Санкт-Петербургский государственный университет
**Кафедра математической теории игр
и статистических решений**

Бакутеев Антон Николаевич

Магистерская диссертация

Приложения теории игр к системам водоснабжения

Направление 01.04.02
«Прикладная математика и информатика»
Магистерская программа «Исследование операций и системный анализ»

Научный руководитель,
доктор физ.-мат. наук,
профессор
Петросян Л. А.

Санкт-Петербург
2017

Содержание

1 Введение

Автоматизация учета потребления в системах водоснабжения позволяет в реальном времени получать статистики о качестве и надежности подачи воды, а также эффективности ее транспортировки. Большой объём данных позволяет применить различные методы машинного обучения и решить ряд оптимизационных задач.

Первая часть работы посвящена оптимизации ценовой политики водоснабжающего предприятия для уменьшения высоты пиков и увеличения прибыли. Вторая часть работы посвящена восстановлению измеренных данных для неавтоматизированных или вышедших из строя приборов учета воды.

1.1 Система водоснабжения

Данная работа выполненна с целью провести исследование и решить ряд поставленных задач для предприятия ГУП «Водоканал Санкт-Петербурга», являющимся заказчиком компании Siemens. ГУП «Водоканал Санкт-Петербурга» (Водоканал) — государственное унитарное предприятие, обеспечивающее услугами водоснабжения (только холодная вода) и водоотведения город Санкт-Петербург.

В систему водоснабжения входят по данным на 2016 год:

- 7104,2 км водопроводных сетей
- 193 повысительных насосных станций
- 9 водопроводных станций (крупнейшие — Южная водопроводная станция, Северная водопроводная станция, Главная водопроводная станция)
- 2 завода по производству гипохлорита натрия

Конечные потребители Водоканала — дома. Совместно с компанией Siemens была внедрена система мониторинга водопотребления. Каждый дом может иметь несколько входов для воды, которые далее будем называть точками предоставления услуг (ТПУ). На каждой ТПУ крепятся

датчики, которые позволяют измерять давление в трубах и объем проходящей воды за интервал времени равный одному часу. С помощью этих датчиков и были получены измерения. Также имеется один дом, в котором датчики установлены в каждой квартире.

2 Снижение пика потребления воды

Согласно статистическим данным потребление воды пользователями в среднем происходит по установившемуся временному портрету, в котором наблюдается пик потребления в дневные часы и минимум потребления в ночные часы. Проблема заключается в том, что требуется постоянная поддержка в трубах высокого давления для того, чтобы в часы максимального потребления обеспечить достаточный напор воды. Вследствие этого возрастают затраты на электроэнергию, а также создается избыточное давление в трубах, что значительно увеличивает утечки воды [1].

2.1 Обзор литературы

В настоящее время известно небольшое количество математических работ по теме снижения пикового потребления в сетях водоснабжения. Однако тема снижения пиковой нагрузки очень популярна в контексте «умных сетей» электроснабжения (smart grids). В [2] представлена распределенная система управления спросом для снижения пикового спроса в умных сетях. В ней доказано, что теоретико-игровая модель системы является потенциальной игрой, а также предложены стратегии, которые сходятся к равновесию по Нэшу. В [3] также исследовался вопрос о снижении пика в умных сетях. В ней авторы предложили решение с использованием батарей. В [4] поднята тема зависимости высоты пика потребления воды от количества пользователей.

В данной работе предлагается теоретико-игровой подход с явно заданным решением в некоторых частных случаях, а также предложен алгоритм и произведена оценка его эффективности в общем случае.

2.2 Постановка задачи

Элементами модели являются компания, которая управляет тарифами на воду, и пользователи (жители домов). Цель компании — назначить тарифы так, чтобы уменьшить ежедневные пики водопотребления. Цели пользователей — потребить определенное количество воды с минимальными затратами, которые они могут снизить, распределив потребление воды в зависимости от тарифов.

2.2.1 Общая модель системы водоснабжения

Итак, пусть имеется $\mathcal{H}$ домов, в каждом доме h находится упорядоченное множество $\mathcal{A}^{(h)}$ приборов, потребляющих воду, которые будут включены за день. При этом время включения некоторых приборов может быть выбрано пользователем, например, время включения стиральной машины. Обозначим это множество приборов как $\mathcal{SA}_h$. Тогда как время включения других приборов $\mathcal{FA}_h$ изменить нельзя, например, открытие крана для утреннего умывания.

Представим день в виде объединения множества $\mathcal{T}$ непересекающихся промежутков, на которых компанией будут установлены цены. Тогда каждый прибор $i \in \mathcal{A}^{(h)}$ должен быть запущен на $d_i^{(h)}$ последовательных промежутках. Обозначим за $ST_i^{(h)}$ временной интервал, на котором прибор может быть включен, а за $ET_i^{(h)}$ — временной интервал, отвечающий времени выключения прибора. Тогда, если прибор относится к множеству $\mathcal{SA}_h$, то $ST_i^{(h)} \leqslant ET_i^{(h)} - d_i^{(h)} + 1$, иначе $ST_i^{(h)} = ET_i^{(h)} - d_i^{(h)} + 1$. Далее последовательность интервалов $ST_i^{(h)}, \ldots, ET_i^{(h)}$ будем называть *допустимым временем работы*.

Рассмотрим теперь потребление воды прибором за время его работы. Оно может быть неравномерно распределено во времени, например, как у стиральной машины. Поэтому обозначим фазы водопотребления этого прибора через $\mathcal{F}$. Это подмножество промежутков разбиения $\mathcal{T}$, на которых водопотребление приборов больше нуля. При этом $d_i^{(h)} = |\mathcal{F}_i^{(h)}|$. В каждой фазе $f \in \mathcal{F}$ потребление воды прибором $i \in \mathcal{A}^{(h)}$ обозначим через $l_i^{(h)}(f)$. Так как рассматриваются только дискретные промежутки времени, то потребление воды прибором за данный промежуток рассчи-

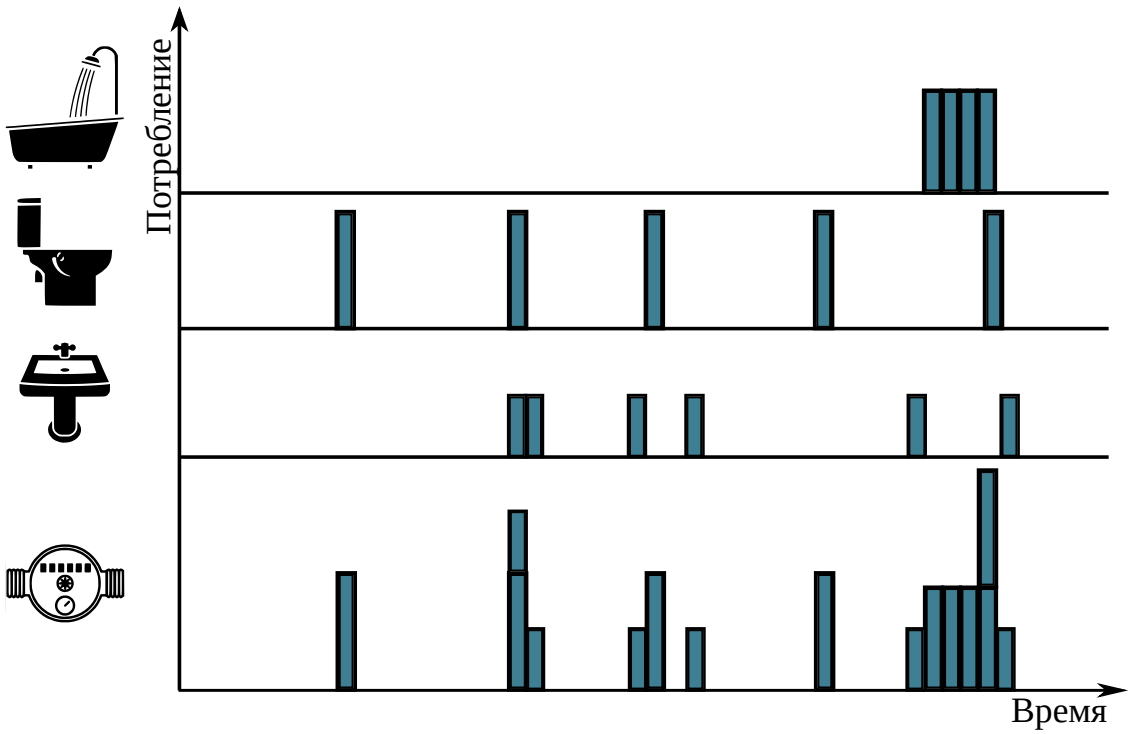


Рис. 1: Возникновение пиков

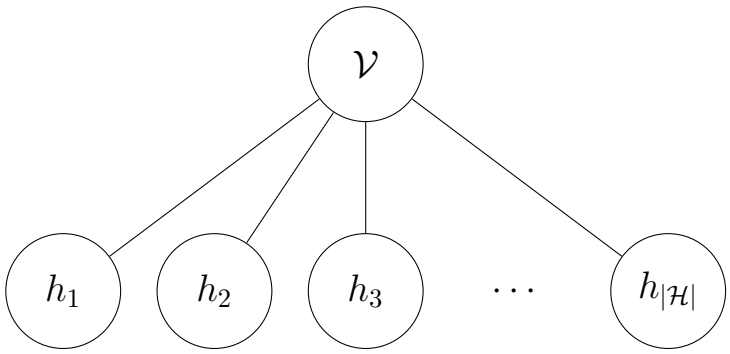
тывается как среднее потребление за весь промежуток.

Со стороны пользователя управление прибором $i \in \mathcal{A}^{(h)}$ в момент времени $t \in \mathcal{T}$ представлено индикаторами $a_i^{(h)}(t)$. Если прибор i пользователь включает в момент t, то $a_i^{(h)}(t) = 1$, иначе $a_i^{(h)}(t) = 0$. Управлять временем включения пользователь может лишь для приборов из $\mathcal{SA}_h$. Для приборов из $\mathcal{FA}_h$ переменные $a_i^{(h)}(t)$ фиксированы.

Компания управляет ценой $c_t(\cdot)$ на воду момент $t \in \mathcal{T}$. Ее целью является снижение ежедневного пика потребления воды, который возникает из-за одновременного включения приборов. Схематическое изображение задачи представлено на Рис. 1.

2.3 Иерархическая модель системы водоснабжения

Рассмотрим игру в рамках иерархической модели Штакельберга, где лидером является водоснабжающая компания $\mathcal{V}$, а последователями пользователи $h \in \mathcal{H}$.



Компания $\mathcal{V}$ может устанавливать разные тарифы для разных типов потребителей, например, для многоэтажных жилых домов и тепловых энергостанций. Стратегия $\mathcal{V}$ это вектор-функция $c(t) \in C$, определенная на дискретном множестве интервалов времени $\mathcal{T}$, где

$$C = \Big\{ c(t) = \big(c_1(t), \ldots, c_{|\mathcal{H}|}(t)\big) : \mathcal{T}^{|\mathcal{H}|} \to R^{|\mathcal{H}|} \;\Big|\; c_i(t) \le b_i, \quad \forall i \in \mathcal{H}, \quad \forall t \in \mathcal{T} \Big\}. \quad (1)$$

Значения b_i можно интерпретировать как ограничения на тариф со стороны государства, например, для того, чтобы не были установлены слишком высокие цены. Других ограничений на тарифы нет.

После того как тарифы заданы водоснабжающей компанией, пользователи выбирают свои стратегии. Стратегия пользователя $h \in \mathcal{H}$ — это вектор-функция $a_t^{(h)} \in A^{(h)}$, определенная на дискретном множестве интервалов времени $\mathcal{T}$, где

$$A^{(h)} = \Big\{ a^{(h)}(t) = \Big(a_1^{(h)}(t), \ldots, a_{|\mathcal{A}^{(h)}|}^{(h)}(t)\Big) : \mathcal{T}^{|\mathcal{A}^{(h)}|} \to \{0,1\}^{|\mathcal{A}^{(h)}|} \;\Big|\; \sum_{t=ST_i^{(h)}}^{ET_i^{(h)}-d_i^{(h)}+1} a_i^{(h)}(t) = 1, \quad \forall i \in \mathcal{A}^{(h)} \Big\}. \quad (2)$$

Ограничение возникает из-за того, что рассматриваются лишь приборы, которые нужно запустить ровно один раз в день. Если прибор в действительности нужно запустить несколько раз, то в данной модели для него создаются фиктивные приборы со своим допустимым временем работы. Заметим, что выбор тарифа компанией $\mathcal{V}$ не накладывает ограничений на множество стратегий пользователя.

Определим функции выигрышей игроков. При фиксированных тарифах $c(\cdot)$ каждый из пользователей $h \in \mathcal{H}$ минимизирует свою целевую функцию, которая отвечает цене за воду

$$U^{(h)}(a^{(h)}(\cdot)|c^{(h)}(\cdot)) = \sum_{i \in \mathcal{A}^{(h)}} \sum_{t=ST_i^{(h)}}^{ET_i^{(h)}} \sum_{f=1}^{\min\{t, d_i^{(h)}\}} c^{(h)}(t) a_i^{(h)}(t-f+1) l_i^{(h)}(f). \quad (3)$$

где $l_i^{(h)}(f)$ — потребление воды прибором $i \in \mathcal{A}^h$ в фазе f. Правая часть равенства имеет такой вид, поскольку в момент t прибор $i \in \mathcal{A}^h$ имеет потребление $l_i^{(h)}(f)$ в том и только том случае, если он был запущен в момент $t-f+1$. Обозначим оптимальную стратегию пользователя $h \in \mathcal{H}$ при установленных тарифах через

$$\overline{a}^{(h)}(\cdot|c^{(h)}) = \min_{a^{(h)}(\cdot) \in A_h} U^{(h)}\left(a^{(h)}(\cdot)|c^{(h)}(\cdot)\right).$$

Оптимальные стратегии, выбранные пользователями при установленных тарифах, обозначим через $I = \left\{\overline{a}^{(h)}(\cdot|c^{(h)})\right\}_{h \in \mathcal{H}}$.

Для компании $\mathcal{V}$ целевой функцией является пик потребления

$$W(c(\cdot)|I) = \max_{t \in \mathcal{T}} \sum_{h \in \mathcal{H}} \left[\sum_{i \in \mathcal{A}^{(h)}} \sum_{f=1}^{\min\{t, d_i^{(h)}\}} \overline{a}_i^{(h)}(t-f+1|c^{(h)}) l_i^{(h)}(f). \right] \quad (4)$$

Обозначим оптимальный тариф через $\overline{c}(\cdot) = \min_{c(\cdot) \in C} W(c(\cdot)|I)$.

Заметим что задача (2), (3) является параметрической задачей целочисленного линейного программирования. Задача (1), (4) носит нелинейный характер. Решением этих задач являются $\overline{a}^{(h)}(\cdot|c^{(h)})$ и $\overline{c}(\cdot)$ соответственно. В [5] доказано, что указанные стратегии образуют ситуацию равновесия по Нэшу при условии, что компании $\mathcal{V}$ известны стратегии пользователей при заданных тарифах.

2.4 Решение задачи минимизации пика

Перепишем (3) в виде

$$U^{(h)}(a^{(h)}(\cdot)|c^{(h)}(\cdot)) = \sum_{i\in\mathcal{A}^{(h)}} \sum_{t=ST_i^{(h)}}^{ET_i^{(h)}} \sum_{f=1}^{\min\{t,d_i^{(h)}\}} c^{(h)}(t)a_i^{(h)}(t-f+1)l_i^{(h)}(f)$$

$$= \sum_{i\in\mathcal{A}^{(h)}} \sum_{f=1}^{\min\{ST_i^{(h)},d_i^{(h)}\}-1} \sum_{t=ST_i^{(h)}}^{ET_i^{(h)}} c^{(h)}(t)a_i^{(h)}(t-f+1)l_i^{(h)}(f)$$

$$+ \sum_{i\in\mathcal{A}^{(h)}} \sum_{f=\min\{ST_i^{(h)},d_i^{(h)}\}}^{d_i^{(h)}} \sum_{t=f}^{ET_i^{(h)}} c^{(h)}(t)a_i^{(h)}(t-f+1)l_i^{(h)}(f)$$

$$= \sum_{i\in\mathcal{A}^{(h)}} \sum_{f=1}^{\min\{ST_i^{(h)},d_i^{(h)}\}-1} \sum_{p=ST_i^{(h)}-f+1}^{ET_i^{(h)}-f+1} c^{(h)}(p+f-1)a_i^{(h)}(p)l_i^{(h)}(f)$$

$$+ \sum_{i\in\mathcal{A}^{(h)}} \sum_{f=\min\{ST_i^{(h)},d_i^{(h)}\}}^{d_i^{(h)}} \sum_{p=1}^{ET_i^{(h)}-f+1} c^{(h)}(p+f-1)a_i^{(h)}(p)l_i^{(h)}(f).$$

Так как $a_i^{(h)}(t) = 0$ для $\forall t \notin [ST_i^{(h)}, ET_i^{(h)} - d_i^{(h)} + 1]$, то последнее равенство можно переписать следующим образом.

$$U^{(h)}(a^{(h)}(\cdot)|c^{(h)}(\cdot)) = \sum_{i\in\mathcal{A}^{(h)}} \sum_{f=1}^{\min\{ST_i^{(h)},d_i^{(h)}\}-1} \sum_{p=ST_i^{(h)}}^{ET_i^{(h)}-d_i^{(h)}+1} c^{(h)}(p+f-1)a_i^{(h)}(p)l_i^{(h)}(f)$$

$$+ \sum_{i\in\mathcal{A}^{(h)}} \sum_{f=\min\{ST_i^{(h)},d_i^{(h)}\}}^{d_i^{(h)}} \sum_{p=ST_i^{(h)}}^{ET_i^{(h)}-d_i^{(h)}+1} c^{(h)}(p+f-1)a_i^{(h)}(p)l_i^{(h)}(f)$$

$$= \sum_{i\in\mathcal{A}^{(h)}} \sum_{f=1}^{d_i^{(h)}} \sum_{p=ST_i^{(h)}}^{ET_i^{(h)}-d_i^{(h)}+1} c^{(h)}(p+f-1)a_i^{(h)}(p)l_i^{(h)}(f)$$

$$= \sum_{i\in\mathcal{A}^{(h)}} \sum_{t=ST_i^{(h)}}^{ET_i^{(h)}-d_i^{(h)}+1} a_i^{(h)}(t) \sum_{f=1}^{d_i^{(h)}} c^{(h)}(t+f-1)l_i^{(h)}(f).$$

Отсюда можно заключить, что минимум целевой функции дости-

гается тогда и только тогда, когда

$$\overline{a}_i^{(h)}(t|c^{(h)}) = \begin{cases} 1, & t = \underset{t\in[ST_i^{(h)},ET_i^{(h)}-d_i^{(h)}+1]}{\arg\min} \sum\limits_{f=1}^{d_i^{(h)}} c^{(h)}(t+f-1)l_i^{(h)}(f), \\ 0, & \text{иначе.} \end{cases}$$

Обозначим оптимальное время включения приборов через

$$\overline{t}_i^{(h)}(c^{(h)}) = \underset{t\in[ST_i^{(h)},ET_i^{(h)}-d_i^{(h)}+1]}{\arg\min} \sum_{f=1}^{d_i^{(h)}} c^{(h)}(t+f-1)l_i^{(h)}(f).$$

Минимальный возможный пик достигается в случае, если пользователи выберут стратегии, которые являются решением задачи целочисленного линейного программирования (5), где введена дополнительная переменная L для линейности целевой функции.

$$\begin{aligned} & L \to \min; \\ & L \geq \sum_{h\in\mathcal{H}} \sum_{i\in\mathcal{A}^{(h)}} \sum_{f=1}^{\min\{t,d_i^{(h)}\}} a_i^{(h)}(t-f+1)l_i^{(h)}(f); \\ & \sum_{t=ST_i^{(h)}}^{ET_i^{(h)}-d_i^{(h)}+1} a_i^{(h)}(t) = 1, \quad \forall h \in \mathcal{H}, \quad \forall i \in \mathcal{A}^{(h)}. \end{aligned} \tag{5}$$

Тогда для минимизации пика компании достаточно установить такой тариф, что $\overline{t}_i^{(h)}(c^{(h)}) = \tau_i^{(h)}$, то есть

$$\underset{t\in[ST_i^{(h)},ET_i^{(h)}-d_i^{(h)}+1]}{\arg\min} \sum_{f=1}^{d_i^{(h)}} c^{(h)}(t+f-1)l_i^{(h)}(f) = \tau_i^{(h)}.$$

Что равносильно

$$\sum_{f=1}^{d_i^{(h)}} c^{(h)}(\tau_i^{(h)}+f-1)l_i^{(h)}(f) \leq \sum_{f=1}^{d_i^{(h)}} c^{(h)}(t+f-1)l_i^{(h)}(f)$$
$$\forall t \in [ST_i^{(h)}, ET_i^{(h)} - d_i^{(h)} + 1].$$

Но данная задача не всегда имеет решения относительно $c^{(h)}$, поскольку для приборов, имеющих одинаковые допустимые интервалы времени запуска и одинаковые профили потребления при фиксированных тарифах время запуска будет одним и тем же, если пользователи действуют оптимально.

2.4.1 Случай единичных профилей потребления.

Так как задача (1), (4) является нелинейной и имеет разрывы, то ее решение стандартными способами затруднено. Для решения методом перебора даже в простейшем случае, когда $d_i^{(h)} = 1$ и $l_i^{(h)}(1) = 1$ для $\forall h \in \mathcal{H}$, $\forall i \in \mathcal{A}^{(h)}$ требуется перебор $|\mathcal{T}|!$ вариантов. Так как в этом случае

$$\overline{a}_i^{(h)}(t|c^{(h)}) = \begin{cases} 1, & t = \underset{t \in [ST_i^{(h)}, ET_i^{(h)}]}{\arg\min} c^{(h)}(t), \\ 0, & \text{иначе.} \end{cases}$$

То есть решение сводится к установлению порядка на $c^{(h)}(t)$. На практике при $|\mathcal{T}| = 24$ число вариантов превосходит $6 \cdot 10^{23}$, и поэтому такое решение является вычислительно слишком сложным.

Опишем алгоритм получения оптимальных тарифов при $|\mathcal{H}| = 1$. Для краткости опустим индекс h.

Алгоритм 1 Снижение пика для единичных профилей

1: Пусть задано множество различных цен, например, $\{c(t)\}_{t \in \mathcal{T}} = \overline{1, |\mathcal{T}|}$.
2: Для любого $t \in \mathcal{T}$ найти количество приборов m_t, чье допустимое время работы содержит t.
3: Установить минимальную цену в любой из моментов t, для которых число m_t минимально.
4: Удалить из расписания запуска те приборы, для которых допустимое время работы содержит t из пункта 3.
5: Перейти к пункту 2 для $\mathcal{T} = \mathcal{T} \setminus t$, если цены установлены не для всех $t \in \mathcal{T}$.

Теорема 1. *Если* $|\mathcal{H}| = 1$, $d_i = 1$, $l_i(1) = 1$ *для* $\forall i \in \mathcal{A}$, *то тариф, полученный с помощью Алгоритма 1 обеспечивает минимум высоты пика потребления.*

Доказательство. При $t = \arg\min_{t\in\mathcal{T}} c(t)$ потребление равно m_t. Поэтому минимальная возможная высота пика потребления всегда не меньше чем $\min_{t\in\mathcal{T}} m_t$ на первой итерации. Так как на каждой последующей итерации рассматриваются лишь приборы, чьё допустимое время работы не содержит t (поскольку, для оставшихся приборов время запуска равно t), то это же рассуждение можно применить и к остальным итерациям. И поэтому минимальная возможная высота пика потребления равна $\min_{t\in\mathcal{T}} m_t$ на одной из итераций. □

Замечание 1. Полученное решение не является единственным относительно порядка $c(t)$.

2.4.2 Случай профилей потребления произвольной высоты.

Случай единичных профилей потребления $d_i = 1$ и $l_i(1) = 1$ для $\forall i \in \mathcal{A}$ легко обобщить до случая профилей потребления произвольной высоты $d_i = 1$ для $\forall i \in \mathcal{A}$, а $l_i(1)$ произвольно. Действительно, два прибора с единичными профилями потребления и одинаковыми допустимыми интервалами запуска влияют на высоту пика абсолютно также, как один прибор с тем же допустимым интервалом запуска и $l_i(1) = 2$. Поэтому для произвольного $l_i(1)$ достаточно выбрать минимальную целочисленную единицу измерения и добавить $l_i(1)$ дополнительных приборов с теми же допустимыми интервалами запуска. Ниже представлен соответствующий Алгоритм 2.

2.4.3 Случай произвольных профилей потребления.

Решение в общем случае представляет собой более сложную задачу. Это связано с тем, что даже при условии совпадения допустимых интервалов запуска оптимальное время запуска приборов с разными профилями потребления может отличаться. Поэтому тарифы, используемые в ранее указанных случаях, не всегда являются оптимальными. Тем не менее на практике Алгоритм 2 может быть применен и в этом случае.

Алгоритм 2 Снижение пика для профилей произвольной высоты

1: Пусть задано множество различных цен, например, $\{c(t)\}_{t\in\mathcal{T}} = \overline{1, |\mathcal{T}|}$.
2: Для любого $t \in \mathcal{T}$ найти максимальное возможное потребление m_t.
3: Установить минимальную цену в любой из моментов t, для которых m_t минимально.
4: Удалить из расписания запуска те приборы, для которых допустимое время работы содержит t из пункта 3.
5: Перейти к пункту 2 для $\mathcal{T} = \mathcal{T} \setminus t$, если цены установлены не для всех $t \in \mathcal{T}$.

При этом максимально возможное потребление в пункте 2 понимается как сумма максимальных значений профилей приборов в данный момент. Покажем, что Алгоритм 2 позволяет снизить высоту пика потребления с помощью компьютерного моделирования.

Генерация случайных расписаний потребления. Построим расписание с помощью генерации случайных профилей потребления и допустимых времен работы. Список допустимых времен работы составим следующим образом: в качестве начала интервала $ST_i^{(h)}$ выберем равновероятно интервал из множества $\mathcal{T}$, а в качестве конца — равновероятно интервал из множества $\mathcal{T}$ не меньше $ST_i^{(h)}$. Список профилей составим так: длину профиля $d_i^{(h)}$ выберем равновероятно в пределах допустимого времени работы, а каждое потребление профиля $l_i^{(h)}(f)$ для заданного интервала зададим как случайную величину с равномерным распределением на промежутке $[0, 1]$.

- $ST_i^{(h)} \sim U(\mathcal{T}), \quad ET_i^{(h)} \sim U(ST_i^{(h)}, |\mathcal{T}|)$;
- $d_i^{(h)} \sim U(1, ET_i^{(h)} - ST_i^{(h)} + 1)$;
- $l_i^{(h)}(f) \sim U([0, 1])$.

Результаты вычислений. Для оценки эффективности Алгоритма 2 реализуем дополнительно функцию, вычисляющую лучший тариф с помощью перебора всех возможных перестановок упорядоченных цен анало-

гично случаю единичных потреблений. В силу ограниченных вычислительных ресурсов произведем расчет для $|\mathcal{T}| = 6$, $|\mathcal{H}| = 1$ и $|\mathcal{A}| = 50$. При фиксированном расписании эксперимент заключается в сравнении высоты пика без оптимизации с высотой пика при оптимизации с помощью Алгоритма 2 и метода перебора. Сгенерируем расписание случайным образом 1000 раз согласно предыдущему параграфу и вычислим среднее процентное соотношение высот пиков. Использование Алгоритма 2 позволяет снизить высоту пика в среднем на 13%. Но при этом высота пика при использовании алгоритма 2 на 6% выше, чем при использовании метода перебора. При $|\mathcal{T}| = 24$ произведена симуляция для 1000 приборов с потреблением от 1 до 3 часов. В этом случае Алгоритм 2 позволяет снизить пиковую нагрузку на 30%.

2.5 Применение к реальным данным

Проверим эффективность модели в реальном мире. По данным потребления для квартир оценим параметры модели и сравним высоту пика до оптимизации и после нее.

2.5.1 Валидация данных

Перед применением модели проведем статистический анализ, для того чтобы убедиться, в качестве данных и их достаточности для построения теоретико-игровой модели.

О данных. Были проанализированы данные по 13 квартирам в период с 1 марта 2012 года по 1 апреля 2013. В квартирах были установлены автоматические датчики интервального потребления. В данных приводится почасовое потребление для каждой квартиры.

Так как данных об имеющихся приборах и времени их запуска нет, то положим, что каждому прибору соответствует пик, то есть последовательные моменты времени на которых потребление больше нуля, а потребление на предшествующих и последующих моментах равно нулю.

Оценки пиков потребления. На Рис. 2 изображен пример реального потребления за сутки, на котором наблюдается пик с 10:00 до 15:00.

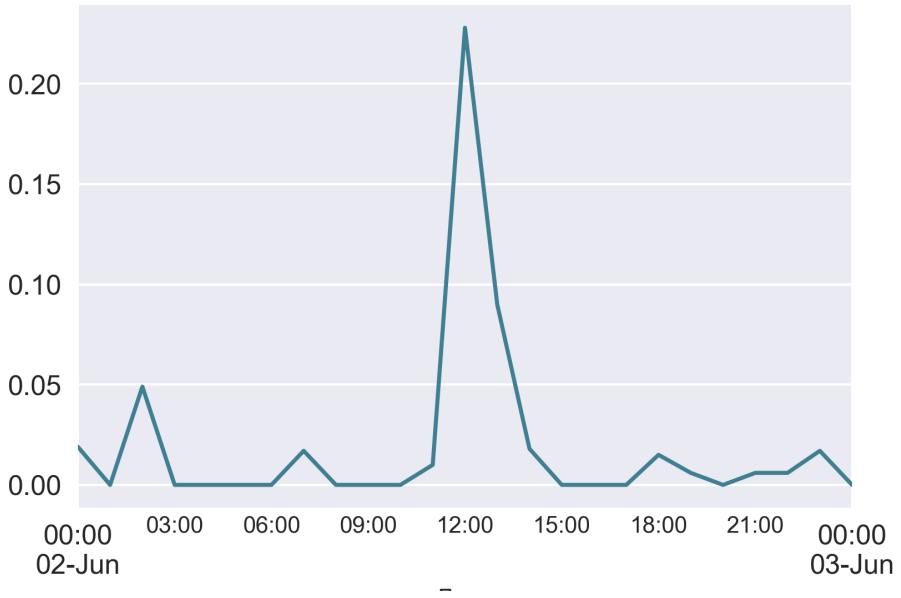


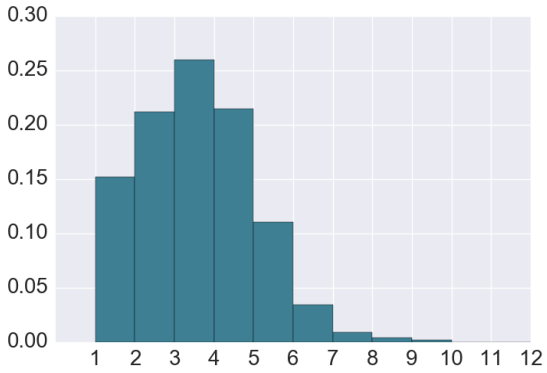
Рис. 2: Пример суточного потребления

Пик потребления характеризуется *временем начала*, *временем конца*, *высотой* (максимумом потребления), *длиной* (разностью в часах между началом потребления и концом) и *площадью* (суммарным потреблением). Также важным параметром является *количество пиков* в течение дня. Исследуем эти характеристики.

Сначала построим гистограммы и вычислим статистики для количества пиков и их длины в течении суток (Рис. 3, 4). При этом выбросим дни, где потребление нулевое, так как в нашей модели фиксированное количество игроков. Так как произведение среднего количества пиков на их среднюю длину меньше, чем количество часов в сутках, то возможно управление тарифами для их сдвига в области, где потребление отсутствует.

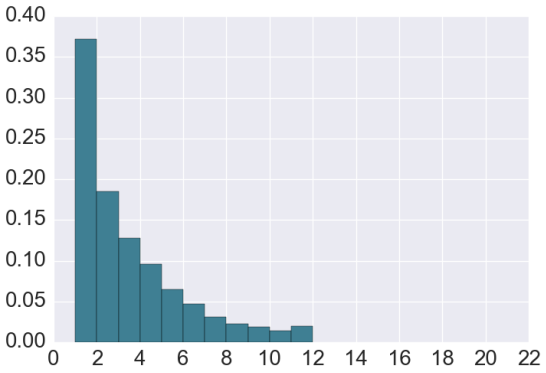
Почасовое потребление. Оценим суммарное потребление воды пользователями в каждый час. На Рис. 5 изображены соответствующие гистограммы, а также соответствующие графики плотностей логнормального распределения (подобранного с помощью оценки максимального правдоподобия).

Поставим нулевую гипотезу о том, что суммарное потребление в фиксированный час распределено в соответствии с логнормальным зако-



Статистика	Значение
Среднее	3.10
Медиана	3.00
Дисперсия	2.13

Рис. 3: Количество пиков



Статистика	Значение
Среднее	3.58
Медиана	2.00
Дисперсия	12.42

Рис. 4: Длина пиков

ном. Проверим эту гипотезу с помощью критерия Колмогорова-Смирнова. Тогда при уровне значимости 0.01 эта гипотеза может быть принята для всех часов, кроме 4. А при уровне значимости 0.05 для всех часов, кроме 3, 4, 7. Для этих часов, выборка представляет собой смесь распределений. Это связано с небольшим количеством исследуемых квартир и различными профилями потребления пользователей. Логнормальность потребления согласуется с результатами исследований в [6].

2.5.2 Генерация расписаний потребления.

Оценим параметры модели с помощью кластеризации исходных данных. Для каждого потребителя определим усредненные допустимые времена работы приборов и профили потребления. Опишем метод на примере одного фиксированного потребителя. Поскольку каждый пик потребления считается отдельным прибором, то представим данные в виде множества точек следующим образом. Для каждого пика потребления создадим точку, первые координаты которой отвечают времени начала пика и времени его конца в часах за день. Если потребление не заканчивается в течение дня, то искусственно разобьем пик потребления на 2 части: до полуночи и после полуночи. Это может привести к созданию лишних приборов, однако это необходимо сделать, так как в

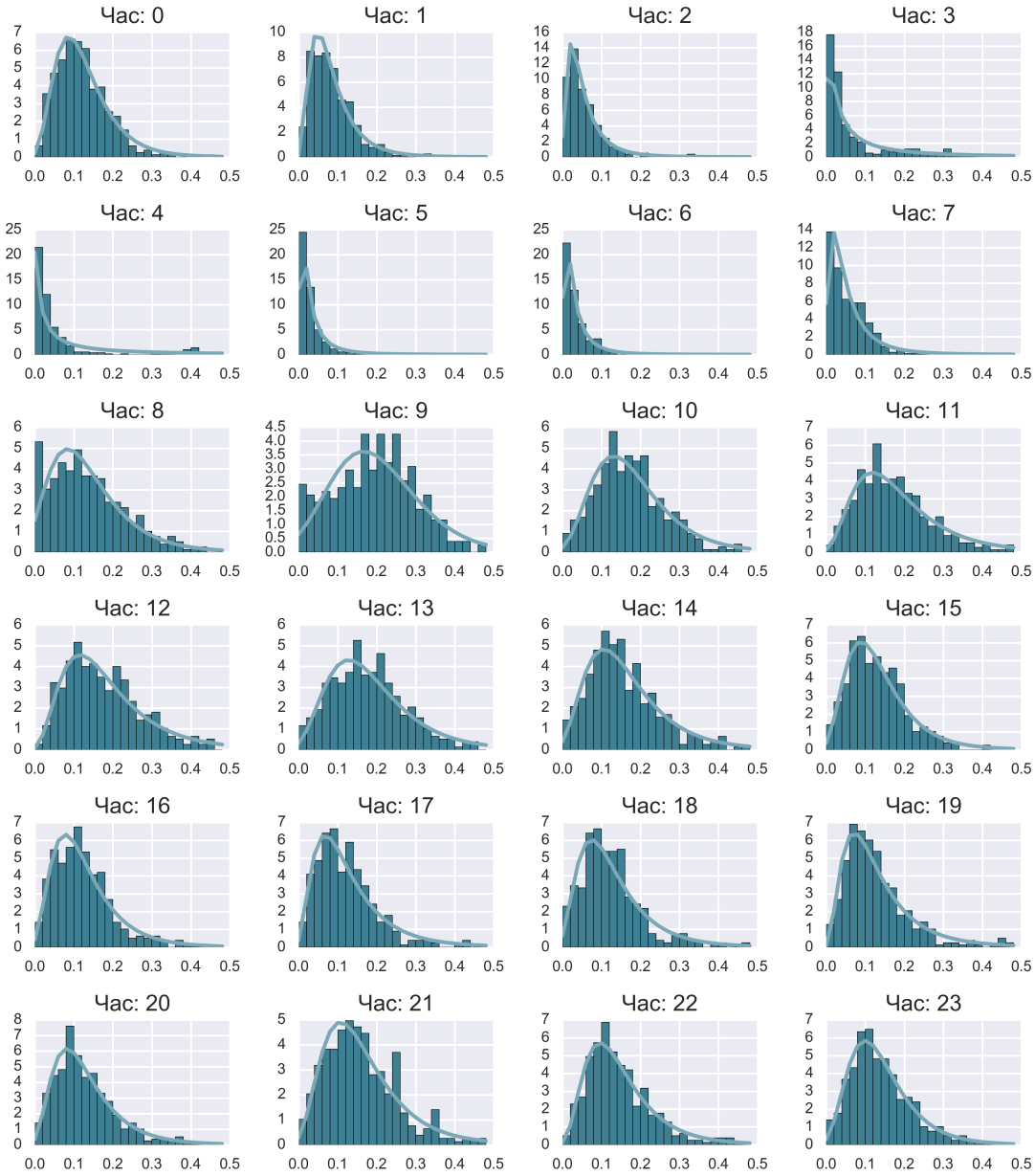


Рис. 5: Суммарное потребление в каждый час

рамках модели рассматривается оптимизация в течение суток. Остальные координаты точки заполним соответствующими последовательными почасовыми потреблениями пика:

$$\left(ST_i^{(h)}, ET_i^{(h)}, l_i^{(h)}(0), \ldots, l_i^{(h)}(d_i^{(h)})\right).$$

К новому набору данных применим алгоритм иерархической кластеризации по методу Варда [7], предварительно изменив данные так, чтобы каждый признак имел среднее равное 0 и дисперсию равную 1. Умножим времена начала и окончания на 2 для того, чтобы увеличить важ-

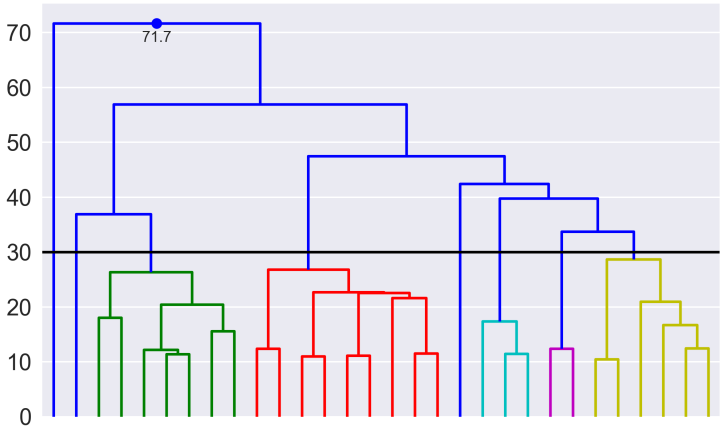


Рис. 6: Дендрограмма иерархической кластеризации (обрезанная)

ность временной зависимости пиков потребления. На Рис. 6 представлена дендрограмма для одного из потребителей, где черной линией отмечено минимальное расстояние, разделяющее кластеры. Для каждого потребителя произведем кластеризацию пиков потребления с 12 кластерами. Такое большое количество кластеров выбрано во избежание слишком длительных интервалов допустимого времени работы, а также на случай нехарактерного потребления пользователя. Например, в случае утечки будет наблюдаться положительное потребление в течение длительных промежутков времени. После применения кластеризации такие случаи попадут в отдельный кластер с небольшим количеством элементов.

Далее для каждого кластера оценим допустимое время начала и конца работы как, соответственно, первый квартиль первой координаты и третий квартиль второй координаты точек из данного кластера. Профили потребления оцениваются с помощью покоординатной медианной агрегации. На Рис. 7 изображены составляющие кластеров в виде прозрачных столбчатых диаграмм, наложенных друг на друга, и их усреднение. Наиболее часто встречающиеся потребления отображаются более ярко. Медианное усреднение позволяет избежать выбросов.

На Рис. 8 изображена гистограмма количества элементов в каждом кластере для выбранного потребителя. Для каждого потребителя отбросим кластеры с количеством элементов менее 5% от общего числа. Тогда среднее количество кластеров для потребителей равно $4,15$, что соот-

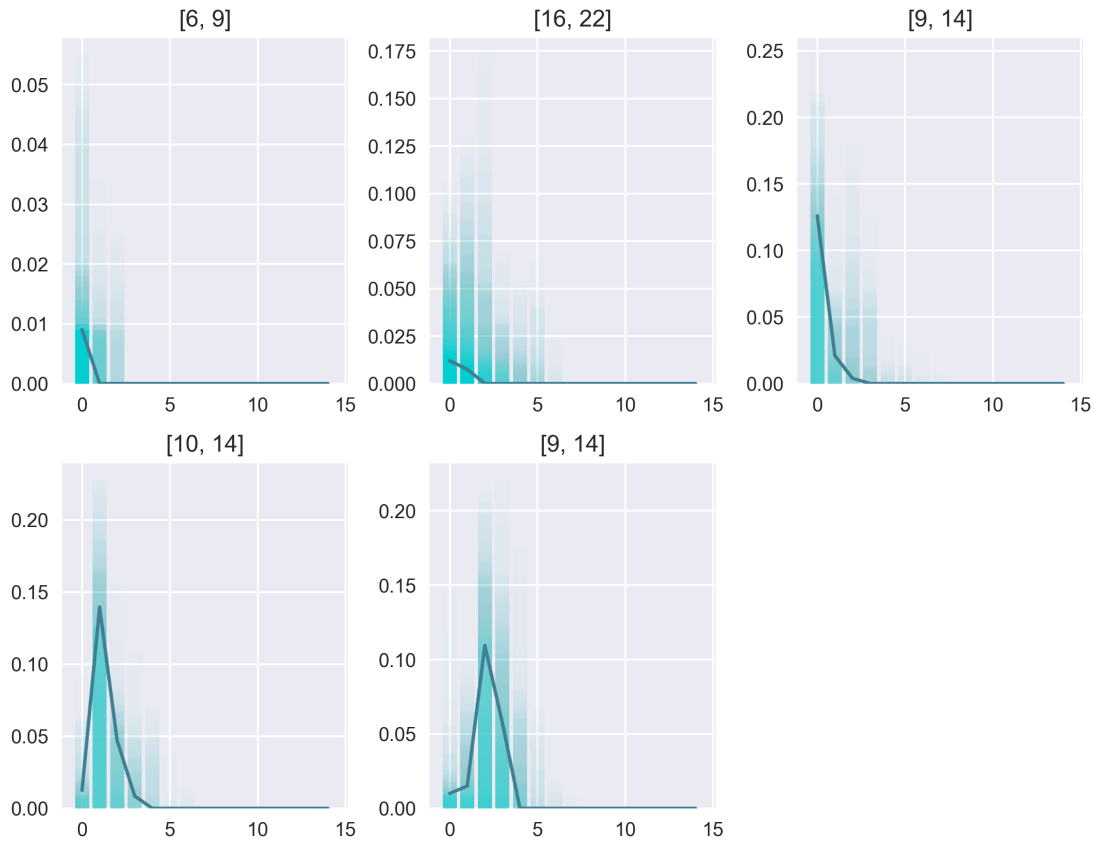


Рис. 7: Оценка параметров модели

ветствует Рис. 3, так как несколько кластеров могут соответствовать одному пику потребления.

Таким образом, данный метод кластеризации позволяет оценить параметры теоретико-игровой модели.

Результаты вычислений. После оценки параметров модели по статистическим данным согласно параграфу 2.5.2 произведем компьютерную симуляцию. Эксперимент заключается в сравнении высоты пиков потребления в случае оптимизации и в случае, когда время запуска прибора выбирается случайно и равновероятно из допустимого времени работы прибора. Было произведено 1000 экспериментов, которые показали, что применение Алгоритма 2 позволяет снизить суммарную высоту пика на 13%.

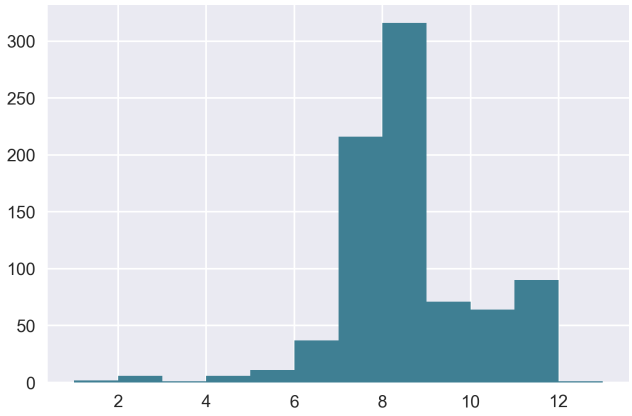


Рис. 8: Количество элементов в кластерах

2.6 Вывод

В данной работе представлена теоретико-игровая модель водоснабжающей компании и предложен алгоритм снижения высоты пика потребления. Также в работе доказано, что данный алгоритм является оптимальным в случае, если длительность потребления не превышает частоту дискретизации времени, выбранной компанией для установления тарифов. В общем случае произведена компьютерная симуляция, показывающая, что данный алгоритм позволяет снизить высоту пика до 30%. Проанализированы данные потребления для 13 квартир, предложен метод оценивания параметров модели и произведена оценка. Рассматриваемый алгоритм позволяет снизить высоту пика на 13% в этом случае. На основе полученных результатов можно заявить, что контроль высоты пикового потребления целесообразен, перспективен и требует дальнейших исследований.

2.6.1 Направления дальнейших исследований

Возможно несколько направлений исследования. Первый направление соответствует аналитическому решению оптимизационной задачи в общем случае, а также рассмотрению других постановок задач. Например, целесообразно рассмотреть кооперативный подход и другие виды равновесия, такие как равновесие в стратегиях наказания.

Второе направление соответствует обобщению модели. Например,

возможно включение дополнительных параметров, таких как тип потребителя, режим его дня, возраст, количества потребителей в квартире и др. В силу большого количества параметров аналитическое решение может быть невозможно. Поэтому целесообразно применить имитационное и агентное моделирование [8]. Возможно использование уже составленных моделей для задачи минимизации пика. Так в [9] была составлена модель потребления воды людьми с учетом размера семьи, привычек, уровня занятости и др. В [10] составлена имитационная модель городского электроснабжения и исследуется ее поведение при изменении параметров модели таких как время года, температура, тип дня и др.

Третье направление заключается в исследовании применимости модели на основе реальных данных. Для этого необходимо проанализировать данные потребления в большем масштабе и получить более точные оценки параметров модели. Поскольку в настоящее время невозможно получить данные потребления непосредственно с каждого прибора, то необходимо разработать способ оценки параметров модели в случае, когда потребления известны лишь на уровне суммарного потребления дома. В настоящий момент известно несколько методов декомпозиции потребления. Например, в [11] используются скрытые марковские модели для разложения потребления для отдельных приборов. Однако предложенный метод обладает большой вычислительной сложностью и непригоден для применения к данным агрегированным до уровня дома. В [12] описан алгоритм AFAMAP, который обладает меньшей ресурсоемкостью, но может быть применим лишь к высокочастотным данным, что также невозможно в настоящее время. В [13] представлен алгоритм декомпозиции на основе нейронных сетей, но его недостатком является наличие тренировочного множества, получение которого затруднительно. Таким образом, необходима разработка вычислительно эффективного алгоритма, не требующего тренировочного множества и высокочастотных данных.

3 Прогноз потребления

Для улучшения качества подачи воды и сокращения объемов утечек необходимо обеспечивать постоянный мониторинг использования воды на всех объектах. Установка автоматизированных датчиков требует значительных материальных и временных ресурсов. Кроме того, в случае выхода из строя автоматизированного прибора учета (ПУ) воды, неизбежна потеря данных. Поэтому в системе мониторинга потребления воды предусмотрена возможность ручного ввода показаний на каждой ТПУ. Данные с автоматизированных ТПУ поступают в систему с частотой равной одному часу. Данные с каналов ручного ввода заносятся в систему кумулятивно (т.е. каждое показание прибора равно сумме предыдущего показания и потребления между текущим показанием и предыдущим) с частотой примерно раз в месяц.

3.1 Постановка задачи

Задача состоит в оценке почасового потребления для неавтоматизированных приборов учета воды по имеющимся данным о доме, в котором расположена ТПУ, показаниям канала ручного ввода и почасовых значениях автоматизированных ТПУ. Предлагаемое ниже решение использует модели машинного обучения, каждая из которых обладает своими достоинствами и недостатками.

3.2 Описание данных

Выборка состоит из показаний 1596 каналов с 1 января 2016 года по 1 ноября 2016 года. Кроме измеренных значений для каналов доступна информация [14] о доме, в котором они расположены (Таблица 1).

3.3 Регрессия по дням

Положим горизонтальную размерность матрицы Y равной 24. Для каждого из интервальных каналов выпишем последовательно вдоль вертикальной оси почасовые показания по дням за весь промежуток време-

Название признака	Тип признака
тип дома (жилье, общежитие и т.п)	категориальный
тип абонента (многокв. дом, адм-ный и т.п.)	категориальный
серия	категориальный
тип проекта	категориальный
год постройки	численный
общая площадь здания	численный
площадь жилых помещений	численный
число этажей	численный
количество проживающих	численный

Таблица 1: Метаданные канала

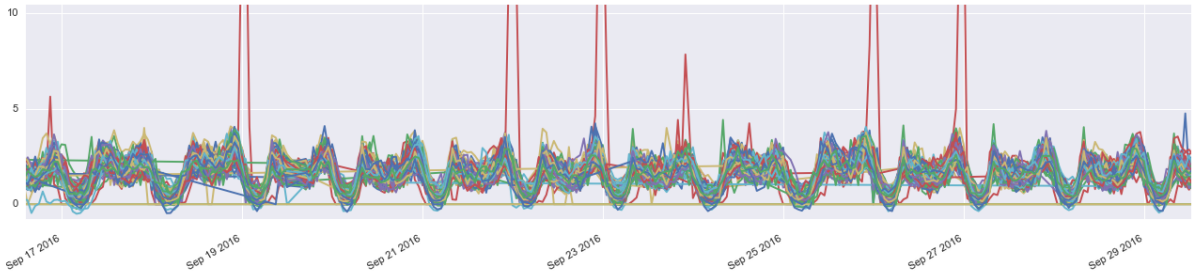


Рис. 9: Потребление

ни. В матрицу X выпишем признаки интервального канала. В качестве признаков возьмем данные о доме, в котором находятся соответствующие интервальный канал и канал ручного ввода. Для категориальных признаков применим прямое кодирование. То есть для каждого значения категории добавим соответствующий ей столбец, значения которого положим равным 1, если этот признак имеет место, иначе 0. А также в качестве признаков возьмем показания соответствующих каналов ручного ввода за последние 3 месяца от даты получения показаний интервального канала (метод временного окна для показаний ручного ввода). Поскольку потребление носит стационарный характер в течение недели (Рис. 9), за исключением различного профиля потребления по будням и выходным, то добавим соответствующий признак для выходных дней. Схематически матрицы X и Y представлены ниже.

Тип дома	...	КРВ1	КРВ2	КРВ3
...	...	...	...	...

Ч0	...	Ч23
...	...	...

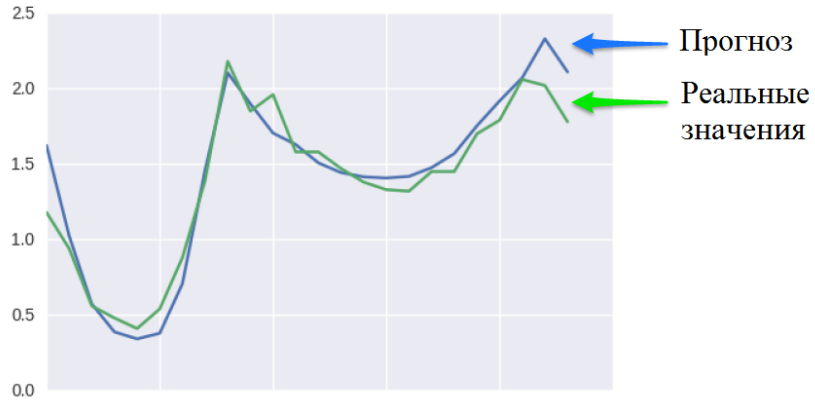


Рис. 10: Регрессия по дням

Таким образом ответом модели для заданного канала с признаками из матрицы X является прогноз потребления на день (Рис. 10)

В [15], [16] предлагается использовать измеренные данные с лагами также и в качестве признаков, что позволяет предсказывать по нескольким предыдущим измерениям следующее. В нашем случае это невозможно, поскольку по для каналов, по которым необходимо получить прогноз, имеются лишь метаданные о доме, в котором находится канал, и средние ежедневные потребления за предыдущие месяцы.

3.4 Используемые методы

Основными критериями при выборе методов являлись: точность, низкая вычислительная сложность и масштабируемость относительно количества данных, возможность получения многомерного ответа, возможность интерпретации результатов, устойчивость к выбросам. Использовался ряд стандартных методов машинного обучения, таких как Random Forest, нейронные сети, метод k ближайших соседей, линейную регрессию и их различные вариации и композиции. Для наглядной оценки эффективности представленных методов был использован модифицированный метод сезонного разложения.

3.4.1 Метод сезонного разложения

Вычислим для каждого типа дома (из матрицы признаков) *профиль потребления*. То есть, отдельно для будних дней и выходных сгруппируем данные по часам и для каждого часа найдем агрегирующую статисти-

ку в зависимости от функции потерь, например, среднее или медиану. Так как известно, что среднее минимизирует среднеквадратичное отклонение, а медиана среднее отклонение по модулю. В данном случае будем использовать медиану, так как она менее чувствительна к выбросам. Далее с периодом равным одной неделе развернем профиль на будние дни и выходные. При поступлении новых данных данный профиль будем перечитывать. Для прогноза неавтоматизированных каналов используем данные каналов ручного ввода. Выберем последние два измерения из соответствующего канала и изменим масштаб профиля потребления так, чтобы среднее дневное потребление по данным каналов ручного ввода было равно среднему дневному потреблению по новому профилю. Обозначим через $m(t)$ значения канала ручного ввода в момент t. Значения профиля потребления обозначим через $c(t)$. Тогда, если последние два измерения канала ручного ввода были получены в моменты t_1 и t_2, то умножим профиль потребления на

$$\frac{m(t_2) - m(t_1)}{\mathrm{days}(t_1, t_2) \sum\limits_{t \in [t_1, t_2]} c(t)}$$

где $\mathrm{days}(t_1, t_2)$ количество дней между t_1 и t_2. Далее этот метод будем называть *прогнозированием среднего ежедневного потребления с масштабированием (СЕПМ)*.

Достоинства:

- Простота и интерпретируемость результатов.
- Возможность обнаружения выбросов в измеренных значениях.
- Высокая скорость вычисления.
- При добавлении новых элементов в выборку не требует пересчета.

Недостатки:

- Высокая чувствительность к выбросам в каналах ручного ввода.
- Не использует метаданные.

- Высокая зависимость от выбора каналов, по которым считается профиль потребления. При этом выбор каналов необходимо произвести вручную.

3.4.2 Random Forest Regressor

Метод регрессии *Random Forest (RF)* [17] является композицией решающих деревьев. Метод обладает следующими достоинствами:

- Слабая зависимость от коррелированности входных данных.
- Высокая скорость вычисления.
- Легкая реализуемость параллельного вычисления.
- Независимость от масштаба признаков и типа переменной (числовая/категориальная).
- Устойчивость к выбросам.
- Оценка важности признаков.

Также имеются и недостатки:

- Большой размер обученных моделей требует значительного объема оперативной памяти.
- Метод плохо справляется с экстраполяцией за пределы выборки [18].

3.4.3 Extra Trees Regressor

Метод регрессии *Extra Trees (ET)* [19] похож на метод RF, за исключением того, что

1. При выборе признака для деления дерева используется вся тренировочная выборка, в отличие от бутстрапированной.
2. Деление происходит полностью случайным образом.

В следствие этих отличий, ET обладает следующими достоинствами и недостатками.

Достоинства:

- ET обладает более высокой обобщающей способностью, чем RF.

Недостатки:

- Требуется значительно больше памяти для хранения модели в сравнении с RF.

3.4.4 Метод k ближайших соседей

Метод k ближайших соседей (kNN) — метрический метод, основывающийся на гипотезе непрерывности.

Достоинства метода:

- Простота и наглядность.
- При увеличении выборки не требует полного пересчета.

Недостатки:

- Плохо работает при большой размерности пространства признаков («проклятье размерности»).
- Чувствительность к выбросам.
- Не создает модели регрессии.

3.4.5 Гребневая линейная регрессия

Метод *гребневой линейной регрессии (Ridge)* [20] является обобщением метода линейной регрессии. В случае мультиколлинеарности признаков у линейной регрессии проявляется высокий разброс коэффициентов и неустойчивость решения относительно шума. Для гребневой регрессии в оптимизирующий функционал вводится регуляризирующее слагаемые с коэффициентами, позволяющие избежать этого. Коэффициенты могут быть подобраны оптимальным образом с помощью поиска по решетке и кросс-валидации.

Данный метод не позволяет экстраполировать нелинейные зависимости, но хорошо проявляет себя в композициях алгоритмов, о которых будет сказано далее.

3.4.6 Нейронные сети

Преимущество нейронных сетей состоит в способности моделировать нелинейные связи без предварительных предположений о структуре данных [21]. Для задачи регрессии на последнем слое в нейронной сети всегда должна быть использована функция активации не изменяющая области допустимых значений для прогноза. Чаще всего используются линейная функция активации и выпрямленная линейная функция активации[1] (rectified linear unit, ReLU). Так как потребление может быть отрицательным (при течении воды в обратную сторону, например, во время аварийной ситуации) мы используем линейную функцию активации. Для обучения нейронной сети использовались алгоритмы Adagrad [22], Adadelta [23], RMSprop[24] и Adam[25]. При этом в текущих условиях наилучшая сходимость была достигнута с помощью алгоритма Adam.

Многослойный персептрон. *Многослойный персептрон* (Рис. 11) — нейронная сеть прямого распространения с полносвязными слоями, обучающаяся с помощью алгоритма обратного распространения ошибки [26].

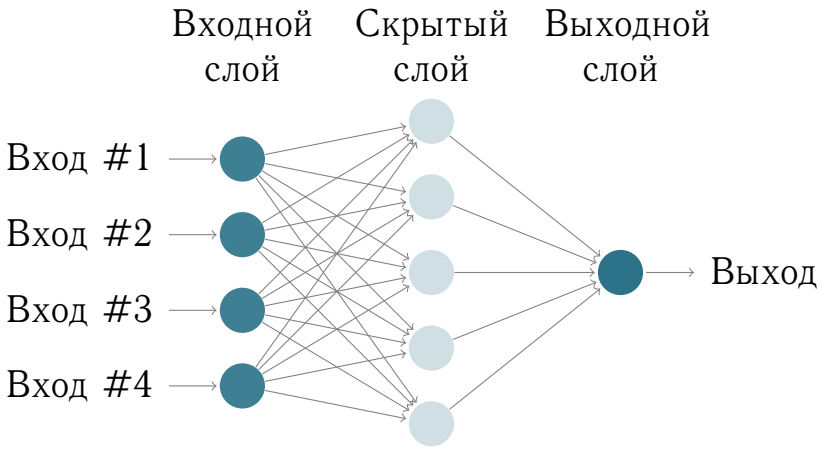


Рис. 11: Многослойный персептрон

Преимущества персептрона:

- Персептрон с по крайней мере одним скрытым слоем является универсальным аппроксиматором [27]. То есть персептрон способен сколь угодно приблизить любую непрерывную функцию на компакте в $\mathbb{R}^n$. Поэтому в нашем случае нет необходимости в боль-

[1] $r(x) = \max(0, x)$

шом количестве скрытых слоев. В частности, персептрон обобщает многие широко известные методы для прогнозирования временных рядов. Например, персептрон с линейной функцией активации для временного ряда с несколькими лаговыми переменными является авторегрессионной моделью [28].

Недостатки:

- Не учитывает последовательность поступления элементов и временные зависимости между ними.
- Требует большого количества данных для обучения.
- Подвержен переобучению.

Для борьбы с последним недостатком используются различные техники усреднения. В задаче регрессии хорошо себя зарекомендовало использование функции активации *maxout* [29], то есть взятие максимума из выходов слоя.

Сверточные нейронные сети. *Сверточные нейронные сети (CNN)* [30] — это нейронные сети, которые содержат один или больше сверточных слоев. Сверточные слои оперируют с 3-х мерными тензорами. Обычно после сверточных слоев используют *max pooling* слои, уменьшающие размер входной карты признаков, что сокращает число параметров и улучшает качество CNN.

Чаще всего CNN используют при работе с изображениями, однако они также могут быть использованы в задачах классификации и регрессии [31], так как они позволяют автоматически производить извлечение признаков (feature extraction) [32].

Также достоинством CNN является поддержка параллельных вычислений на GPU.

3.4.7 Композиции алгоритмов

Композиции алгоритмов машинного обучения позволяют существенно повысить качество прогноза.

Взвешенное голосование. Одним из способов композиции алгоритмов машинного обучения является метод *взвешенного голосования*. Пусть имеется набор алгоритмов машинного обучения $b_i : \mathbb{R}^n \to \mathbb{R}^m, i = \overline{1, K}$. Тогда их взвешенным голосованием является следующий алгоритм:

$$b : \mathbb{R}^n \to \mathbb{R}^m,\ b(x) = A(b_1(x), \ldots, b_t(x))^T, \quad A \in \mathbb{R}^{m \times (m \cdot K)}$$

Матрицу A подберем так, чтобы минимизировать среднеквадратичную ошибку алгоритма b. Для этого разобьем тренировочную часть выборки на два множества. На первом множестве обучим алгоритмы $b_i, i = \overline{1, K}$, а на втором множестве вычислим их ответы. Для множества ответов составим алгоритм взвешенного голосования, параметры которого подберем с помощью линейной регрессии (или гребневой линейной регрессии). Схематически алгоритм представлен на Рис. 12.

Достоинства подхода:

- Алгоритмы можно обучать по отдельности.
- Сохранив ответы алгоритмов, больше не требуется их перерасчет.

Недостатком является то, что при обучении алгоритмов не используется часть выборки.

Стекинг. Метод был представлен в [33]. Также как в предыдущем случае разобьем тренировочное множество на две части. На первой части обучим один из алгоритмов. Далее его предсказания на второй части используем как дополнительные признаки для второго алгоритма, обучающегося на второй части тренировочного множества. Итоговым предсказанием является предсказание второго алгоритма на тестовой части. Схематически алгоритм представлен на Рис. 13. Недостатком этого метода также является то, что алгоритмы обучаются только на части тренировочной выборки.

3.5 Полученные результаты

Перед применением моделей машинного обучения необходимо очистить данные от выбросов и произвести нормировку.

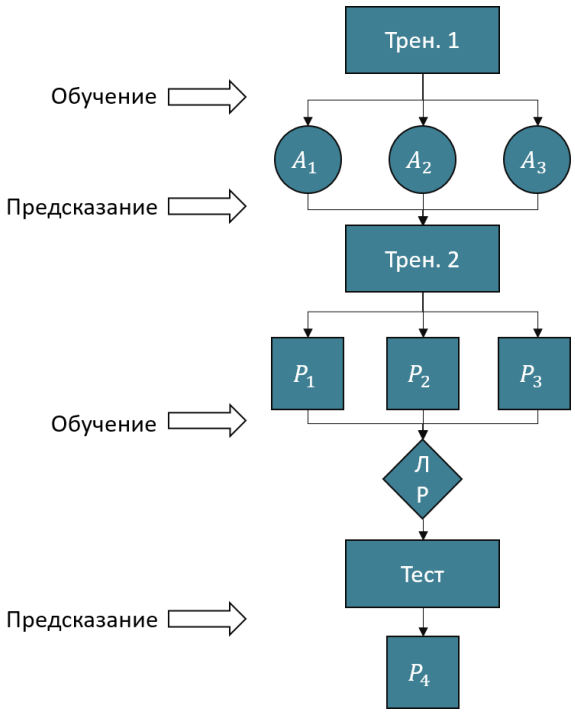


Рис. 12: Взвешенное голосование

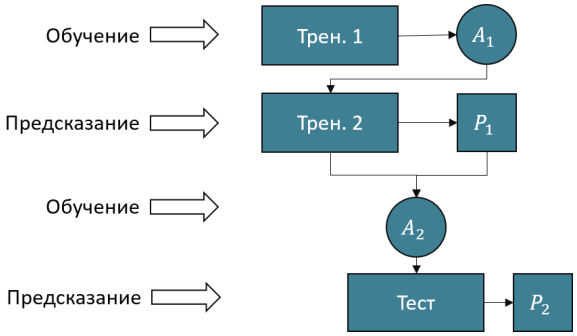


Рис. 13: Стекинг

3.5.1 Предварительная обработка данных

Выбросы в измеренных значениях могут появиться по различным причинам. Например, в связи со сбоем в ПУ или в связи с аварийной ситуацией. Также в системе имеются низкочувствительные ПУ, посылающие только целочисленные значения. Эти данные препятствуют качественному обучению модели и искажают оценку ее эффективности.

В системе присутствуют данные о радиусе труб для ТПУ. Воспользуемся ими для удаления выбросов. По указанным в [34] нормативах о скорости течения воды в трубах, рассчитаем верхнюю оценку потребления. Например, скорость движения воды в трубопроводах внутренних водопроводных сетей, в том числе при пожаротушении, не должна превышать 3 м/с. Отсюда, зная радиус трубы, на которой установлен счетчик получим верхнюю оценку почасового потребления H:

$$H = \pi r^2 \cdot 3 \cdot 3600 \frac{\text{м}^3}{\text{ч}}$$

Далее удалим из данных строки, для которых измеренное почасовое потребление в матрице Y превышает допустимое.

Также выбросы могут встречаться и в метаданных канала. И для

некоторых методов прогнозирования это может значительно ухудшить качество прогноза. Например, возможна следующая ситуация. Была произведена замена датчика ручного ввода показаний, и на новом датчике показания отличаются от уже имеющихся, при этом факт замены не был отражен в системе учета. Также возможно введение ошибочных показаний для каналов ручного ввода. В этом случае, например, для метода СЕПМ ошибка прогноза будет пропорциональна ошибке для показания канала ручного ввода. Поскольку в выборке уже присутствуют подобные ошибки, то удалим из нее те строки, для которых ошибка прогноза СЕПМ превышает три стандартных отклонения. Применим к этим данным нормировку со средним равным 0 и дисперсией равной 1.

3.5.2 Подбор параметров и оценка точности моделей

Для обучения моделей множество каналов было разбито на тренировочную и тестовую часть. Разбиение производилось по каналам, так как при применении модели в реальной ситуации доступны лишь метаданные о каналах. В качестве оценки точности модели для канала было выбрано среднеквадратичная ошибка (RMSE), а также средняя абсолютная ошибка в процентах (MAPE):

- *Среднеквадратичная ошибка* (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - t_i)^2} \tag{6}$$

- *Средняя абсолютная ошибка в процентах* (MAPE):

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{p_i - t_i}{t_i}\right| \tag{7}$$

где p_i — предсказанное значение, а t_i — реальное значение.

С помощью выбранных метрик оценим точность перечисленных выше регрессоров. Более приоритетной будем считать среднюю абсолютную ошибку.

СЕПМ. Хотя метод СЕПМ не является методом машинного обучения, поскольку не происходит обучения как такового, вычислим для него ошибку на кросс-валидации и на тестовой выборке для сравнения с остальными моделями. Результаты представлены в таблице ниже.

Ошибка	**Кросс-валидация**	**Тест**
RMSE	0.43	0.43
MAPE	0.34	0.34

Также стоит отметить, что на ошибку в большей части влияет сам метод усреднения, а не использование данных каналов ручного ввода. Поскольку даже при использовании значений каналов ручного ввода за прогнозируемый интервал времени (то есть, заглядывая в будущее на месяц вперед) качество модели улучшается незначительно.

Random Forest Regressor. В качестве меры качества разделения было использовано среднеквадратичное отклонение. По результатам кросс-валидации была подобрана модель со следующими параметрами:

Параметр	**Значение**
Кол-во деревьев	300
Минимальное кол-во элементов в листе	10

Модель имеет следующее качество прогноза. Как видно, на тестовой выборке ошибка MAPE на 6% меньше, чем у метода СЕПМ, при этом на кросс-валидации оценка хуже. Таким образом, качество модели Random Forest Regressor сравнимо с качеством модели СЕПМ, при этом требует больших вычислительных ресурсов.

Ошибка	**Кросс-валидация**	**Тест**
RMSE	0.49	0.35
MAPE	0.39	0.28

Тем не менее важным достоинством модели является ее способность оценки важности признаков. Ниже перечислены 10 наиболее важных признаков. Как видно, наиболее важные признаки входят и в состав модели СЕПМ.

№	Признак	Важность
1	дата постройки	0.45
2	среднее ежедневное потребление 1 месяц назад	0.36
3	признак жилья	0.03
4	среднее ежедневное потребление 3 месяца назад	0.03
5	признак выходного дня	0.03
6	среднее ежедневное потребление 2 месяца назад	0.02
7	площадь здания	0.01
8	день года	0.01
9	количество этажей	0.01
10	количество проживающих	0.01

Оставим в только эти признаки и заново обучим модель. Ниже представлен результат. Как видно, качество модели почти не изменилось. Поэтому далее и для других моделей будем использовать только эти наиболее важные признаки.

Ошибка	Кросс-валидация	Тест
RMSE	0.49	0.35
MAPE	0.40	0.26

Многослойный персептрон. Мы использовали модель со следующей структурой. Веса инициализировались из нормального распределения с масштабированием по количеству входов нейрона [35]. Количество нейронов выходного слоя определяется размерностью выхода. Поэтому в нашем случае число нейронов выходного слоя равно 24. Размер пакета (batch) для оптимизатора был выбран равным 1000. Всего было произведено 2000 итераций.

Первый слой: 400 нейронов с функцией активации softsign.
Второй слой: 200 нейронов с функцией активации tanh.
Третий слой: 24 нейронов с функцией активации linear.

Модель имеет следующую точность. Как видно, ее точность хуже, чем у предыдущих моделей.

Ошибка	Кросс-валидация	Тест
RMSE	0.37	0.37
MAPE	0.32	0.32

Метод k ближайших соседей. Для метода kNN использовались следующие параметры.

Параметр	Значение
Метрика	L_1
Количество соседей	30
Веса	обратно проп-ны расстоянию

Подбор количества соседей осуществлялся с помощью перебора значений [5, 10, 30, 50, 100] и кросс-валидации. Модель имеет следующее качество прогноза. В целом оно хуже, чем у RF.

Ошибка	Кросс-валидация	Тест
RMSE	0.5	0.39
MAPE	0.39	0.27

Стекинг: Random Forest + СЕПМ. Применим метод стекинга для моделей RF и СЕПМ. Из оценок прогноза ниже видно, что модель стала вести себя более устойчиво на кросс-валидации.

Ошибка	Кросс-валидация	Тест
RMSE	0.39	0.35
MAPE	0.33	0.30

Взвешенное голосование: RF + kNN + MLP + ET + СЕПМ. Воспользуемся предыдущей моделью (RF + СЕПМ), и по аналогии составим модели (MLP + СЕПМ) и (ET + СЕПМ). Объединим эти модели и метод kNN в композицию с помощью алгоритма взвешенного голосования с гребневой линейной регрессией. Финальная модель показывает следующие результаты.

Ошибка	Кросс-валидация	Тест
RMSE	0.32	0.32
MAPE	0.26	0.25

Данная модель превосходит все остальные по точности как на кросс-валидации, так и на тестовой выборке. Тем не менее для ее использования требуются значительные вычислительные ресурсы.

Сверточная нейронная сеть. Исходя из полученных результатов, композиции алгоритмов машинного обучения позволяют существенно повысить точность. Воспользуемся этой же логикой по отношению к нейронным сетям. На Рис. 14 представлена архитектура используемой нейронной сети. Нейронная сеть состоит из трех ветвей, выходы которых конкатенируются и используются последними полносвязными слоями. Ветви создавались так, чтобы быть максимально различными. Также в каждой ветви используется разное количество подряд идущих сверточных слоев для возможности подстройки глубины нейронной сети. В первой ветви используются наборы из одного или двух последовательных сверточных слоев с разными фильтрами. На вход каждому входному слою в этой ветви подается только метаинформация о каналах. Цель создания этой ветви — выделение и создание важных признаков. Вторая ветвь имеет три подряд идущих сверточных слоя. Ей на вход подается метаинформация по каналам и соответствующий прогноз метода СЕПМ. Эта ветвь позволяет выделять признаки на основе комбинации прогноза и метаинформации. В последней ветви используются различные комбинации полносвязных и локально связных слоев. Листьям на вход в ней подается метаинформация и комбинация метаинформации с прогнозом. Данная ветвь позволяет получить отличные от остальных ветвей прогнозы, поскольку в ней не используются сверточные слои, что в результате усреднения на последнем этапе улучшает качество прогноза. Подробные сведения представлены в листинге кода Python с библиотекой Keras, где `trainX` — конкатенация метаинформации с прогнозом СЕПМ. Код рассматриваемой сети представлен далее в Приложении А.

Модель имеет следующую погрешность.

Ошибка	**Кросс-валидация**	**Тест**
RMSE	0.35	0.35
MAPE	0.27	0.27

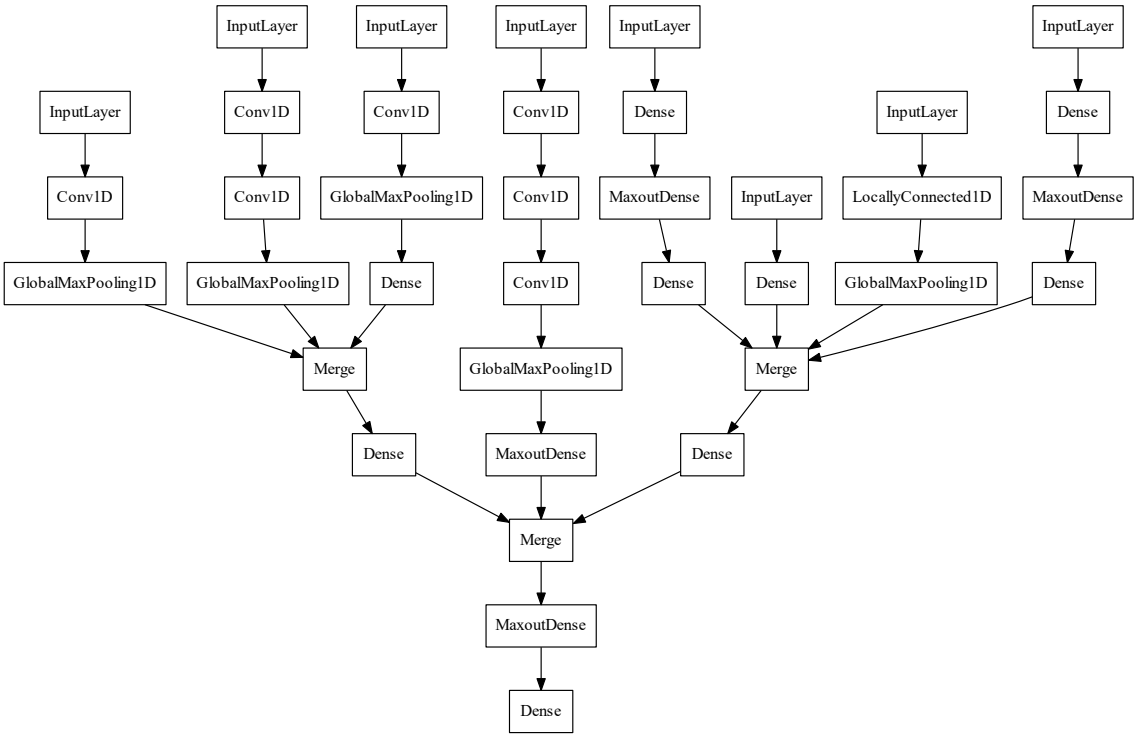


Рис. 14

Таким образом, данная сверточная нейронная сеть имеет точность, сравнимую с точностью предыдущей модели, но требует значительно меньше оперативной памяти и при использовании GPU позволяет значительно снизить время вычисления.

3.6 Вывод

В Таблице 2 представлена сводная информация об эффективности алгоритмов. Таким образом, метаданные о доме, в котором находится канал, позволяют улучшить качество прогноза потребления. Наиболее высокую точность обеспечивают композиции методов машинного обучения, однако они требуют значительных вычислительных ресурсов. Наиболее перспективным является применение сверточных нейронных сетей, поскольку они обеспечивают сравнимую точность, но требуют значительно меньше вычислительных ресурсов при использовании GPU.

	Кросс-валидация		Тест	
	RMSE(m^3)	MAPE(%)	RMSE(m^3)	MAPE(%)
ADU	0.43	0.34	0.43	0.34
RF	0.49	0.39	0.35	0.28
MLP	0.37	0.31	0.37	0.32
kNN	0.5	0.39	0.39	0.27
RF+ADU(s)	0.39	0.33	0.35	0.3
RF+kNN+ MLP+ET(sv)	0.32	0.26	0.32	0.25
CNN	0.35	0.27	0.35	0.27

Таблица 2: Эффективность алгоритмов

3.6.1 Направления дальнейших исследований

Для улучшения качества прогноза необходимо дальнейшее исследование применения нейронных сетей, поскольку рассмотренная CNN показала лишь второй результат. В данном случае на прогноз влияет архитектура нейронной сети. Тем не менее в настоящее время нет общих способов для подбора оптимальной архитектуры и ее выбор является *state of the art*. Кроме сверточных нейронных сетей возможно использование и других их видов, например, рекуррентных или генеративных. Однако в этом случае возрастает вычислительная сложность.

4 Заключение

В настоящее время благодаря внедрению автоматизированных приборов учета потребления воды становится возможен непрерывный мониторинг качества подачи воды. Предоставляемых данных достаточно для постановки оптимизационных задач и их решения, в частности для решения задачи минимизации ежедневного пика потребления. Также при использовании методов машинного дополнительная метаинформация о доме позволяет значительно улучшить качество прогноза потребления для неавтоматизированных приборов учета.

Список литературы

[1] ABB, “Leakage monitoring Reducing leakage through effective flow measurement”, p. 11, 2011.

[2] Barbato, A. et al., “A power scheduling game for reducing the peak demand of residential users”, in *Online Conference on Green Communications*, pp. 137–142, IEEE, 2013.

[3] Nguyen, H. K., Song, J. B. and Han, Z., “Demand side management to reduce peak-to-average ratio using game theory in smart grid”, in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pp. 91–96, IEEE, 2012.

[4] Tricarico, C. et al., “Peak residential water demand”, in *Proceedings of the Institution of Civil Engineers-Water Management*, vol. 160, pp. 115–121, Thomas Telford Ltd, 2007.

[5] Petrosyan, L. A. and Zenkevich, N. A., *Game Theory*. World Scientific, 2016.

[6] Surendran, S., Tanyimboh, T. T. and Tabesh, M., “Peaking demand factor-based reliability analysis of water distribution systems”, *Advances in Engineering Software*, vol. 36, no. 11, pp. 789–796, 2005.

[7] Ward J. H., Jr., “Hierarchical grouping to optimize an objective function”, *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963.

[8] Бутырский, Е. Ю., “Теоретические основы моделирования”, *Издательский дом «Palmarium Academic Publishing», Germany*, 2012.

[9] Linkola, L., Andrews, C. J. and Schuetze, T., “An agent based model of household water use”, *Water*, vol. 5, no. 3, pp. 1082–1100, 2013.

[10] Pruckner, M., Eckhoff, D. and German, R., “Modeling country-scale electricity demand profiles”, in *Simulation Conference (WSC), 2014 Winter*, pp. 1084–1095, IEEE, 2014.

[11] Zoha, A. et al., “Low-power appliance monitoring using factorial hidden markov models”, in *Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on*, pp. 527–532, IEEE, 2013.

[12] Fiol Arguimbau, A., “Algorithms for energy disaggregation”, Master’s thesis, Universitat Politècnica de Catalunya, 2016.

[13] Kelly, J. and Knottenbelt, W., “Neural nilm: Deep neural networks applied to energy disaggregation”, in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, pp. 55–64, ACM, 2015.

[14] “Технико-экономические паспорта многоквартирных домов”, *http://data.gov.spb.ru/opendata/7840013199-passports_houses*, 2016.

[15] Busseti, E., Osband, I. and Wong, S., “Deep learning for time series modeling”, *Technical report, Stanford University*, 2012.

[16] Kane, M. J. et al., “Comparison of arima and random forest time series models for prediction of avian influenza h5n1 outbreaks”, *BMC bioinformatics*, vol. 15, no. 1, p. 276, 2014.

[17] Breiman, L., “Random forests”, *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[18] Shah, A. D. et al., “Comparison of random forest and parametric imputation models for imputing missing data using mice: a caliber study”, *American journal of epidemiology*, vol. 179, no. 6, pp. 764–774, 2014.

[19] Geurts, P., Ernst, D. and Wehenkel, L., “Extremely randomized trees”, *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.

[20] Стрижов, В. В. and Крымова, Е. А., “Методы выбора регрессионных моделей”, *М.: ВЦ РАН*, vol. 60, p. 2, 2010.

[21] Kourentzes, N. and Crone, S. F., "Forecasting high-frequency time series with neural networks-an analysis of modelling challenges from increasing data frequency",

[22] Duchi, J., Hazan, E. and Singer, Y., "Adaptive subgradient methods for online learning and stochastic optimization", *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[23] Zeiler, M. D., "Adadelta: an adaptive learning rate method", *arXiv preprint arXiv:1212.5701*, 2012.

[24] Hinton, G., Srivastava, N. and Swersky, K., "Lecture 6a overview of mini–batch gradient descent", *Coursera Lecture slides https://class. coursera. org/neuralnets-2012-001/lecture,[Online*, 2012.

[25] Kingma, D. and Ba, J., "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980*, 2014.

[26] Rosenblatt, F., "Principles of neurodynamics. perceptrons and the theory of brain mechanisms", tech. rep., DTIC Document, 1961.

[27] Hornik, K., Stinchcombe, M. and White, H., "Multilayer feedforward networks are universal approximators", *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[28] Amaldi, E., Mattavelli, M. and Vesin, J.-M., "A perceptron-based approach to piecewise linear modeling with an application to time series", 1997.

[29] Goodfellow, I. J. et al., "Maxout networks", *arXiv preprint arXiv:1302.4389*, 2013.

[30] LeCun, Y. et al., "Backpropagation applied to handwritten zip code recognition", *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[31] Cui, Z., Chen, W. and Y., Chen, "Multi-scale convolutional neural networks for time series classification", *CoRR*, vol. abs/1603.06995, 2016.

[32] Yosinski, J. et al., “How transferable are features in deep neural networks?”, in *Advances in neural information processing systems*, pp. 3320–3328, 2014.

[33] Jahrer, M., Töscher, A. and Legenstein, R., “Combining predictions for accurate recommender systems”, in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 693–702, ACM, 2010.

[34] “СНиП 2.04.01-85*: Внутренний водопровод и канализация зданий”, 1997.

[35] He, K. et al., “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”, in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Приложение

A Сверточная нейронная сеть

```
time_period = 24
X_meta = trainX[:, 0:(trainX.shape[1] - time_period)]
X_adu = trainX[:, (trainX.shape[1] - time_period):trainX.shape[1]]
X_meta_conv = np.reshape(X_meta, (X_meta.shape[0], X_meta.shape[1], 1))
X_conv = np.reshape(trainX, (trainX.shape[0], trainX.shape[1] // 2, 2))
br1 = Sequential()
br1.add(Dense(60, input_dim=X_meta.shape[1], init='he_normal',
    ↪ activation='softsign'))
br1.add(MaxoutDense(30, init='he_normal'))
br1.add(Dense(24, activation='softsign', init='he_normal'))
br2 = Sequential()
br2.add(Dense(24, input_dim=X_adu.shape[1], init='he_normal',
    ↪ activation='relu'))
br3 = Sequential()
br3.add(LocallyConnected1D(24, 3, activation='softsign', init='
    ↪ he_normal', input_shape=(X_meta_conv.shape[1], X_meta_conv.shape
    ↪ [2])))
br3.add(GlobalMaxPooling1D())
br4 = Sequential()
br4.add(Dense(60, input_dim=trainX.shape[1], init='he_normal',
    ↪ activation='softsign'))
br4.add(MaxoutDense(30, init='he_normal'))
br4.add(Dense(24, activation='softsign', init='he_normal'))
model1 = Sequential()
model1.add(Merge([br1, br2, br3, br4], mode='concat'))
model1.add(Dense(24, init='he_normal', activation='relu'))
br5 = Sequential()
br5.add(Convolution1D(24, 5, activation='relu', init='he_normal',
    ↪ input_shape=(X_meta_conv.shape[1], X_meta_conv.shape[2])))
br5.add(GlobalMaxPooling1D())
```

```
br5.add(Dense(24, init='he_normal', activation='relu'))
br6 = Sequential()
br6.add(Convolution1D(30, 7, activation='softsign', init='he_normal',
    ↪ input_shape=(X_meta_conv.shape[1], X_meta_conv.shape[2])))
br6.add(GlobalMaxPooling1D())
br7 = Sequential()
br7.add(Convolution1D(40, 3, init='he_normal', activation='relu',
    ↪ input_shape=(X_meta_conv.shape[1], X_meta_conv.shape[2])))
br7.add(Convolution1D(24, 3, init='he_normal', activation='softsign'))
br7.add(GlobalMaxPooling1D())
br8 = Sequential()
br8.add(Convolution1D(40, 3, init='he_normal', activation='relu',
    ↪ input_shape=(X_conv.shape[1], X_conv.shape[2])))
br8.add(Convolution1D(30, 3, init='he_normal', activation='softsign'))
br8.add(Convolution1D(24, 3, init='he_normal', activation='softsign'))
br8.add(GlobalMaxPooling1D())
br8.add(MaxoutDense(12, init='he_normal'))
model2 = Sequential()
model2.add(Merge([br5, br6, br7], mode='concat'))
model2.add(Dense(24, init='he_normal', activation='relu'))
model3 = Sequential()
model3.add(Merge([model1, model2, br8], mode='concat'))
model3.add(MaxoutDense(40, init='he_normal'))
model3.add(Dense(24, init='he_normal', activation='linear'))
model3.compile(loss='mae', optimizer='adam')
model3.fit([X_meta, X_adu, X_meta_conv, trainX, X_meta_conv,
    ↪ X_meta_conv, X_meta_conv, X_conv], trainY, nb_epoch=1000,
    ↪ batch_size=1000)
```