Санкт–Петербургский государственный университет
КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И СИСТЕМ

БОЙКОВ Артем Сергеевич

Выпускная квалификационная работа

Прогнозирование электропотребления по историческим данным

Уровень образования: магистратура
Направление 02.04.02
«Фундаментальная информатика и информационные технологии»
Основная образовательная программа
ВМ.5786.2019 «Цифровые технологии и системы»

Научный руководитель:
кандидат физ.-мат. наук,
доцент
Балыкина Ю. Е.

Санкт-Петербург
2021 г.

Содержание

Введение

На современном рынке электроэнергии между ее производителями и потребителями используются контракты. Эти контракты могут быть как долгосрочными, на несколько месяцев, и даже лет, так и краткосрочными с горизонтом в одни сутки. Отклонение фактического потребления электроэнергии от спрогнозированного объема влечет необходимость для предприятия продажи излишнего или закупки недостающего объема электроэнергии по заведомо невыгодным ценам. Поэтому задача планирования и прогнозирования энергопотребления является достаточно значимой в электроэнергетике, и повышение точности этого прогнозирования может существенно снизить затраты на покупку электроэнергии предприятием.

С развитием вычислительных технологий данная задача все чаще переходит от экспертных систем к системам автоматизированным, а новые методы и алгоритмы машинного обучения и статистические модели позволяют повышать точность этого прогноза, учитывать множество разных факторов и находить сложные нелинейные зависимости в данных.

Таким образом, имеющей большое практическое применение и актуальной научно-технической задачей является разработка методик прогнозирования потребления электрической энергии на основе исторических и других имеющихся данных.

Постановка задачи

Целью данной работы является построение модели для прогнозирования почасового потребления электроэнергии предприятием с горизонтом прогноза в два дня на основе имеющихся исторических данных об электропотреблении и метеофакторах. Также, за это время имеются плановые значения электропотребления, которые будут использоваться для вычисления метрик качества построенной прогностической модели.

В связи с поставленной целью были рассмотрены следующие вопросы:

- Анализ различных методов и подходов к задаче прогнозирования потребления электроэнергии;
- Предобработка данных для обучения и тестирования моделей;
- Программная реализация моделей;
- Сравнение, визуализация и анализ результатов.

Обзор литературы

Развитие алгоритмов анализа данных и информационных технологий создало беспрецедентную возможность изучения моделей прогнозирования электропотребления с точки зрения используемых данных. В литературных источниках для исследования краткосрочного прогнозирования электропотребления рассматривается применение как классических методов анализа и прогнозирования данных, так и методов глубокого обучения на основе нейронных сетей.

Тао Хонг в своей диссертации «Short Term Electric Load Forecasting» [1] описывает специфику данной задачи, рассматривает способы обработки данных и существующие подходы к прогнозированию на основе различных моделей.

В диссертации А. С. Грицая [2] также рассматриваются существующие методы и подходы, применимые к задаче прогнозирования электропотребления, приводится их классификация, а также предлагается гибридный метод краткосрочного прогнозирования для энергосбытового предприятия, основанный на аппроксимации временного ряда электропотребления с использованием синусоидальных функций для дневного и ночного циклов потребления, коэффициенты которых подбираются адаптивно при помощи искусственной нейронной сети. Данный метод показал наилучшую точность среди рассмотренных в работе.

Наиболее распространенными подходами для прогнозирования электропотребления являются модели, основанные на линейной регрессии (MLR) и регрессии опорных векторов (SVR). Они хорошо подходят как для долгосрочного прогнозирования (LTLF) с горизонтом прогнозирования больше месяца, так и для краткосрочного (STLF) с горизонтом до нескольких дней [3]. Данные методы показывают хорошую точность прогноза при небольших вычислительных затратах в сравнении с другими подходами [4–6]. Так, в соревновании, которое проводила компания EUNITE network и задачей которого было прогнозирование ежедневных пиковых нагрузок на ближайшие 31 день, победителями стали Chen, B.-J., Chang, M.-W., Lin, C.-J., представившие модель на основе регрессии опорных векторов [7].

В статьях [8], [9] авторы предлагают вероятностный подход для построения краткосрочного прогнозирования нагрузки на основе ансамблей гауссовских процессов. В статье [10] М. Блум и М. Рейдмиллер описывают улучшение данного подхода, используя три ядра, отражающих суточные и недельные модели нагрузки, а также данные о погоде. В результатах они отмечают хорошие результаты прогнозирования для различных наборов данных.

Еще одним подходом к прогнозированию является применение статистических моделей временных рядов [11]. Классические авторегрессионные (AR) модели временных рядов основываются только на исторических данных. При наличии дополнительных данных применяется авторегрессия с экзогенными переменными (ARX) [12] и множественная линейная регрессия [13]. Учитывая сезонность временных рядов электропотребления, А. Тарситано и А. Америзе в статье [14] предложили использовать двухэтапную сезонную интегрированную модель авторегрессии — скользящего среднего с экзогенными переменными (SARIMAX).

Также, в последнее время в литературе наблюдается рост интереса к разработке моделей прогнозирования, основанных на глубоком обучении, благодаря их преимуществам в улавливании нелинейностей, скрытых в данных. Для прогнозирования нагрузки использовались многие типы нейронных сетей, такие как нейронные сети с прямой связью, сети радиальных базисных функций [15], спайковые нейронные сети [16] и рекуррентные нейронные сети [17, 18]. Наиболее популярным методом обучения является алгоритм обратного распространения ошибки. Авторы этих статей сообщают о довольно хороших результатах данных моделей. Однако исследования применения таких методов в задаче краткосрочного прогнозирования нагрузки все еще относительно невелики по сравнению с классическими методами, и в большинстве работ представляется только общее описание модели и ее результаты, что делает их воспроизведение довольно затруднительным.

В статьях [19], [20] производится сравнение моделей на основе нейронных сетей (ANN, RNN/CNN) и нелинейной авторегрессии с экзогенными переменными (NARX) с моделями временных рядов, такими как SARIMAX. Из них следует, что нейросетевые модели в большинстве случаев показывают

лучший результат. Однако модели временных рядов имеют меньше параметров и лучшую вычислительную эффективность и более применимы для прогнозирования с ограниченным объемом доступных данных.

Важным фактором при решении задачи прогнозирования потребления электроэнергии являются погодные условия, особенно при наличии большого количества кондиционеров или систем отопления, питаемых от электричества. В литературе чаще всего рассматривается влияние таких факторов как температура воздуха, скорость ветра, влажность и атмосферное давление [21–23]. Также, в некоторых статьях [24, 25] дополнительно рассматриваются температура мокрого термометра, температура точки росы, направление ветра, индекс температуры-влажности (THI) и ветро-холодовой индекс (WCI). Однако, влияние метеоданных на значения потребления электроэнергии не всегда значимо и сильно зависит от объекта потребления.

Для увеличения точности применяют ансамбли моделей или гибридные методы, которые сочетают в себе преимущества нескольких подходов. Большинство таких моделей объединяют линейные и нелинейные методы для построения более эффективного прогноза. В последние годы разрабатываются различные ансамблевые модели, в основном сочетающие традиционные статистические методы и алгоритмы машинного обучения [26–29].

Помимо методов прогнозирования, отдельной важной задачей является отбор признаков для входных данных модели. Обычно такими признаками в задаче прогнозирования нагрузки являются исторические данные (значения лага), различные календарные переменные и, при наличии, данные о погоде и других внешних факторах. В статье [30] рассматривается применение различных методик выбора признаков и их влияние на точность различных моделей построения прогноза.

Глава 1. Обзор методов прогнозирования

1.1 Множественная линейная регрессия

Регрессионные методы широко используются для задач прогнозирования. Они позволяют учитывать множество внешних факторов, влияющих на значение целевой переменной. Прогноз в множественной линейной регрессии (Multiple linear regression) строится по формуле:

$$\hat{y}_t = \beta_0 + \beta_1 x_{1,t} + ... + \beta_m x_{x,t},$$

где $\hat{y}_t$ — спрогнозированное значение зависимой переменной в момент времени t, $x_{1,t}, x_{2,t}...x_{m,t}$ — m независимых переменных, а $\beta = \{\beta_0, \beta_1, ..., \beta_m\}$ вектор из $m+1$ параметров, получаемых на этапе обучения с помощью метода наименьших квадратов.

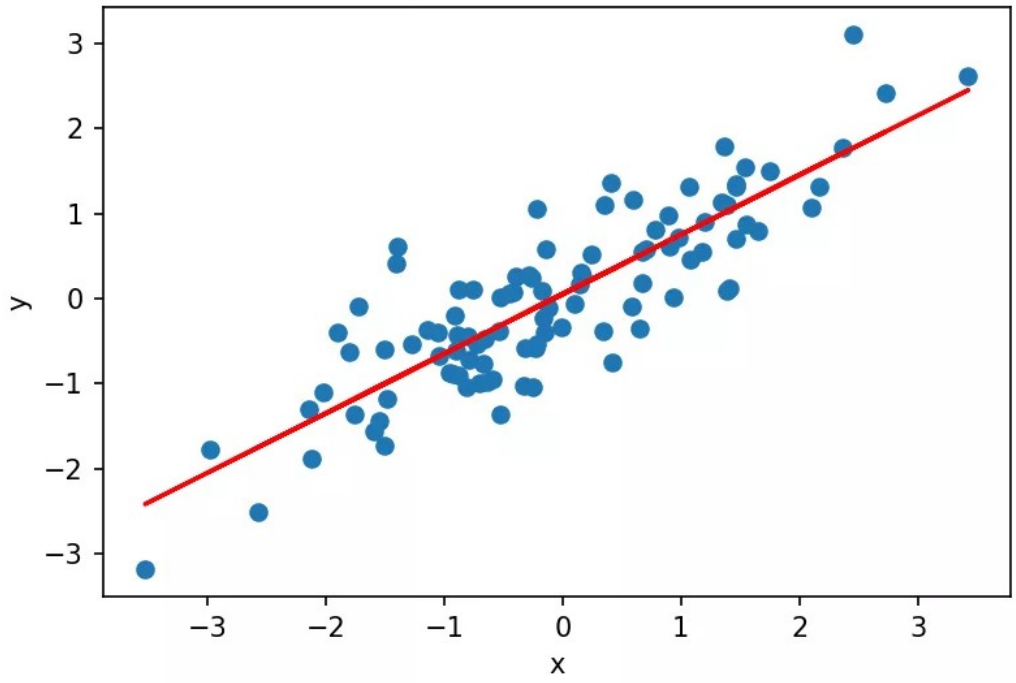

Рис. 1: Линейная регрессия для двух переменных.

Данный метод является легко моделируемым и особенно полезен при создании не очень сложных зависимостей, а также при небольшом количестве данных, а его интерпретация интуитивно-понятна. Однако этот метод сильно чувствителен к выбросам и склонен к переобучению.

Для борьбы с переобучением линейной модели используется регуляризация —– это ограничение весов у признаков. Когда линейная модель переобучается, веса у признаков становятся большими по модулю и разными по знаку. Ограничивая значение этих весов по модулю, можно до какой-то степени побороться с переобучением. Существует два основных способа регуляризации:

L2-регуляризатор добавляет к функционалу потерь слагаемое, равное сумме квадратов весов нашей линейной модели с множителем λ:

$$J_{RIDGE} = \sum_{i=1}^{N}(y_n - \hat{y}_n)^2 + \lambda\|w\|_2^2.$$

L1-регуляризатор использует вместо суммы квадратов сумму модулей весов:

$$J_{LASSO} = \sum_{i=1}^{N}(y_n - \hat{y}_n)^2 + \lambda\|w\|_1.$$

Регрессия с L2-регуляризатором называется ридж-регрессией или гребневой регрессией, а с L1-регуляризатором — лассо.

1.2 Регрессия опорных векторов

Регрессия опорных векторов (Support vector regression) является модернизированным методом опорных векторов, который в классическом виде используется для задачи классификации.

Первоначальная задача регрессии опорных векторов представляется в виде :

$$\begin{aligned} &\min_{w,b,\xi_n\xi_n^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_n + \xi_n^*) \\ &y^i - (w^T x^{(i)} + b) \leq \epsilon + \xi_i, i = 1, ..., N \\ &(w^T x^{(i)} + b) - y^{(i)} \leq \epsilon + \xi_i^*, i = 1, ..., N \\ &\xi_i\xi_i^* \geq 0, i = 1, ..., N \end{aligned} \tag{1}$$

где w и b — параметры гиперплоскости, а ξ_i и ξ_i^* — слабые переменные для i-й точки $x^{(i)}$. C и ϵ являются настраиваемыми гиперпараметрами модели. C

отвечает за регуляризацию, которая штрафует точки за пределами ϵ-трубки, а ϵ — за ширину этой трубки. Для решения этой оптимизационной задачи первичная форма (1) обычно преобразуется в двойственную форму путем введения ядра, которое отображает точки данных в многомерное пространство. В итоге модель регрессии опорных векторов имеет вид:

$$\hat{y} = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) k(x^{(i)}, x) + b$$

где α_i и α_i^* — двойственные переменные для точки $x^{(i)}$, а $k(\cdot, \cdot)$ — функция ядра, вычисляющая корреляции между двумя точками.

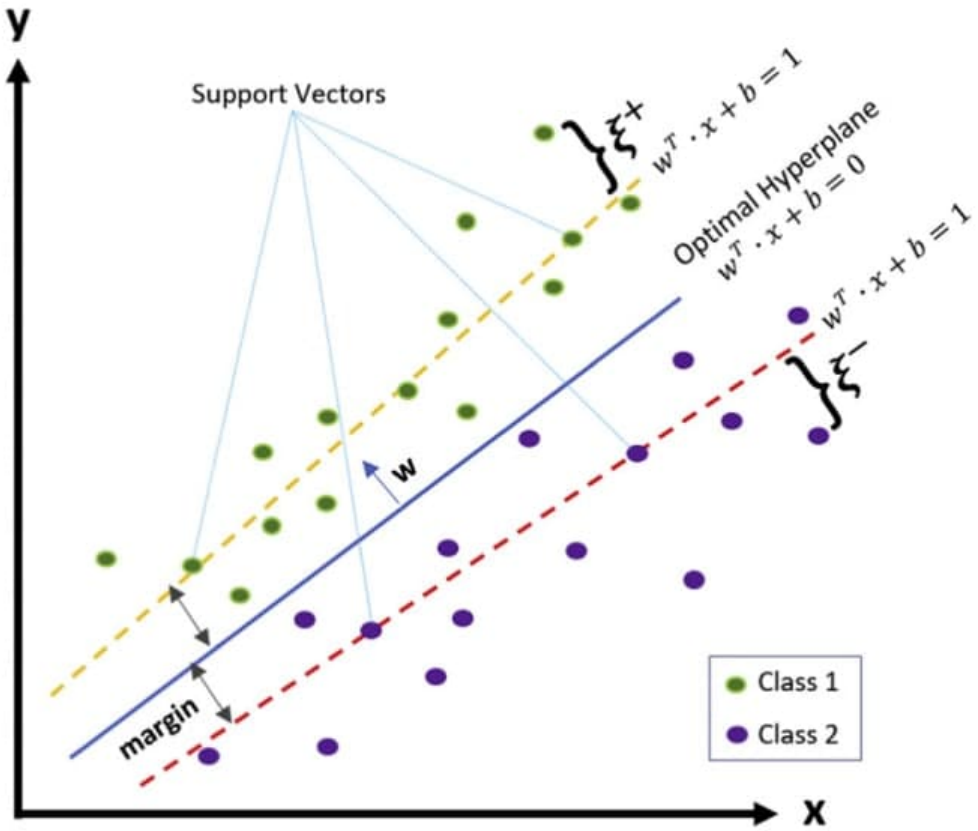

Рис. 2: Геометрическое представление метода опорных векторов [31].

Применение ядерных функций позволяет строить модели с использованием разделяющих поверхностей различной формы.

Ядром может являться любая симметричная, положительно полуопределенная матрица K, которая состоит из скалярных произведений пар векто-

ров x_i и x_j: $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$, характеризующих меру их близости. ϕ здесь является произвольной преобразующей функцией, формирующей ядро.

В качестве ядерных функций чаще всего используют:

- линейное ядро: $K(x_i, x_j) = x_i^T x_j$, что соответствует исходному пространству;
- полиномиальное ядро со степенью p: $K(x_i, x_j) = (1 + x_i^T x_j)^p$;
- сигмодиное ядро: $K(x_i, x_j) = tanh(\gamma x_i^T x_j + \beta_0)$;
- гауссово ядро с радиальной базовой функцией (RBF):

  $K(x_i, x_j) = exp(\gamma \|x_i - x_j\|^2)$.

Параметры p, γ, β_0 и т.д. подлежат оптимизации.

Достоинства данного метода:

- хорошо работает с пространством признаков большого размера;
- хорошо работает с данными небольшого объема;
- так как алгоритм сводится к решению задачи квадратичного программирования в выпуклой области, то такая задача всегда имеет единственное решение
- построение нелинейных зависимостей с помощью ядер.

Ограничения:

- неустойчивость к шуму: выбросы в обучающих данных становятся опорными объектами-нарушителями и напрямую влияют на построение разделяющей гиперплоскости;
- долгое время обучения (для больших наборов данных);
- не описаны общие методы построения ядер и спрямляющих пространств, наиболее подходящих для конкретной задачи.

1.3 Случайный лес

На сегодняшний день случайный лес является одним из самых распространенных алгоритмов машинного обучения. Он был предложен Л. Брейманом [32] для решения задачи классификации. Однако, модификации данного метода также используются для решения задач регрессии, кластеризации, поиска аномалий, селекции признаков и т.д.

Алгоритм регрессии методом случайного леса заключается в построении N деревьев и усреднении их результатов.

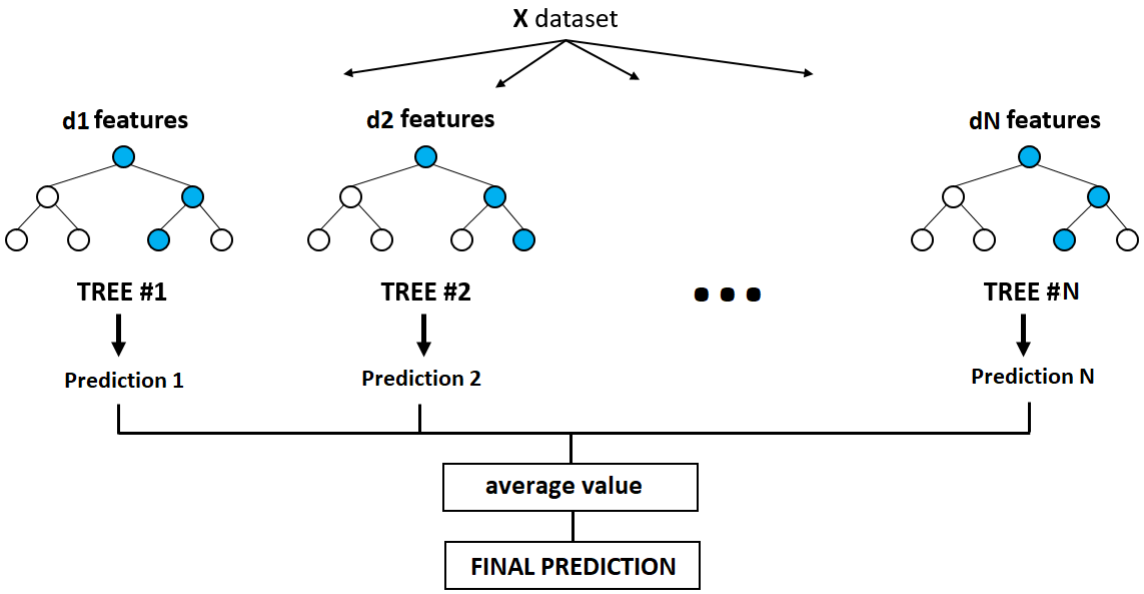

Рис. 3: Случайный лес.

Для каждого $i = 1, ..., N$ с помощью метода бутстрэпа генерируется выборка X_i, затем по этой выборке строится решающее дерево b_i:

— по заданному критерию выбирается лучший признак, по нему производится разбиение в дереве и так до исчерпания выборки;

— дерево строится до тех пор, пока в каждом листе не более n_{min} объектов или пока не достигнем определенной высоты дерева;

— при каждом разбиении сначала выбирается m случайных признаков из n исходных, и оптимальное разделение выборки ищется только среди них.

Итоговое значение прогноза получается по формуле:

$$\hat{y}(t) = \frac{1}{N} \sum_{i=1}^{N} b_i(t)$$

В качестве достоинства данного подхода можно отметить его универсальность и устойчивость к выбросам.

1.4 Нейронные сети

Еще одним подходом к прогнозированию временных рядов является использование нейронных сетей. В частности, распространенным решением является применение сетей с радиальными базисными функциями (РБФ) и сети долгой краткосрочной памяти (LSTM).

Сети РБФ имеют ряд преимуществ перед другими видами нейронных сетей. Во-первых, они моделируют произвольную нелинейную функцию с помощью всего одного промежуточного слоя, тем самым избавляя разработчика от необходимости решать вопрос о числе слоев. Во-вторых, параметры линейной комбинации в выходном слое можно полностью оптимизировать с помощью хорошо известных методов линейной оптимизации, которые работают быстро и не испытывают трудностей с локальными минимумами, мешающими при обучении с использованием алгоритма обратного распространения ошибки.

На рис. 4 приведена структура стандартной сети с РБФ. Она состоит из входного слоя, на который подается входной вектор, скрытого слоя с нейронами радиального типа и выходного слоя, состоящего из одного или нескольких линейных нейронов.

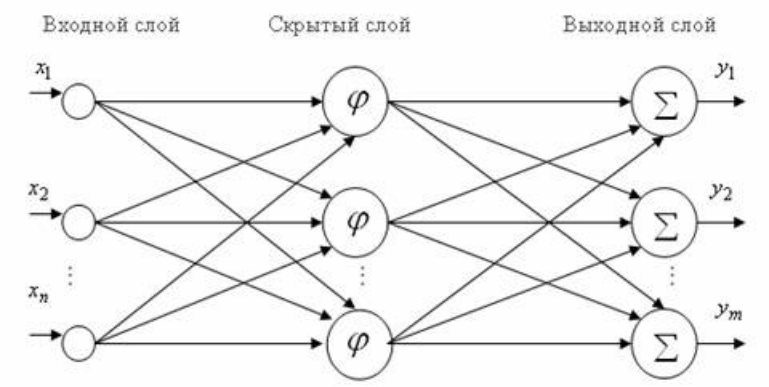

Рис. 4: Архитектура сети с радиальными базисными функциями.

Вход функции активации радиального базисного слоя определяется как модуль разности вектора весов и вектора входа, умноженный на смещение. Т.е. выход радиального базисного слоя определяется по формуле:

$$a_i = F(\left\| IW_i^{11} - P \right\| b_i^1)$$

где P — вектор входа, IW_i^{11} — вектор весов i-го нейрона, b_i^1 — смещение i-го нейрона, F – радиальная базисная функция активации:

$$F(x) = e^{-x^2}.$$

А выход линейного слоя является выходом всей сети:

$$y = LW^{21}a^1 + b^2$$

где LW^{21} — матрица весов от первого ко второму слою сети, b^2 — смещение линейного нейрона, а a^1 — выход радиального базисного слоя.

Преимуществом сетей РБФ является быстрое обучение —- на порядок быстрее, чем с использованием алгоритма обратного распространения ошибки.

Недостатками данного метода являются плохие аппроксимирующие свойства и большой размер при большой размерности вектора входов.

Глава 2. Подготовка данных

Рассматриваемая задача решалась на примере реальных данных, предоставленных энергосбытовой компанией. Исходный набор данных состоит из почасовых записей фактического потребления электроэнергии с января 2019 года по октябрь 2020 (15336 записей), а также плановых значений за данный период, которые будут использоваться только для вычисления метрики качества прогноза. Также будет использован набор данных о почасовом прогнозе четырех метеофакторов: температуры, влажности воздуха, атмосферном давлении и скорости ветра за период с июля 2019 года по октябрь 2020.

На рис. 5–6 приведен пример данных за три недели и почасовое потребление электроэнергии по дням недели.

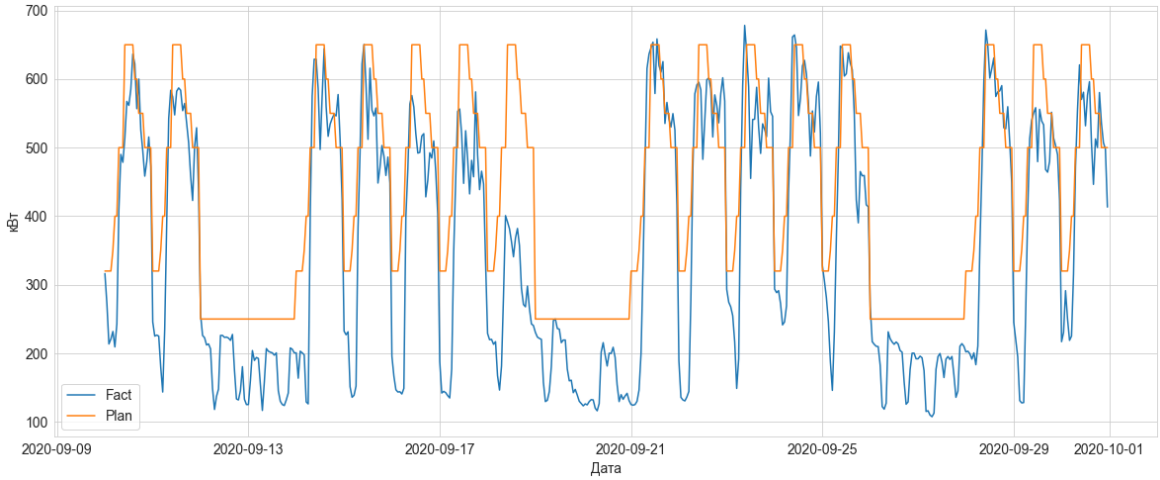

Рис. 5: Пример данных за три недели.

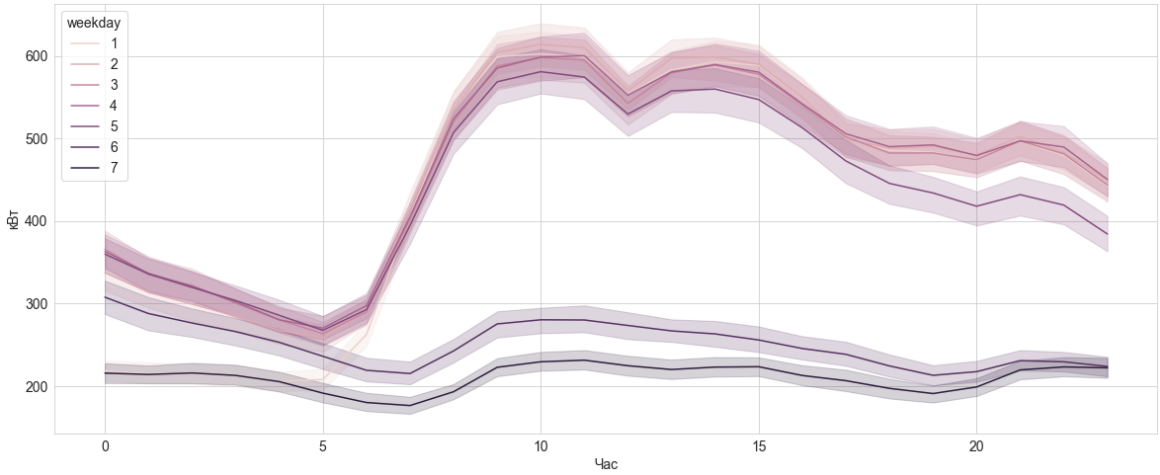

Рис. 6: Почасовое потребление по дням недели.

На основе имеющихся данных можно выделить признаки, относящиеся к трем категориям: исторические данные, календарные и данные о погоде. Далее в этой главе будут описаны различные варианты получения этих признаков.

2.1 Исторические данные

Так как имеющиеся значения потребления электроэнергии имеют ярко выраженную недельную периодичность, наиболее значимыми для построения прогноза будут данные со значением лага в одну (или несколько) недель, т. е. кратные 168 часам.

Далее будут использоваться 3 варианта:

- Только значение недельного лага L_{t-168};
- Значения трёх недельных лагов L_{t-168}, L_{t-336}, L_{t-504};
- Среднее значение трёх недельных лагов $\frac{L_{t-168}+L_{t-336}+L_{t-504}}{3}$.

2.2 Календарные данные

Соответствующий механизм кодирования календарной информации для точного указания закономерностей периодических колебаний может быть

полезен для прогнозирования электропотребления. Можно выделить 3 вида календарных паттернов: ежедневное электропотребление, еженедельное электропотребление и годовое.

На рис. 7 представлен пример почасового потребления электроэнергии за два будних дня. Здесь четко выделяется высокое электропотребление в рабочие часы и низкое в ночное время.

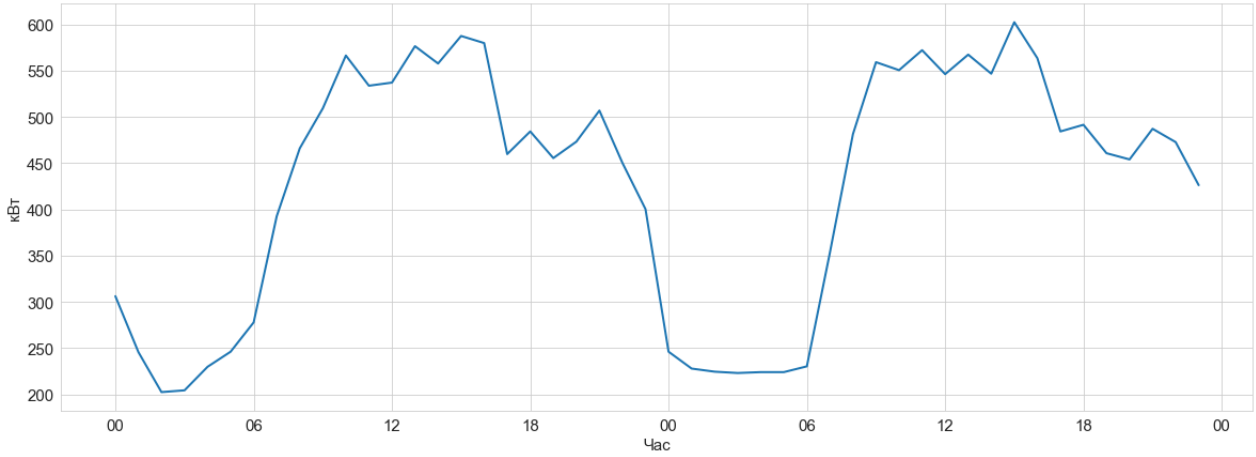

Рис. 7: Суточное электропотребление.

На рис. 8 изображены данные за две недели, начиная с понедельника. Ежедневные модели изменения электропотребления с понедельника по пятницу аналогичны в течение двух недель, в то время как в выходные дни наблюдается тенденция к снижению.

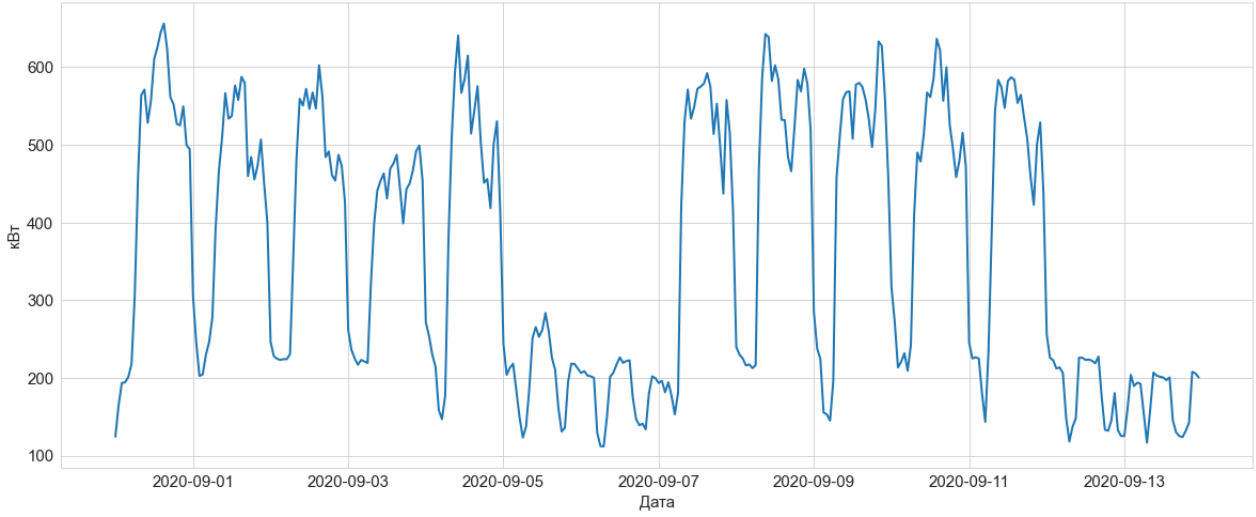

Рис. 8: Недельное электропотребление.

На рис. 9 показано потребление электроэнергии в 10:00 каждого дня за весь период с января 2019 по октябрь 2020. Можно заметить, что наиболее значительные спады нагрузки происходили в выходные и в период новогодних и других праздников, а среднее значение в летний период ниже, чем в зимний.

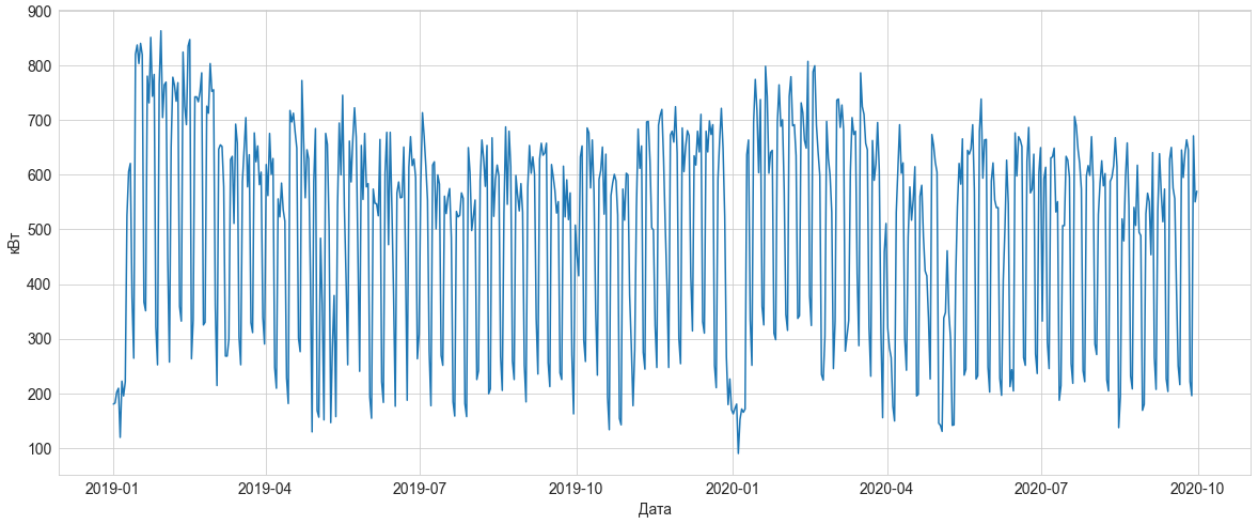

Рис. 9: Годовое электропотребление в 10:00.

В исходном виде дата и время не могут быть использованы моделями прогнозирования, поэтому применяются описанные ниже способы кодирования календарной информации о времени, дне недели и дне года:

- Кодирование порядковым номером

Самым очевидным способом является кодирование порядковым номером. Он может использоваться для кодирования времени, дня недели и дня года. Так, переменная для обозначения времени будет принимать целочисленные значения от 0 до 23, а дня недели от 1 до 7. При таком механизме кодирования две более близкие временные точки будут иметь меньшее расстояние и более высокое сходство. Однако этого недостаточно, чтобы отразить периодический эффект. Например, 23:00 и 0:00 на следующий день соседствуют, в то время как расстояние между ними равно 23. Поэтому, чаще всего используются другие методы, учитывающие цикличность.

- Полярное кодирование

Данный способ заключается в кодировании календарных значений на основе полярной системы координат. Периодический цикл можно рассматривать как единичный круг в полярной системе координат. Календарные переменные кодируются в виде единичного круга, а их координаты указывают время. Процесс кодирования описывается формулой (2):

$$x = (cos(g), sin(g)), g = \frac{2\pi t}{p} \tag{2}$$

где t — время, а p — длина цикла. Т. е. $p = 24$ для времени суток и $p = 7$ для дня недели. Данный подход также может использоваться для кодирования времени, дня недели и дня года. Он точно описывает периодические особенности, и различие между точками может быть обоснованно закреплено евклидовым расстоянием. На рис. 10 графическое представление данного метода.

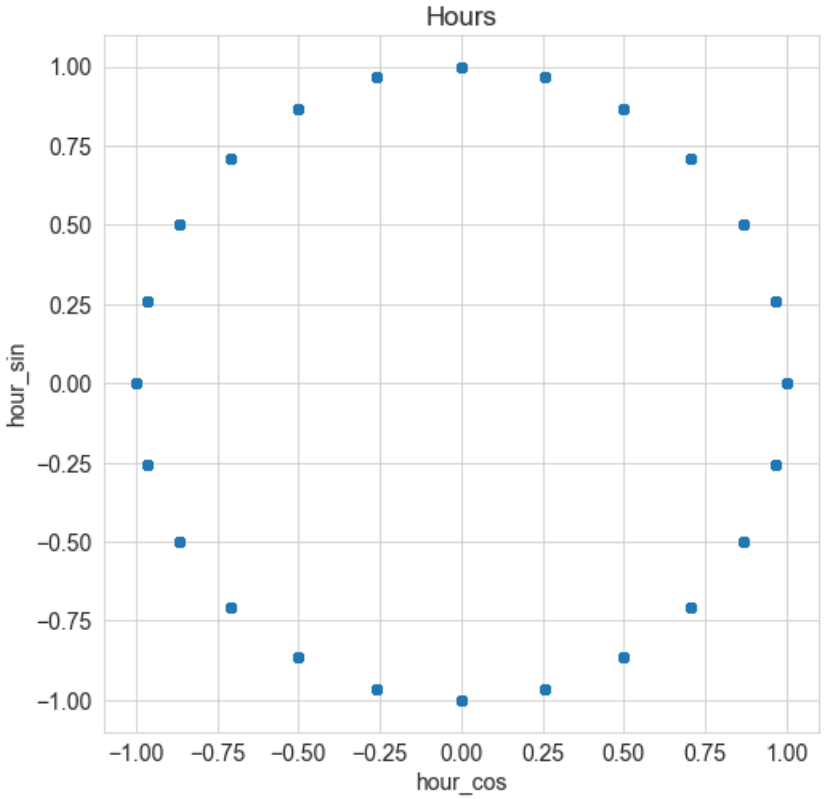

Рис. 10: Полярное кодирование времени суток.

- Dummy-кодирование

При таком кодировании индекс календарных переменных преобразуется в двоичный вектор с одним элементом, равным 1, указывающим требуемое время, и другими элементами, равными 0. Так, для кодирования дня недели будет использоваться вектор из семи значений, где 6 нулей и единица на месте нужного дня недели. Например, понедельник будет кодироваться вектором $x = [1, 0, 0, 0, 0, 0, 0]$.

Такой механизм кодирования полезен для идентификации двух одинаковых точек времени, однако расстояние между ними не учитывается и всегда равно $\sqrt{2}$ для различных значений и 0 для одинаковых. Еще одним недостатком такого подхода является сильно увеличивающаяся размерность пространства признаков если переменная может принимать большое количество значений, как например значение дня года, имеющее 365 значений. Поэтому данный способ кодирования будем использовать только для времени и дня недели.

- Типы дней недели

Так как работа и электропотребление предприятия связаны с распорядком недели, можно выделить четыре типа дней: начало рабочей недели (понедельник), середина рабочей недели (вторник—четверг), конец рабочей недели (пятница) и выходные (суббота, воскресенье). Данные категории можно закодировать вектором из четырех значений аналогично прошлому пункту.

- Количество дней до/после выходных

Также, день недели можно закодировать двумя значениями, которые будут отражать количество дней до следующего выходного и количество дней прошедшее с прошлого. Так например, понедельник будет закодирован парой чисел $[5, 1]$, а вторник $[4, 2]$.

- Построение нескольких моделей

Данный подход заключается в построении локальных моделей путем разделения данных на подмножества на основе календарных значений. Путем разделения набора данных по часам будет разработано 24 локальных модели, и каждая из них должна предсказать электропотребление соответствующего часа. Например, чтобы спрогнозировать будущее электропотребление в 0:00, модель должна быть обучена только с использованием исторических данных, записанных в 00:00. Аналогично, используя для разделения набора данных дни недели, можно построить семь локальных моделей для раздельного прогнозирования нагрузки в течение семи дней.

2.3 Метеоданные

Особенности погодных факторов также часто использовались в предыдущих исследованиях [21–25]. Обычно рассматривают четыре погодных фактора: температура воздуха, давление, влажность и скорость ветра.

Как показано на рис. 9, характер изменения годового электропотребления может быть частично объяснен сезонными изменениями. Например, использование кондиционеров и систем отопления может быть увеличено в течение лета и зимы. С этой точки зрения температура и скорость ветра являются двумя основными факторами, которые могут повлиять на потребление энергии. Влажность и атмосферное давление связаны с температурой и скоростью ветра, поэтому также могут рассматриваться в качестве потенциальных факторов.

Будем рассматривать следующие варианты использования метеоданных:

- не добавлять данные о погоде;
- только данные о температуре воздуха;
- данные о температуре воздуха и скорости ветра;
- данные о всех четырех факторах: температуре воздуха, давлении, влажности и скорости ветра;

- преобразование четырех погодных факторов с помощью метода главных компонент (PCA) в один погодный фактор.

В таблице 1 приведены все рассматриваемые варианты выбора признаков, которые будут использоваться в моделях:

Таблица 1: Варианты признаков

Категория	Вариант	Обозначение
Исторические значения	• Значение недельного лага	L^{w}
	• Значения трёх недельных лагов	L^{3w}
	• Среднее значение трёх недельных лагов	L^{3w_m}
Время суток	• Полярное кодирование	T^{polar}
	• Dummy-кодирование	T^{dummy}
	• Построение 24 моделей для каждого часа	T^{24}
День недели	• Полярное кодирование	D^{polar}
	• Dummy-кодирование	D^{dummy}
	• Тип дня недели	D^{tod}
	• Количество дней до/после выходных	D^{baw}
	• Построение 7 моделей для каждого дня недели	D^{7}
Метеоданные	• Не учитывать метеоданные	W^{0}
	• Только значение температуры воздуха	W^{t}
	• Значения температуры и скорости ветра	W^{tw}
	• Все метеоданные	W^{all}
	• Значение, полученное с помощью метода главных компонент	W^{PCA}

2.4 Удаление недель с выбросами

Для того чтобы модель не обучалась на нетипичных данных-выбросах, дополнительно был рассмотрен вариант исключения из обучающей выборки недель, в которых присутствуют дни с необычным для своего дня недели объемом потребления электроэнергии. Критерием для отбора таких дней выберем попадание 7,5 перцентиль для будних дней и выход из 92,5 перцентиля для выходных, что обычно соответствует нерабочим будним и рабочим выходным соответственно.

2.5 Кросс-валидация

Кросс-валидация (перекрёстная проверка, cross-validation) — это метод оценки качества построенной модели. При однократном разделении данных на обучающую (train) и тестовую (test) выборку не всегда учитывается общая картина предоставленных данных. Результат оценки качества будет очень сильно зависеть от разбиения. Например, могло получиться так, что в наборе данных есть особые объекты, которые при разделении на обучающую и тестовую, попали только в тестовую. Тогда на этих особых объектах мы получим плохое качество, т.к. алгоритм при обучении не видел таких объектов.

Суть метода заключается в разбиении данных на n блоков, затем на $(n-1)$ блоках данных производится обучение модели, а оставшаяся часть данных используется для тестирования. Данная процедура повторяется так, чтобы каждый из n блоков был использован в качестве тестового. В результате получается оценка качества модели с наиболее равномерным использованием имеющихся данных.

Однако, при работе с временными рядами стоит учитывать их временную структуру, и случайно перемешивать в блоках значения всего ряда без сохранения этой структуры нельзя, иначе в процессе потеряются все взаимосвязи наблюдений друг с другом. Поэтому, для временных рядов применяется кросс-валидация на скользящем окне.

Так, в начале модель обучается на отрезке временного ряда, от начала до некоторого t, и производится прогноз на $t+n$ шагов вперед и считается метрика. Затем, обучающая выборка расширяется до $t+n$ значений и строится

прогноз для отрезка с $t + n$ до $t + 2 * n$, так тестовый отрезок двигают до конца имеющихся наблюдений.

Варьируемыми параметрами являются размер тестового отрезка и количество итераций.

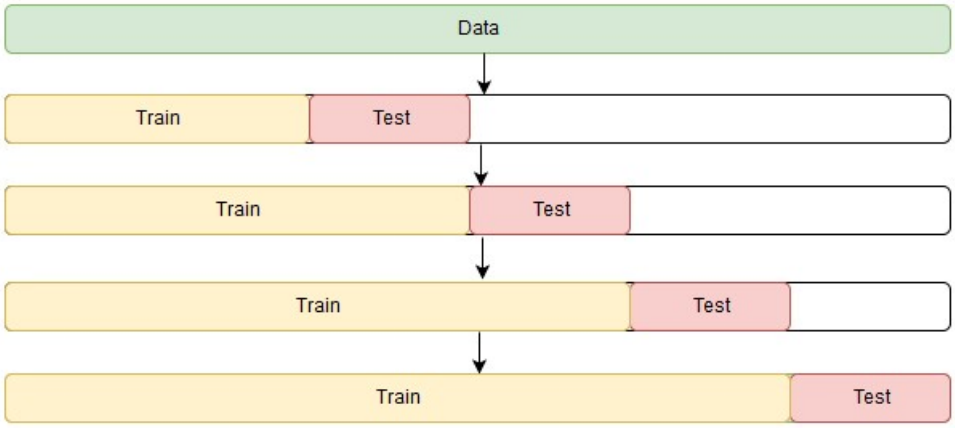

Рис. 11: Кросс-валидация временных рядов.

Глава 3. Результаты

В данной главе будет рассмотрена программная реализация алгоритмов прогнозирования, проведен анализ, визуализация и сравнение полученных результатов.

Для реализации программной части был использован язык программирования Python 3.6 и библиотеки NumPy для работы с массивами и математическими функциями, Pandas для представления и обработки данных в виде таблиц, Scikit Learn для реализации алгоритмов машинного обучения, Keras для реализации нейронных сетей и Matplotlib для визуализации данных. Программный код реализации слоя сети с радиальными базисными функциями приведен в Приложении 1.

3.1 Разработка моделей

Для построения прогноза были реализованы 4 регрессионнные модели:

- множественная линейная регрессия (MLR);
- регрессия опорных векторов (SVR);
- случайный лес (RF);
- нейронные сети с радиальными базисными функциями (RBN).

Гиперпараметры для алгоритмов регрессии опорных векторов, случайного леса и сетей RBN подбирались случайным поиском по сетке (random grid search).

Регрессия опорных векторов:

- $C = 1000$ — коэффициент регуляризации;
- $\epsilon = 0.05$ — размер трубки;
- $kernel =' rbf'$ — ядро радиальных базисных функций.

Случайный лес:

- $n\_estimators = 250$ — число деревьев;
- $criterion =' gini'$ — критерий выбора;
- $max\_depth = 15$ — максимальная глубина дерева;
- $min\_samples\_leaf = 5$ — минимальное число объектов для деления;
- $min\_samples\_leaf = 5$ — минимальное число объектов в листьях.

Нейронные сети RBN:

- $units = 10$ — количество юнитов;
- $\gamma = 0.5$ — коэффициент экспоненты.
- $epoch = 50$ — число эпох.

3.2 Анализ результатов прогнозирования

Для сравнения результатов прогнозирования были использованы средние значения по трем метрикам при кросс-валидации на трех блоках по три недели (504 часа):

$$Metric = \frac{|fact - plan| - |fact - predict|}{|fact - plan|} * 100$$

$$MAPE = \frac{100}{n} \sum_{t=1}^{N} \left| \frac{fact - predict}{fact} \right|$$

$$RMSE = \sqrt{\frac{\sum_{t=1}^{N} (predict - fact)^2}{N}}$$

где fact, plan и predict — фактические, плановые и спрогнозированные значения соответственно.

В табл. 2—4 представлены лучшие результаты прогнозирования по значению трех метрик на основе только исторических данных с 1 января 2019

по 30 сентября 2020 года. Лучшие значения: RMSE = 59.63, MAPE = 16.24, Metric = 53.99. Алгоритм регрессии опорных векторов показал лучшие результаты по значеню RMSE и MAPE, а по значению плановой метрики лучшие результаты у случайного леса.

Таблица 2: Лучшие результаты по значению RMSE

Model	L	T	D	RMSE	MAPE	Metric
SVR	L^{3w_m}	T^{dummy}	D^{baw}	59.6279	16.2844	50.2053
SVR	L^{3w_m}	T^{dummy}	D^{tod}	59.7457	16.2794	50.6990
SVR	L^{3w_m}	T^{polar}	D^{baw}	59.8999	16.2417	52.6232
SVR	L^{3w_m}	T^{dummy}	D^{7}	59.9702	16.3011	51.8672
SVR	L^{3w_m}	T^{dummy}	D^{dummy}	60.0416	16.3372	51.6165
SVR	L^{3w_m}	T^{polar}	D^{7}	60.1305	16.4537	53.5987
SVR	L^{3w_m}	T^{polar}	D^{polar}	60.1393	16.4313	52.2256
SVR	L^{3w_m}	T^{dummy}	D^{polar}	60.1449	16.3291	50.9758
SVR	L^{3w_m}	T^{24}	D^{dummy}	60.2143	16.5283	51.2412
SVR	L^{3w_m}	T^{polar}	D^{tod}	60.3035	16.4654	50.6912

Таблица 3: Лучшие результаты по значению MAPE

Model	L	T	D	RMSE	MAPE	Metric
SVR	L^{3w_m}	T^{polar}	D^{baw}	59.8999	16.2417	52.6232
SVR	L^{3w_m}	T^{dummy}	D^{tod}	59.7457	16.2794	50.6990
SVR	L^{3w_m}	T^{dummy}	D^{baw}	59.6279	16.2844	50.2053
SVR	L^{3w_m}	T^{dummy}	D^{7}	59.9702	16.3011	51.8672
SVR	L^{3w_m}	T^{dummy}	D^{polar}	60.1449	16.3291	50.9758
SVR	L^{3w_m}	T^{dummy}	D^{dummy}	60.0416	16.3372	51.6165
SVR	L^{3w}	T^{dummy}	D^{baw}	61.1334	16.4286	46.4510
SVR	L^{3w_m}	T^{polar}	D^{polar}	60.1393	16.4313	52.2256
SVR	L^{3w_m}	T^{polar}	D^{7}	60.1305	16.4537	53.5987
SVR	L^{3w_m}	T^{polar}	D^{tod}	60.3035	16.4654	50.6912

Таблица 4: Лучшие результаты по значению плановой метрики

Model	L	T	D	RMSE	MAPE	Metric
RF	L^{3w_m}	T^{24}	D^{baw}	63.8124	17.8037	53.9896
RF	L^{3w_m}	T^{24}	D^{dummy}	63.7460	17.7910	53.9548
RF	L^{3w_m}	T^{polar}	D^{7}	64.7063	18.2016	53.9402
RF	L^{3w_m}	T^{polar}	D^{baw}	63.5354	17.7793	53.9163
SVR	L^{3w_m}	T^{polar}	D^{7}	60.1305	16.4537	53.5987
RF	L^{3w_m}	T^{polar}	D^{polar}	63.6788	17.8443	53.4554
RF	L^{3w_m}	T^{polar}	D^{dummy}	63.5507	17.7932	53.3830
RF	L^{3w_m}	T^{24}	D^{polar}	63.9626	17.8504	53.2736
RF	L^{3w_m}	T^{dummy}	D^{dummy}	63.7889	17.8636	52.7102
SVR	L^{3w_m}	T^{polar}	D^{baw}	59.8999	16.2417	52.6232

Наблюдения в наборе данных с метеофакторами начинаются на 6 месяцев позже наблюдений об электропотреблении, поэтому размер обучающей выборки для следующего эксперимента будет меньше, что может повлиять на точность прогноза.

В табл. 5—7 представлены результаты прогнозирования с использованием данных о погоде. Не смотря на уменьшение обучающей выборки, результаты метрик RMSE и MAPE несколько улучшились до значений 57.73 и 14.85 соответственно. Однако, на значение плановой метрики добавление метеоданных повлияло незначительно и некоторые из лучших моделей не использовали данные о погоде для построения прогноза.

Наилучшим вариантом использования метеоданных оказался вариант с добавлением только информации о температуре воздуха.

Таблица 5: Лучшие результаты по значению RMSE

Model	L	T	D	W	RMSE	MAPE	Metric
SVR	L^{3w_m}	T^{dummy}	D^{tod}	W^{t}	57.7331	14.8496	42.4052
SVR	L^{3w}	T^{dummy}	D^{baw}	W^{t}	58.2446	15.0626	43.9853
SVR	L^{3w_m}	T^{dummy}	D^{baw}	W^{t}	58.3682	15.1059	45.1717
SVR	L^{3w_m}	T^{dummy}	D^{polar}	W^{t}	58.4450	15.0351	44.1712
SVR	L^{3w_m}	T^{dummy}	D^{baw}	W^{PCA}	58.4890	15.5042	45.7074
SVR	L^{3w_m}	T^{dummy}	D^{dummy}	W^{t}	58.5237	15.0254	44.2438
RF	L^{3w_m}	T^{polar}	D^{dummy}	W^{all}	58.7209	15.4088	50.0996
RF	L^{3w_m}	T^{polar}	D^{baw}	W^{all}	58.7248	15.4716	49.6927
SVR	L^{3w}	T^{dummy}	D^{polar}	W^{t}	58.7927	15.3149	43.6990
RF	L^{3w_m}	T^{polar}	D^{polar}	W^{all}	58.8817	15.4742	49.4724

Таблица 6: Лучшие результаты по значению MAPE

Model	L	T	D	W	RMSE	MAPE	Metric
SVR	L^{3w_m}	T^{dummy}	D^{tod}	W^{t}	57.7331	14.8496	42.4052
SVR	L^{3w_m}	T^{dummy}	D^{dummy}	W^{t}	58.5237	15.0254	44.2438
SVR	L^{3w_m}	T^{dummy}	D^{polar}	W^{t}	58.4450	15.0351	44.1712
SVR	L^{3w}	T^{dummy}	D^{baw}	W^{t}	58.2446	15.0626	43.9853
SVR	L^{3w_m}	T^{dummy}	D^{baw}	W^{t}	58.3682	15.1059	45.1717
SVR	L^{3w_m}	T^{polar}	D^{baw}	W^{t}	59.6934	15.1439	44.8734
SVR	L^{3w_m}	T^{dummy}	D^{7}	W^{t}	59.4571	15.1674	42.6698
SVR	L^{3w}	T^{dummy}	D^{tod}	W^{t}	58.9756	15.2086	42.2090
SVR	L^{3w_m}	T^{polar}	D^{tod}	W^{t}	59.7223	15.2190	42.9989
SVR	L^{3w}	T^{dummy}	D^{dummy}	W^{t}	59.1691	15.2543	44.2720

Таблица 7: Лучшие результаты по значению плановой метрики

Model	L	T	D	W	RMSE	MAPE	Metric
RF	L^{3w_m}	T^{dummy}	D^{baw}	W^{t}	60.8342	16.3006	54.3703
RF	L^{3w_m}	T^{polar}	D^{baw}	W^{0}	62.5355	17.3995	53.5910
RF	L^{3w_m}	T^{dummy}	D^{dummy}	W^{t}	60.9013	16.2909	53.5705
SVR	L^{3w_m}	T^{polar}	D^{7}	W^{0}	60.4792	16.4713	53.4700
RF	L^{3w_m}	T^{polar}	D^{7}	W^{0}	63.6030	17.7360	53.3065
RF	L^{3w_m}	T^{dummy}	D^{baw}	W^{PCA}	59.7655	16.1304	53.1985
RF	L^{3w_m}	T^{dummy}	D^{dummy}	W^{PCA}	60.2198	16.2173	53.1826
RF	L^{3w_m}	T^{polar}	D^{dummy}	W^{0}	62.4446	17.3435	53.1654
RF	L^{3w_m}	T^{24}	D^{baw}	W^{0}	62.6294	17.3516	53.1024
RF	L^{3w_m}	T^{dummy}	D^{dummy}	W^{tw}	61.3301	16.3386	53.0565

3.3 Визуализация прогноза

Наилучшую точность по совокупности значений метрик показала модель на основе регрессии опорных векторов с набором признаков L^{3w_m}, T^{dummy}, D^{tod}, W^{t} (среднее значение трёх недельных лагов, dummy-кодирование времени суток, кодирование дня недели его типом, значение температуры воздуха).

На рис. 12 представлен результат прогноза лучшей из построенных моделей на данных последних четырех недель с горизонтом в одну неделю.

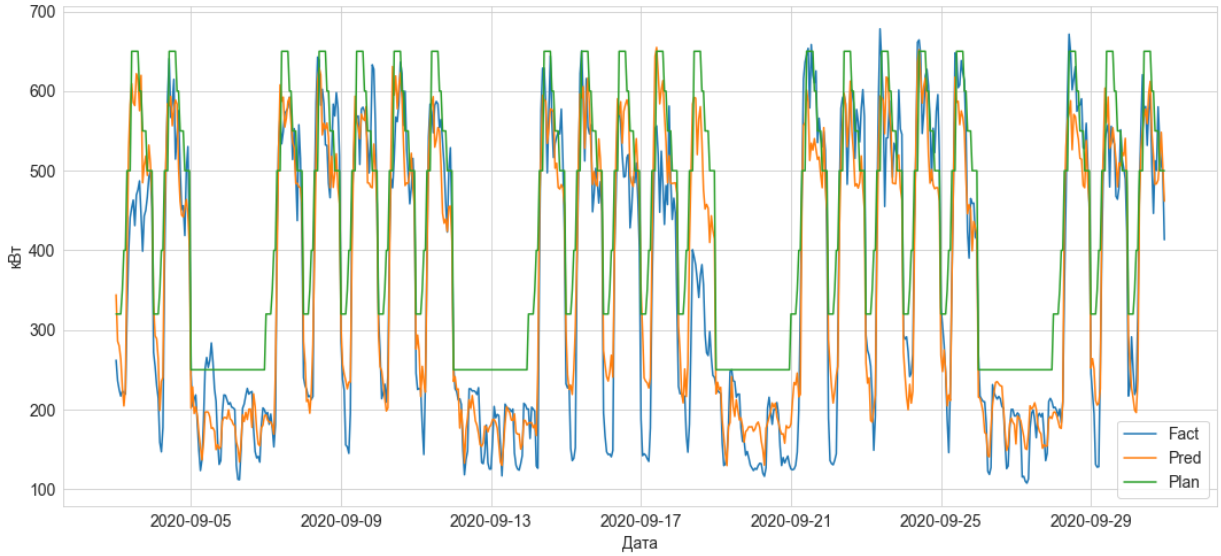

Рис. 12: Прогноз лучшей модели.

На рис. 13 представлено распределение остатков результата прогнозирования модели и плановых значений. Остатки прогноза распределены симметрично нулю, в то время как плановые значения чаще больше реальных, что также видно на графике рис. 12.

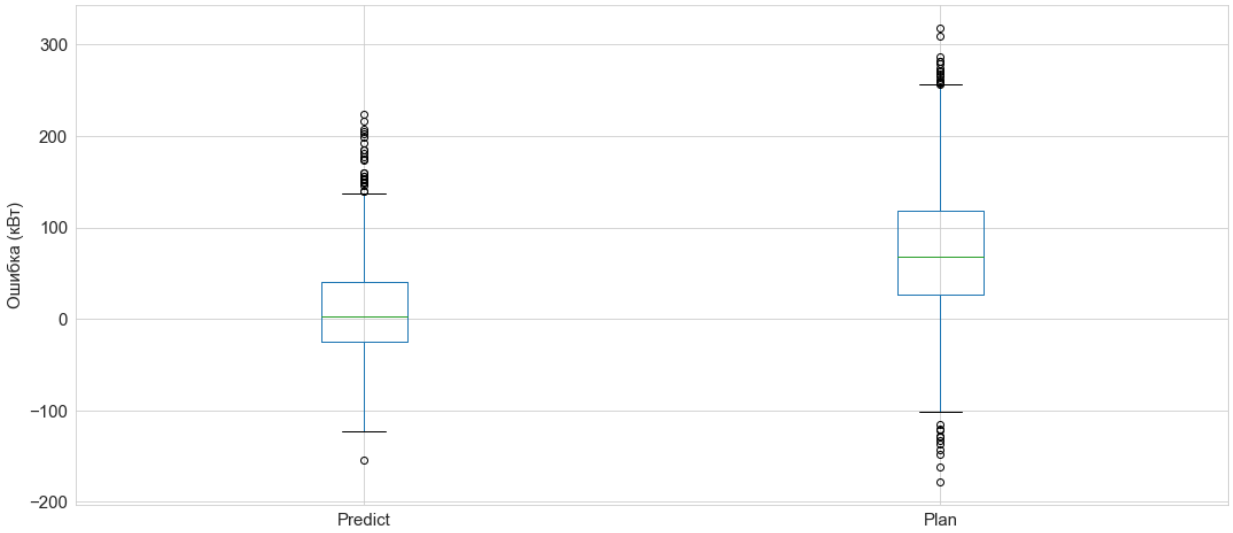

Рис. 13: Ошибки прогноза и плановых значений.

На рис. 14 показан модуль дневной ошибки результата прогнозирования модели и плановых значений. В среднем ошибка прогноза меньше плановой более чем на 1000 кВт.

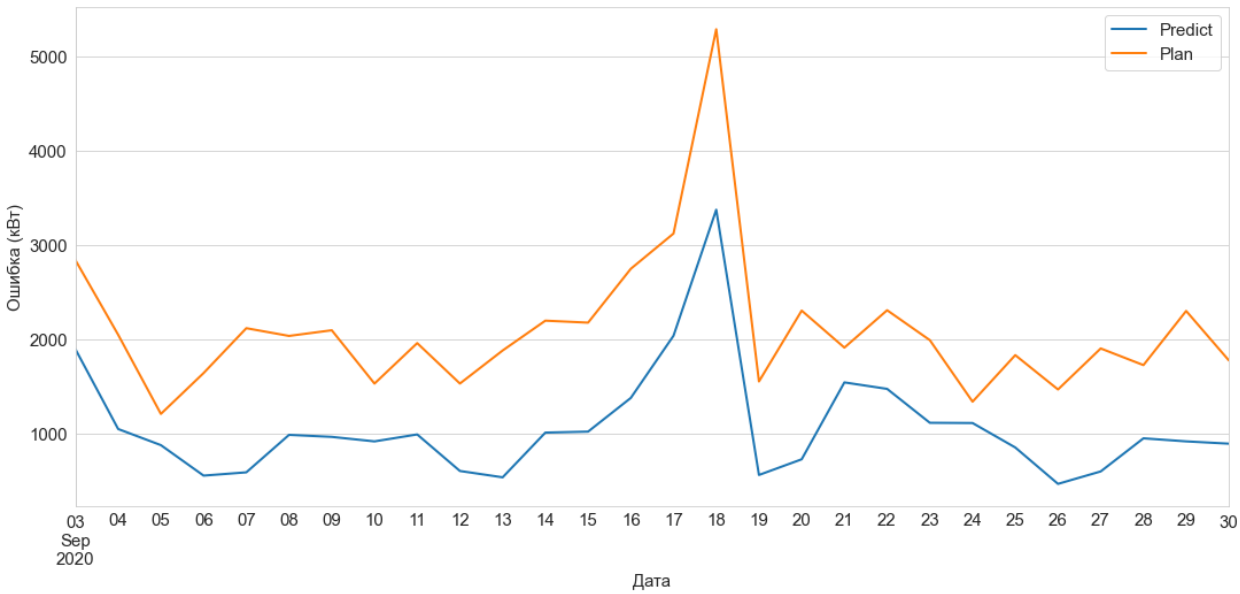

Рис. 14: Ошибки прогноза и плановых значений.

Выводы

На основе полученных результатов можно заключить, что регрессия опорных векторов и алгоритм случайного леса хорошо подходят для решения задачи прогнозирования потребления электроэнергии и превосходят в точности модели на основе множественной линейной регрессии и нейронных сетях радиальных базисных функций.

Прогноз, полученный с помощью описанной в работе модели, превосходит плановые значения по метрике в среднем более чем на 50%.

Из таблиц результатов (табл. 5—7) также следует, что при добавлении данных о метеофакторах положительно на точность прогноза влияют только значения температуры воздуха.

Заключение

В данной работе был проведен обзор и анализ методов, применяемых для краткосрочного прогнозирования потребления электроэнергии. Также, были реализованы модели на основе множественной линейной регрессии, регрессии опорных векторов, случайного леса и радиальных базисных нейронных сетей. Построенные модели были протестированы на основе имеющихся реальных данных почасового потребления электроэнергии предприятием. Был проведен анализ, визуализация и сравнение полученных результатов.

Для реализации программной части был использован язык программирования Python 3.6 и библиотеки NumPy для работы с массивами и математическими функциями, Pandas для представления и обработки данных в виде таблиц, Scikit Learn для реализации алгоритмов машинного обучения, Keras для реализации нейронных сетей и Matplotlib для визуализации данных.

Для улучшения полученного результата можно рассмотреть добавление дополнительных данных, таких как график работы предприятия и данные о внутренних процессах. Также можно рассмотреть комбинированные методы и разбиение данной задачи на несколько подзадач и построение отдельных моделей для каждой из этих подзадач, например, строить отдельный прогноз для базового электропотребления предприятия и электропотребления оборудования.

Список литературы

[1] Hong T. Short Term Electric Load Forecasting : дис. – North Carolina State University, 2010.

[2] Грицай А. С. Гибридный метод краткосрочного прогнозирования потребления электрической энергии для энергосбытового предприятия с учетом метеофакторов : дис. – Омск, 2017.

[3] Hong T., Fan S. Probabilistic electric load forecasting: A tutorial review //International Journal of Forecasting. – 2016. – Т. 32. – №. 3. – С. 914-938.

[4] Bracale A. et al. Short-term industrial reactive power forecasting //International Journal of Electrical Power & Energy Systems. – 2019. – Т. 107. – С. 177-185.

[5] Zhang W. Y. et al. Application of SVR with chaotic GASA algorithm in cyclic electric load forecasting //Energy. – 2012. – Т. 45. – №. 1. – С. 850-858.

[6] Duan P. et al. Short-term load forecasting for electric power systems using the PSO-SVR and FCM clustering techniques //Energies. – 2011. – Т. 4. – №. 1. – С. 173-184.

[7] Chen B. J. et al. Load forecasting using support vector machines: A study on EUNITE competition 2001 //IEEE transactions on power systems. – 2004. – Т. 19. – №. 4. – С. 1821-1830.

[8] Alamaniotis M., Ikonomopoulos A., Tsoukalas L. H. A Pareto optimization approach of a Gaussian process ensemble for short-term load forecasting //2011 16th International Conference on Intelligent System Applications to Power Systems. – IEEE, 2011. – С. 1-6.

[9] Mori H., Ohmi M. Probabilistic short-term load forecasting with Gaussian processes //Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems. – IEEE, 2005. – С. 6 pp.

[10] Blum M., Riedmiller M. Electricity demand forecasting using gaussian processes //Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence. – 2013.

[11] Weron R. Modeling and forecasting electricity loads and prices: A statistical approach. – John Wiley & Sons, 2007. – T. 403.

[12] Yun K. et al. Building hourly thermal load prediction using an indexed ARX model //Energy and Buildings. – 2012. – T. 54. – C. 225-233.

[13] Gaillard P., Goude Y., Nedellec R. Additive models and robust aggregation for GEFCom2014 probabilistic electric load and electricity price forecasting //International Journal of forecasting. – 2016. – T. 32. – №. 3. – C. 1038-1050.

[14] Tarsitano A., Amerise I. L. Short-term load forecasting using a two-stage sarimax model //Energy. – 2017. – T. 133. – C. 108-114.

[15] Chang W. Y. et al. Short-term load forecasting using radial basis function neural network //Journal of Computer and Communications. – 2015. – T. 3. – №. 11. – C. 40.

[16] Kulkarni S., Simon S. P., Sundareswaran K. A spiking neural network (SNN) forecast engine for short-term electrical load forecasting //Applied Soft Computing. – 2013. – T. 13. – №. 8. – C. 3628-3635.

[17] Ding N. et al. Neural network-based model design for short-term load forecast in distribution systems //IEEE transactions on power systems. – 2015. – T. 31. – №. 1. – C. 72-81.

[18] Khwaja A. S. et al. Improved short-term load forecasting using bagged neural networks //Electric Power Systems Research. – 2015. – T. 125. – C. 109-115.

[19] Cai M., Pipattanasomporn M., Rahman S. Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques //Applied Energy. – 2019. – T. 236. – C. 1078-1088.

[20] Kim Y., Son H., Kim S. Short term electricity load forecasting for institutional buildings //Energy Reports. – 2019. – T. 5. – C. 1270-1280.

[21] Osczevski R., Bluestein M. The new wind chill equivalent temperature chart //Bulletin of the American Meteorological Society. – 2005. – T. 86. – №. 10. – C. 1453-1458.

[22] Mao Y., Yang F., Wang C. Application of BP network to short-term power load forecasting considering weather factor //2011 International Conference on Electric Information and Control Engineering. – IEEE, 2011. – C. 172-175.

[23] Bunnoon P., Chalermyanont K., Limsakul C. Mid-term load forecasting: Level suitably of wavelet and neural network based on factor selection //Energy Procedia. – 2012. – T. 14. – C. 438-444.

[24] Papalexopoulos A. D., Hao S., Peng T. M. An implementation of a neural network based load forecasting model for the EMS //IEEE transactions on Power Systems. – 1994. – T. 9. – №. 4. – C. 1956-1962.

[25] Rahman S. Formulation and analysis of a rule-based short-term load forecasting algorithm //Proceedings of the IEEE. – 1990. – T. 78. – №. 5. – C. 805-816.

[26] Amjady N., Daraeepour A. Mixed price and load forecasting of electricity markets by a new iterative prediction method //Electric power systems research. – 2009. – T. 79. – №. 9. – C. 1329-1336.

[27] Niu D., Shi H., Wu D. D. Short-term load forecasting using bayesian neural networks learned by Hybrid Monte Carlo algorithm //Applied Soft Computing. – 2012. – T. 12. – №. 6. – C. 1822-1827.

[28] Singh P., Dwivedi P. Integration of new evolutionary approach with artificial neural network for solving short term load forecast problem //Applied energy. – 2018. – T. 217. – C. 537-549.

[29] Sudheer G., Suseelatha A. Short term load forecasting using wavelet transform combined with Holt–Winters and weighted nearest neighbor models //International Journal of Electrical Power & Energy Systems. – 2015. – T. 64. – C. 340-346.

[30] Liu X., Zhang Z., Song Z. A comparative study of the data-driven day-ahead hourly provincial load forecasting methods: From classical data mining to deep learning //Renewable and Sustainable Energy Reviews. – 2020. – T. 119. – C. 109632.

[31] İskenderoğlu F. C. et al. Comparison of support vector regression and random forest algorithms for estimating the SOFC output voltage by considering hydrogen flow rates //International Journal of Hydrogen Energy. – 2020. – T. 45. – №. 60. – C. 35023-35038.

[32] Breiman L. Random forests //Machine learning. – 2001. – T. 45. – №. 1. – C. 5-32.

Приложение

Приложение 1

Программный код реализации слоя сети с радиальными базисными функциями с помощью библиотеки Keras.

```
class RBFLayer(Layer):
    def __init__(self, units, gamma, **kwargs):
        super(RBFLayer, self).__init__(**kwargs)
        self.units = units
        self.gamma = K.cast_to_floatx(gamma)

    def build(self, input_shape):
        self.mu = self.add_weight(name='mu',
                                  shape=
                                  (int(input_shape[1]), self.units),
                                  initializer='uniform',
                                  trainable=True)
        super(RBFLayer, self).build(input_shape)

    def call(self, inputs):
        diff = K.expand_dims(inputs) - self.mu
        l2 = K.sum(K.pow(diff, 2), axis=1)
        res = K.exp(-1 * self.gamma * l2)
        return res

    def compute_output_shape(self, input_shape):
        return (input_shape[0], self.units)
```