей CNN и LSTM.

Инструменты Машинного Обучения:

Scikit-learn [4] — бесплатная библиотека машинного обучения программного обеспечения для Python.

Пакет кластеризации K-методов [5] в рамках пакета scikit-learn-extra.

scikit-learn-extra — модуль Python для машинного обучения, являющийся расширением scikit-learn. Он включает в себя полезные алгоритмы, не удовлетворяющие критериям включения в scikit-learn из-за их новизны или более низкого числа цитирований.

НОВИЗНА ИССЛЕДОВАНИЯ

Новизна этой работы заключается в использовании сверточных нейронных сетей для прогнозирования цен на электроэнергию, основываясь на исторических данных о цене и нагрузке и прогнозируемого значения нагрузки на интересующий нас день. Насколько нам известно, предсказание цен на электроэнергию исходя из электрической нагрузки с помощью CNN еще не было рассмотрено в литературе. В работе [6] была предпринята попытка спрогнозировать цену электроэнергии путем сопоставления исторических данных о ценах и нагрузках. Однако отличие нашего подхода от подхода, описанного в [6], заключается в том, что мы используем глубокие CNN вместо обратной сети из двух скрытых слоев. Наш алгоритм принимает меньше параметров, чем предложенный в [6] (5 против 9). Это сделано чтобы избежать переобучения и позволить скрытым слоям CNN выявить более точные зависимости.

Еще одним новым вкладом является улучшение классического PSF путем исключения дней, содержащих выбросы вне диапазона, равного 1,5 межквартильный диапазон (Interquartile Range (IQR)) исторических данных, из процесса прогнозирования. Это улучшение привело к снижению значений всех четырех погрешностей. Мы также обнаружили, что день, удаленный от прогнозируемого дня на неделю/две /три недели и т. д., имеет важное значение для прогнозирования нагрузки. Наконец, анализ паттернов кластеризации показал, что алгоритмы кластеризации способны улавливать сезонные колебания временных рядов электрической нагрузки.

КЛЮЧЕВЫЕ СЛОВА

Прогнозирование временных рядов (*Time Series Forecasting (TSF)*), прогнозирование электрической нагрузки (*Electrical Load Prediction*), прогнозирование цен на электроэнергию (*Electricity Price Prediction*), глубокое обучение (*Deep Learning*), машинное обучение (*Machine Learning*), сети долговременной краткосрочной памяти (*Long Short Term Memory Networks*), сверточные нейронных сетей (*Convolutional Neural Network*), методы k-средних (*k-means*), k-медиодов (*k-medoids*), нечетких C-средних (*Fuzzy C-Means*), иерархическая (агломеративная) кластеризация (*Hierarchical clustering*), самоорганизующиеся карты (*Self Organizing Maps*).

ГЛАВА 1. ОБЗОР ЛИТЕРАТУРЫ

Наиболее распространенным и популярным способом прогнозирования электрической нагрузки является анализ временных рядов. Он в значительной степени опирается на тщательный анализ прошлых наблюдений, для вывода закономерностей, которые содержат информацию о лежащих в их основе процессах. Например, многие временные ряды электрической нагрузки демонстрируют периодическую закономерность, когда пик спроса приходится на дневные и вечерние часы. Другие факторы также оказывают очень сильное влияние на спрос, например, потребление электроэнергии может следовать сезонным закономерностям, увеличиваясь с понижением температуры.

Большинство известных методов прогнозирования спроса на электроэнергию могут быть сгруппированы в следующие классы, в зависимости от длины горизонта прогнозирования (т. е. периода времени, подлежащего прогнозированию [7]):

- *Прогнозирование очень краткосрочных нагрузок (Very Short-Term Load Forecasting (VSTLF))*: для временных интервалов от секунд или минут до нескольких часов. Методы VSTLF применимы для управления потоком и полагаются на ближайшие входные переменные, такие как минуты и часы.
- *Краткосрочное прогнозирование нагрузки (Short-Term Load Forecasting (STLF))*: для прогнозирования чуть более длительных временных интервалов, варьирующихся от часов до недель. Методы STLF контролируют выработку электроэнергии в соответствии со спросом и широко используются на электротехническом рынке. Диапазон входных переменных для методов STLF составляет несколько дней.
- *Среднесрочное и долгосрочное прогнозирование нагрузки (Medium-Term and Long-Term Load Forecasting (MTLF and LTLF))*: применимо для длительных интервалов от месяцев до лет. Эти методы используются для разработки сетевых активов и утилит. Диапазон входных переменных для методов MTLF и LTLF охватывает недели и даже месяцы.

Компании больше заинтересованы в часовом, дневном и недельном горизонтах прогнозирования, так как это позволяет им планировать производство и продажу электроэнергии. Учитывая количество прогнозируемых значений можно выделить два основных класса моделей:

1. Модели, прогнозирующие только одно значение (например, общую нагрузку на следующий день или пиковую нагрузку).
2. Модели, прогнозирующие множественные значения, такие как агрегированная нагрузка или профиль нагрузки (формы) на следующий день.

Исходя из теоретического подхода, используемого в модели, можно выделить три основных класса моделей: линейные, нелинейные и модели, сочетающие в себе как линейные, так и нелинейные методы. Среди линейных моделей различают модели для предсказания только пика нагрузки и модели для предсказания формы (профиля) нагрузки. Модели профиля можно разделить на временные и динамические.

Модели времени суток имеют дело со значениями нагрузки $L(t)$ для каждого момента времени t , $t \in [1, T]$, в заданных временных рядах нагрузки, где T — период прогнозирования (например, $T = 24$ часа, если мы имеем дело с ежедневной нагрузкой). В общем случае модели времени суток можно представить формулой:

$$L(t) = \sum_{i=1}^N w_i f_i(t) + v(t)$$

т. е. нагрузка в момент времени t , $t \in [1, T]$ аппроксимируется взвешенной суммой явных временных функций $f_i(t)$ а термин $v(t)$ является коррекцией ошибки модели. Веса w_i можно оценить с помощью линейной регрессии. В некоторых случаях замена функций $f_i(t)$ собственными функциями для корреляционной функции временных рядов может привести к лучшим результатам [8]. Некоторые модели времени суток, такие как обычные наименьшие квадраты (Ordinary Least Squares(OLS)), обычно требуют более длительных исторических данных, например 12 лет для ежегодного прогнозирования [9].

Динамические модели предполагают связь между величиной нагрузки и внешними факторами, такими как температура или другие погодные условия. Наиболее известные динамические модели скользящего среднего значения (Moving Average (MA)) и интегрированная модель авторегрессии — скользящего среднего (Autoregressive Integrated Moving Average (ARIMA)). В общем случае модели ARIMA имеют следующий вид:

$$L(t) = y_p(t) + y(t),$$

где $y_p(t)$ коррелирует со временем суток и погодными условиями, а $y(t)$ - остаточный срок. В модели ARIMA температурная переменная преобразуется нелинейной функцией, чтобы учесть нелинейность задачи прогнозирования спроса.

В большинстве случаев традиционные статистические методы, такие как скользящая средняя (*Moving Average (MA)*) и интегрированная модель авторегрессии — скользящего среднего (*Autoregressive Integrated Moving Average (ARIMA)*), не могут выявить большую часть нелинейных зависимостей в данных о нагрузке на электроэнергию [10].

Кроме того, методы автоматизированного анализа временных рядов опираются на большой объем исторических данных для обучения моделей. Однако вовлечение большого объема данных в анализ и прогнозирование делает применение неавтоматизированных методов совершенно нецелесообразным, что обуславливает необходимость появления нетрадиционных автоматизированных методов прогнозирования временных рядов [11].

В отличие от традиционных подходов к прогнозированию временных рядов, нелинейные модели обеспечивают лучшее представление внутренних тенденций и влияния внешних факторов на электрическую нагрузку [12]. Например, Мартинес-Альварес (*Martínez-Álvarez*) и др. предложили интересный нетрадиционный подход к прогнозированию электрической нагрузки в [11], позже распространенный на прогнозирование цен на электроэнергию в [13]. Их метод основывается на предположении о существовании последовательности паттернов в исторических данных, аналогичной последовательности паттернов непосредственно перед предсказуемым днем. Алгоритм прогнозирования на основе меток (*Label-Based Forecasting (LBF)*), известный как алгоритм прогнозирования на основе последовательности паттернов (*Pattern Sequence-based Forecasting (PSF)*), впервые упомянут в [11] и содержит два этапа: кластеризацию и прогнозирование. Этап кластеризации предполагает применение алгоритма k -средних для группирования аналогичных фрагментов данных временных рядов. Число кластеров определяется значениями показателей индекса силуэта [14], Данна (*Dunn*) [15], и Дэвиса-Булдина (*Davies-Bouldin*) [16] методом большинства голосов (то есть, если индекс силуэта показывает что оптимальная сегрегация достигается при $k = 3$, индекс Данна при $k = 5$ и индекс Дэвиса-Булдина при $k = 3$ тоберется значение $k = 3$). В этом случае весь набор данных сегментируется метками, заменяя таким образом реальные данные последовательностями меток. Эти метки используются для прогнозирования значений целевой временной точки на этапе прогнозирования. Преимущества LBF/PSF алгоритма заключаются в его простоте и способности редуцировать размерность данных, тем самым уменьшая время обработки. Количество параметров невелико, что облегчает

процесс подбора оптимальных параметров. Метод также позволяет прогнозировать более чем одну выборку, что облегчает его развертывание для задач онлайн-обучения [11].

Несмотря на все преимущества, алгоритм PSF обладает рядом недостатков, которые были рассмотрены в [17]. В этой работе авторы подчеркивают неоднозначность использования большинства голосов трех индексов для определения оптимального размера кластера. И действительно, из (Таблицы 1) [18] очевидно, что иногда три индекса дают совершенно противоречивые значения для оптимального числа кластеров. Кроме того, последовательность кластеров для предсказуемого дня может быть аналогична последовательности перед предыдущим днем, особенно это характерно для рабочих дней. Наличие паттернов, отличных от предыдущего рабочего дня, может внести дополнительный шум при усреднении, что приводит к серьёзным различиям в прогнозировании [17]. Авторы предложили улучшить PSF, рассмотрев частоту кластеров для интересующего дня, причем более часто встречающемуся кластерному паттерну придавалось более высокое весовое значение при усреднении [17].

Вариация алгоритма PSF для расчета потребления электроэнергии при зарядке электромобиля была предложена в [19]. Вместо усреднения по всем найденным последовательностям модифицированная версия алгоритма PSF, т.н. (MPSF), для генерации прогноза использует последнюю найденную последовательность. В случае отсутствия совпадающей последовательности алгоритм MPSF рассматривает в качестве прогнозируемого значения центроид самого большого кластера, а не предыдущий день. Кроме того, алгоритм начинается с числа кластеров, равного 10% от общего числа выборок для выбора параметров. Модифицированная версия алгоритма PSF продемонстрировала повышенную точность для прогнозирования потребления энергии при зарядке электромобилей (*Electric Vehicle (EV)*) [19].

В последнее время было предложено несколько более усовершенствованных версий PSF алгоритма. Например, в [18] авторы дополнили алгоритм PSF другими методами кластеризации (k - медоиды, агломеративная кластеризация, нечеткие С-средние и самоорганизующиеся карты). Авторы также предложили ансамблевую модель, включающую различные алгоритмы кластеризации на основе PSF, для предсказаний методом итераций. Точность ансамбля превышала точность каждого отдельно взятого эксперта, а модель обеспечивала точный и более надежный прогноз данных о нагрузке на электроэнергию. Еще одно усовершенствование PSF было предложено в работе [20]. Вместо алгоритма k -средних авторы применили

положительную тензорную факторизацию для прогнозирования спроса и предложения энергии. Предложенное двумерное расширение PSF продемонстрировало более высокую точность, чем обычная структура.

PSF может быть реализован не только для прогнозирования стационарных временных рядов, но и для обнаружения аномалий (пиков), как это продемонстрировано в [21]. В этой работе PSF используется для обнаружения мотивов - последовательностей паттернов, которые предшествуют аномальным данным. Целевые данные, непосредственно следующие за мотивами, рассматривались отдельно. Алгоритм обнаружения аномалий (пиков) был включен в улучшенную версию PSF. Этот подход продемонстрировал высокую эффективность для обнаружения аномалий в большинстве электрических данных.

Целый класс нелинейных моделей основывается на употреблении искусственных нейронных сетей (*Artificial Neural Networks* (ANN)), которые получили широкое распространение со второй половины 80-х. Искусственные нейронные сети способны извлекать существенные признаки из данных, выводить нелинейные связи и изучать более общие понятия из предыдущего опыта. В зависимости от модели нейрона, архитектуры и алгоритма обучения ANN могут быть гибридными: (например, контрпропагандирование (*counterpropagation*)), контролируемыми: (Нечеткая Когнитивная Карта (*Fuzzy Cognitive Map*), Перцептрон (*Perceptron*), Радиальная Базисная Функция (*Radial Basis Function*), Метод Опорных Векторов (*Support Vector Machine*), неконтролируемыми (Адаптивная Резонансная Теория (*Adaptive Resonance Theory*), Самоорганизующиеся Карты (*Self-Organizing Maps*), Анализ Главных Компонент (*Principal Component Analysis*) и с подкреплением (ассоциативная награда/наказание). По сравнению с обычными методами, ANN способны обнаруживать нелинейные объекты даже с помощью небольшого набора обучающих данных, что приводит к более надежному и точному прогнозу. Будущие модели для прогнозирования временных рядов нагрузки будут либо включать в себя ANN наряду с обычными алгоритмами (т.е. гибридные модели) или будут основываться только исключительно на ANN [7].

Временные ряды могут быть смоделированы как комбинация детерминированной составляющей и некоторого шума. Минимизация шумовой составляющей приводит к более аккуратному прогнозированию модели. Для эффективного шумоподавления обычно используется вейвлет-преобразование. В [22] авторы разработали искусственную нейронную сеть с вейвлет-шумоподавляющим

алгоритмом для прогнозирования нагрузки в энергосистеме Болгарии. Их подход использует вейвлет-преобразование для извлечения сигналов различных частот из исторических данных по электрической нагрузки. После реконструкции сигнал содержит меньше шума и сохраняет паттерны исходных данных. В целом вейвлет-преобразование повышает точность прогнозирования и считается эффективным методом для краткосрочного прогнозирования.

В ряде работ сообщается о внедрении нейронных сетей обратного распространения (*Backpropagation Neural Network*) для краткосрочного прогнозирования данных временных рядов. Например, в [23] авторы применяют нейронную сеть обратного распространения для прогнозирования почасовой нагрузки и температурные данные североамериканских и словацких сетей. С помощью вейвлет-преобразования, авторы разложили данные по электрической нагрузке на низкочастотные и высокочастотные компоненты для последующего использования в качестве входных данных нейронной сети. Также учитывались и другие факторы, такие как температура, влажность и плотность облачного покрова.

Еще одна интересная реализация искусственных нейронных сетей, направленная на повышение точности алгоритма PSF для прогнозирования электрической нагрузки, была принята в [24]. Авторы использовали нейронные сети обратного распространения, с входными данными по нагрузке для предыдущего дня и предыдущей недели для уточнения прогноза алгоритма PSF.

Несмотря на эффективность искусственных нейронных сетей, довольно долгое время они уступали пальму первенства методам ядра, таким как опорная векторная регрессия (*Support Vector Regression (SVR)*). Возрождение нейронных сетей произошло в 2006 году, с публикацией основополагающей работы Джейфри Хинтона (*Geoffrey Hinton*) и др. [25]. В этой работе было показано, что сети глубокого обучения могут работать лучше, чем обычные методы машинного обучения. Она способствовала распространению методов глубокого обучения в различных задачах машинного обучения, включая компьютерное зрение, обработку естественных языков и другие области техники и науки. По сравнению с традиционным неглубоким обучением, сети глубокого обучения включают в себя несколько скрытых слоев и основываются на стохастической оптимизации для выявления более сложных нелинейных зависимостей [26]. Различное количество слоев позволяет кодировать различные уровни абстракции, что приводит к повышению производительности. Будучи сравнительно новой отраслью машинного обучения, сети глубокого обучения применимы к широкому спектру

сложных проблем. Множество примеров применения сетей глубокого обучения можно найти в классификации изображений, обнаружении и распознавании объектов, распознавании речи, машинном переводе, синтезировании голоса и т.д.

Самое первое упоминание сетей глубокого обучения для прогнозирования временных рядов приведена в [27]. Авторы применили сеть доверия глубокого обучения (*Deep Learning Believe Network* (DBN)) для прогнозирования данных временных рядов путем объединения выходов из всех DBN методом регрессии опорного вектора (*Support Vector Regression* (SVR)). Эффективность предложенного способа прогнозирования была продемонстрирована на трех независимых наборах данных. В работе [28] авторы использовали различные методы моделирования, чтобы показать, что рекуррентные сети потенциально могут превосходить обычные нелинейные методы с ядром. Они также рассмотрели, как точность предсказания сети меняется в зависимости от ее архитектуры (количество скрытых слоев и нейронов на каждом слое). Было выяснено, что дополнительная сложность не улучшает точность, а как раз наоборот, оптимальная точность может быть достигнута при малом количестве слоев. В другой работе [29] авторы разработали сеть глубокого обучения с контрастной дивергенцией для прогнозирования потребления природного газа. Другой пример использования сети глубокого обучения с ограниченной машиной Больцмана (*Restricted Boltzmann Machine* (RBM)) для прогнозирования временных рядов можно найти в работе [30]. Этот новый подход по аккуратности предсказаний превзошел традиционную многослойную нейронную сеть персептрона (*Multy-Layer Perceptron* (MLP)) и статистическую интегрированную модель авторегрессии - скользящего среднего ARIMA. Другое эффективное применение условно-ограниченной машины Больцмана (*Conditional Restricted Boltzmann Machine* (CRBM)) и факторизованной условно-ограниченной машины Больцмана (*Factored Conditional Restricted Boltzmann Machine* (FCRBM)) для индивидуального прогнозирования потребления электроэнергии в жилых домах рассмотрено в работе [31]. Метод был протестирован на эталонном наборе данных о потреблении электроэнергии отдельными потребителями с временным разрешением в одну минуту. Авторы пришли к выводу, что FCRBM превосходит ANN, метод опорных векторов (SVM), рекуррентные нейронные сети (RNN) и условно-ограниченную машину Больцмана (CRBM) [31].

В целом, обычные методы, основанные на ANN, подвержены уязвимостям, вызываемым затухающим или взрывным ростом градиента, которые возникают на стадии обратного распространения в многослойных архитектурах. Кроме того,

традиционные искусственные нейронные сети и традиционные методы машинного обучения, такие как SVM, не способны извлечь последовательную информацию из временных рядов [10]. Для преодоления этих проблем были разработаны рекуррентные нейронные сети (*Recurrent Neural Networks (RNN)*) как основа для выявления нестационарных паттернов во временных рядах, а также долгосрочных зависимостей. RNN способны объединять параметры с разных временных интервалов, и, таким образом, требуют меньше времени для обучения, что приводит к более эффективным вычислениям. Эти характеристики послужили становлению RNN в качестве надежного метода для долгосрочного и краткосрочного прогнозирования электрической нагрузки. Например, в работе [32] для прогнозирования электрической нагрузки домашних хозяйств была предложена RNN с объединяющим слоем. Эта сеть способна смягчить последствия переобучения путем объединения различных профилей нагрузки клиента в один вход, тем самым диверсифицируя данные. Предложенная глубокая RNN на основе объединения пулинга (*pooling-based deep RNN*), продемонстрировала более высокую точность по сравнению с традиционными методами, такими как ARIMA, SVR и обычные глубокие RNN. В работе [10] авторы разработали гибридную модель RNN, исходя из концепции пошагового обучения. Сначала RNN был обучен на временных рядах электрической нагрузки, которые позже были переданы гибридной модели RNN для изучения нелинейных зависимостей. Предложенная модель продемонстрировала превосходную точность по сравнению с известными методами.

Однако глубокие RNN уязвимы перед затухающим или взрывным ростом градиента, пагубно влияющих на их точность. Сети долговременной кратковременной памяти (*Long-Short Term Memory Networks (LSTMN)*) были первоначально введены Зеппом Хохрейтером (*Sepp Hochreiter*) и Юргеном Шмидхубером (*Jurgen Schmidhuber*) [33] в качестве вариации RNN. Вскоре они стали популярным инструментом для предсказания временных рядов благодаря своей способности справляться с затухающим или взрывным ростом градиента. Сети LSTM успешно применяются для прогнозирования временных рядов в различных областях. Например, в работе [34] авторы успешно адаптировали комбинацию *Deep Learning Networks, AutoEncoder* и LSTM для прогнозирования мощности возобновляемых источников энергии и продемонстрировали, что методы глубокого обучения работают лучше, чем ANN. В [35], дополнительные характеристики, такие как температура, влажность и скорость крыла, были переданы в сеть LSTM с данными временных рядов электрической нагрузки. Обученная на этих данных сеть LSTM была использована для

прогнозирования электрической нагрузки как с короткими, так и с длинными горизонтами, варьирующиеся от 24 часов до 30 дней. Авторы также показали, что прогноз LSTM является более точным, чем прогноз на основе привычных методов прогнозирования.

Сети LSTM также могут быть модифицированы для прогнозирования индивидуального потребления электроэнергии. Этот тип прогнозирования является очень сложной задачей из-за изменчивого характера отдельных нагрузок, на которые влияют различные внешние факторы, такие как климат, производительность тепловых систем и характеристики отдельно взятого домохозяйства. Марино (*Marino*) и др. [36] были одни из первых первыми, кто применил сети LSTM для прогноза нагрузки от одного счетчика. Они использовали тот же эталонный набор данных, что и в [36] достигли аналогичной производительности для факторизованной условно-ограниченной машины Больцмана (*Factored Conditional Restricted Boltzmann Machine (FCRBM)*). В другой работе [37] сеть LSTM была приспособлена для долгосрочного прогнозирования электрической нагрузки. Авторы применили кластеризацию на основе плотности, для получения вывода о противоречивом характере профилей электрической нагрузки для индивидуального домохозяйства. Также авторы обсудили проблемы связанные с прогнозированием нагрузки в жилых районах. В работе показано, что сети LSTM, в большинстве случаев, способны улавливать тонкие паттерны в индивидуальных профилях нагрузки, что является необходимым для решения похожих задач.

Другой класс методов глубокого обучения, сверточные нейронные сети (*Convolutional Neural Networks (CNN)*), успешно используются для распознавания объектов благодаря их предрасположенности к выявлению скрытых зависимостей в необработанных данных. CNN также может обнаружить локальные трендовые характеристики и масштабно-инвариантные характеристики для тесно взаимосвязанных соседних данных, например локальные трендовые паттерны в ближайшие часы. Эти обстоятельства обуславливают широкое применение CNN для прогнозирования электрической нагрузки [38]. Пример прогнозирования нагрузки с помощью CNN можно найти в работе [39], где авторы демонстрируют превосходство подхода, основанного на CNN, позволяющего достичь наименьшей ошибки прогнозирования среди известных методов. В предлагаемой архитектуре CNN использовались три последовательных сверточных слоя (*convolutional layers*). Субдискретизирующие слои (*слои подвыборки, pooling layers*) уменьшают размерность карты объектов и позволяют

обнаружить интересные характеристики из глубоких слоев. Прогнозирование производится на выходном слое, где уплотненные значения проходят через полностью связную структуру между слоями уплотнения и выходным слоями, к входам сигмоидной функции. Этот алгоритм позволяет сделать прогноз электрической нагрузки на ближайшие три дня.

Амарасинг (*Amarasinghe*) и др. [40] исследовали применение CNN для прогнозирования электрической нагрузки на уровне отдельных зданий. Исторические нагрузки пропускались через несколько сверточных слоев и подавались на полностью соединенные слои для выполнения регрессии и получения прогноза. Как и в аналогичной работе по LSTM [36], для оценки эффективности подхода на основе CNN в сравнении с ANN, SVM, LSTM и FCRBM был использован эталонный набор данных для индивидуальных клиентов. Представленная сеть CNN продемонстрировала сопоставимые результаты с FCRBM [31] и *Sequence to Sequence* LSTM [36] на одном и том же наборе данных. Все приведенные выше примеры демонстрируют жизнеспособность CNN для точного прогнозирования электрической нагрузки.

В прошлом поставщики электроэнергии не были ориентированы на рынок, и цены на нее устанавливались правительствами. Однако после коммерциализации сетей цена начала определяться вновь возникшим рынком. Другими словами, цена устанавливается в соответствии с колебаниями спроса и предложения на рынке, где электроэнергия рассматривается в качестве товара. По сравнению с нагрузкой, цена на электроэнергию обладает большей волатильностью, т. е. подвержена существенным изменениям за данный промежуток времени. Более того, данные о нагрузке относительно однородны и демонстрируют циклические вариации, в то время как данные о ценах лишены такой однородности, и их вариации следуют довольно быстрому циклическому паттерну [6].

В условиях динамичного электроэнергии рынка, предсказание цены на электроэнергию имеет важное значение для принятия решений. На самом деле от точных прогнозов цен выигрывают все субъекты рынка. Независимые производители электроэнергии (*Independent Power Producers (IPP)*) смогут регулировать выработку электроэнергии в соответствии с ожидаемой ценой, снижая себестоимость производства. Участники рынка смогут разрабатывать стратегии продажи и покупки электроэнергии на рынке. Конечные потребители будут переносить свою основную деятельность по потреблению электроэнергии на часы, когда цены на электроэнергию являются низкими, тем самым снижая нагрузку сети в часы пик. Проблема

прогнозирования цен опирается на ряд факторов, таких как корреляции между спросом и предложением электроэнергии, затраты на генерацию, структура рынка, влияние других участников рынка и др. Эти факторы используются для создания моделей, отражающих причинно-следственную связь между историческими ценами на электроэнергию и будущими ценами. Знание этой причинно-следственной связи позволяет точно прогнозировать цены на электроэнергию в будущем [41].

Поскольку электроэнергия не является складируемым товаром, баланс между спросом и предложением электроэнергии может быть достигнут только при стабильной энергосистеме. Поэтому в большинстве случаев цены на электроэнергию очень волатильны, и легко подвержены влиянию таких факторов, как внезапные изменения погодных условий, проблемы с передачей электроэнергии в энергосистеме, перебои на крупных электростанциях, стратегии участников торгов, перегрузка сетей при передачи электроэнергии, цены на топливо, стоимость единичных операций и многие другие. Все эти факторы вносят дополнительные сложности в задачу прогнозирования цен [41], [42].

Подобно прогнозированию нагрузки в электросетях, существуют традиционные статистические методы и современные подходы к прогнозированию цен на электроэнергию, основанные на машинном обучении. Пример статистической модели, основанной на интегрированной модели авторегрессии — скользящего среднего (*Autoregressive Integrated Moving Average (ARIMA)*) в сочетании с вейвлет-преобразованием, описан в [43], где прогноз на ближайшие 24 часа был получен путем применения обратного вейвлет-преобразования. Модель была протестирована на рынках материковой Испании и Калифорнии, демонстрируя вполне надежные результаты. В [44] ARIMA вместе с обобщенной авторегрессионной условной гетероскедастичностью (*Generalized Autoregressive Conditional Heteroskedasticity (GARCH)*) применена к временным рядам, пропущенным через вейвлет-преобразование для разложения их на две части, которые могут быть предсказаны отдельно. Для получения окончательного прогноза было применено обратное вейвлет-преобразование для объединения компонент в один временной ряд. Эффективность предложенного подхода была оценена на данных с рынков электроэнергии Испании и PJM, и полученные результаты были сопоставлены с другими методами прогнозирования.

Традиционные статистические системы хорошо работают с линейными данными, однако, когда речь заходит о нелинейных временных рядах, их точность прогнозирования существенно падает. Чтобы справиться с экстремальной нелинейностью данных ценового временного ряда, Воронин и др. [45] разработали

итерационную модель, состоящую из модулей для прогнозирования нормальной цены и ценовых пиков (*всплесков, аномалий*). Модуль для нормального прогнозирования цены представляет собой смесь вейвлет-преобразования, линейной ARIMA и нелинейных нейросетевых моделей. Поиск пиков производился с помощью гибридных классификаторов (*модулями пикового прогнозирования*). Прогноз цен производится путем объединения прогнозов как от модулей нормального прогнозирования цен, так и от модулей пикового прогнозирования. Точность предсказаний комбинированной модели превзошла другие известные методы прогнозирования цен.

Способность искусственных нейронных сетей выявлять нелинейные зависимости в данных временных рядов была использована несколькими авторами для создания успешных моделей прогнозирования цен на электроэнергию. В [46] для прогнозирования краткосрочной электрической нагрузки и цены авторы использовали нейронную сеть радиальных базисных функций (*Radial Basis Function Neural Network (RBFNN)*). На вход модели в качестве дополнительного параметра подавалась информация о дне недели (например, воскресенье, понедельник). Согласно заключению авторов, включение информации о дне недели в входные данные обуславливает более высокую точность предсказания. Другая работа [6] опирается на искусственные нейронные сети для извлечения сложных зависимостей между ценой электроэнергии, нагрузкой и другими факторами. Модель принимает в качестве входных данных несколько функций, таких как день недели, временной интервал дня, прогнозируемый спрос, изменение спроса, исторические цены за день, неделю, две недели, три недели и месяц назад. Горизонт прогнозирования может варьироваться от нескольких часов до нескольких дней и недель. Модель оказалась эффективной для обычных дней, но продемонстрировала низкую точность для дней с ценовыми пиками.

В [41] авторы предлагают систему прогнозирования цен на электроэнергию, основанную на сочетании сверточной нейронной сети (*Convolutional Neural Network (CNN)*) и долговременной краткосрочной памяти (*Long Short Term Memory (LSTM)*). Сеть CNN состоит из двух сверточных слоев 1D, с пакетной нормализацией после второго сверточного слоя. Прогнозирование осуществляется блоком LSTM с функцией активации ReLU. Гибридная система принимает на входе данные о ценах за последние 24 часа и выдает прогноз на следующий час. По сравнению с традиционными подходами машинного обучения представленная гибридная модель демонстрирует наилучшую точность прогнозирования.

Применение системы глубокого обучения для прогнозирования цен на электроэнергию также описано в [47], где авторы рассматривают четыре различные модели глубокого обучения и сравнивают их точности прогнозирования. Точность прогнозирования методов глубокого обучения как правило превосходит точность от обычных методов. Авторы также демонстрируют, что подходы машинного обучения работают лучше, чем традиционные статистические модели. Традиционные подходы не позволяют достичь высокой точности прогноза для ценовых временных рядов, особенно для моделей, основанных на скользящих средних. Кроме того, гибридные модели не превосходят по точности свои более простые компоненты. Таким образом, предпочтение следует отдавать простым нелинейным моделям, например сетям глубокого обучения с небольшим количеством нейронов в каждом слое.

В целом, методы глубокого обучения демонстрируют более высокую эффективность в отношении данных о ценах на электроэнергию благодаря их способности выявлять сложные нелинейные зависимости и закономерности. Рассматривая все вышеупомянутые примеры можно прийти к следующим заключениям:

- *Сети долговременной кратковременной памяти (LSTM) [33] лучше подходят для работы с нелинейными последовательностями и, таким образом, обеспечивают лучшую точность прогнозирования цен на электроэнергию*
- *Сеть с более чем одним слоем способна моделировать ту же зависимость, что и однослойная сеть, но с меньшим числом нейронов, что упрощает архитектуру сети глубокого обучения. Это также улучшает способность сети к обобщению на большой выборке данных [47].*

Таким образом, подходы глубокого обучения представляют собой весьма перспективную технологию для прогнозирования цен на электроэнергию.

ГЛАВА 2. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ

ПОСТАНОВКА ЗАДАЧИ

Формально временные ряды можно представить в виде последовательности векторов, зависящих от времени t : $X(t)$, $t = 0, 1, \dots$. Компонентами вектора $X(t)$ являются наблюдаемые переменные, такими как температура, скорость ветра, цена акций, электрическая нагрузка и т. д. На самом деле $X(t)$ - непрерывная функция, однако на практике мы имеем дело с дискретными временными интервалами. $X(t)$ определенные на таких дискретных временных интервалах, называются временными последовательностями или временными рядами. В случае непрерывных переменных, таких как температура, значения переменной замеряются в определенных точках временного интервала (например, измерение температуры один раз в час). Количество измеренных точек на данном временном интервале называется частотой дискретизации. Это важный параметр, поскольку различные частоты дискретизации могут влиять на характеристики временного ряда.

Если вектор $X(t)$ состоит только из одной компоненты, то временные ряды называются одномерными. В случае многокомпонентных векторов $X(t)$ мы имеем дело с многомерными временными рядами. Анализ многомерных временных рядов основан на выявлении зависимостей между наблюдаемыми компонентами вектора $X(t)$, например, влияния температуры на цену электроэнергии, и использовании этих паттернов для прогнозирования одного из компонентов.

Прогнозирование временных рядов, т. е. прогнозирование будущей динамики интересующей нас переменной, пользуется наибольшим спросом в различных отраслях науки и техники. Большинство индустриальных задач связано с предсказанием будущих значений вектора $X(t)$, например, прогнозирование спроса на электрическую энергию для оценки нагрузки сети и планирования производства электроэнергии. Формально задачу прогнозирования временных рядов можно определить следующим образом [48]: для данного временного ряда $X(t)$, $t = 0, 1, \dots$, найти функцию $f: \mathbb{R}^{k \times n+l} \rightarrow \mathbb{R}^k$, где $k = \dim(X(t))$. $k = 24$, если мы заинтересованы в прогнозировании почасовой нагрузки, а n - это число отдельных векторов временных рядов для обучения, например число дней для обучения модели. Конечная цель состоит в получении оценки $\hat{X}(t + d)$ вектора X , учитывая значения $X(t)$ до времени t , а также дополнительные зависящие от времени

экзогенные признаки $\pi_i, i = 1, \dots, l$, которые используются для учета влияния внешних факторов, например: динамики плотности населения в интересующем районе:

$$\hat{X}(t + d) = f(X(t), X(t - 1), \dots, X(0), \pi_1, \dots, \pi_l),$$

где d – лаг прогноза (обычно, $d = 1$). В данной работе мы пренебрегаем экзогенными признаками $\pi_i, i = 1, \dots, l$.

ЭКСПЕРТНЫЕ АЛГОРИТМЫ

В данной работе, в качестве традиционных автоматических методов прогнозирования, мы рассматриваем алгоритмы на основе PSF для прогнозирования нагрузки со следующими методами кластеризации:

- *k - средних*
- *k - медоиды*
- *агломеративная кластеризация*
- *нечеткая кластеризация c-средних*
- *самоорганизующиеся карты*

Мы приводим сравнение традиционных методов машинного обучения для краткосрочного прогноза нагрузки с методами глубокого обучения - сверточными нейронными сетями (*Convolutional Neural Networks (CNN)*) и сетями долговременной кратковременной памяти (*Long-Short Term Memory Networks (LSTM)*). Сверточная нейронная сеть (CNN) может извлекать локальные тренды, скрытые характеристики и интересные паттерны, а сети долговременной краткосрочной памяти (LSTM) способны извлечь временные корреляции во временных рядах [49].

***k* -средние**

Цель кластеризации *k*-средних состоит в минимизации целевой функции J , которая представляет собой совокупное расстояние между центром кластера и множеством C_j из N_j точек данных, принадлежащих определенному кластеру k . Другими словами, *k* -средних пытается разбить N наблюдений на k отдельных подмножеств C_j , содержащих N_j точек данных. Алгоритм *k*-средних является примером бесконтрольного обучения, так как он не требует помеченных данных. Это также

жадный алгоритм, производительность которого зависит от выбора параметра k и соответствующего выбора начальных центров кластера [50].

Целевая функция J может быть задана как [18]:

$$J = \sum_{k=1}^K \sum_{n \in C_j} w_{i,n} d(x_n, \mu_k)$$

где μ_k - центроид кластера для точек в C_k и $d(x_n, \mu_k)$ - метрика расстояния между точкой данных x_n и центром кластера μ_k . Наиболее популярной метрикой расстояния является Евклидово расстояние.

В общем случае алгоритм k -средних включает в себя следующие шаги:

1. инициализация параметров алгоритма, задание количества кластеров k ,
2. перетасовка данных, для исключения возможных упорядоченностей,
3. выбор случайным образом k отдельных точек данных в качестве начальных центроидов кластера,
4. вычисление расстояния $d(x_n, \mu_k)$ для каждой точки данных x_n и для каждого кластера k между этой точкой данных и центром кластера μ_k ,
5. назначение каждой точки данных ближайшему кластеру (центроиду) на основе полученной метрики расстояния,
6. вычисление центроидов для кластеров как среднее значение всех точек данных, принадлежащих каждому кластеру.

Алгоритм продолжает итерировать до тех пор, пока не будет достигнута стабильность в изменении положений центроидов, т. е. не будет зафиксировано распределение точек по соответствующим кластерам.

Алгоритм k -средних прост в реализации и вычисляется быстрее, чем другие известные методы (например, иерархическая кластеризация). Однако начальное состояние рандомизации и порядок расположения данных могут повлиять на кластеризацию (именно поэтому необходим шаг 2). Кроме того, нормализация набора данных может существенно повлиять на результаты.

k -медоиды

Алгоритм k -медоидов [51] опирается на представлении о медоиде как он центре кластера, в отличие от центроида, принимаемого за центр кластера в алгоритме k -

средних. В отличие от центроида, то есть среднего значения точек данных в кластере C_j , медоид является экземпляром данных в этом кластере. Для определения медоидов алгоритм ищет точку данных i внутри кластера, которая минимизирует сумму $\sum_{j \in C_j} d(i, j)$, где C_j – кластер объекта i , а $d(i, j)$ - расстояние между объектами i и j .

Алгоритм k -medoids может быть описан следующим образом [51]:

1. инициализация параметров алгоритма, с указанием количества кластеров k ,
2. перетасовка данные, для исключения возможных упорядоченностей,
3. случайной выборке k отдельных точек данных в качестве исходных кластерных медоидов,
4. назначении каждой точке данных кластеру, связанному с ближним медоидом
5. пересчет позиции k -медоидов.

Алгоритм продолжает итерировать до тех пор, пока не перестанут наблюдаться изменения положений медоидов, т. е. не будет зафиксировано назначение точек данных кластерам.

Есть несколько преимуществ использования существующих объектов в качестве центров кластеров (медоидов):

- как представитель данных, медоид служит для описания кластера лучше, чем абстрактный центроид,
- k -средних пересчитывает все расстояния между точками данных и центроидами, так как новые центроиды создаются на каждой итерации. В случае k -медоидов нет необходимости пересчитывать расстояния между точками данных, и алгоритм может получить их из матрицы расстояний, ускоряя сходимость алгоритма k -медоидов за фиксированное число шагов,
- k -медоиды менее чувствителен к аномалиям и выбросам, чем другие алгоритмы кластеризации.

Однако алгоритм k -медоиды может сходиться к разным разделам для разных запусков в одном и том же наборе данных, поскольку первые k -медоиды выбираются случайнным образом.

Агломеративная кластеризация

Алгоритм агломеративной (иерархической) кластеризации инициализирует каждую точку данных как отдельный кластер, рекурсивно объединяя аналогичные точки в соответствующие кластеры. Нет необходимости предоставлять заранее заданное

количество кластеров [52]. Единственным параметром, необходимым для предотвращения слияния всех кластеров в один является расстояние отсечения или оптимальное количество кластеров. Его можно определить по дендрограмме — дереву, представляющему отношения между подобными кластерами. Одна ось дендрограммы представляет собой точки данных и кластеры, другая — расстояние или расхождение между кластерами. Интуитивно понятным способом определения расстояния отсечения является выбор расстояния, соответствующего максимальному расхождению между кластерами (где ветви длиннее). Существует также несколько вычислительных методов, таких как метод динамического разреза [53] который обрабатывает топологию дерева.

Алгоритм агломеративной (иерархической) кластеризации начинается с вычисления матрицы расстояний M , содержащей все попарные расстояния между n точками данного набора данных $D = (x_1, x_2, \dots, x_n)$. Затем происходит переход к следующим шагам:

1. *выбор двух ближайших точек $x_i, x_j \in D$ т. е. $i, j = \underset{i, j: i \neq j}{\operatorname{argmin}} d(x_i, x_j)$* ,
2. *агломерация этих двух точки в новую точку (кластер) с. Эта новая точка может быть получена путем вычисления среднего значения x_i, x_j или с помощью какого-либо другого метода,*
3. *замена x_i, x_j в D на эту новую точку с и пересчет матрицы расстояний M ,*
4. *рекурсивные повтор шагов 1 – 3 до тех пор пока D не будет содержать только одну точку данных.*

Иерархическая кластеризация создает более информативную иерархическую структуру данных, которая позволяет более точно определить количество кластеров. Недостатком метода является то, что он может сходиться к нереалистичным дендрограммам, и может не отражать существующие корреляции между точками данных. Алгоритм чувствителен к выбросам и инициализации кластеров на первом этапе. Кроме того, вычисление матрицы расстояний M требует больших вычислительных ресурсов, что может быть неосуществимо при больших наборах данных.

Нечеткая кластеризация С-средних

Методы нечеткой логики могут быть использованы для кластеризации многомерных данных путем присвоения каждой точке значения принадлежности, варьирующегося от 0 до 100%, и отражающего близость точки данных к конкретному

центру кластера [54]. Таким образом, точка данных может принадлежать одновременно нескольким кластерам, что позволяет достигать лучших результаты при кластеризации смешанных наборов данных. Это придает методу нечеткой кластеризации c -средних большую гибкость по сравнению с традиционными подходами, основанными на использовании пороговых значений.

Для набора данных $D = (x_1, x_2, \dots, x_n)$ из N точек и множества $V = \{v_1, v_2, \dots, v_c\}$, алгоритм нечеткого c -среднего состоит из следующих шагов:

1. произвольный выбор центров кластеров c ,
2. определение матрицы нечеткого множества $U = (u_{i,j})_{n \times c}$, где это $u_{i,j}$ является нечетким множеством i -ых точечных данных к j -ых кластера:

$$u_{i,j} = \frac{1}{\sum_{k=1}^c \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{2}{m-1}}}$$

где n -число точек данных, d_{ij} – расстояние от точки данных i -ых до кластера j -ых, $m \geq 1$ – индекс нечеткости,

3. вычисление нечетких центров v_j по формуле: $v_j = \frac{\sum_{i=1}^n x_i (u_{ij})^m}{\sum_{i=1}^n (u_{ij})^m}$, для каждого $j = 1, 2, \dots, c$.
4. повторение шагов 2 и 3, для минимизации целевой функции:

$$J(U, V) = \sum_{i=1}^n \sum_{j=1}^c (u_{ij})^m (x_i, v_j)$$

или по достижении порога β , где $\beta > d(U^{(k+1)}, U^{(k)})$ и $\beta \in [0, 1]$.

Самоорганизующиеся карты (Self-Organizing Map (SOM))

Самоорганизующиеся карты (Self-Organizing Map(SOM)) - это искусственная нейронная сеть для бесконтрольного обучения, способная уменьшить размерность данных, отображая их в двумерное пространство, но при этом сохраняя топологические свойства данных (входное пространство). Заметим что сохраняются только топологические свойства (т.е. паттерны соотношений между данными), а не фактические расстояния [55]. Другими словами, узлы в выходном пространстве сети отражают статистические особенности входного пространства (данных).

Используя процесс обучения путем наблюдений, самоорганизующиеся карты способны выявлять скрытые закономерности и зависимости в данных [56]. Для этой цели SOM используют политику конкурентного обучения, т. е. заставляют выходные узлы конкурировать между собой за лучшее представление концепций из данных [57]. Качество представления оценивается дискриминантной функцией, например блоком наилучшего соответствия (*Best Matching Unit* (BMU)), который сравнивает входной вектор с взвешенным вектором каждого выходного узла, так что выбирается узел с весами соединений, наиболее близкими к исходному входному экземпляру.

Соседние узлы в пространстве вывода содержат похожие паттерны входных данных. Таким образом, узел – победитель создает окрестности узлов с общими признаками (сотрудничающие узлы). В результате соревновательной подготовки происходит модификация победившего узла и его окружения: победитель становится центром локальной группы взаимодействующих узлов, которые вносят свой вклад в процесс корректировки веса. Другими словами, веса узла – победителя и его соседей обновляются, чтобы максимизировать их дискриминантные функции, тем самым адаптируя соответствующие узлы к входным данным. Эти узлы способны индуцировать взаимную активацию, и это делает их более отзывчивыми к аналогичным функциям во входных данных.

SOM - это полностью связанная сеть в том смысле, что каждый узел во входном пространстве связан с каждым узлом в выходном слое [58]. Узлы входного слоя не участвуют в обучении сети. Они непосредственно передают информацию от входных узлов к узлам выходного слоя, так что каждое значение входных данных связано с одним входным узлом. Каждый узел инициализируется вектором весов $w = [w_1, w_2, \dots, w_n]^T$, где n -размерность входного пространства. Данные могут быть непосредственно переданы в выходной слой, как это делается в картах Кохонена [59] или пройти через несколько скрытых слоев (более общие случаи SOM). Двумерная сеточная структура принята для удобства процесса понижения размерности данных путем проецирования с n - мерной на двумерную плоскости. Нейроны выходного слоя расположены в узлах двумерной сети прямоугольной или шестиугольной структуры. Это сделано потому что евклидовы метрики легче определить на шестиугольной, а не прямоугольной структуре. Сама по себе сеть очень чувствительна к выбору метрик, так как степень взаимодействия между узлами зависит от расстояния между ними, а количество узлов определяет способность карты к обобщению.

Алгоритм кластеризации SOM состоит из следующих шагов:

1. *Инициализация* включает в себя установку следующей конфигурации сети:

- количество узлов в сети,
- скорость обучения η
- начальный радиус обучения R (количество соседей узла-победителя).

Определение правильного набора параметров зависит от компромисса между степенью детализации и скоростью обучения. Может возникнуть ситуация, когда количество нейронов превышает количество объектов в обучающей выборке данных. В этом случае результат обучения зависит от выбора начального радиуса, что влияет на способность сети к обобщению. Большое количество узлов приведет к более длительному обучению. Кроме того, во время инициализации весам узлов присваиваются определенные значения. Можно инициализировать все узлы малыми случайными числами, или случайно выбранными значениями признаков из обучающего набора, или инициализировать линейно упорядоченными векторами в линейном подпространстве между двумя основными собственными векторами исходного набора данных.

2. *Обучение.* Обучение начинается с активации сети вектором активации X_n , случайным образом выбранным из набора данных. На этом этапе каждый узел конкурирует с другими узлами, чтобы стать более похожим на обучающие данные. В конце этого процесса, известного как векторное квантование, узел-победитель является самым близким к X_n . Если мы обозначим весовой вектор узла-победителя как w_c , то метрика расстояния, характеризующая близость узла c к вектору активации X_n , равна $\|X_n - w_c\| = \min_i \|X_n - w_i\|$, где $i = 1, \dots, n$. После выбора узла-победителя происходит консолидация других узлов в пределах радиуса обучения R . На этом этапе обучения веса узла-победителя и его соседей движутся в направлении вектора X_n . Новые веса для итерации $t + 1$ инициализируются как $w_i^{t+1} = w_i^t + \eta \|X_n^t - w_i^t\|$, где η скорость обучения, а w_i^t – веса из предыдущей итерации (см. Рис. 1).
3. На этапе *коррекции* радиус обучения R и скорость обучения η настраиваются для достижения оптимальной производительности. Во-первых, берутся большие значения радиуса и скорости обучения, чтобы обеспечить соответствие

местоположения сетевого узла данным в обучающем наборе. Затем радиус и скорость обучения уменьшаются для достижения более точной настройки.

Обученная самоорганизующая сеть содержит все соотношения между узлами из выходного слоя и паттернами во входных данных. Таким образом, все паттерны из входных данных могут быть сопоставлены с величинами на выходном слое. В этой работе для кластеризации данных с помощью SOM мы устанавливаем число узлов в выходном слое равным числу кластеров и обучаем сеть так, чтобы каждый узел был центроидом кластера, таким образом идентифицируя кластеры с нейронами сети.

Несмотря на необходимость указания оптимального количества кластеров и структуры сети (топологии узлов), самоорганизующиеся карты имеют ряд преимуществ перед традиционными методами уменьшения размерности, такими как анализ главных

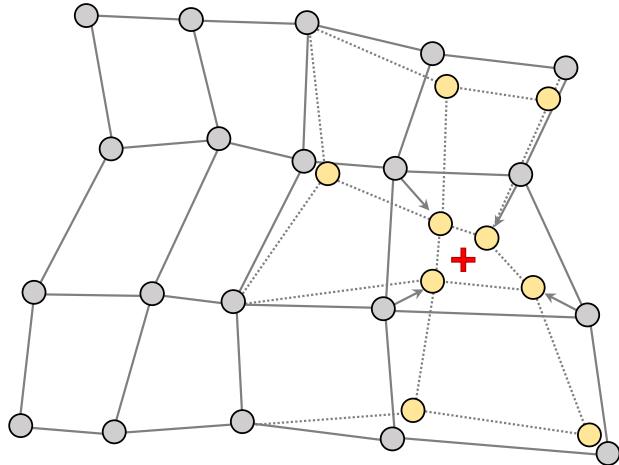


Рис. 1. Настройка весов узла-победителя и его соседей. Положение вектора активации отмечено крестиком, узлы карт до настройки серые, после настройки желтые. Сеть после настройки обозначается пунктирными линиями.

компонент и многомерное масштабирование. Во-первых, они могут работать с любыми данными, т. е. нет никаких требований к тому, чтобы экземпляры данных были независимыми.

Самоорганизующиеся карты также не делают никаких предположений о распределении данных [57] [60]. Далее, самоорганизующиеся карты способны решать задачи высокой сложности и не

требуют больших усилий при их реализации [61]. И наконец, самоорганизующиеся карты более эффективны для борьбы с выбросами, шумом, пропущенными значениями [62], они могут обрабатывать экземпляры очень низкой размерности [57] и хорошо работать с большими данными [63]. Способность самоорганизующихся карт уменьшать размерность данных при сохранении связей между точками данных делает их отличным инструментом для применения при прогнозировании нелинейных временных рядов [64].

АЛГОРИТМ ПРОГНОЗИРОВАНИЯ НА ОСНОВЕ ПОСЛЕДОВАТЕЛЬНОСТИ ПАТТЕРНОВ

Схема алгоритма прогнозирования на основе последовательности паттернов (*Pattern Sequence-Based Forecasting (PSF)*) [11] (известный также как алгоритм прогнозирования на основе меток (*Label Based Forecasting Algorithm (LBF)*)) представлена на Рис. 2

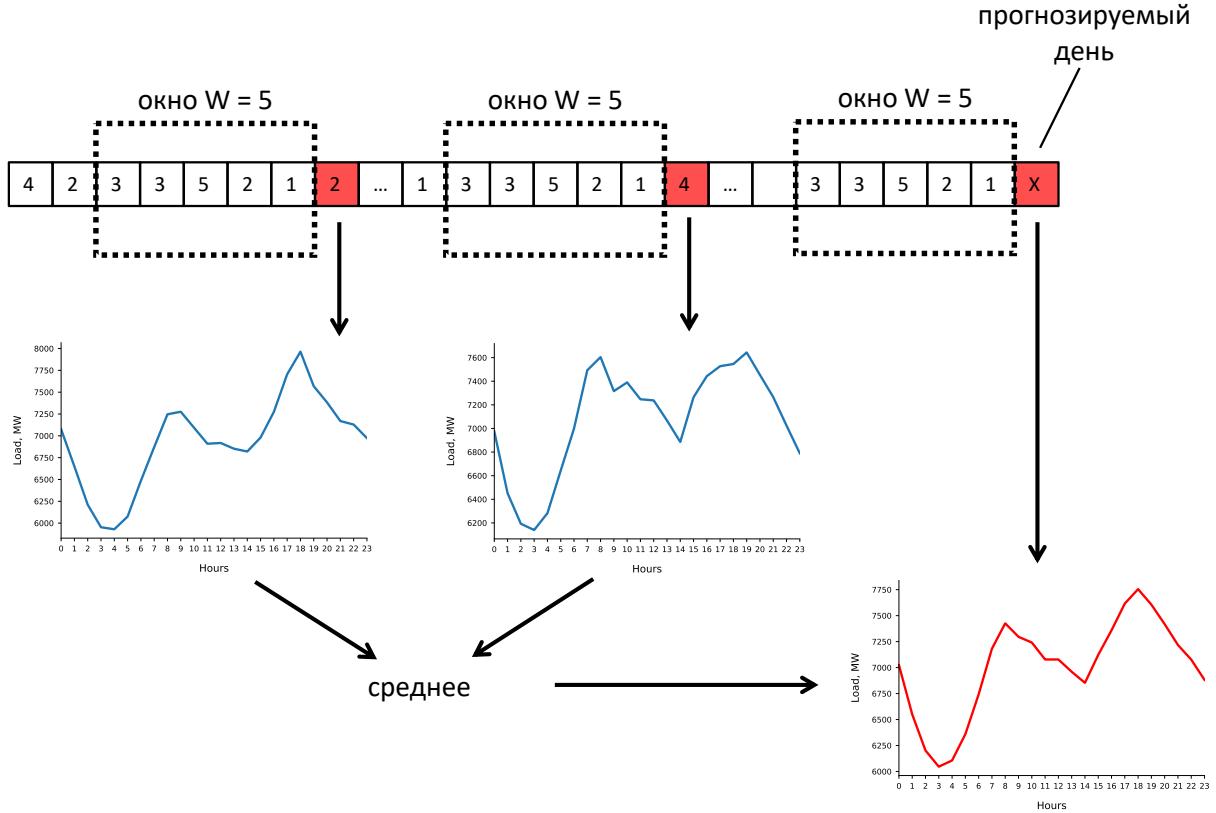


Рис. 2. Алгоритм прогнозирования на основе последовательности паттернов (см. Рис. 3 [11]).

Электрическая нагрузка для каждого дня i представлена векторами $E_i = \langle e_1, e_2, \dots, e_{24} \rangle$, размерности 1×24 , где e_i - значения нагрузки для каждого часа. После применения метода кластеризации ко всем этим N векторам мы получаем набор меток $L = \langle L_1, L_2, \dots, L_N \rangle$. Обозначим через $S_W^d = \langle L_{i-W+1}, L_{i-W+2}, \dots, L_{d-1}, L_d \rangle$ подпоследовательность меток нагрузок W последовательных дней, начиная со дня d назад. Размер окна определяется в [11] и обычно составляет около 5 дней. Алгоритм проходит через предыдущие дни, помеченные алгоритмом кластеризации, чтобы найти точную последовательность W (Exact Sequence(ES)) меток, предшествующую дню, который мы собираемся предсказать. Другими словами, нам нужно найти множество ES , определяемое уравнением:

$$ES = \{ \text{набор показателей } j \text{ таких что } S_W^j = S_W^d \}$$

При отсутствии какой-либо подпоследовательности, равной S_W^d , алгоритм ищет подпоследовательности меток S_{W-1}^d . Другими словами, мы уменьшаем длину окна, состоящего из подпоследовательности меток.

24-х часовые нагрузки дня $d + 1$ предсказываются путем усреднения нагрузок дней, следующих за теми, что были в ES . То есть,

$$E_{d+1} = \frac{1}{\text{size}(ES)} \sum_{j \in ES} E_{j+1},$$

где $\text{size}(ES)$ - количество элементов в наборе ES .

СЕТИ ГЛУБОКОГО ОБУЧЕНИЯ ДЛЯ ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ

Глубокое обучение – область машинного обучения, которая рассматривает применение многослойных нейронных сетей для получения характеристик более высокого уровня из входных данных [65]. Чем более глубже расположены слои, тем более значимыми становятся выведенные характеристики и закономерности. Глубокое обучение основывается на последовательности слоев представлений, которые вносят вклад в модель (т.н. глубина модели). Модель глубокого обучения может содержать сотни последовательных слоев. Модели, использующие несколько слоев, называются моделями поверхностного обучения. Другими словами, модель глубокого обучения – это многоступенчатая структура, которая перегоняет информацию через последовательные фильтры, чтобы очистить от шума и сделать ее пригодной для обработки сетью в целях извлечения характеристик и обучению представлениям [66].

При глубоком обучении представления извлекаются с помощью нейронных сетей, состоящих из слоев, наложенных друг на друга. Связанный со слоем весовой вектор параметризует преобразование, реализуемое данным слоем. Весовые векторы называются параметрами слоя и назначаются случайным образом на начальном этапе обучения. Для правильного сопоставления входных данных с их целевыми объектами, сеть должна рассчитать параметры для каждого слоя. Для определения производительности сети выходные данные должны быть сопоставлены с целевым значениями.

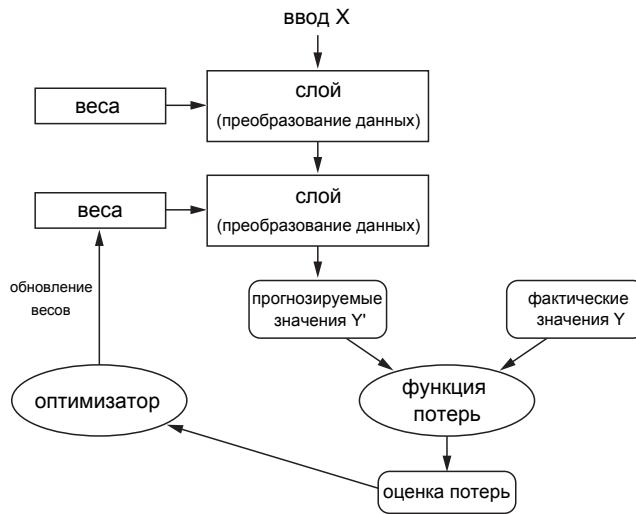


Рис. 3. Схематическое представление подхода к глубокому обучению (см. [66] Рис. 1.9)

Функция потерь оценивает разницу между целевым значением и выходом, вычисляя метрику расстояния. Эта метрика показывает, насколько хорошо сеть работает на данном конкретном экземпляре и используется в качестве обратной связи для корректировки значений весов, чтобы минимизировать функцию потерь. Это достигается оптимизатором, запускающим алгоритм обратного распространения (см. Рис. 3). В рамках

цикла обучения веса корректируются с каждым новым экземпляром обработанных данных, и функция потерь уменьшается [66].

Идея создания нейронной сети была вдохновлена нейроном в человеческом мозге. Нейрон содержит клеточное тело, или сому, входы-дendриты и выходы – аксоны. При получении электромеханических входов от соседних нейронов, соединенных dendритами, нейрон может активироваться или нет, в зависимости от силы входного сигнала. Активированный нейрон обрабатывает сигнал в теле (*соме*) и затем передает его по аксону к dendритам ближайших нейронов. Поэтому нейрон может находиться в двух состояниях-либо он срабатывает, либо нет, то есть он может выполнять бинарные операции [67].

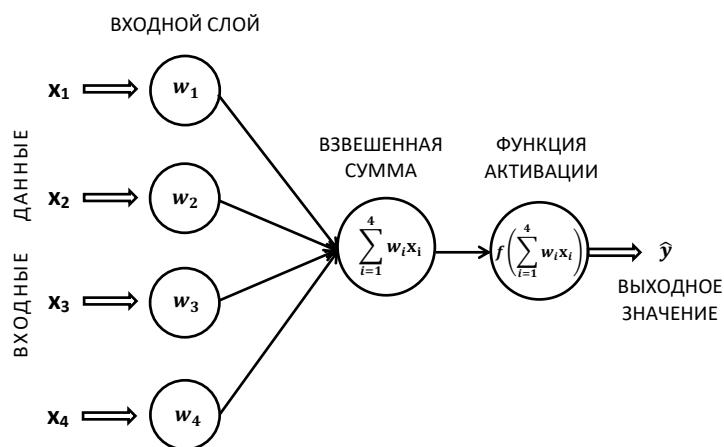


Рис. 4. Схематическое представление искусственной нейронной сети. Рис. 10.3 [67]

Несмотря на то, что идея искусственной нейронной сети была заимствована из нейробиологии, сети глубокого обучения не являются моделями мозга, поскольку нет никаких доказательств того, что нейрофизиологические и нейробиологические механизмы, ответственные за обработку данных в нашем мозге, напоминают алгоритмы,

используемые в моделях глубокого обучения. Глубокое обучение – всего лишь математическая основа для выявления характеристик и представлений из данных [66].

Предположим, что наши данные хранятся в виде многомерной матрицы проектирования, содержащей точки данных и связанные с ними метки классов. Базовая искусственная нейронная сеть выполняет взвешенное суммирование входных данных (точки данных x_1, x_2, x_3 на Рис. 4). Четвертый вход-смещение, добавляемое к расчетной матрице (постоянное значение 1). Каждая точка данных x_i связана с нейроном через весовой вектор $w = (w_1, w_2, \dots, w_n)$. Выходной узел передает взвешенную сумму через функцию активации f , чтобы получить выходное значение $f(\sum_{i=1}^n w_i x_i)$.

Широко используются следующие функции активации [67]:

- Шаговая функция: $f(z) = \begin{cases} 1, & \text{если } z > 0 \\ 0, & \text{в противном случае} \end{cases}$
- Сигмоидная функция: $f(z) = \frac{1}{1+e^{-z}}$.
- Гиперболический тангенс: $f(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$.
- усеченное линейное преобразование (Rectified Linear Unit (ReLU)): $f(z) = \max(0, z)$
- Неплотный ReLU: $f(z) = \begin{cases} z, & \text{если } z \geq 0 \\ \alpha \times z, & \text{в противном случае} \end{cases}$.
- Экспоненциальные линейные преобразования (Exponential Linear Units (ELUs)): $f(z) = \begin{cases} z, & \text{если } z \geq 0 \\ \alpha \times (e^z - 1), & \text{в противном случае} \end{cases}$ где $\alpha = \text{const}$

Основной целью сетевого обучения является изучение вектора весов w при заданном входном векторе $x = (x_1, x_2, \dots, x_n)$ и фактическом значении на выходе y . Это достигается с помощью обратного распространения, которое включает в себя оптимизацию функции потерь методом градиентного спуска:

$$L(y, \hat{y}) = (y - \hat{y})^2,$$

где $\hat{y} = f(z) = f(\sum_{i=1}^n w_i x_i)$ – сеть прогнозирования y , $z = \sum_{i=1}^n w_i x_i$, а f – функция активации (см. Рис. 4 для случая $n = 4$).

Сначала вычислим первую частичную производную функции потерь по отношению к выходу сети \hat{y} :

$$\frac{\partial L}{\partial \hat{y}} = \frac{\partial}{\partial \hat{y}} (y - \hat{y})^2 = -2(y - \hat{y}).$$

Затем вычислить $\frac{\partial L}{\partial z}$ как функцию от $\frac{\partial L}{\partial \hat{y}}$. Вид этой производной зависит от математического представления функции активации $f(z)$. Мы рассматриваем простейший случай функций *sigm*, *tanh* или *ReLU* чтобы производные также были элементарными функциями. Применив цепное правило мы получим:

$$\frac{\partial L}{\partial z} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} = \frac{\partial L}{\partial \hat{y}} \frac{\partial f}{\partial z} = f'(z) \frac{\partial L}{\partial \hat{y}}$$

В случае сигмоидной активации $f(z) = \frac{1}{1+e^{-z}}$ мы получим: $f'(z) = f(z)(1 - f(z))$.

Далее, используя производные $\frac{\partial L}{\partial z}$, мы применяем цепное правило для вычисления частных производных весов:

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial z} \frac{\partial z}{\partial w_i} = \frac{\partial L}{\partial z} \frac{\partial}{\partial w_i} \left(\sum_{i=1}^n w_i x_i \right) = x_i \frac{\partial L}{\partial z}$$

Наконец, мы вычисляем производные функции потерь по отношению к входным активациям $\frac{\partial L}{\partial z}$:

$$\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial z} \frac{\partial z}{\partial x_i} \left(\sum_{i=1}^n w_i x_i \right) = w_i \frac{\partial L}{\partial z}.$$

Этот подход может быть расширен для нескольких слоев. В этом случае выходы предыдущего слоя используются в качестве входных активаций x_i [68].

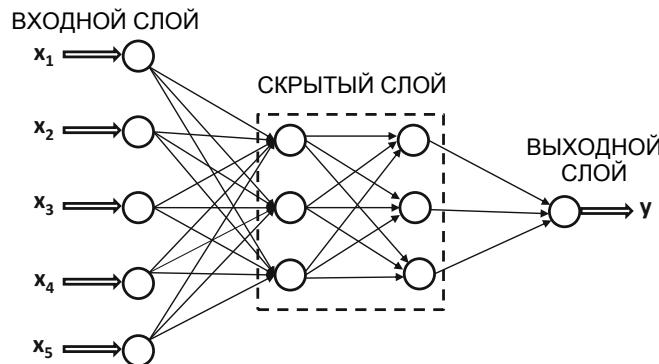


Рис. 5. Базовая архитектура сети прямого распространения с двумя скрытыми слоями и одним выходным слоем. См. Рис. 1.11 (а) [69].

прямого распространения (*feedforward network*), где узлы в слое i связаны только с узлами в слое $i + 1$.

Рекуррентные нейронные сети – те же сети прямого распространения, но только со связями, идущими обратно к входам (обратные связи) [67]. Архитектура прямого распространения полностью определена, и функция потерь оптимизируется в выходном

нейронные сети, состоящие из более чем одного вычислительного слоя, известны как многослойные сети. Слои между входным и выходным слоями называются так называемыми "скрытыми слоями", поскольку они скрывают вычисления от пользователя [69].

Наиболее часто используемая многослойная сеть – это сеть

слое. Нейроны смещения могут присутствовать как в скрытом, так и в выходном слоях. Например, на Рис. 5 представлена базовая архитектура сети прямого распространения с двумя скрытыми слоями и одним выходным слоем. Сеть содержит три слоя (обычно входной слой не учитывается). Количество единиц измерения в каждом слое определяет размерность сети.

Обозначим через $W_1[d \times p_1]$ весовую матрицу связей между входным слоем d нейронов и первым скрытым слоем, содержащим нейроны p_1 , а весовую матрицу между r -ым скрытым слоем (нейронами p_r) и $r + 1$ -ым скрытым слоем (нейронами p_{r+1}) как $W_r[p_r \times p_{r+1}]$. Итоговая матрица между k -ым скрытым слоем (p_k нейронов) и выходным слоем (o нейронов) будет $W_r[p_k \times o]$. Учитывая функцию активации f , преобразование вектора $X = (x_1, x_2, \dots, x_n)$ в выходной вектор \hat{X} можно формально записать в виде [69]:

- $h_1 = f(W_1^T X)$ (от входного слоя до первого скрытого слоя)
- $h_{p+1} = f(W_{p+1}^T h_p) \forall p = 1, \dots, k - 1$ (между скрытыми слоями)
- $\hat{X} = f(W_{k+1}^T h_k)$ (от последнего скрытого слоя до выходного слоя)

Эти уравнения справедливы только для многослойных сетей прямого распространения и не учитывают другие нетрадиционные архитектуры [69].

Обучение однослойной сети упрощается тем, что функция потерь является функцией весов, что существенно упрощает вычисление ее градиента. В многослойных сетях функция потерь представляет собой композицию функций весов от каждого слоя, и ее градиент может быть вычислен с помощью алгоритма обратного распространения, включающего динамическое программирование и цепное правило дифференцирования. Алгоритм обратного распространения состоит из прямой и обратной фазы. Во время прямой фазы данные из тренировочного набора поступают на входы сети, и соответствующие веса вычисляются для всех слоев сети. Величина на выходе сравнивается с целевым значением и вычисляется производная от функции потерь. Во время обратной фазы производная функции потерь вычисляется для всех слоев используя различные веса и весь процесс происходит в обратном направлении, начиная с выходного слоя. Этот двухэтапный процесс тренировки с целью получения весов для всех узлов, является вычислительно сложным, и повторяется в течении нескольких эпох.

Сеть глубокого обучения может:

- *автоматически выявлять произвольные зависимости между входами и выходами,*

- поддерживать нескольких входов и выходов.
- сверточные нейронные сети (*Convolutional Neural Networks (CNN)*) способны выявлять характеристики и концепции из данных,
- сети долговременной и кратковременной памяти (*Long-Short Term Memory Networks (LSTM)*) могут выявлять временные зависимости в данных,
- гибридные модели включают в себя возможности различной сетевой архитектуры для более эффективного прогнозирования [70].

Сверточные Нейронные Сети (*Convolutional Neural Networks (CNNs)*)

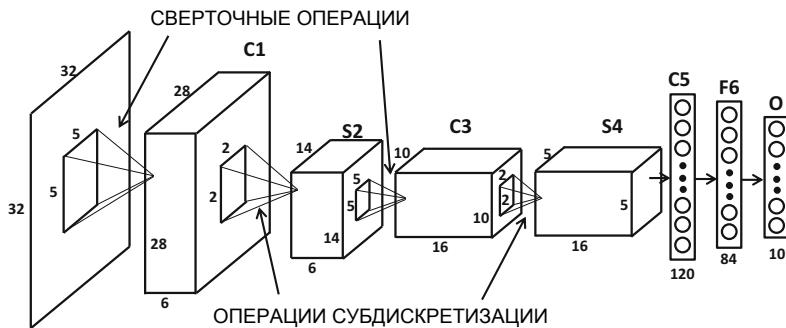


Рис. 6. Пример сверточной нейронной сети (LeNet-5). См. Рис. 1.18 [69].

состоят из трехмерных слоев, которые способны кодировать как пространственные, так и глубинные представления, в зависимости от количества признаков.

Архитектура CNN отражает организацию цветовых каналов RGB (*Red Green Blue*) во входном слое (*свертка*), и поэтому CNN широко используется для классификации изображений. Скрытые слои (субдискретизирующие слои или слои подвыборки) CNN представляют примитивные формы в нижних слоях, постепенно продвигаясь к представлению более сложных форм, таких как петли, в более высоких слоях. В случае ввода оттенков серого входной слой имеет глубину 1, но более высокие слои сохраняют свою трехмерную структуру. Слой свертки получил свое название в честь операции свертки, которая использует фильтр для отображения активаций от одного слоя к другому. Во время свертки используется 3D-фильтр весов, и выделение пространственной области в слое, прошедшем через функцию активации (например, ReLU), определяет значение скрытого состояния следующего слоя. В следующем слое активация сохраняет свои пространственные соотношения с предыдущим слоем. Поскольку любая активация в каждом скрытом слое является функцией ограниченной

Сверточные нейронные сети (*Convolutional Neural Networks (CNNs)*) основаны на концепции обработки информации в зрительной коре головного мозга. CNN являются наиболее успешным из всех подходов к глубокому обучению и

области в предыдущем слое, сверточная сеть связана слабо. Для сжатия остаточных пространственных величин, поступающих из предыдущих слоев, добавляется дополнительный слой подвыборки, усредняющий значения локальных областей размером 2×2 (операция подвыборки). Типичная сверточная сеть из 6 слоев представлена на Рис. 6. изначально CNN были разработаны для работы с двумерными картинками и приобрели популярность в применении к распознаванию изображений, локализации объектов и обработки текста [69]. В этой работе мы демонстрируем применение CNN для прогнозирования временных рядов.

Сети долговременной кратковременной памяти (*Long-Short Term Memory Networks (LSTM)*)

Временные ряды состоят из последовательно зависимых значений, т. е. значения соседних временных меток тесно взаимосвязаны. Чтобы избежать потери информации,

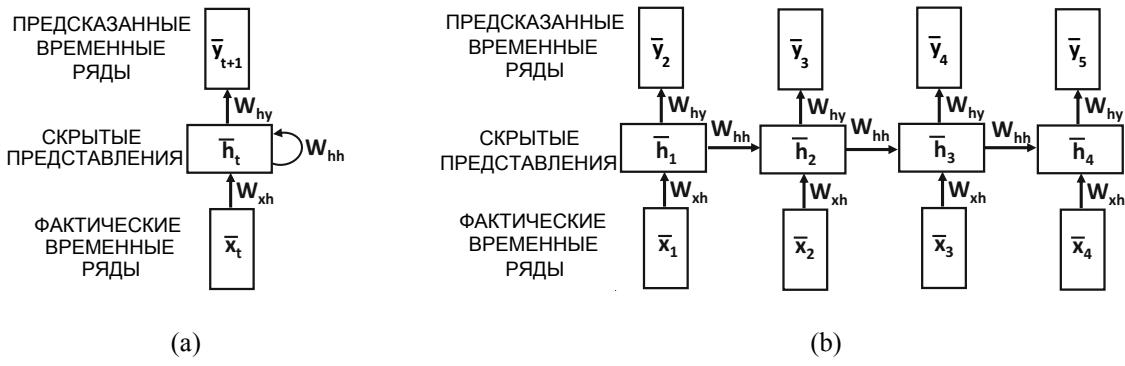


Рис. 7. Рекуррентная нейронная сеть (а) и ее многоуровневое представление во времени (б) (см. Рис. 7.2, [69]).

значения во временных рядах не должны рассматриваться как независимые переменные. Другими словами, сеть должна иметь возможность:

- *Приема и обработки входных данных в том же порядке, в каком они присутствуют в последовательности.*
- *Обработка входных данных на каждой временной метке по аналогии с предыдущими входными данными [69].*

Эта функциональность достигается рекуррентными нейронными сетями (*Recurrent Neural Networks (RNN)*), которые создают взаимно однозначное соответствие между слоями сети и временными метками во временных рядах. Терминология указывает на архитектуру сети, в которой каждый слой использует один и тот же набор фиксированных параметров для обеспечения сходства при моделировании каждой временной метки. Эта однородная послойная структура повторяется во времени, т. е.

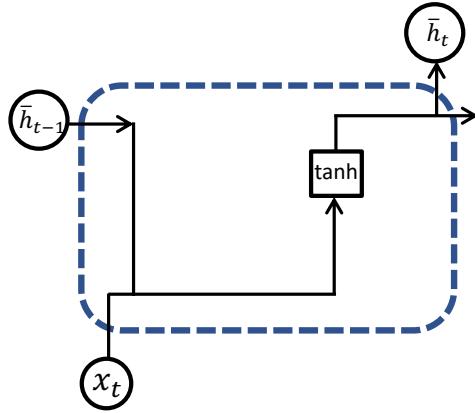


Рис. 8. Структурная схема сетевого блока RNN.
См. Рис. 3а [41].

рекуррентно. Другими словами, RNN, в отличие от других сетей обратной связи, принимают в качестве входных данных не только имеющиеся в настоящее время данные, но и данные, которые были обработаны ранее. Этот подход к распределению времени в структуре обратной связи RNN позволяет сетям преобразовывать последовательность входов в последовательность выходов, что делает их подходящим инструментом для последовательного обучения (например, прогнозирования временных рядов, машинного перевода и т. д.)) [69].

Схематическое представление архитектуры RNN приведено на Рис. 7(а). Наличие цикла в структуре приводит к изменению скрытого состояния в сети после ввода каждого нового экземпляра данных с временной меткой. Этот цикл может быть развернут во временную многоуровневую сеть с другим узлом для скрытого состояния в каждой временной метке, а цикл преобразуется в сеть обратной связи (см. Рис. 7(б) и Рис. 8). Здесь x_t , $\dim(x_t) = d$ – это входной вектор в момент времени t , \bar{h}_t , $\dim(\bar{h}_t) = p$ – скрытое состояние в момент времени t , и \bar{y}_{t+1} , $\dim(\bar{y}_{t+1}) = d$ – выходной вектор, т.е. прогноз на следующий момент времени $t + 1$. Скрытое состояние в момент времени t задается как функция входного вектора в момент времени t и скрытого вектора в момент времени $(t - 1)$, т. е. $\bar{h}_t = f(\bar{h}_{t-1}, x_t)$. Функция f выражается в терминах весовых матриц и активационных функций так, что на каждой временной отметке используются одни и те же веса. Функция вывода, $\bar{y}_{t+1} = g(\bar{h}_t)$ выдает выходные значения из скрытых состояний. Если мы определим скрытую входную матрицу как W_{xh} , $\dim(W_{xh}) = p \times d$, скрытую матрицу как W_{hh} , $\dim(W_{hh}) = p \times p$ и скрытую выходную матрицу как W_{hy} , $\dim(W_{hy}) = d \times p$, то условия для выходов будут следующими:

$$\begin{aligned}\bar{h}_t &= \tanh(W_{xh}\bar{x}_t + W_{hh}\bar{h}_{t-1}) \\ \bar{y}_{t+1} &= W_{hy}\bar{h}_t\end{aligned}$$

В самой первой временной метке \bar{h}_{t-1} инициализируется нулями, так как на первом временном шаге ряда нет входных данных из скрытого слоя. Хотя скрытые состояния

изменяются на каждой временной метке, весовые матрицы остаются фиксированными на различных временных метках [69]. Приведенные выше математические формулировки могут быть расширены для случая нескольких слоев (например, скрытое состояние k -го слоя можно обозначить как $\bar{h}_t^{(k)}$).

В основе RNN лежит механизм обратного распространения во времени, однако, в самих RNN может возникать затухающий или взрывной рост градиента. Обратное распространение основано на цепном правиле дифференцирования сложной функции для вычисления градиента функции потерь по отношению к весам сети. В зависимости от архитектуры, с увеличением количества слоев, обновления в более ранних слоях могут быть бесконечно малыми (затухающий рост градиента) или бесконечно большими (взрывной рост градиента). Этот тип неустойчивости является прямым результатом последовательного умножения с помощью (рекуррентной) весовой матрицы на различных временных метках. Другими словами, сети, опирающиеся на мультиплексные обновления, хорошо обучаются на коротких последовательностях, т. е. имеют хорошую кратковременную память, но плохую долговременную память [69].

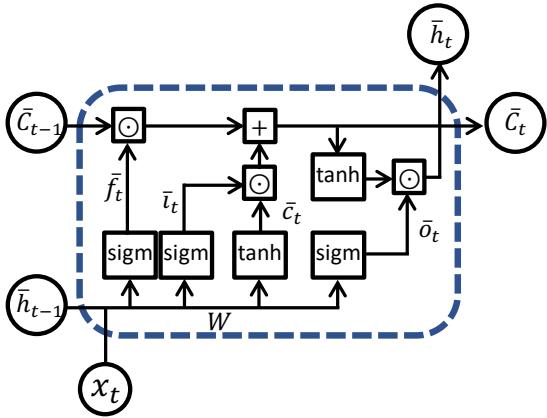


Рис. 9. Структурная схема ячейки рекуррентной нейронной сети LSTM. См. Рис. 2 [79]

Сети с долговременной - кратковременной памятью (*Long-Short Term Memory (LSTM)*) решают проблему затухающего или взрывного роста градиента путем внесения небольших изменений в рекуррентное уравнение для скрытых слоев. Предположим, что $\bar{h}_t^{(k)}$ – это скрытое состояние k -го слоя (см. Рис. 9). LSTM имеет многослойную архитектуру RNN с одной лишь разницей в условиях рекуррентности при

распространении скрытых состояний $\bar{h}_t^{(k)}$. В LSTM вводится вектор состояния ячейки $\bar{c}_t^{(k)}$, $\dim(\bar{c}_t^{(k)}) = p$ для сохранения информации о более ранних состояниях. Матрица обновления $W^{(k)}$ в LSTM имеет размер $4p \times 2p$, т. е. умножение вектора столбца $[\bar{h}_t^{(k-1)}, \bar{h}_{t-1}^{(k)}]^T$ размера $2p$ приводит к вектору размера $4p$. Обновления основаны на четырех p -мерных векторных переменных \bar{i} (*input*), \bar{f} (*forget*), \bar{o} (*output*), и \bar{c} . Чтобы определить вектор скрытого состояния $\bar{h}_t^{(k)}$ и вектор состояния ячейки $\bar{c}_t^{(k)}$, сначала

вычисляются промежуточные переменные \bar{t} , \bar{f} , \bar{o} , и \bar{c} , чтобы получить скрытые переменные из этих промежуточных переменных.

$$\begin{array}{ll} \text{Входной вентиль:} & \begin{bmatrix} \bar{t} \\ \bar{f} \\ \bar{o} \\ \bar{c} \end{bmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^{(k)} \begin{bmatrix} \bar{h}_t^{(k-1)} \\ \bar{h}_{t-1}^{(k)} \end{bmatrix} \begin{array}{l} \text{Инициализация} \\ \text{промежуточных} \\ \text{значений} \end{array} \\ \text{Вентиль забывания:} & \\ \text{Выходной вентиль:} & \\ \text{Новый С-вентиль} & \end{array} \quad (1)$$

$$\bar{C}_t^{(k)} = \bar{f} \odot \bar{c}_{t-1}^{(k)} + \bar{t} \odot \bar{c} \quad \begin{array}{l} \text{Выборочное забвение и добавление} \\ \text{в долгосрочную память} \end{array} \quad (2)$$

$$\bar{h}_t^{(k)} = \bar{o} \odot \tanh(\bar{C}_t^{(k)}) \quad \begin{array}{l} \text{Выборочная утечка из долгосрочной памяти} \\ \text{в скрытое состояние} \end{array} \quad (3)$$

“ \odot ” является поэлементное произведение векторов и “сигм”, является сигмоидная функция. В случае $k = 1$ (первый слой) $\bar{h}_t^{(k-1)} = \bar{x}_t$ и $\dim(W^{(k)}) = 4p \times (p + d)$.

В начале обучения промежуточные переменные векторы \bar{t} , \bar{f} , \bar{o} , и \bar{c} инициализируют свои величины в интервале $(0,1)$. Первые три из них, т. е. \bar{t} , \bar{f} , и \bar{o} , называются входным вентилем, вентилем забывания и выходным вентилем. Вентиля концептуально рассматриваются как булевые переменные, хотя они непрерывны на интервале $(0,1)$. Это сделано для того чтобы обеспечить дифференцируемость для обновления градиента. Вектор \bar{c} используется для инкрементирования состояний ячеек и содержит значения p , изменяющиеся в интервале $[-1, +1]$ (из-за функции \tanh). Эти три вектора определяют: а) добавлять ли состояние ячейки, б) забывать ли состояние ячейки или с) допускать ли утечку в скрытое состояние из состояния ячейки. Вектор \bar{c} хранит недавно обновленное содержимое состояния ячейки, а входные и выходные элементы регулируют степень изменений в состоянии ячейки. Все четыре промежуточных вектора переменных \bar{t} , \bar{f} , \bar{o} , и \bar{c} вычисляются с использованием весовых матриц $W^{(k)}$ для k -го слоя в (1).

Уравнение (2) состоит из двух членов. Первый член, $\bar{f} \odot \bar{c}_{t-1}^{(k)}$ использует значения p в векторе \bar{f} для принятия решений о том, какие состояния p -ячейки, поступающие из предыдущих временных шагов, будут обнулены. Он также принимает значения p из вектора \bar{t} , чтобы решить, будут ли соответствующие компоненты из вектора \bar{c} добавлены в каждое из состояний ячейки. Аддитивный характер обновлений предотвращает затухающий рост градиента. Таким образом, вектор ячейки \bar{c} обновляет долговременную память, и элементы забвения и ввода решают, необходимо ли сбросить

состояния ячеек с предыдущей временной метки и забыть прошлое; и следует ли увеличивать состояния ячеек с предыдущей временной метки, чтобы включить новую информацию в долговременную память. На последнем этапе скрытые состояния $\bar{h}_t^{(k)}$ обновляются и уравнение (3) выполняет копирование значений из каждого состояния r - ячейки в каждое из r -скрытых состояний, в зависимости от того, является ли выходной элемент \bar{o} 1 или 0. Иными словами, (3) описывает утечку информации из $\bar{c}_t^{(k)}$ в $\bar{h}_t^{(k)}$. Поскольку \bar{o} и другие переменные непрерывны, происходит только частичное стробирование, причем только часть сигнала копируется из каждого состояния ячейки в соответствующее скрытое состояние. Для обеспечения медленного затухания градиентных потоков, смещения вентиля забывания инициализируются большими значениями. Выражение скрытых состояний в терминах состояний ячеек с помощью уравнения (3) обеспечивает простую операцию вывода, что облегчает утечку градиентов в скрытые состояния. Все эти улучшения приводят к лучшей организации градиентных потоков в LSTM по сравнению с RNN [69].

ПОКАЗАТЕЛИ ЭФФЕКТИВНОСТИ

Для оценки эффективности предложенного в данной работе подхода к прогнозированию используются следующие стандартные метрики ошибок:

Средняя абсолютная процентная погрешность (*Mean Absolute Percentage Error (MAPE)*)

Средняя абсолютная процентная погрешность (MAPE) определяется как:

$$MAPE = \frac{100\%}{N} \sum_{n=1}^N \left| \frac{x_n - \hat{x}_n}{x_n} \right|$$

где N - количество выборок. x_n является фактическим значением и \hat{x}_n - прогнозируемым значение.

Средняя относительная погрешность (*Mean Relative Error (MRE)*)

Средняя относительная погрешность (MRE) определяется как:

$$MRE = 100 \times \frac{1}{N} \sum_{n=1}^N \frac{|\hat{x}_n - x_n|}{\bar{x}}$$

где \hat{x}_n и x_n – прогнозируемый и фактический спрос в час h соответственно, \bar{x} – средняя нагрузка/цена за день а N – количество прогнозируемых часов.

Средняя абсолютная погрешность (Mean Absolute Error (MAE))

Средняя абсолютная погрешность (MAE) определяется как:

$$MAE = \frac{1}{N} \sum_{n=1}^N |\hat{x}_n - x_n|$$

где \hat{x}_n и x_n – прогнозируемый и фактический спрос в час h соответственно \bar{x} – средняя нагрузка/цена за день, а N – количество прогнозируемых часов.

Среднеквадратическая погрешность (Root Mean Square Error (RMSE))

Среднеквадратическая погрешность (RMSE) определяется как:

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^N (\hat{x}_n - x_n)^2}$$

где \hat{x}_n и x_n – прогнозируемый и фактический спрос в час h соответственно, \bar{x} – средняя нагрузка/цена за день, а N – количество прогнозируемых часов.

ПРОГНОЗИРОВАНИЕ ЭЛЕКТРИЧЕСКОЙ НАГРУЗКИ В СЕТЯХ LSTM И CNN

В этой работе мы применяем сети глубокого обучения LSTM и CNN для прогнозирования электрической нагрузки. Обе сети состоят из последовательности скрытых слоев. LSTM сеть (см. Рис. 10) содержит два скрытых слоя с блоками LSTM в 500 и 700 нейронов и плотный слой (слой с каждым входным нейроном, соединенным с



Рис. 10. Структурное представление сети глубокого обучения LSTM.

выходным нейроном). Плотный слой имеет линейную активацию, т. е. он выполняет линейную операцию над входным вектором слоя, например $f(x) = x$. В качестве функции активации для скрытых слоев LSTM используется усеченное линейное



Рис. 11. Структурное представление сети глубокого обучения CNN.

преобразование (ReLU). Экспериментируя с архитектурой, мы заметили, что добавление большего количества слоев не на много улучшает точность прогнозирования. Это может быть обусловлено переобучением модели [32].

Метод основанный на CNN (Рис. 11) ставит целью нахождение соотношения между историческими данными и будущим значением. CNN состоит из входного слоя, который принимает информацию о прошлой электрической нагрузке, а выходной слой выдает собой прогнозируемую будущую нагрузку. Поиск характеристик происходит в первом слое свертки и следующим субдискретизирующем слоем. Сверточный слой включает в себя 15 различных окон (фильтров), каждое окно (сверточное ядро) имеет размер 2. Свертываясь с входным сигналом слоя по одному измерению, это ядро приводит к выходному тензору. Последовательные временные ряды преобразовываются в образцы для тренировки CNN таким образом, что на входной слой сети поступает нагрузка от предшествующих 5 дней. Входной слой применяет сверточного ядра к этим данным для получения карты распределения характеристик. Извлеченная карта распределения характеристик затем пропускается через функцию активации ReLU.

Следующим идет субдискретизирующий слой (слой подвыборки), который выполняет понижающую дискретизацию, чтобы уменьшить размерность карты характеристик, передаваемой в последующие слои. Несмотря на существенное понижение размерности в карте характеристик, преобразованной из предыдущего слоя, она все еще содержит информацию для изучения важных зависимостей. Мы установили размер максимального окна субдискретизации равным 2.

Чтобы выполнить прогнозирование, мы сглаживаем выходные данные из слоя объединения, чтобы создать полностью связанную структуру между объединением и выходным слоем. Затем значения переносятся в два плотных слоя, один из которых содержит 1500 нейронов с активацией ReLU, а другой-линейный слой с 24 нейронами, т. е. равен размерности выходного сигнала.

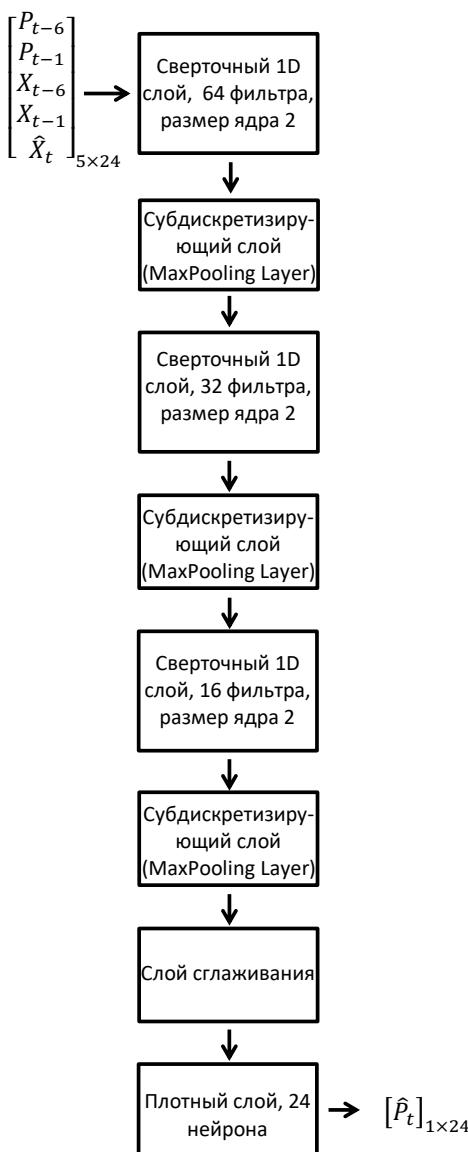
Обе сети в качестве оптимизатора используют стохастический алгоритм градиентного спуска (Adam) с среднеквадратичной погрешностью (MSE) в качестве функции потерь.

ГЛУБОКАЯ CNN ДЛЯ ПРОГНОЗИРОВАНИЯ ЦЕН НА ЭЛЕКТОЭНЕРГИЮ

Идея многослойного CNN для прогнозирования цен была вдохновлена работой [6], где автор рассматривает трехслойную нейронную сеть обратного распространения, принимающую в качестве входных данных временные индексы, прогнозируемую нагрузку и историческую информацию о фактических ценах для прогнозирования будущего значения цены. В данной работе мы используем глубокую сверточную нейронную сеть с тремя 1D сверточными слоями, тремя промежуточными слоями субдискретизации и плотным слоем на выходе (см. Рис. 12). На выбор CNN влияет тот факт, что данные о ценах на электроэнергию и электрической нагрузке коррелируют, хотя эта связь и является нелинейной [71]. На цену влияют различные внешние факторы, в том числе колебания потребления конечных потребителей, сезонность, погодные условия и другие социальные, экономические и политические факторы, а также динамика рынка, что приводит к высокой волатильности ценовых данных. Большинство из этих факторов внутренне закодированы в данных о нагрузке и исторических ценах. Поскольку CNN способны выявлять сложные отношения, мы ожидаем, что CNN с многослойной архитектурой сможет выяснить точное отображение между данными о нагрузке и цене. Насколько нам известно, основанный на глубоком обучении подход CNN для прогнозирования цены по значениям нагрузки в литературе еще не рассматривался. Отличие нашего подхода от подхода, предложенного в [6] заключается в том, что мы используем глубокий CNN вместо сети обратного распространения из двух скрытых слоев. Наш алгоритм принимает меньше параметров, чем алгоритм из [6] (5 против 9), чтобы избежать переобучения и позволить скрытым слоям CNN выводить зависимости от входных данных. Поскольку еще не существует теоретического способа определения оптимального количества слоев и нейронов в сети глубокого обучения, мы применили метод проб и ошибок для разработки архитектуры многослойного глубокого CNN для отображения данных о нагрузке и цене. Исходя из нашего предположения, первые два слоя сети извлекают общий и дневной тренд из исторических данных по нагрузке и ценам. Третий слой отвечает за учет других внешних факторов, таких как сезонные колебания и динамичность рынка. Кроме того, как указано в работе [47], сеть с более чем одним слоем но с меньшим числом нейронов способна моделировать ту же зависимость, что и однослойная сеть с большим числом нейронов. Поэтому мы решили

принять простую многослойную архитектуру, чтобы избежать чрезмерного переобучения.

Эксперименты с числом сетевых слоев показали, что три сверточных 1D-слоя для изучения временных зависимостей и три субдискретизирующих слоя (слоя подвыборки) для уменьшения размерности, приводят к оптимальной производительности модели. На входе сеть принимает фактические цены за день до интересующего нас дня P_{t-1} и за аналогичный день неделю назад P_{t-6} вместе с фактическими нагрузками за день до интересующего нас дня X_{t-1} и за аналогичный день неделю назад X_{t-6} . Пятый параметр — прогнозируемая нагрузка на интересующий нас день \hat{X}_t . А на выходе получается

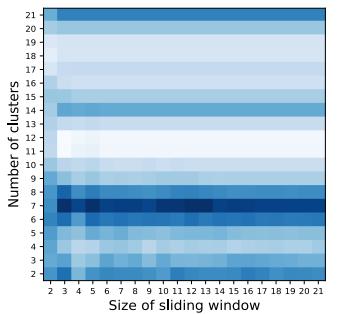


прогнозируемая цена \hat{P}_t на интересующий нас день t . Сеть была обучена по данным за один год, предшествующий дню t , чтобы выявить связь между входными $[P_{t-6}, P_{t-1}, X_{t-6}, X_{t-1}, \hat{X}_t]_{5x24}^T$ и выходными векторами $[P_t]_{1x24}$, где P_t — это фактические значения цены. Выбор нагрузки и цены за неделю до интересующего нас дня обосновывается наличием недельного тренда в паттернах изменения цены и нагрузки. Также мы рассматриваем нагрузку и цену за день до интересующего дня из-за сходства профиля нагрузки с прогнозируемым днем. Добавление большего количества исторических данных в виде цены/нагрузке две/три/четыре недели назад не оказывает существенного влияния на точность прогнозирования, но может привести к переобучению сети. В качестве функции активации для всех слоев мы выбрали усеченное линейное преобразование (ReLU), так как оно более эффективно для многослойных сетей. Значительное убывание функции потерь было обнаружено после 10-й эпохи, однако из-за периодических всплесков функции потерь во время валидации мы решили увеличить количество

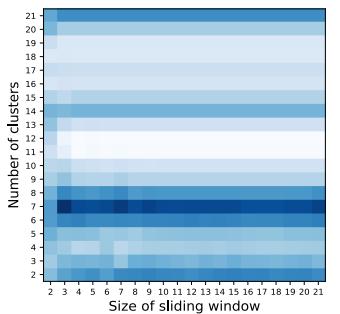
Рис. 12. Структурная схема предлагаемой сети глубокого обучения CNN для прогнозирования цен на электроэнергию.

обучающих эпох до 100. В целом предложенная архитектура продемонстрировала оптимальную точность прогнозирования на ценовых временных рядах за 2019-ый год для района Нью-Йорка (NYISO, [2]).

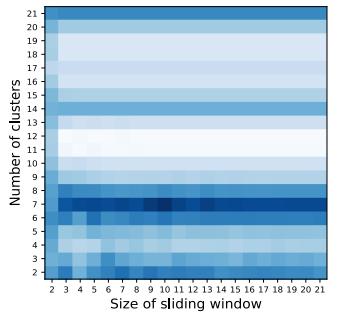
Mean absolute percentage error, K-means



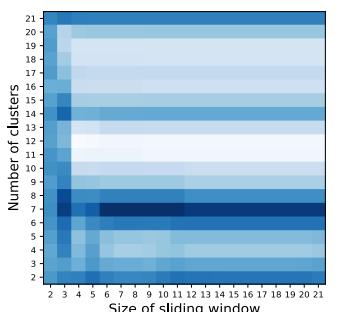
Mean absolute percentage error, K-medoids



Mean absolute percentage error, Fuzzy C-Means



Mean absolute percentage error, Hierarchical



Mean absolute percentage error, SOM

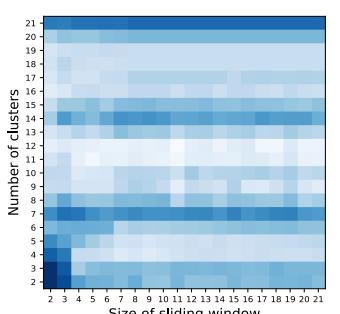


Рис. 13. Поиск по сетке наилучших параметров для каждого из пяти алгоритмов кластеризации.

ГЛАВА 3. РЕЗУЛЬТАТЫ

ПОИСК ОПТИМАЛЬНЫХ ПАРАМЕТРОВ

Основными параметрами алгоритмов кластеризации на основе PSF являются количество кластеров и размер скользящего окна. Существуют различные подходы к определению этих величин. Например, в [11] авторы применили индекс силуэта [14] для определения количества кластеров. В другой работе [18], наряду с индексом силуэта использовали индексы Данна (*Dunn*) [15] и Дэвиса-Булдина (*Davies-Bouldin*) [16]. В случае нечеткого алгоритма *c*-средних для определения числа кластеров использовалась степень нечеткости. Хотя эти индексы являются отличным инструментом для оценки количества кластеров, в некоторых случаях они могут давать неоднозначные оценки, как это было указано в [17]. Кроме того, размер окна все еще определяется с помощью 12-кратной кросс-валидации. Поэтому мы решили использовать 12-кратную кросс-валидацию для оценки оптимального количества кластеров и размера окна путем построения сетки из значений этих двух параметров. В качестве тренировочного набора мы выбрали временные ряды электрической нагрузки на 2015 год, полученные от независимого системного оператора Нью-Йорка (*New York Independent System Operator* (NYISO)) [2]. Для каждого дня в выбранном месячном интервале (*фолде*) для двух параметров перебиралось более 20 значений, вычисляя четыре меры погрешности (MAE, MAPE, MRE и RMSE), а затем усредняли по месячному фолду. Чтобы получить окончательную меру погрешности для данной пары (количество кластеров, размер окна), мы усреднили эти ежемесячные меры погрешности по всем 12 фолдам. Результаты представлены на Рис. 13.

Из полученных результатов можно сделать вывод, что наибольшая точность предсказаний (наименьшее значение

MAPE) алгоритмов k - средних и k - медоидов достигается с 7 кластерами и окном в 3 дня (значения MAPE также низки для размеров окон 5, 12 и 13. Мы заинтересованы в малых величинах параметров для обеспечения наибольшей точности предсказания при малом объеме вычислений. В случае нечеткого алгоритма C -средних MAPE принимает минимальное значение для 7 кластеров и для окон в 10 дней. В иерархической кластеризации низкое значение MAPE достигается с 6-7 днями и 7 кластерами, и, наконец, для SOM минимальное значение MAPE приходится на 2-х дневное окно и 2-3 кластера.

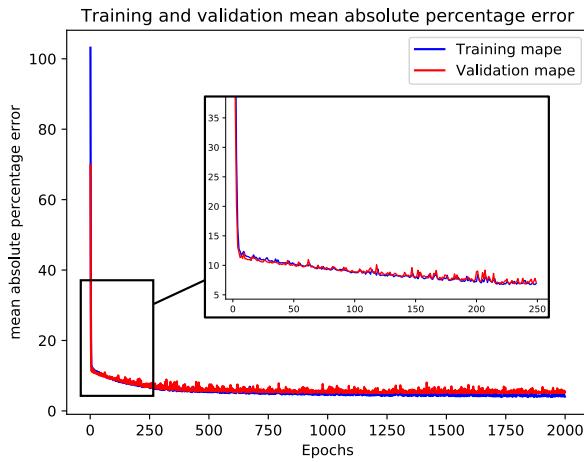


Рис. 14. Определение количества эпох для CNN

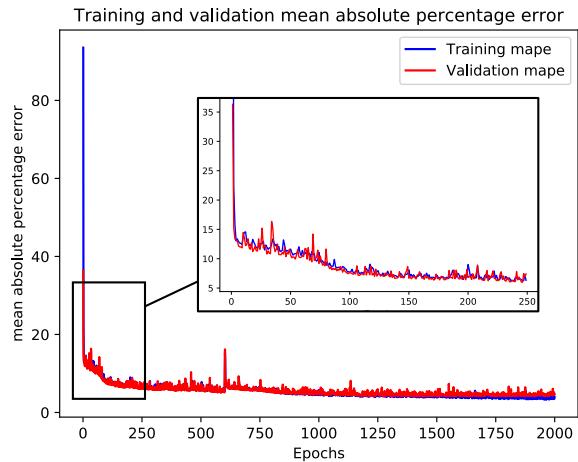


Рис. 15. Определение количества эпох для сети LSTM

Отметим, что наилучшие показатели с кластером из 7 дней, по крайней мере для 4-х из 5-и алгоритмов кластеризации можно объяснить наличием недельных паттернов во временных рядах электрической нагрузки. Другими словами, день $d - 7 / d - 14/d - 21$, то есть тот же самый день неделю/две/три назад, важен для прогнозирования нагрузки на день d . Это предположение также подтверждается наличием относительно низких значений MAPE для кластеров из 14 дней (двухнедельные паттерны) и 21 дня (трехнедельные паттерны) (более темные горизонтальные полосы на тепловых картах). Даже в случае SOM несмотря на то, что наилучшая производительность достигается для относительно небольшого числа кластеров и размеров окон, периодическое падение MAPE в кластерах 7, 14 и 21 дней также наблюдается для некоторых размеров окон (см. Рис. 13). Таким образом, алгоритмы кластеризации позволяют сделать вывод о еженедельных изменениях электрической нагрузки.

Для двух сетей глубокого обучения, то есть LSTM и CNN, выбор оптимальных параметров является более сложной задачей. Прежде всего, необходимо правильно

установить количество эпох, чтобы не допустить переобучения. Для определения оптимального количества эпох мы обучили сети LSTM и CNN на данных по нагрузке за 2014 год и протестировали на данных за 2015-ый. Рис. 14 и Рис. 15 демонстрируют, как одна из метрик ошибок, а именно средняя абсолютная процентная погрешность (MAPE), изменяется с количеством эпох.

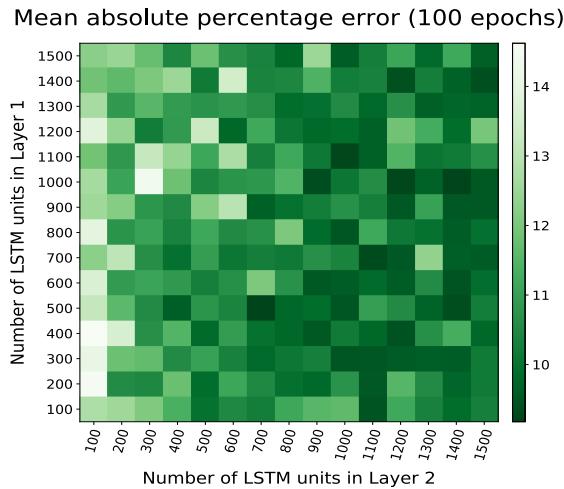


Рис. 16. Средние абсолютные процентные значения погрешности, усредненные за 100 эпох для набора валидации, рассчитаны для различных значений чисел LSTM блоков первого и второго скрытых слоев.

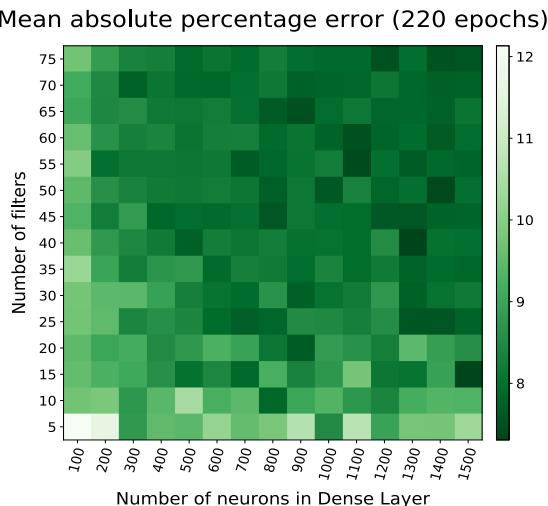


Рис. 17. Средние абсолютные процентные значения ошибок усреднялись за 220 эпох для набора валидации, рассчитанного для различных фильтров и количества нейронов в плотном слое сети CNN.

Как можно заметить, для сети LSTM,

средняя абсолютная процентная погрешность, как тренировочной, так и на валидационной выборке данных достигает плато после первых 100 эпох. Для CNN, MAPE постепенно релаксируется до отметки в 220 эпох. Поэтому нет смысла использовать более 100 эпох для LSTM и более 220 для CNN.

Для определения количества LSTM блоков в обоих скрытых слоях сети, а также количества фильтров с количеством нейронов в плотном слое для CNN, мы использовали поиск по сетке из этих параметров. Обе сети были натренированы на данных о нагрузке за 2014 год и валидированы на временных рядах нагрузки за 2015-й. Все четыре погрешности рассчитаны для оптимального числа эпох (100 для LSTM и 220 для CNN) и усреднены по эпохам. Результаты представлены на Рис. 16 и Рис. 17. Обратите внимание что аналогичные тепловые карты могут быть построены и для других погрешностей. MAPE используется просто для удобства, аналогичные выводы об оптимальных параметрах могут быть достигнуты с помощью других показателей.

Из этих тепловых карт легко сделать вывод, что LSTM достигает наименьшей погрешности с 500 блоками в первом скрытом слое и с 700 блоками во втором. Для CNN минимальное значение MAPE достигается с помощью 15 фильтров и 1500 нейронов в

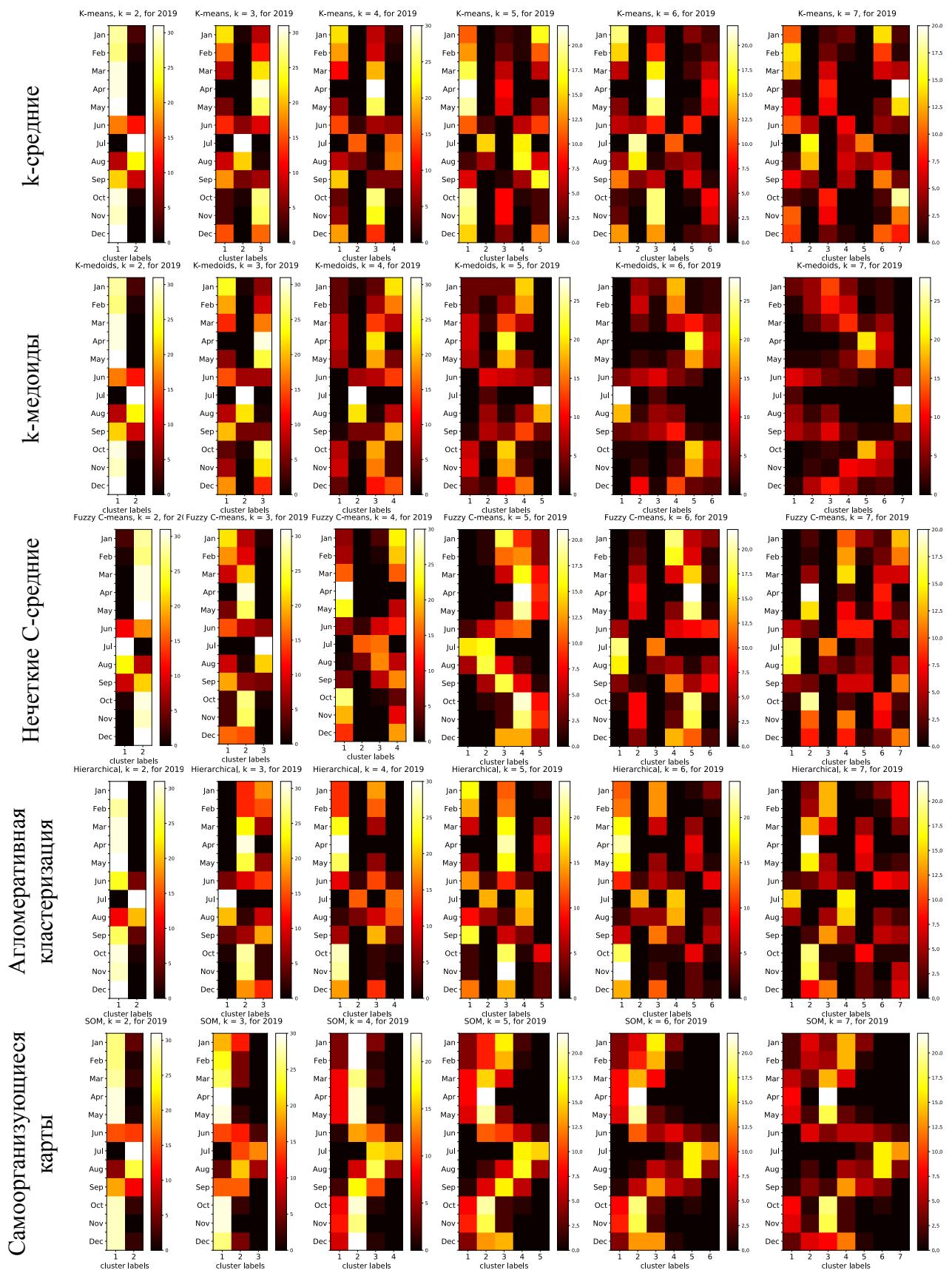
плотном слое. Эти параметры мы использовали в сетях LSTM и CNN для прогнозирования электрической нагрузки.

На данный момент не существует единого теоретического подхода к выбору количества нейронов в скрытых слоях. Общепринятый подход предполагает, что при небольшом количестве нейронов, ошибка обучения достаточно высока из-за отсутствия хорошей подгонки. В этом случае модель запоминает только наиболее общие характеристики. И наоборот, если в скрытом слое слишком много нейронов, то ошибка обучения будет низкой, но модель не будет обобщаться на новых данных из-за переобучения.

АНАЛИЗ МОДЕЛЕЙ КЛАСТЕРИЗАЦИИ

Рассматривая тепловые карты в Таб. 1, легко заметить, что существует интересная особенность в распределении дней по каждому кластеру. Для $k = 2$ кластер 1, по-видимому, отражает потребление электроэнергии в осенние, зимние и весенние месяцы, но второй кластер представляет собой летние месяцы. Аналогичная картина повторяется для всех алгоритмов кластеризации, причем оба кластера захватывают либо осень-зиму-весну, либо только лето. Для $k = 3$ кластер 2 k -средних и k -медиан в основном представляет собой летние месяцы с июня по сентябрь. Напротив него, третий кластер содержит ежедневные нагрузки от весенних и осенних месяцев. Кроме того, картина распределения дней между различными кластерами идентична для k -средних и k -медиан в случае 2 и 3 кластеров, а это означает, что центроиды, генерируемые k -средними, очень близки к фактическим точкам данных, выбранным в качестве медиан k -медианами. Нечеткие C-средние и самоорганизующиеся карты для $k = 3$ также достигают сходных распределений, помещая все летние дни в кластер 3, а иерархическая кластеризация помещает летние дни в кластер 1. В случае $k = 4$ подобная картина сохраняется, но теперь кластеры улавливают более тонкие различия между днями.

Например, алгоритм k -средних теперь имеет два кластера (2 и 4) для представления летних месяцев, однако кластер 2 больше отражает самый жаркий месяц года в Нью-Йорке (июль), в то время как дни в кластере 4 распределяются более или менее равномерно по теплым месяцам между июнем и сентябрем, слегка достигая пика в июле-августе. Для SOM кластеры 2 и 4 кодируют соответственно более прохладные и более теплые дни года, и в этом смысле их можно считать взаимными. Таким образом,



Таб. 1. Кластеризованные значения суточной электрической нагрузки, распределенные по дням для каждого месяца, для каждого из 5 алгоритмов кластеризации (на 2019 год).

независимо от индекса кластера, мы наблюдаем очень похожие закономерности в случае меньшего числа кластеров во всех методах кластеризации.

В случае большего числа кластеров различия между методами кластеризации становятся более выраженным, каждый алгоритм выявляет все более глубокие взаимосвязи между ежедневным потреблением электрической нагрузки в течение разных месяцев. Например, для $k = 5$ кластеры 2 и 4 в k -средних представляют потребление электроэнергии летом, но кластер 4 захватывает электрическую нагрузку в более теплые месяцы июля и августа. Кластер 3 представляет собой нагрузку для весны и поздней осени, в то время как в кластере 5 трудно заметить какой-либо паттерн.

Следует также отметить, что увеличение k приводит к повторению одних кластеров другими. Например, в k -медидах ($k = 6$) кластеры 5 и 6 представляют электрическую нагрузку с марта по май и с октября по ноябрь. Это может быть связано со слишком близкими границами между обнаруженными соседними кластерами, т. е. разделение не является оптимальным. Это наблюдение подчеркивает важность определения оптимального числа кластеров перед прогнозированием.

ОЦЕНКА ЭФФЕКТИВНОСТИ АЛГОРИТМОВ ПРОГНОЗИРОВАНИЯ НАГРУЗКИ

Для оценки средней точности предсказаний каждая модель тестировалась по отдельности на выборке данных по нагрузке за один год с применением 12-и кратной кросс валидации. Эти данные по электрической нагрузке были получены от независимого системного оператора Нью-Йорка (*New York Independent System Operator (NYISO)*) [2] для города Нью-Йорк. Частота измерений нагрузки в исходном наборе данных составляет 12 измерений в час (каждые пять минут). Чтобы уменьшить размер данных, мы решили сократить его до одного измерения в час, оставив только измерения, собранные в начале каждого часа, таким образом, сократив число измерений до 24x в день. На этапе предварительной обработки данных наличие тренда в данных подавляется путем нормализации данных по всем измерениям:

$$X_j = \frac{X_j}{\frac{1}{N} \sum_{i=1}^N X_j}$$

где X_j -нагрузка за j -й час суток и N -количество рассмотренных проб в сутки. В этом случае, $N = 24$, так как каждая выборка представляет собой один час дня [11], [21].

Однако для того, чтобы вычислить знаменатель, нам нужно знать данные за все 24 часа, что может накладывать дополнительные ограничения. Поэтому мы приняли подход нормализации на основе единства [12], позволяющий сопоставить все значения интервалу [0,1]:

$$X'_j = \frac{X_j - \min(X_j)}{\max(X_j) - \min(X_j)}$$

где X'_j -нормированная нагрузка для j -го часа суток, $i = 1, \dots, N$.

Еще одно отличие от классического алгоритма PSF заключается в том, что мы удалили выбросы от прогнозируемых дневных измерений. Используя в качестве критерия 1,5 межквартильного диапазона (*interquartile range* (IQR)), мы исключили из предсказания все те дни, абсолютные значения которых превышали 1,5 IQR (т. е. если они были найдены во главе аналогичной последовательности, предшествующей интересующему дню, они были удалены из среднего значения). Чтобы продемонстрировать, что выбросы могут влиять на точность прогноза, мы применили алгоритм k -средних (оригинальный LBF) для прогнозирования электрической нагрузки с выбросами и без выбросов на 2017 год (используя оптимальные параметры, найденные в предыдущем параграфе). В то время как средние значения MAE, MAPE, MRE и RMSE среднеквадратическое отклонение для данных с выбросами были 740.68, 11.78, 11.92 и 812.74 соответственно, погрешность значения незначительно снизились на данных с удаленными (MAE = 681.33, MAPE = 10.93, MRE = 11.04 и MSE = 751.65). Это наблюдение побудило нас адаптировать обнаружение выбросов для будущих анализов. В общем случае внешние границы составляют около 5% набора данных, и обычно это значения, превышающие верхнюю границу 1,5 IQR.

После удаления выбросов мы применили каждый алгоритм кластеризации для прогнозирования значений электрической нагрузки на каждый день в 2019 году. Полученные результаты приведены в Таб. 2.

Из таблицы видно, что самые низкие средние ошибки достигаются при прогнозировании с помощью LSTM. CNN также демонстрирует довольно высокую точность. Самая низкая точность достигается алгоритмом k -медиан MAPE которого превышает 9%.

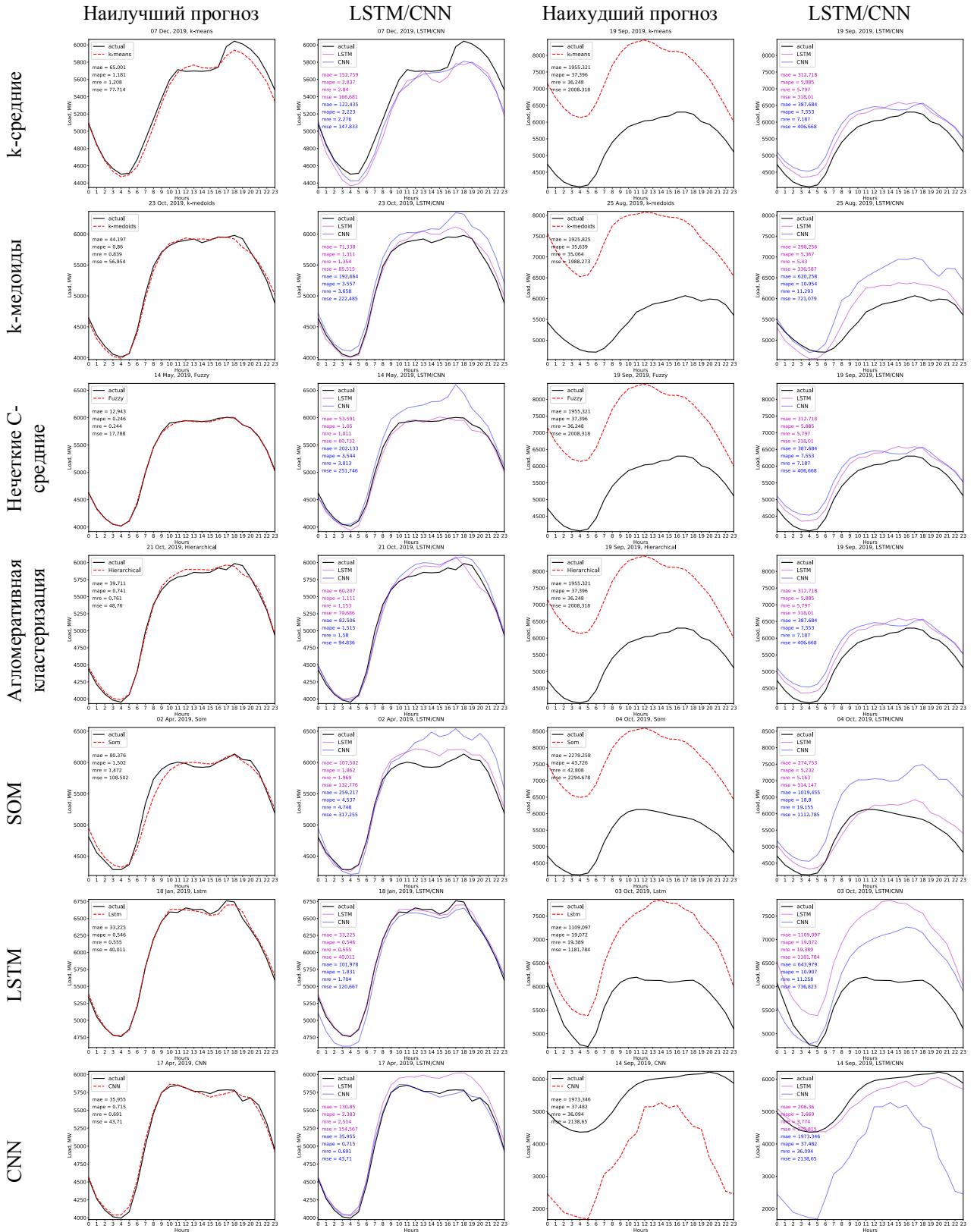
Для большинства алгоритмов сентябрь - самый трудный месяц для прогнозирования нагрузки. Это может быть объяснено изменением повседневной жизни людей, когда значительное число населения возвращается обратно после летних

каникул и школы / университеты возобновляют свою работу, что приводит к непредсказуемым пикам потребления электроэнергии. Интересно, что нагрузка в ранние весенние месяцы - март и апрель, предсказывается довольно таки легко.

В Таб. 3 приведены лучшие и худшие результаты прогнозирования для каждого алгоритма вместе с прогнозами от LSTM и CNN.

		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Среднее
k-средние	MAE	366.5	477.16	393.83	270.1	344.58	742.34	800.73	761.78	658.89	406.77	254.08	399.74	489.71
	MAPE	6.17	8.15	7.11	5.6	6.51	11.22	9.51	11.48	10.79	7.76	4.64	7.13	8.01
	MRE	6.17	8.2	7.2	5.5	6.51	11.35	9.66	11.37	10.79	7.78	4.73	7.16	8.04
	RMSE	405.41	506.39	436.08	304.59	394.86	793.3	870.92	817.4	723.63	456.89	284.27	426.92	535.06
k-медионы	MAE	470.4	583.4	461.92	380.37	467.16	692.05	843.1	773.55	758.88	486.84	335.79	573.06	568.88
	MAPE	7.87	10.15	8.34	7.45	8.7	10.35	10.24	11.54	12.28	9.17	6.17	10.03	9.36
	MRE	8	10.2	8.5	7.63	8.89	10.46	10.25	11.5	12.25	9.37	6.3	10.22	9.46
	RMSE	522.59	639.91	524.27	438.32	538.62	753.4	906.12	846.12	848.1	560.56	383.45	641.83	633.61
Нечеткие C-средние	MAE	314.58	373.33	260.5	141.73	209.59	736.83	799.34	648.96	639.28	338.99	222.36	405.49	424.25
	MAPE	5.24	6.49	4.84	2.8	3.7	11.31	9.6	9.88	10.5	6.36	4.13	7.26	6.84
	MRE	5.21	6.47	4.82	2.79	3.78	11.34	9.75	9.65	10.5	6.33	4.14	7.28	6.84
	RMSE	344.92	398.64	280.8	157.85	244.33	785.25	885.81	706.71	701.57	384.92	249.12	429.01	464.08
Агрометрическая классификация	MAE	325.72	366.67	236.45	184.98	242.29	736.83	798.69	642.33	638.29	344.59	255.37	377.3	429.13
	MAPE	5.41	6.38	4.38	3.69	4.43	11.31	9.38	9.64	10.48	6.48	4.73	6.76	6.92
	MRE	5.38	6.36	4.35	3.65	4.44	11.34	9.64	9.52	10.49	6.43	4.76	6.78	6.93
	RMSE	361.25	392.8	257.38	198.9	274.47	785.25	883.79	703.26	700.86	389.15	282.14	399.34	469.05
SOM	MAE	477.22	395.56	360.88	323.12	382.64	596.01	845.57	625.08	812.82	475.14	289.06	420.57	500.31
	MAPE	8	6.79	6.57	6.7	7.38	9.23	10.28	8.86	13.18	9.02	5.36	7.38	8.23
	MRE	8.05	6.86	6.66	6.57	7.28	9.42	10.35	8.93	13.25	8.93	5.43	7.52	8.27
	RMSE	524.91	438.3	410.26	359.05	431.54	652.36	905.77	678.44	880.36	526.05	328.17	468.49	550.31
LSTM	MAE	195.5	161.81	125.14	121.62	212.04	333.75	407.69	333.67	401.21	189.24	140.78	183.3	233.81
	MAPE	3.3	2.73	2.24	2.3	3.75	5.1	5	4.67	6.21	3.33	2.62	3.25	3.71
	MRE	3.36	2.8	2.27	2.39	3.89	5.3	5.18	4.83	6.45	3.48	2.67	3.35	3.83
	RMSE	226.17	193.29	146.86	146.81	248.84	396.16	482.98	401.92	480.11	224.12	168.18	217.24	277.72
CNN	MAE	297.94	214.14	179.45	185.19	308.45	379.31	596.96	405.64	612.88	371.35	208.59	246.41	333.86
	MAPE	4.89	3.71	3.28	3.54	5.56	6	7.34	5.71	9.87	6.92	3.92	4.36	5.42
	MRE	5.03	3.78	3.29	3.63	5.64	6.12	7.52	5.79	10.01	7	3.96	4.44	5.52
	RMSE	351.92	251.78	210.24	224.55	353.03	439.41	695.6	468.23	714.89	436	243.24	282.82	389.31

Таб. 2. Ежедневные ошибки прогнозирования электрической нагрузки каждым отдельным экспертом на 2019 год, усредненные по каждому месяцу и за год.



Таб. 3. Лучшие и худшие прогнозы каждой модели.

На Рис. 18 и Рис. 19 сопоставлены настоящие величины электрической нагрузки и прогноз на один месяц (октябрь 2019 года).

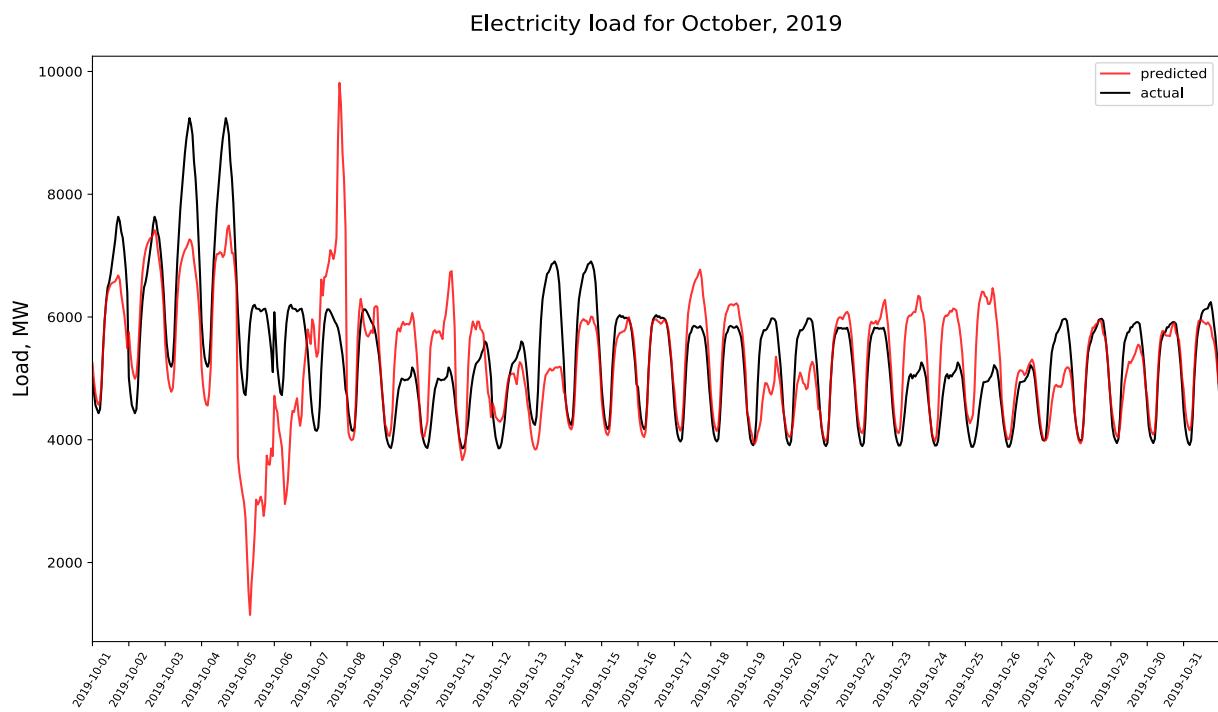


Рис. 18. Фактические и спрогнозированные CNN алгоритмом значения электрической нагрузки на октябрь 2019 года.

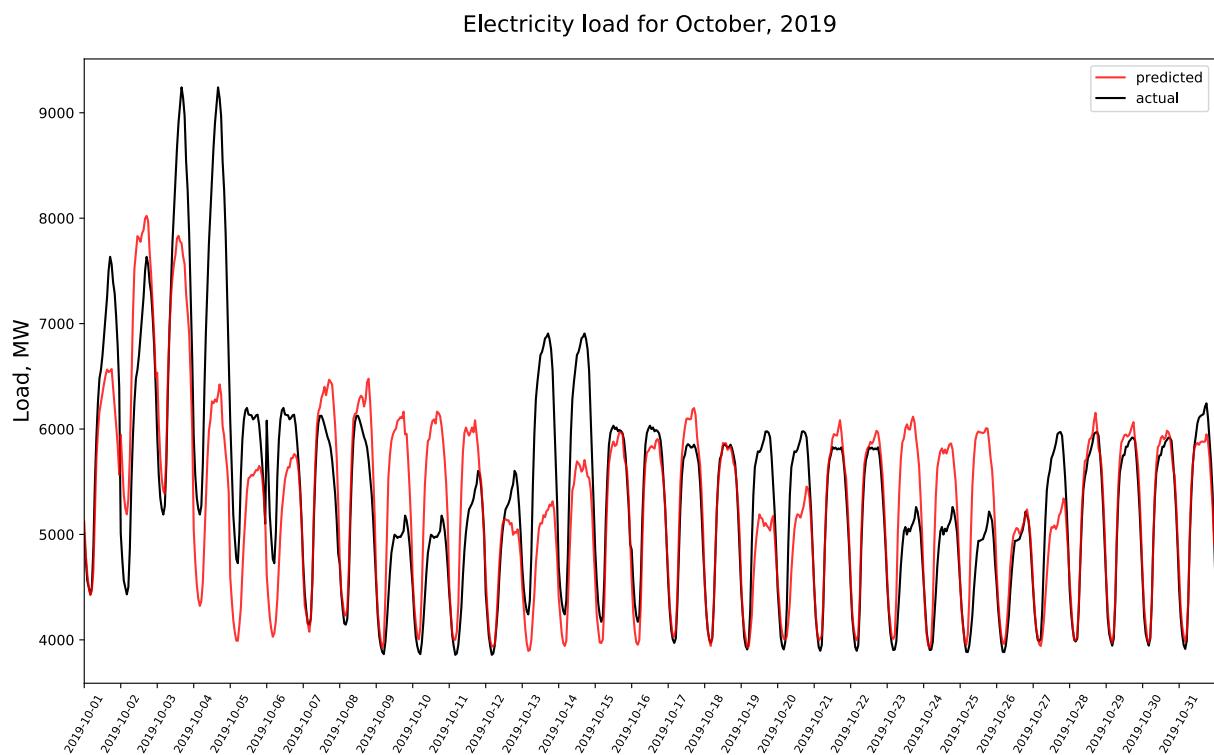


Рис. 19. Фактические и спрогнозированные LSTM алгоритмом значения электрической нагрузки на октябрь 2019 года

Из Рис. 18 и Рис. 19 нетрудно заметить, что сеть LSTM лучше подходит для обработки скачков нагрузки (произошедших 5 октября). Несмотря на неточный прогноз, LSTM удалось уловить общую тенденцию на этот день, в то время как сеть CNN не смогла даже дать приблизительный прогноз, вместо этого генерируя некоторые случайные выходные данные. Это объясняется способностью сетей LSTM улавливать внутренние временные зависимости в данных временных рядов. В среднем алгоритм на основе LSTM продемонстрировал лучшие результаты, чем любой другой метод прогнозирования, рассмотренный в данной работе.

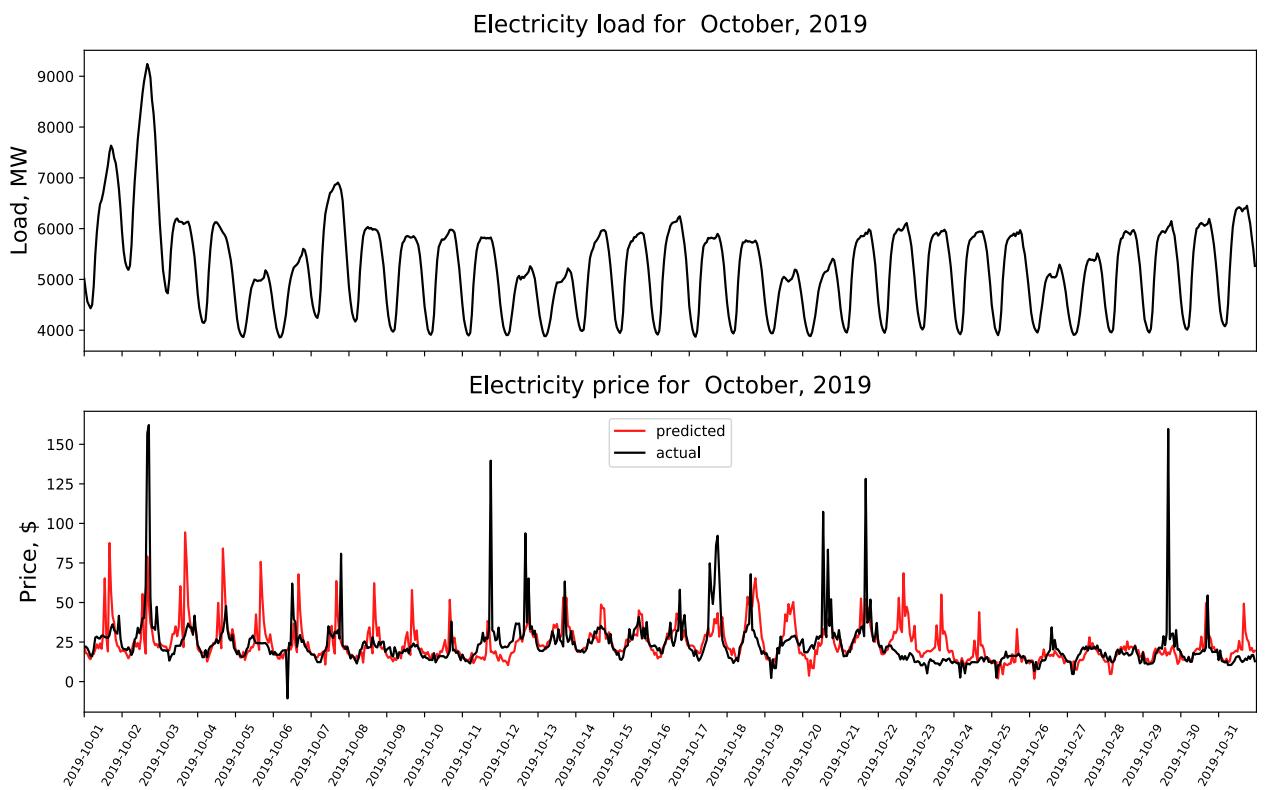


Рис. 20. Фактическая нагрузка и прогнозируемые / фактические значения цен на электроэнергию на октябрь 2019 года.

ПРОГНОЗИРОВАНИЕ ЦЕНЫ НА ЭЛЕКТРОЭНЕРГИЮ ИЗ ДАННЫХ О НАГРУЗКЕ

Прогнозирование цен на электроэнергию является сложной задачей из-за волатильности, нестационарности и неоднородности структуры ценовых временных рядов. В отличие от электрической нагрузки, данные о ценах на электроэнергию имеют менее выраженный тренд (см. Рис. 20). Кроме того, ценовые временные ряды подвержены сильному влиянию внешних факторов. Например, перегрузка передачи данных в системе может вызвать сильный скачок цен. После скачка цена обычно

возвращается к более стабильному уровню [6]. Несмотря на волатильный характер, цена электроэнергии не является случайной величиной. Из Рис. 20 также заметна периодическая структура и вариации суточного тренда ценовых рядов, коррелирующие с электрической нагрузкой. Таким образом, исходя из нагрузки и исторических ценовых данных можно выяснить скрытые паттерны и соотношения, определяющие поведение будущей цены. Знание этих скрытых зависимостей поможет более точному прогнозированию цен на электроэнергию.

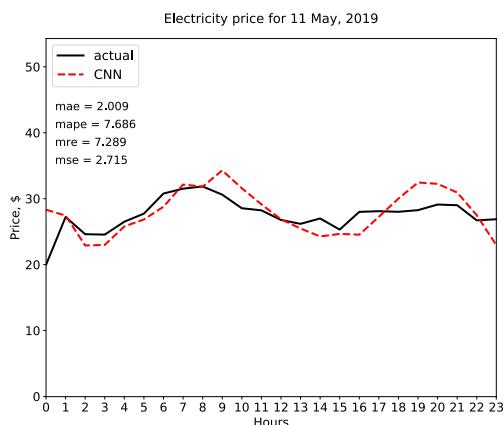


Рис. 21. Лучший прогноз цен в 2019-м году.

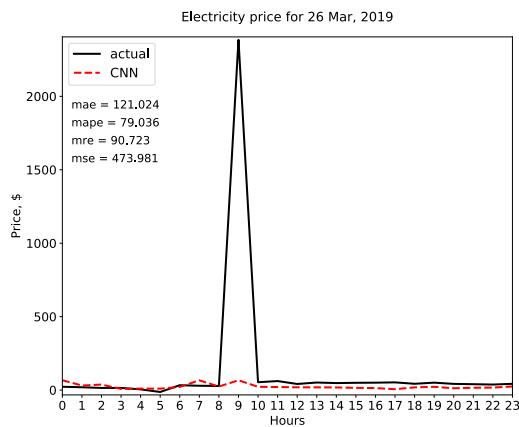


Рис. 22. Худший прогноз цен в 2019-м году.

Различные внешние факторы могут оказывать существенное влияние на нелинейную корреляцию между электрической нагрузкой и ценой. Нагрузка в основном зависит от структуры потребления, погодных условий, социальных событий (например, праздников) и сезонности. Невозможность хранения электроэнергии обуславливает организацию производства электроэнергии и влияет на колебания в нагрузке сети. В случае цены наряду с факторами, влияющими на нагрузки появляются еще и факторы, навязанные рынком, например регулирование процесса ценообразования со стороны органов власти, схемы ценообразования конкурентов, а также многие другие макро - и микроэкономические факторы. Все эти факторы способствуют высокой волатильности ценовых данных и должны учитываться для более точного сопоставления нагрузки и цен.

Мы применили сверточную нейронную сеть для выявления соответствия между электрической нагрузкой и ценой. В качестве ценовых данных мы использовали установленную маржу для данной локации (*Location Based Marginal Pricing (LBMP)*), то есть, стоимость предоставления следующего МВт нагрузки в определенном месте сети, предоставленной независимым системным оператором Нью-Йорка [2] для той же самой локации, что и данные по нагрузке (город Нью Йорк, N. Y. C.).

Тестирование основанного на CNN алгоритма прогнозирования цен на электроэнергию была проведена на данных за 2019 год. Для обучения модели были

использованы данные о нагрузке и ценах за предыдущий год. В качестве прогнозируемых значений нагрузки мы использовали выход из сети LSTM, так как он демонстрировал более высокую среднюю точность предсказаний по сравнению с остальными моделями, рассмотренными в данной работе. На этапе тестирования были входные данные для каждого дня из тестовой выборки были использованы для получения прогноза. После получения прогноза сеть была переобучена принимая во внимание соответствующие входные данные и фактическое значение прогнозируемого дня. Эта процедура известна как онлайн-обучение и позволяет генерировать прогнозы для более длинных горизонтов прогнозирования (в этом случае нам нужно использовать прогнозируемую цену на данный день вместо фактической цены, так как мы предполагаем, что фактическое значение неизвестно). В целом предложенный алгоритм позволил достичь среднего значения MAE в 15,75 балла а средняя RMSE 32.48 по годовым данным. На Рис. 22 и Рис. 21 показаны лучшие и худшие прогнозы в 2019-м году.

Как легко заметить, модель не в состоянии дать полноценный прогноз для резких скачков цен на электроэнергию. Обычно такие скачки появляются в результате временного расхождения между спросом и предложением электроэнергии. Даже расхождения в масштабе секунды могут вызвать значительные всплески в цене. К другим факторам, приводящим к ценовым скачкам, относятся нарушения в производстве электроэнергии, перегруженность, поведение субъектов рынка, манипуляции на рынке и многие другие.

Значительная волатильность ценовых данных создает трудности для точного прогнозирования будущих цен. В результате точность прогнозирования цен обычно ниже, чем при прогнозировании электрической нагрузки. Однако требования к прогнозированию цен не столь строги, как к электрической нагрузке, и некоторая степень неточности в прогнозе является вполне допустимой.

ВЫВОДЫ И РЕКОМЕНДАЦИИ

Прогнозирование нагрузки и цен на электроэнергию очень важно для индустрии и рынка электроэнергетики. На самом деле эти две проблемы взаимосвязаны: знание будущих значений нагрузки жизненно важно для планирования энергосистемы, что приводит к снижению себестоимости производства и, следовательно, к снижению цен на электроэнергию. Точный прогноз цены на электроэнергию важен для всех участников рынка: производителей энергии, трейдеров, поставщиков и потребителей. Надежный механизм прогнозирования нагрузки применяется в настройках интеллектуальной сети, где сама сеть, анализируя характерное поведение на рынке электроэнергии поставщиков и конечных пользователей, пытается оптимизировать производство и распределение электроэнергии.

В данной работе мы рассмотрели автоматизированные методы прогнозирования нагрузки и цен на электроэнергию. Первоначально мы рассматриваем традиционные алгоритмы машинного обучения, такие как алгоритм PSF и его расширения с различными методами кластеризации. Мы внесли улучшение в классический PSF, исключив из процесса прогнозирования дни, содержащие выбросы, лежащие вне диапазона, равного 1,5 межквартильного диапазона (IQR) исторических данных. Это улучшение привело к снижению значений всех четырех погрешностей и повысило точность предсказания.

Глубокое обучению является перспективной технологией для прогнозирования цен на электроэнергию. Мы исследовали две различные архитектуры глубокого обучения - сверточные нейронные сети (CNN) и сети долговременной кратковременной памяти (LSTM). В то время как CNN эффективны для выявления скрытых характеристик в входных данных, сети LSTM более подходят для извлечения временных паттернов. Тестирование обеих сетей на данных нагрузки NYISO продемонстрировало стабильную точность предсказаний сети LSTM, которая смогла обрабатывать нетривиальные паттерны, такие как пики электрической нагрузки. Основанный на LSTM подход также продемонстрировал более высокую среднюю точность прогнозирования по сравнению с другими традиционными методами. В будущем будет интересно рассмотреть гибридные модели, которые используют CNN для выявления характеристик и LSTM для прогнозирования.

Важным достижением данной исследовательской работы является разработка модели на основе CNN для поиска прямого отображения между нагрузкой и ценами на

электроэнергию. Поскольку цену очень трудно предсказать из-за ее волатильности, большинство традиционных моделей не в состоянии охватить все скрытые зависимости, приводящие к непредсказуемому поведению ценовых временных рядов. Предлагаемый подход к глубокому обучению на основе CNN способен выявить некоторые из этих зависимостей, что приводит к более или менее адекватной точности предсказания. В будущей работе было бы интересно попытаться изучить особенности, вызывающие пики ценовых данных и способы их прогнозирования. Это может быть достигнуто путем применения вейвлет-преобразования для извлечения детерминированной и шумовой составляющей и индивидуального анализа каждой компоненты. Было бы также интересно изучить эффективность более продвинутых методов глубокого обучения, таких как генеративные состязательные нейросети (*Generative Adversarial Networks* (GAN)) и глубокое обучение с подкреплением (*Deep Reinforcement Learning*) для прогнозирования временных рядов.

БИБЛИОГРАФИЯ

- [1] S. Stoft, Power system economics: designing markets for electricity, Piscataway, NJ: IEEE Press, 2002.
- [2] «ENERGY MARKET & OPERATIONAL DATA,» New York Independent System Operator, [В Интернете]. Available: <https://www.nyiso.com/energy-market-operational-data>. [Дата обращения: 20 04 2020].
- [3] F. Chollet, «Keras,» 2015. [В Интернете]. Available: <https://keras.io>. [Дата обращения: 20 04 2020].
- [4] F. Pedregosa, V. Gael, G. Alexandre, M. Vincent, T. Bertrand, G. Olivier, M. Blondel, P. Prettenhofer, W. Ron, D. Vincent, V. Jake, P. Alexandre, C. David и B. Matthieu, «Scikit-learn: Machine learning in Python.,» *Journal of Machine Learning Research*, p. 2825–2830, 2011.
- [5] T. Erkkilä, A. Lehmussola, K. Kiełczewski и Z. Dufour, «K-medoids clustering,» 2019. [В Интернете]. Available: https://github.com/scikit-learn-contrib/scikit-learn-extra/blob/master/sklearn_extra/cluster/_k_medoids.py. [Дата обращения: 20 04 2020].
- [6] D. Singhal и S. Swarup, «Electricity price forecasting using artificial neural networks,» *International Journal of Electrical Power & Energy Systems*, т. 33, pp. 550-555, 2011.
- [7] H. Hippert, C. Pedreira и R. C. Souza, Neural Networks for Short-Term Load Forecasting: A review and Evaluation, т. 16, *IEEE Transactions on Power Systems*, 2001, pp. 44-55.
- [8] D. Belik, D. Nelson и D. Olive, Use of the Karhunen- Loeve expansion to analyze hourly load requirements for a power utility, т. A78, *IEEE Power Engineering Society Winter Meeting*, 1987, pp. 225-230.
- [9] M. Meng и W. Shang, Research on Annual Electric Power Consumption Forecasting Based on Partial Least-Squares Regression, т. 2, Proceedings of International Seminar on Business and Information Management, 2008, pp. 125-127.
- [10] L.-Y. Wei, T. Chih-Hung, Y.-C. Chung, L. Kuo-Hsiung, C. Hao-En и L. Jyh-Shyan, «A Study of the Hybrid Recurrent Neural Network Model for Electricity Loads Forecasting,» *International Journal of Academic Research in Accounting, Finance and Management Sciences*, т. 7, № 2, pp. 21-29, 2017.
- [11] F. Martínez-Alvarez, A. Troncoso, J. Riquelme и J. S. Aguilar-Ruiz, LBF: A labeled-based forecasting algorithm and its application to electricity price time series,, *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM'08)*, 2008.
- [12] G. Box, G. Jenkins и G. Reinsel, Time Series Analysis: Forecasting and Control, т. 734, Hoboken, NJ: John Wiley & Sons, 2011.
- [13] F. Martínez-Alvarez, A. Troncoso, J. C. Riquelme и J. S. A. Ruiz, Energy time series forecasting based on pattern sequence similarity, *IEEE Transactions on Knowledge and Data Engineering*, 2011.
- [14] P. Rousseeuw и L. Kaufman, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley & Sons, 2009.
- [15] J. Dunn, Well-Separated Clusters and Optimal Fuzzy Partitions, т. 4, *Journal of Cybernetics and Systems*, 1974, pp. 95-104.
- [16] D. Davies и D. Bouldin, A Cluster Separation Measure, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1979, pp. 224 - 227.

- [17] C. Jin, G. Pok, H. W. Park и K. Ryu, Improved Pattern Sequence-Based Forecasting Method for Electricity Load, IEEJ Transactions on Electrical and Electronic Engineering, 2014.
- [18] W. Shen, V. Babushkin, Z. Aung и W. Woon, An ensemble model for day-ahead electricity demand time series forecasting, In Proceedings of the 4th ACM Conference on Future Energy Systems (e-Energy), 2013.
- [19] M. Mostafa, Q. Charlie, C. Peter, G. Rajit и P. H. R., «Modified pattern sequence-based forecasting for electric vehicle charging stations.» в *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2014.
- [20] Y. Fujimoto и Y. Hayashi, «Pattern sequence-based energy demand forecast using photovoltaic energy records.» в *International Conference on Renewable Energy Research and Applications (ICRERA)*, 2012.
- [21] F. Martínez-Alvarez, A. Troncoso, J. Riquelme и J. S. Aguilar-Ruiz, «Discovery of motifs to forecast outlier occurrence in time series.» в *Pattern Recognition Letters*, т. 32, p. 1652–1665, 2011.
- [22] L. Ekonomou, C. Christodoulou и V. Mladenov, «A short-term load forecasting method using artificial neural networks and wavelet analysis.» в *International Journal of Power Systems*, т. 1, pp. 64-68, 2016.
- [23] A. Reis и A. da Silva, «Feature extraction via multiresolution analysis for short-term load forecasting.» в *IEEE Transactions on Power Systems*, 2005.
- [24] I. Koprinska, M. Rana, A. Troncoso и F. Martínez-Álvarez, «Combining Pattern Sequence Similarity with neural networks for forecasting electricity demand time series.» в *Proceedings of the International Joint Conference on Neural Networks*, 2013.
- [25] G. E. Hinton, O. Simon и T. Yee-Whye, «A fast learning algorithm for deep belief nets.» *Neural computation*, т. 18, pp. 1527-54, 2006.
- [26] S. Bouktif, A. Fiaz, A. Ouni и M. Serhani, «Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches.» *Energies*, т. 11, p. 1636, 2018.
- [27] X. Qiu, Z. Le, Y. Ren, P. N. Suganthan и A. Gehan, «Ensemble deep learning for regression and timeseries forecasting.» в *IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL)*, 2014.
- [28] E. I. O. a. S. W. Busseti, «Deep learning for time series modeling, Technical report.» Stanford University, 2015.
- [29] G. Merkel, R. Povinelli и R. Brown, «Short-Term Load Forecasting of Natural Gas with Deep Neural Network Regression.» *Energies*, т. 11, 2008.
- [30] T. Kuremoto, S. Kimura, K. Kobayashi и M. Obayashi, «Time Series Forecasting Using Restricted Boltzmann Machine.» *Communications in Computer and Information Science*, т. 304, pp. 17-22, 2012.
- [31] E. Mocanu, P. Nguyen, M. Gibescu и W. Kling, «Deep Learning For Estimating Building Energy Consumption.» *Sustainable Energy, Grids and Networks*, Т. %1 из %21-10, 2016.
- [32] H. Shi, M. Xu и R. li, «Deep Learning for Household Load Forecasting – A Novel Pooling Deep RNN.» *IEEE Transactions on Smart Grid*, т. PP, pp. 1-1, 2017.
- [33] S. Hochreiter и J. Schmidhuber, «Long-short term memory.» *Neural Computation*, т. 9, № 8, pp. 1735-1780, 1997.

- [34] A. a. H. J. Gensler, B. Sick и N. Raabe, «Deep Learning for solar power forecasting — An approach using AutoEncoder and LSTM Neural Networks,» в *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016.
- [35] M. Shahzad и A. Afshin, «Short-Term Load Forecasts Using LSTM Networks,» *Energy Procedia*, т. 158, pp. 2922-2927, 2019.
- [36] D. L. Marino, K. Amarasinghe и M. Manic, «Building energy load forecasting using Deep Neural Networks,» в *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016.
- [37] W. Kong, Z. Dong, Y. Jia, D. Hill, Y. Xu и Y. Zhang, «Electric load forecasting in smart grids using Long-Short-Term-Memory based Recurrent Neural Network.,» *IEEE Transactions on Smart Grid*, т. PP, 2017.
- [38] S. Du, T. Li, X. Gong, Y. Yang и S.-J. Horng, «Traffic flow forecasting based on hybrid deep learning framework,» в *12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, 2017.
- [39] P.-H. Kuo и C.-J. Huang, «A High Precision Artificial Neural Networks Model for Short-Term Energy Load Forecasting,» *Energies*, т. 11, p. 213, 2018.
- [40] A. Kasun, M. D. L. и M. Milos, «Deep neural networks for energy load forecasting,» в *IEEE 26th International Symposium on Industrial Electronics (ISIE)*, 2017.
- [41] P.-H. Kuo и C.-J. Huang, «An Electricity Price Forecasting Model by Hybrid Structured Deep Neural Networks,» *Sustainability*, т. 10, p. 1280, 2018.
- [42] L. Hu, G. Taylor, H.-B. Wan и M. Irving, «A review of short-term electricity price forecasting techniques in deregulated electricity markets,» в *Proceedings of the Universities Power Engineering Conference*, 2009.
- [43] J. Contreras, R. Espinola, F. Nogales и A. Conejo, «ARIMA models to predict next-day electricity prices,» *Power Engineering Review, IEEE*, т. 22, 2002.
- [44] Z. Tan, J. Zhang, J. Wang и J. Xu, «Day-ahead electricity price forecasting using wavelet transform combined with ARIMA and GARCH models,» *Applied Energy*, т. 87, № 11, pp. 3606-3610, 2010.
- [45] S. Voronin и J. Partanen, «Price forecasting in the day-ahead energy market by an iterative method with separate normal price and price spike frameworks,» *Energies*, т. 6, pp. 5897-5920, 2013.
- [46] N. K. Singh, M. Tripathy и A. K. Singh, «A radial basis function neural network approach for multi-hour short term load-price forecasting with type of day parameter.,» в *6th International Conference on Industrial and Information Systems*, 2011.
- [47] L. Jesus, D. R. Fjo и D. S. Bart, «Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms,» *Applied Energy*, т. 221, pp. 386-405, 2018.
- [48] G. Dorffner, Neural Networks for Time Series Processing, т. 6, Neural Network World, 1996, pp. 447-468.
- [49] C. Tian, J. Ma, C. Zhang и P. Zhan, «A Deep Neural Network Model for Short-Term Load Forecast Based on Long Short-Term Memory Network and Convolutional Neural Network,» *Energies*, т. 11, 2018.
- [50] K. Hammouda и F. Karay, A comparative study of data clustering techniques, tech. rep., Pattern Analysis and Machine Intelligence Research Group, University of Waterloo, 2000.

- [51] A. Reynolds, G. Richards и V. Rayward-Smith, The application of k-medoids and PAM to the clustering of rules, Springer, 2004.
- [52] P. Rodrigues, J. Gama и J. Pedroso, LBF: Hierarchical time-series clustering for data streams, Proceedings of the 1st International Workshop on Knowledge Discovery in Data Streams (IWKDDS'04), 2004.
- [53] P. Langfelder, B. Zhang и S. Horvath, Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R, т. 24, Bioinformatics, 2007, p. 719–720.
- [54] R. Cannon, J. Dave и J. Bezdek, Efficient Implementation of the Fuzzy C-Means Clustering Algorithms, т. 8, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, pp. 248-255.
- [55] J. Vesanto и E. Altoniemi, Clustering of the self-organizing map, т. 11, IEEE Transactions on Neural Networks, 2000, pp. 586-600.
- [56] J. Han, J. Pei и M. Kamber, Data mining: concepts and techniques., Elsevier, 2011.
- [57] J. Mazanec, Positioning Analysis with Self-Organizing MapsAn Exploratory Study on Luxury Hotels, Cornell Hotel and Restaurant Administration Quarterly, 1995.
- [58] D. Larose, Discovering Knowledge in Data, New Jersey: John Wiley & Sons, 2005.
- [59] T. Kohonen, Self-Organizing Map, 2-nd ред., Springer Verlag, 1997.
- [60] S. Wang и H. Wang, Knowledge Discovery Through Self-Organizing Maps: Data Visualization and Query Processing, т. 4, Knowl. Inf. Syst., 2002, pp. 31-35.
- [61] J. Dayhoff, Neural network architectures — An introduction, Van Nostrand Reinhold, 1990.
- [62] A. Richardson, C. Risien и F. Shillington, Using self-organizing maps to identify patterns in satellite imagery, т. 59, Progress In Oceanography, 2003, pp. 223-239.
- [63] J. Mazanec, Neural market structure analysis: Novel topology-sensitive methodology, т. 35, European Journal of Marketing, 2001, pp. 894-914.
- [64] A. Chitra и S. Uma, An ensemble model of multiple classifiers for time series prediction, International Journal of Computer Theory and Engineering, 2010, pp. 454-458.
- [65] L. Deng и D. Yu, Deep Learning: Methods and Applications, т. 7, Foundations and Trends in Signal Processing, 2014, pp. 1-199.
- [66] C. François, Deep learning with Python, Manning, 2017.
- [67] R. Adrian, Deep Learning for Computer Vision with Python: Starter Bundle, PyImageSearch, 2017.
- [68] J. Patterson и A. Gibson, Deep Learning: A Practitioner's Approach, O'Reilly Media, Inc., 2017.
- [69] C. Aggarwal, Neural Networks and Deep Learning: A Textbook, Springer, Cham, 2018.
- [70] B. Jason, Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python, Machine Learning Mastery, 2018.
- [71] B. Neupane, W. Woon и Z. Aung, Ensemble Prediction Model with Expert Selection for Electricity Price Forecasting, Energies, 2017.