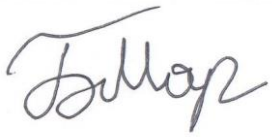
**Федеральное государственное
бюджетное образовательное учреждение высшего образования
«Воронежский государственный технический университет»**

На правах рукописи



МАРТЫНЕНКОВ Борис Витальевич

МОДЕЛИ И АЛГОРИТМЫ ИНТЕЛЛЕКТУАЛЬНОЙ ПОДДЕРЖКИ УПРАВЛЕНИЯ РАБОЧЕЙ НАГРУЗКОЙ СИСТЕМ ОБРАБОТКИ ИНФОРМАЦИИ НА ОСНОВЕ РЕТРОСПЕКТИВНЫХ ДАННЫХ

Специальность 2.3.1. Системный анализ, управление и обработка информации, статистика

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
д.т.н., доцент
Цветков Александр Васильевич

Воронеж – 2026

Оглавление

ВВЕДЕНИЕ

Актуальность темы

Развитие и совершенствование информационных сервисов, как общего назначения, так и со специализированной функциональностью стало основой для разработки и широкого использования виртуализированных центров обработки данных (ВЦОД), инфраструктура которых основана на парадигме виртуализации вычислительных ресурсов и предоставления их по требованию в составе виртуальных машин и/или контейнеров. Такой подход повышает адаптивность ВЦОД к запросам на использование их вычислительных ресурсов за счет процесса миграции виртуальных машин/контейнеров между физическими машинами.

Важной задачей администрирования ВЦОД является динамическая реконфигурация его инфраструктуры в зависимости от текущей рабочей нагрузки, определяемой запросами программного обеспечения (ПО) сервисов, функционирующих в составе виртуальных машин/контейнеров. Очевидно, что рабочая нагрузка ВЦОД должна быть адекватна запросной нагрузке, показатели которой могут носить, как периодический (сезонный: время суток, время года), так и случайный (реакция на события и т.д.) характер. В общем случае можно говорить о некоторых типовых и не типовых шаблонах рабочей нагрузки.

Процесс реконфигурации при этом основан на анализе результатов мониторинга использования (утилизации) вычислительных ресурсов ВЦОД. Сохраненные результаты мониторинга в предыдущие периоды функционирования именуются ретроспективными данными рабочей нагрузки. Их анализ позволяет реализовывать, как функцию текущего (реактивного) управления рабочей нагрузкой, так и функцию проактивного управления ею. Последняя базируется на решении класса задач прогнозирования рабочей нагрузки.

Одной из проблем эффективного решения задачи прогнозирования рабочей нагрузки является «зашумление» значений временного ряда ретроспективных данных. Источниками зашумления могут выступать, как сами приложения или сервисы, так и другие приложения и сервисы, функционирующие в составе ВЦОД.

Известные подходы к решению задачи прогнозирования рабочей ВЦОД не в полной мере рассматривают возможные условия ее зашумления, а также модели, методы и алгоритмы снижения влияния шумовых факторов на значения показателей временного ряда ретроспективных данных.

Таким образом, актуальность исследования обосновывается необходимостью разработки и исследования новых эффективных моделей и алгоритмов прогнозирования рабочей нагрузки ВЦОД на основе ее ретроспективных данных с учетом их зашумления.

Тематика диссертационной работы соответствует научному направлению ФГБОУ ВО «Воронежский государственный технический университет» «Вычислительные комплексы и проблемно-ориентированные системы управления».

Степень разработанности темы. Фундаментальными исследованиями в теории прогнозирования временных рядов методами статистического и машинного обучения посвящены работы Дж. Бокса (George Box), Г. Дженкинса (Gwilym Jenkins), Я. Лекуна (Yann LeCun), З. Хохрайтера (Sepp Hochreiter), Ю. Шмидхубера (Jürgen Schmidhuber). Теоретическими и прикладными исследованиям обработки зашумленных сигналов в различных предметных областях посвящены работы Д. Гильберта (David Hilbert), Н. Хуанга (N. Huang), Г. де Мораеса (G. de Moraes), М.Д. Сенюка, С.А. Ерошенко, Х. Джанга (H. Zhang).

Задача прогнозирования рабочей нагрузки на основе анализа значений временного ряда ее ретроспективных данных в указанных исследованиях рассматривается в основном для контролируемых условий функционирования ВЦОД и носит узкий характер за счет получения специальных решений, или рассматривается в общем случае, что не позволяет получить эффективные модели и алгоритмы формирования решения.

Вопросы разработки моделей и алгоритмов снижения влияния факторов зашумления на значения временного ряда рабочей нагрузки ВЦОД для решения задачи ее прогнозирования в рассматриваемых исследованиях, либо выносятся в область ограничений, либо рассматриваются недостаточно полно. В связи с этим

предлагается разработка моделей и алгоритмов предварительной обработки зашумленного временного ряда ретроспективных данных рабочей нагрузки, а также моделей и алгоритмов машинного обучения для решения задачи прогнозирования рабочей нагрузки на основе полученного «очищенного» временного ряда.

Объектом исследования является временной ряд ретроспективных данных рабочей нагрузки в условиях влияния шумовых факторов и процедура получения его прогнозных значений.

Предметом исследования являются модели представления временных рядов, алгоритмы их разложения, процедуры машинного обучения для получения их прогнозных значений.

Цель и задачи исследования. Целью диссертационного исследования является разработка модели временного ряда рабочей нагрузки и алгоритмов, снижающих факторы ее зашумления, а также моделей, алгоритмов и специального программного обеспечения управления системы обработки информации для прогнозирования рабочей нагрузки.

Для достижения поставленной цели необходимо решить следующие частные научные задачи:

1) Разработать модель модовой декомпозиции временного ряда рабочей нагрузки системы обработки информации, на основе анализа формирования временных рядов ее ретроспективной информации, влияния на нее факторов зашумления и существующих методов модовой декомпозиции сигналов.

2) Создать комплексный алгоритм предварительной обработки временного ряда рабочей нагрузки на основе анализа способов интеграции эмпирического и вариационного методов декомпозиции сигнала.

3) Разработать гибридный алгоритм системы управления прогнозированием временного ряда рабочей нагрузки, на основе анализа методов глубокого машинного обучения, поддерживающих анализ временных рядов.

4) Модифицировать существующую архитектуру системы управления прогнозированием рабочей нагрузки на основе анализа функциональных

возможностей существующих фреймворков модовой декомпозиции сигналов и гибридных сетей глубокого машинного обучения.

Методология и методы исследования. Для решения поставленных задач используются методы математической статистики, модовой декомпозиции сигналов, методы искусственного интеллекта, теория прогнозирования, теория сложных систем. Общей методологической основой является системный подход.

Научная новизна. В диссертации получены следующие результаты, характеризующиеся научной новизной:

– модель модовой декомпозиции временного ряда рабочей нагрузки, отличающаяся совместным использованием эмпирического и вариационного подходов получения множества колебательных модовых функций и обеспечивающая снижение влияния факторов зашумления на значения временного ряда;

– комплексный алгоритм предварительной обработки временного ряда рабочей нагрузки, отличающийся наличием этапа вторичной вариационной модовой декомпозиции базовой колебательной модовой функции, полученной методом эмпирической модовой декомпозиции, и позволяющий формировать множества обучающей и тестовой выборок для системы прогнозирования значений временного ряда на основе методов глубокого обучения;

– гибридный алгоритм прогнозирования временного ряда рабочей нагрузки для системы глубокого обучения, отличающийся ансамблевым способом выделения значимых признаков шаблонов рабочей нагрузки и каскадным режимом ее прогнозирования, обеспечивающий получение разномасштабных прогнозных значений временного ряда рабочей нагрузки;

– архитектура системы прогнозирования рабочей нагрузки, отличающаяся интеграцией модуля предварительной обработки временного ряда рабочей нагрузки и обеспечивающая прогнозирование рабочей нагрузки с пригодной точностью прогноза.

Теоретическая значимость заключается в том, что предлагаемые новые подходы к предварительной обработке временного ряда рабочей нагрузки

и разработанные алгоритмы прогнозирования его значений, основанные на методах модовой декомпозиции и глубокого обучения, могут быть применены при формировании экспериментальных моделей временных рядов в других предметных областях и для разработки программных комплексов их прогнозирования.

Практическая значимость работы заключается в повышении точности прогноза значений временного ряда, необходимого в различных областях человеческой деятельности. Разработано специальное программное обеспечение, позволяющее осуществлять прогнозирование временного ряда рабочей нагрузки (на примере использования процессорных ядер). Предложены рекомендации по исследованию ретроспективных временных рядов вычислительных ресурсов системы обработки информации для решения задачи прогнозирования шаблонных типов рабочей нагрузки.

Достоверность результатов подтверждается использованием при разработке моделей известных математических методов и результатами вычислительных экспериментов.

Положения, выносимые на защиту:

1) Модель модовой декомпозиции временного ряда обеспечивает снижение влияния факторов зашумления на значения временного ряда.

2) Комплексный алгоритм предварительной обработки временного ряда рабочей нагрузки обеспечивает формирование множеств обучающей и тестовой выборок для системы прогнозирования элементов временного ряда на основе методов глубокого обучения.

3) Гибридный алгоритм прогнозирования временного ряда рабочей нагрузки для системы глубокого обучения обеспечивает получение разномасштабных прогнозных значений временного ряда рабочей нагрузки.

4) Архитектура системы прогнозирования рабочей нагрузки, обеспечивает прогнозирование рабочей нагрузки с пригодной точностью прогноза.

Апробация работы. Основные результаты диссертационной работы докладывались и обсуждались на следующих конференциях и семинарах: Международная научно-практическая конференция «Общество – Наука Инновации

(Уфа, 2021), Материалы круглого стола «Хроники цифровых трансформаций» (Волгоград, 2022), III ежегодной национальная научно-практическая конференция «Кибербезопасность: технические и правовые аспекты защиты информации» (Москва, 2024), Международная научно-практическая конференция «Стратегии успеха: инновационные методы, технологии и практики в науке для достижения глобального прогресса» (Уфа, 2025), XXXI International Open Science Conference «Modern informatization problems in simulation and social technologies (Yelm, WA, USA, 2026), а также на научных семинарах кафедры кибернетики в системах организационного управления ВГТУ (2023-2026 гг.).

Реализация и внедрение результатов работы. Результаты диссертации внедрены в практическую деятельность ФГУП «Научно-технический центр «Орион» (г. Москва) и ООО «Научно-технический центр «Разработка сложных систем» (г. Орёл), а также в образовательный процесс Московского технического университета связи и информатики (МТУСИ).

Соответствие паспорту специальности. Содержание диссертации соответствует п. 4 «Разработка методов и алгоритмов решения задач системного анализа, оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта», п. 5 «Разработка специального математического и алгоритмического обеспечения систем анализа, оптимизации, управления, принятия решений, обработки информации и искусственного интеллекта», п. 6 «Методы идентификации систем управления на основе ретроспективной, текущей и экспертной информации» паспорта специальности 2.3.1. Системный анализ, управление и обработка информации, статистика.

Публикации. По материалам диссертации опубликовано 24 печатные работы, в том числе 4 статьи опубликовано в изданиях, рекомендованных ВАК при Минобрнауки России, 1 статья в издании, индексируемом в WoS, разработано 12 программ для ЭВМ, зарегистрированных в Федеральной службе по интеллектуальной собственности.

В работах, опубликованных в соавторстве и лично соискателем предложены: [1, 3, 6-8, 16, 23] – подход к предварительной обработке ретроспективных данных

об уровне загрузки вычислительных ресурсов; [2, 4, 5, 9, 12, 13, 21] – гибридная модель глубокого обучения для прогнозирования рабочей нагрузки в виртуализированных центрах обработки данных; [10, 11, 14, 15, 22] – алгоритмы системы прогнозирования рабочей нагрузки, функционирующей в условиях зашумления ее ретроспективных данных; [18, 19, 20, 24] – архитектура системы прогнозирования рабочей нагрузки; [17] – процедура интеграции модуля предварительной обработки временного ряда рабочей нагрузки в систему прогнозирования рабочей нагрузки.

Структура и объем работы. Диссертация состоит из введения, четырех глав, заключения, списка литературы из 130 наименований. Работа изложена на 154 страницах машинописного текста (основной текст занимает 125 страниц, содержит 71 рисунок, 4 таблицы).

Глава 1. Разработка модели модовой декомпозиции временного ряда рабочей нагрузки виртуализированного центра обработки данных

1.1. Анализ предметной области управления рабочей нагрузкой в виртуализированных центрах обработки данных

Развитие и совершенствование информационных сервисов, как общего назначения, так и со специализированной функциональностью в таких предметных областях, как электронная коммерция, электронное правительство, инженерное проектирование и анализ, финансы, здравоохранение, веб-хостинг и социальные сети, системы искусственного интеллекта (ИИ) прикладного и специализированного назначения стало основой для разработки и широкого использования виртуализированных центров обработки данных (ВЦОД), являющихся вычислительной основой указанных сервисов.

Технологии, реализуемые в ВЦОД предоставляют экономически эффективные масштабируемые решения благодаря гибкости и эластичности в предоставлении вычислительных ресурсов [1, 2]. ВЦОД основаны на парадигме виртуализации вычислительных ресурсов физических машин (ФМ) и предоставления этих виртуализированных ресурсов потребителям в составе виртуальных машин (ВМ) и/или контейнеров [3]. Это коренным образом отличает архитектуру ВЦОД от архитектуры ЦОД традиционного типа, основанных на парадигмах «stand-alone server» и доменной инфраструктуры [4], выделяющих потребительскому ПО непосредственные вычислительные ресурсы ФМ, часто в режиме их разделения с ПО других потребителей и системы администрирования ЦОД.

На рисунке 1.1 дается представление о структурах традиционного ЦОД, ВЦОД и их месте в многоуровневой схеме обработки разнородных потребительских запросов [5].

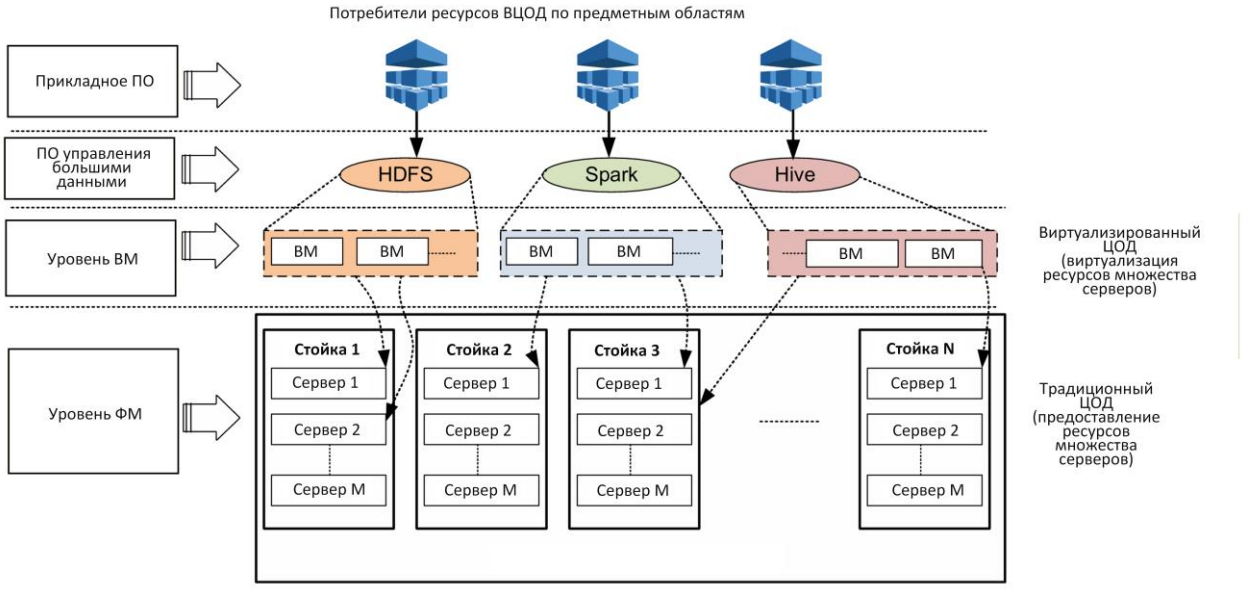


Рис. 1.1 Место и функции традиционного ЦОД и ВЦОД в многоуровневой схеме обработки потребительских запросов

На рисунках 1.2 и 1.3 представлены: сравнительные расходы на развертывание и эксплуатацию традиционных (hardware and software) ЦОД и ВЦОД (рисунок 1.2), а также прогнозируемый рост финансовых вложений в инфраструктуру ВЦОД на ближайшее десятилетие (рисунок 1.3) [6,7].

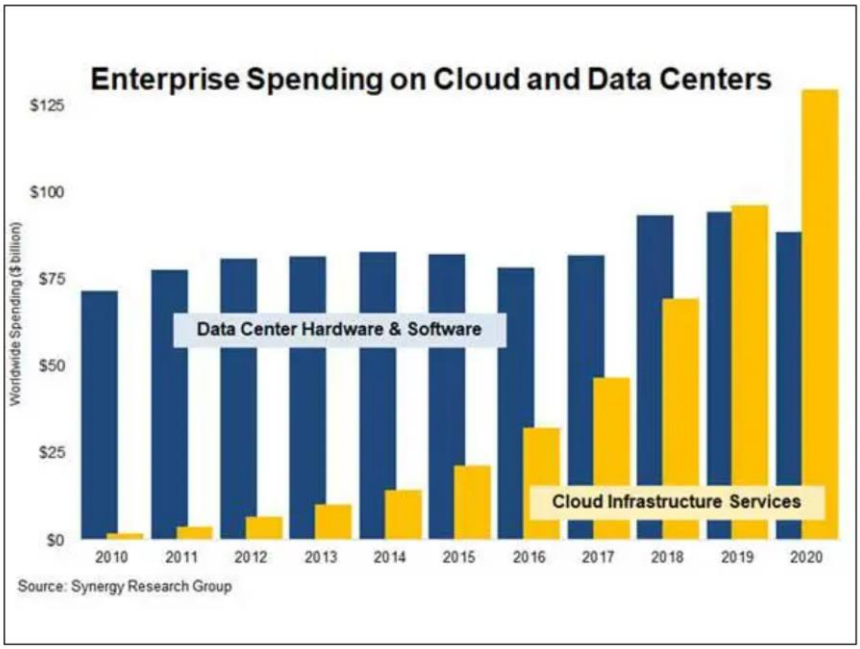


Рис. 1.2 Сравнительный рост расходов на развертывание и эксплуатацию традиционных ЦОД и ВЦОД

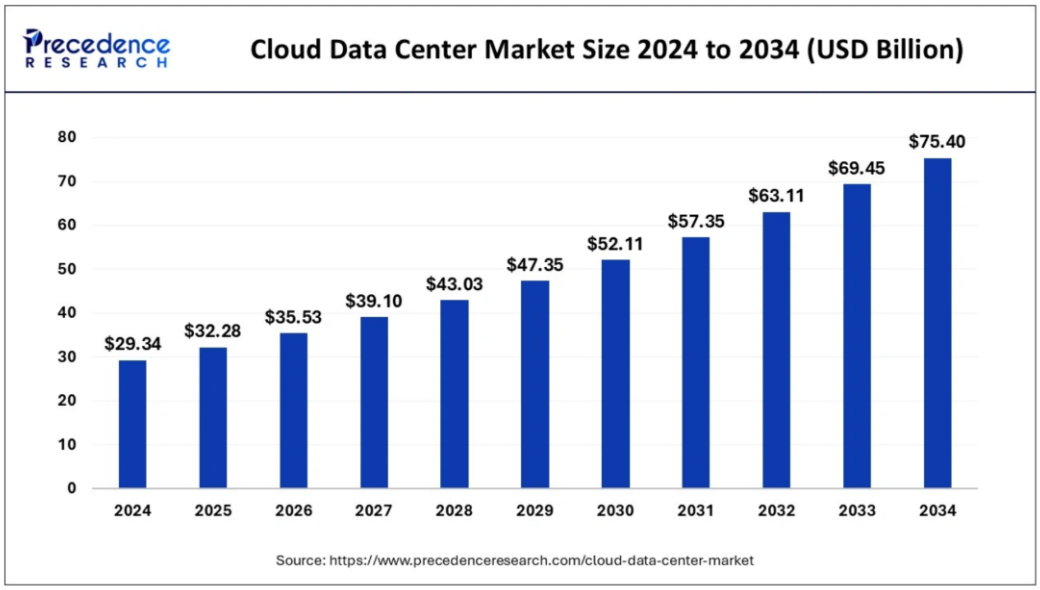


Рис. 1.3 Прогнозируемый рост финансовых вложений в инфраструктуру ВЦОД

Виртуализация вычислительных ресурсов повышает адаптацию ВЦОД к потребительским запросам. В первую очередь такая адаптация основана на функциях:

– приостановки функционирования ВМ/контейнеров с сохранением данных о их текущем состоянии;

– миграции ВМ/контейнеров между ФМ.

Указанные функции обеспечивают динамическое перераспределение вычислительных ресурсов ФМ и их высвобождение в условиях снижения интенсивности потребительских запросов.

При этом потребительские запросы ВЦОД зачастую затрагивают функционирование множества видов ПО и сервисов, каждый из которых характеризуется своими собственными требованиями к производительности и вычислительным ресурсам, а также ограничениями, указанными в виде соглашений об уровне обслуживания (SLA) [4]. Кроме этого, существенное влияние на эффективность функционирования ВЦОД оказывает его рабочая нагрузка.

1.1.1. Исследование особенностей представления рабочей нагрузки ВЦОД

Понятие рабочая нагрузка (workload) ВЦОД заимствовано из предметной области многозадачных и многомашинных вычислительных систем, и получило дальнейшее доопределение в соответствии с инфраструктурными особенностями ВЦОД.

В целом под рабочей нагрузкой понимается все множество приложений и сервисов, выполняющих обработку данных и потребляющих вычислительные ресурсы ВЦОД. Очевидно, что эффективность функционирования ВЦОД определяется соотношением выполняемых им приложений и сервисов (а также числом потребительских запросов к ним, поскольку затрагивается и сетевая подсистема ВЦОД) и выделенными для этих приложений и сервисов вычислительными ресурсами ВЦОД.

В силу этого множество исследований посвящено рассмотрению особенностям рабочей нагрузки вычислительных систем в целом и ВЦОД в частности. Так, в [8] выделяются общие категориальные признаки, характеризующие рабочую нагрузку. В [9] делается обобщение методов и способов управления рабочей нагрузкой известных крупномасштабных ВЦОД. В [10] рассматриваются теоретические подходы к моделированию рабочей нагрузки ВЦОД.

В общем случае выделяют следующие категории, характеризующие рабочую нагрузку ВЦОД:

– модель обработки данных;

– архитектура рабочей нагрузки.

Так модель обработки данных, как характеристика рабочей нагрузки ВЦОД, соответствует парадигмам обработки данных классических вычислительных систем и подразделяется на интерактивную (современная интерпретация – онлайн) и пакетную (современная интерпретация – оффлайн) обработку данных. Очевидно, что, как и в случае классических вычислительных систем, эти модели применительно к ВЦОД характеризуются разными требованиями

к производительности его инфраструктуры, принципами управления, связанными с планированием и распределением вычислительных ресурсов. Интерактивная рабочая нагрузка обычно состоит из кратковременных задач обработки данных, одновременно выполняемых некоторым подмножеством потребителей. Пакетная рабочая нагрузка характерна для долговременных и, обычно, ресурсоемких вычислительных задач.

Архитектура рабочей нагрузки в [8] рассматривается применительно к модели обработки потоков данных, характеризующих сервисы, реализуемые в рамках ВЦОД. Фактически требуется рассматривать количество и типы приложений, порождаемых каждым из сервисов ВЦОД, а также зависимости, возникающие между ними. Как и в случае классических вычислительных систем архитектура рабочей нагрузки ВЦОД может быть: последовательной (жестко определенный порядок и приоритетность выполнения приложений сервиса); параллельной (допускающая одновременное или псевдоодновременное выполнение приложений в режиме временного разделения вычислительных ресурсов); гибридной (сочетающей особенности последовательной и параллельной архитектур).

В рамках развития и совершенствования современных сервисов ВЦОД следует рассматривать гибридную архитектуру рабочей нагрузки. Ее теоретическим представлением является структура направленного ациклического графа потока данных (data workflow graph) [9], узлы которого определяют обрабатываемые приложениями данные, а ребра – взаимосвязи в процессе обработки (рисунок 1.4). Из рисунка видно, что потоки d1-d2, d2-d4, d3-d6-d8, d5-d7-d8 являются последовательными, в то время как d2-d3-d4-d5 – параллельным.

Очевидно, что превалирование в гибридной модели последовательных или параллельных ветвей оказывает существенное влияние на производительность сервиса и ВЦОД в целом. Так, большее число параллельных ветвей может вести к снижению производительности, что обусловлено конкуренцией за вычислительные ресурсы, возникающими при совместном выполнении разнородных приложений в одной ФМ, а также накладными расходами, вызванными применяемыми политиками управления ресурсами.

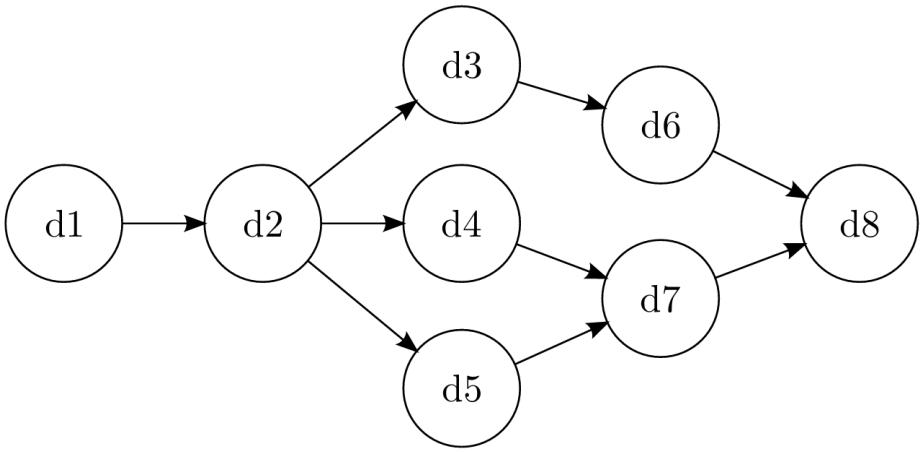


Рис. 1.4 Пример графа потоков данных гибридной архитектуры рабочей нагрузки

Применительно к особенностям использования методов виртуализации – это может вести к непредсказуемому влиянию на производительность ВЦОД из-за несовместимых временных шаблонов использования виртуализированных ресурсов [10].

В рамках рассмотренных выше архитектур рабочей нагрузки ВЦОД рассматриваются их технологические реализации, такие как:

– модель Map-Reduce [11];

– многоуровневая (multi-tier) архитектура [12].

Если модель Map-Reduce изначально ориентировалась на интерактивные сервисы обработки потока запросов потребителей на структурированные обработку (индексирование) и поиск информации (поисковые системы), то многоуровневая модель является более универсальной и предполагает, что каждый уровень сервиса, развернутый на одной и более ВМ реализует его определенную функциональность, например: уровень web-взаимодействия, уровень баз данных, уровень основной логики ПО, уровень балансировки нагрузки и т.д. В рамках такой архитектуры рассматривается ее горизонтальное (изменение числа используемых ВМ) и вертикальное (изменение количества вычислительных ресурсов, выделяемых отдельным ВМ) масштабирование. Пример горизонтального и вертикального масштабирования многоуровневой архитектуры представлен на рисунке 1.5 [13].

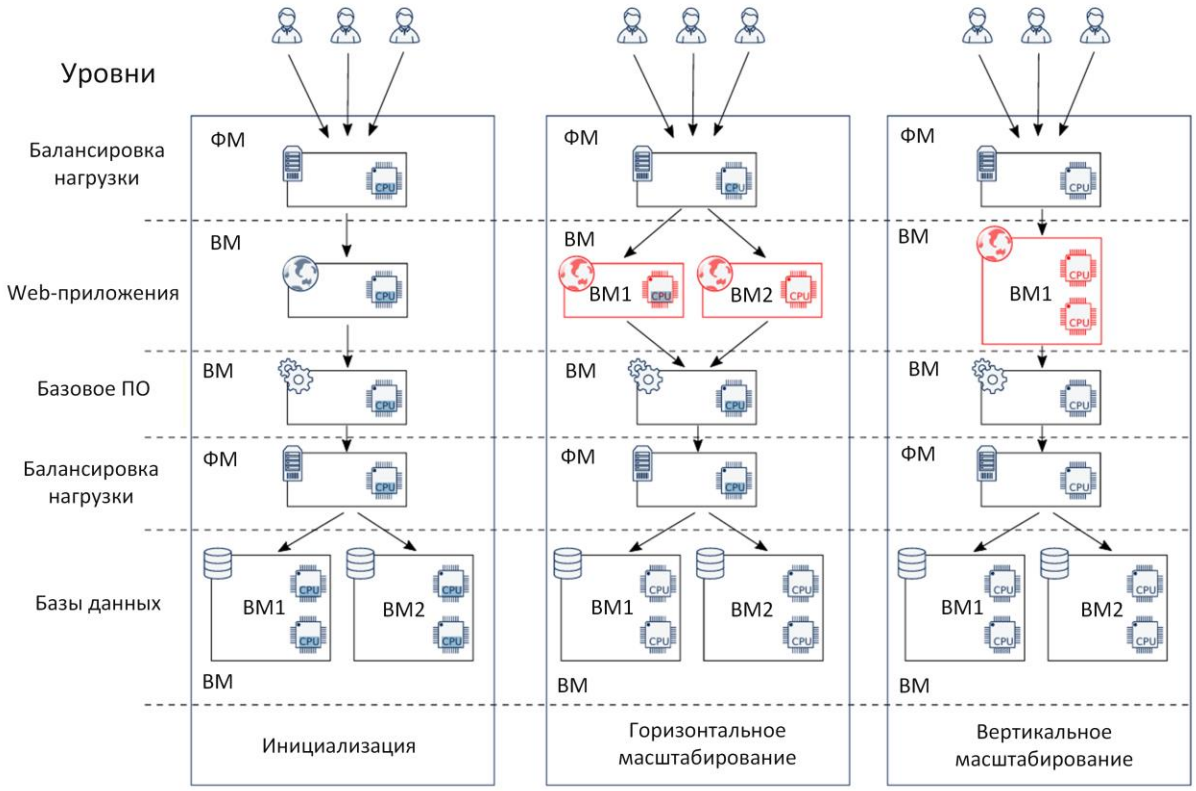


Рис. 1.5 Пример моделей масштабирования в многоуровневой архитектуре рабочей нагрузки ВЦОД

1.1.2. Анализ исследований, связанных с управлением рабочей нагрузкой ВЦОД

Процесс управления рабочей нагрузкой ВЦОД является частным случаем процесса управления рабочей нагрузкой ЦОД традиционного типа, ориентированным на рассмотренные в п. 1.1.1 особенности представления рабочей нагрузке в виртуализированной ресурсной среде.

В общем случае в исследованиях [14,15,16] и ряде других, подход к управлению может подразделяться на пассивный и активный. Пассивное управление основано на периодическом перераспределении вычислительных ресурсов ФМ между ВМ, обслуживающих текущие потребительские запросы, путем приостановки их функционирования и проведения реконфигурации ВМ. Очевидно, что подобные схемы управления ориентированы в основном на пакетную обработку данных и сложно реализуемы для интерактивных потребительских запросов.

Активное управление ориентировано на динамическую реконфигурацию ВМ без их приостановки. В свою очередь активное управление рабочей нагрузкой может носить реактивный характер – инициализацию процесса реконфигурации в ответ на определенные события, связанные с процессом обслуживания потребительских запросов или анализ текущих данных, характеризующих рабочую нагрузку, и проактивный характер, основанный на анализе ретроспективных данных (historical data), характеризующих рабочую нагрузку и выработке управляющего воздействия по результатам такого анализа. Обобщение указанных способов управления и поддерживающий их методический аппарат представлены на рисунке 1.6. В дальнейшем ряд представленных методов конкретизируется в интересах решения задачи исследования.

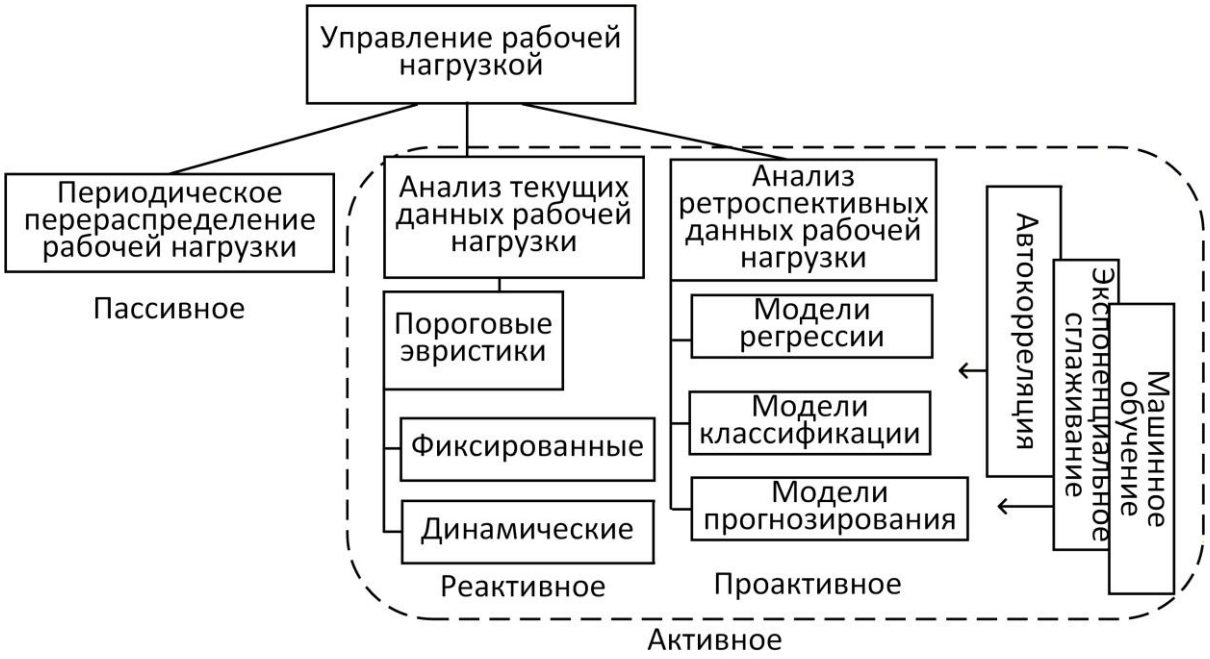


Рис. 1.6 Обобщение подходов к управлению рабочей нагрузкой ВЦОД

Вне зависимости от подхода к управлению рабочей нагрузкой ВЦОД, методы его реализации опираются на базовые характеристики рабочей нагрузки. В [14] к ним относят:

– интенсивность рабочей нагрузки (workload intensity);

– использование ресурсов (resource usage).

В рассматриваемых исследованиях, например, [17] подход к количественной оценке интенсивности рабочей нагрузки производится с точки зрения скорости поступления потребительских запросов с выполнением их кластеризации с целью группирования и оценивания динамики скорости поступления групп однородных запросов. В [18] для моделирования интенсивности рабочей нагрузки применяется методология стохастических процессов. В исследовании обосновывается невозможность использования в предметной области ВЦОД широко применяемой в телекоммуникациях модели пуассоновского процесса. В [19] при этом в модели рабочей нагрузки рассматриваются эффект пачечности и свойство самоподобия, а также обосновывается их влияние на процесс управления рабочей нагрузкой ВЦОД. В [20, 21] динамика рабочей нагрузки моделируется на основе фрактальных методов представления потока потребительских запросов. В качестве методологической основы используются дробные дифференциальные уравнения с дополнительнsм доопределением параметров на основе статистического распределения коэффициентов использования базовых вычислительных ресурсов (процессорное время и объем оперативной памяти). В дополнение к этому, в [22] для описания самоподобного поведения рабочей нагрузки ВЦОД выполняется комбинация использования таких метрик, как коэффициент вариации и индекс дисперсии рабочей нагрузки и ее модели, основанной на марковском процессе с двумя состояниями, описывающими параметры уровней интенсивности рабочей нагрузки (рисунок 1.7). В [23] для описания производительности ВЦОД марковский процесс потока потребительских запросов объединен с моделью очередей.

В отличие от интенсивности рабочей нагрузки ВЦОД, имеющей методологическую базу схожую с подобной характеристикой в области телекоммуникаций, характеристика использования ресурсов ВЦОД (resource usage) является достаточно сложной для моделирования и существенно зависит от поставленных исследовательских целей. При этом очевидно, что учет количественных значений показателей использования ресурсов является чрезвычайно важным, как при активном, так и при проактивном управлении рабочей нагрузкой (рисунок 1.6), поскольку позволяет избежать, как

недостаточного, так и избыточного выделения, как вычислительных (процессорное время, оперативная память), так и сервисных (объем системы хранения данных, пропускная способность сети) ресурсов.

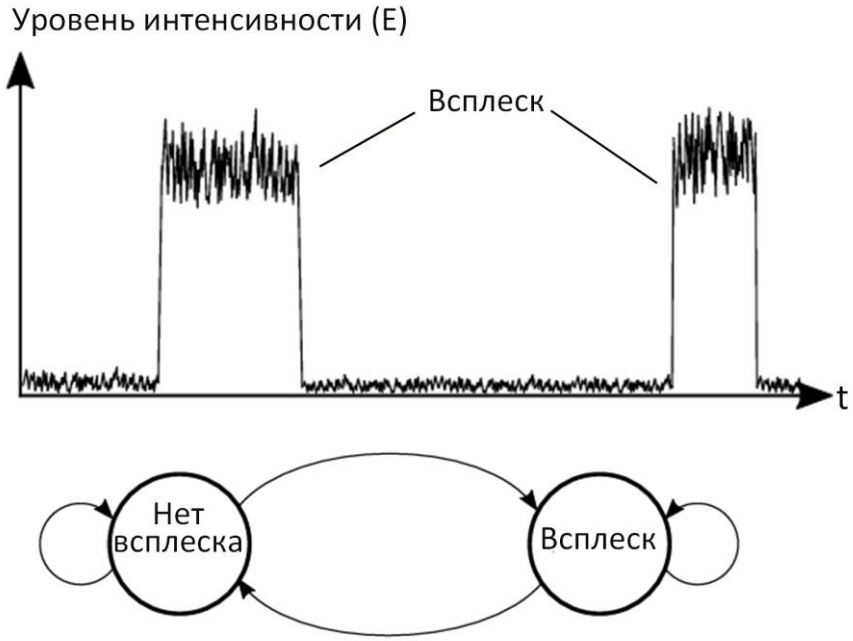


Рис. 1.7 Моделирование интенсивности рабочей нагрузки ВЦОД марковским процессом с двумя состояниями

С точки зрения характеристики объема используемых ресурсов в [17] рабочую нагрузку разделяют на два класса:

– критичную к базовым вычислительным ресурсам (процессорное время, оперативная память);

– критичную к вводу-выводу ВЦОД, к которому относят, как внутренний ввод-вывод вычислительной системы, связанный, в первую очередь с системой хранения данных (СХД), так и сетевой ввод-вывод, связанный с пропускной способностью сетевой подсистемы с интенсивными вычислениями или вводом-выводом.

В общем случае, первый класс рабочей нагрузки характерен для ее пакетных видов, в то время, как второй больше присущ интерактивным видам рабочей нагрузки.

Также, в отличие от пакетных рабочих нагрузок, характеризующихся относительно равномерным требование к выделению ресурсов, интерактивные виды рабочей нагрузки, например, порождаемые различного вида онлайн-сервисами, могут иметь сложные временные характеристики, такие как периодичность смены доступа к вычислительным ресурсам и ресурсам ввода-вывода, наличие всплесков доступа к различным видам ресурсов (рисунок 1.7), а также рост и падение доступа к ним, описываемые разными законами распределения.

Поскольку рабочая нагрузка ВЦОД формируется потоком потребительских запросов, который, в общем случае, имеет недетерминированный характер, то использование ресурсов ВЦОД может существенно зависеть от времени суток, быть связано с различными сезонными явлениями (праздничные дни), или явлениями, основанными на внезапных событиях (феномен «выброса пепла» (flash crowd)) [17].

Таким образом, рассмотренные исследования показывают, что представленные характеристики рабочей нагрузки ВЦОД оказывают существенное влияние, как на производительность его отдельных подсистем, так и на производительность ВЦОД в целом.

В связи с этим, важную роль в структуре системы администрирования ВЦОД играет подсистема мониторинга рабочей нагрузки, основанная на мониторинге ее характеристик.

1.1.3. Системы мониторинга рабочей нагрузки ВЦОД

Подсистема мониторинга рабочей нагрузки ВЦОД в общем случае решает две задачи администрирования:

1. Отслеживает действий рабочей нагрузки, порождаемой потоком потребительских запросов. К таким действиям можно отнести: запуск, остановку, приостановку ВМ, запуск и завершение конкретных приложений в рамках ВМ, запросы на реконфигурацию ресурсов ВМ и т.д.

2. Контролирует выделенные и доступные для выделения вычислительные и коммуникационные ресурсы ВЦОД.

Решение этих задач подсистемой мониторинга является основой таким высокоуровневых задач службы администрирования ВЦОД, как:

– оплата использования ресурсов ВЦОД (billing);

– вопросы информационной и технологической безопасности;

– устранение сбоев и отказов компонентов ВЦОД;

– соблюдение условий SLA;

– планирование развития компонентов ВЦОД.

Очевидно, что в силу гибкости подходов предоставления ресурсов ВЦОД потребителям, а также высокой динамики этого процесса, особенно в современных онлайн-сервисах, решение является достаточно сложныv и требует исследования и разработки соответствующих подходов. Обзор таких подходов дается, например, в [24].

В целом, в исследованиях, связанных с мониторингом рабочей нагрузки [15, 25-28] отмечается, что процесс мониторинга должен затрагивать, как минимум, три объекта: множество ФМ, множество ВМ, множество потребителей. При этом базовое свойство систем виртуализации – изоляция отдельных ВМ и отделение управления ими от управления ресурсами ФМ требует специфических подходов к процессу мониторинга, отличающихся от мониторинга ресурсов традиционных ЦОД. Так, мониторинг производительности ВМ, как объект потребления ресурсов ФМ дополняется мониторингом ресурсов их гостевых операционных систем, путем выполнения корреляции данных внутренних (по отношению к ВМ и внешних журналов мониторинга).

В общем случае средства мониторинга рабочей нагрузки разрабатываются с использованием модели «агент-менеджер». При этом множество распределенных программных и/или программно-аппаратных агентов, периодически собирающих информацию об использовании вычислительных (процессорное время, оперативная память) и коммуникационных (подсистема ввода-вывода, сетевая подсистема) ресурсов, развертываются на ключевых уровнях ВЦОД и передают данные программному менеджеру/менеджерам подсистемы мониторинга, которая сохраняется в специализированной части СХД – базе ретроспективных данных

мониторинга. Обычно основой таких средств мониторинга могут быть, как системные компоненты host и гостевой операционной системы такие, как vmstat, iostat, netstat [15], так и специализированные продукты. При этом, в зависимости от функциональных возможностей средств мониторинга, могут собираться, как высокоуровневые данные, связанные, например, с планированием и подготовкой ВМ (количество, типы ВМ, количественные характеристики выделенных им ресурсов) [27], так и низкоуровневые количественные данные (время использования и ожидания процессорных ядер, число операций подкачки страниц виртуальной памяти, число и тип прерывания ввода-вывода и т.д.) [28]. Обобщенная схема функционирования подсистемы мониторинга ВЦОД представлена на рисунке 1.8.

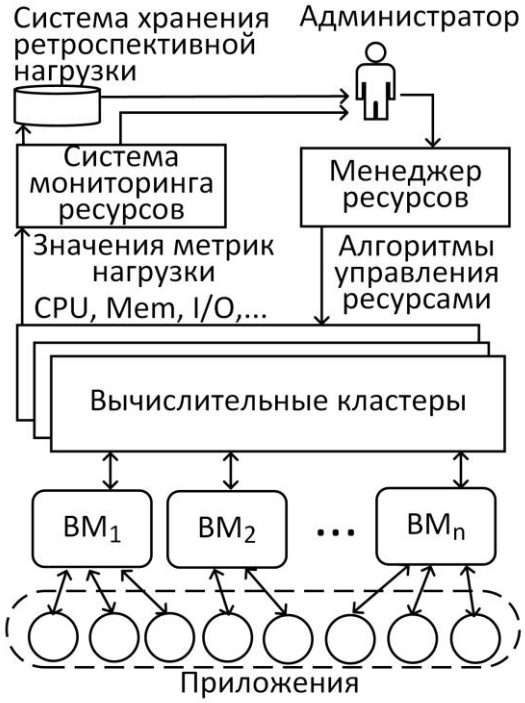


Рис. 1.8 Обобщенная схема функционирования подсистемы мониторинга ВЦОД

Наиболее известными реализациями систем мониторинга крупномасштабных коммерческих ВЦОД являются: Google Resource Manager ВЦОД Google Cloud Platform [29], AWS CloudWatch ВЦОД Amazon Web Services, Azure Monitor ВЦОД Microsoft Azure, IBM Tivoli Monitoring, Rightscale, Cloudify, Rackspace [30].

Развернутый обзор средств мониторинга ВЦОД представлен в [31]. Среди отечественных крупномасштабных ВЦОД можно выделить систему мониторинга Yandex Monitoring ВЦОД Yandex Cloud [32].

Существуют реализации систем мониторинга ВЦОД с открытым исходным кодом. К ним можно отнести: проект Nagios, являющийся частью облачной платформы OpenStack [33], а также Ganglia, Collectl и MonALISA [29]. К переспективным направлениям развития систем мониторинга ВЦОД можно отнести исследования в области разработки концепции MaaS (Monitoring as a Service) [34, 35].

1.1.4. Представление рабочей нагрузки ВЦОД временными рядами использования ресурсов

Рассмотренные в п. 1.1.3 системы мониторинга ориентируются на периодический опрос заданных администратором ВЦОД показателей, характеризующих использование вычислительных и коммуникационных ресурсов ВЦОД. В большинстве систем мониторинга программные агенты для получения этих показателей, именуемые «счетчики производительности» (performance counter), создаются для мониторинга различных физических (процессор, оперативная память) и логических (процесс, поток) объектов. На рисунке 1.9 представлены примеры счетчиков производительности системы мониторинга Microsoft Performance Monitor.

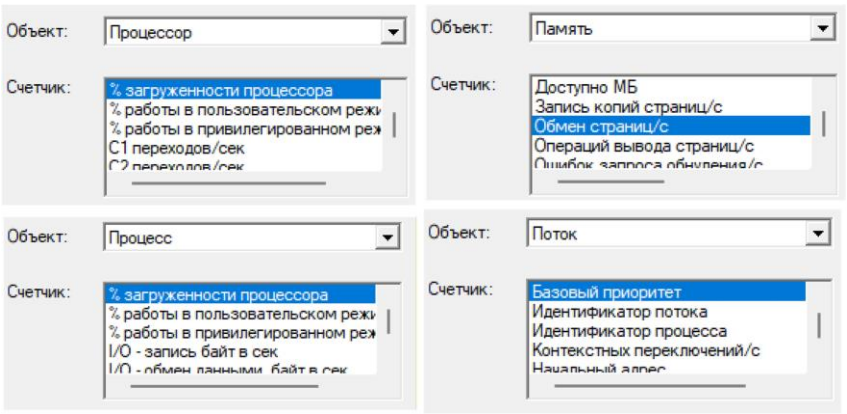


Рис. 1.9 Пример показателей счетчиков производительности системы мониторинга Microsoft Performance Monitor

Для каждого из счетчиков задается определенная скважность (интервал) опроса, соответствующий логике функционирования объекта мониторинга.

Результатом такого периодического опроса является одномерный массив значений выбранного показателя, каждое значение которого привязано к временной метке интервала опроса. Таким образом, в общем случае, результатом мониторинга заданного объекта является временной ряд (time series) значений выбранных показателей мониторинга.

Очевидно, что выбор интервала мониторинга, в зависимости от логики функционирования объекта мониторинга, существенно влияет на формируемый временной ряд значений показателя. На рисунке 1.10 представлен фрагмент временного ряда показателя «% загруженности процессора» для одного и того же объекта «Поток» многопоточного приложения, полученный с интервалами мониторинга 1 секунда и 0,01 секунда соответственно.

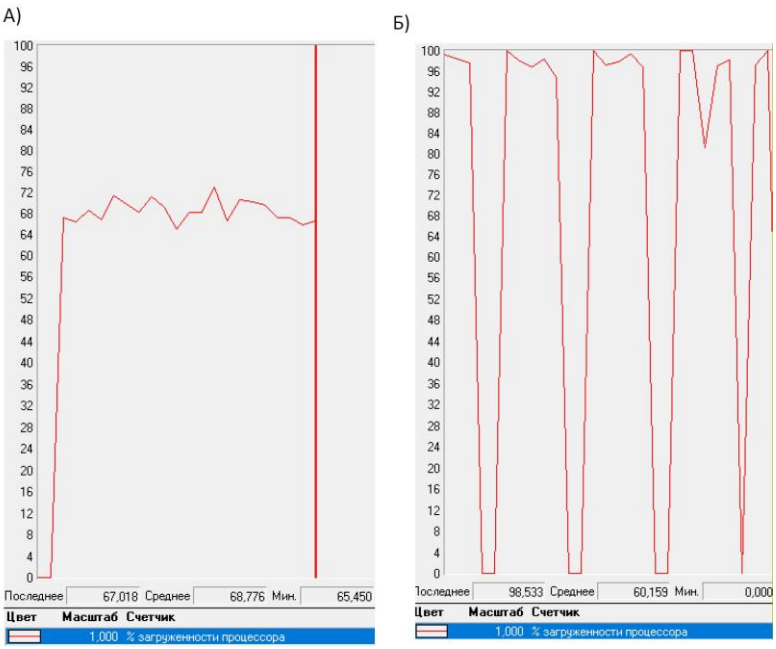


Рис. 1.10 Значения временного ряда показателя «% загруженности процессора» для объекта «Поток» с интервалами: а) 1 сек. и б) 0,01 сек.

При этом логика функционирования объекта «Поток» известна: периодическое 100% задействование процессора (состояние выполнения) и периодический переход в состояние ожидания. Из рисунка 1.10 видно, что выбор

интервала мониторинга в 1 секунду не позволяет не только не получить достоверный результат мониторинга, но и не отражает логику функционирования объекта «Поток». В то время, как выбор интервала мониторинга 0,01 секунды дает достоверные значения временного ряда. Это обусловлено относительной скоростью реализации функций объекта «Поток», при которой на длительном интервале мониторинга возможно получение только усредненного значения состояний «выполнение» и «ожидание».

Сохранение результатов мониторинга показателей рабочей нагрузки ВЦОД позволяет службе администрирования решать такую важную задачу, как профилирование (шаблонирование) рабочей нагрузки (workload profiling) [36]. Задача профилирования основывается на том факте, что при выполнении типовых потребительских запросов выделение и использование ФМ и ВМ, вычислительных и коммуникационных ресурсов в среднем являются одинаковыми. То есть, в общем случае, можно говорить о некоторых типовых профилях (шаблонах) рабочей нагрузки, в случае штатного процесса функционирования ВЦОД. Пример шаблонов показателя рабочей нагрузки «% использования процессора», полученные в ходе исследования для системы контейнеризации Kubernetes и ежесуточного функционирования ВМ, поддерживающей выполнение Web-сервера, представлен на рисунке 1.11.

Из рисунка видно, что задействование процессора («% использования процессора»), как в случае выполнения контейнера, так и в случае функционирования ВМ с развернутым на ее базе Web-сервисом имеют достаточно четкие фазы. В случае контейнера Kubernetes эти фазы отражают логику его функционирования от момента запуска до выхода на некоторый стационарный режим функционирования. В случае ВМ (результат получен по среднесуточному оцениванию показателя), фазы отражают изменение потребностей ВМ в ресурсе процессорного времени в зависимости от логики функционирования поддерживаемого WEB-сервиса, которая, в свою очередь, отражает динамику интенсивности потребительских запросов к данному Web-сервису.

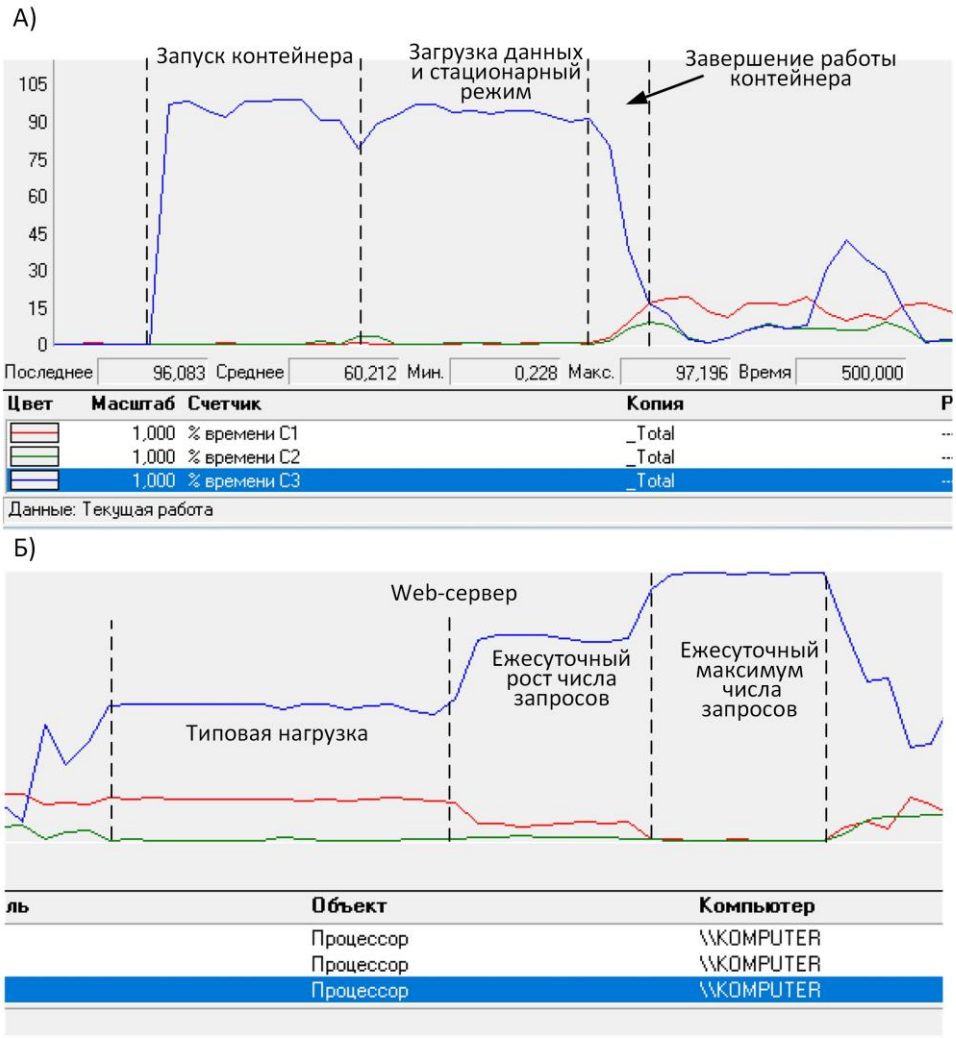


Рис. 1.11 Пример периодических шаблонов рабочей нагрузки: А) Выполнение контейнера Kubernetes, Б) Ежесуточное функционирование виртуализированного Web-сервера

Сохранение временных рядов заданных показателей рабочей нагрузки позволяют оценить особенность функционирования отдельных компонентов ВЦОД на выбранных интервалах времени. Так, на рисунке 1.12 представлены числовые ряды показателя «% загруженности процессора) для трех ВМ с различным ПО, функционирующим в их составе, полученные за один и тот же 15-минутный временной интервал [37].

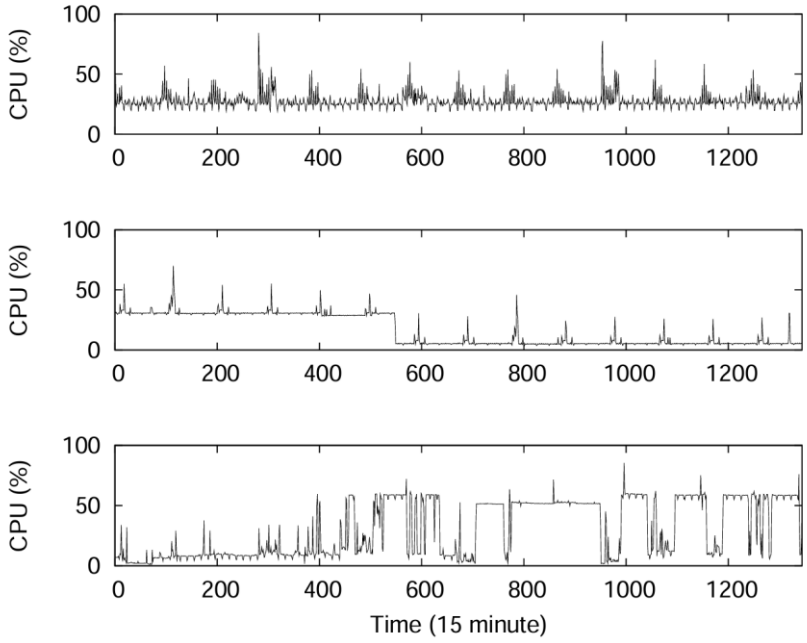


Рис. 1.12 Пример шаблонов рабочей нагрузки (показатель «% использования процессора) трех ВМ за единый временной интервал

Из рисунка 1.12 видно, что логика функционирования первой ВМ носит периодический характер обращения к ресурсу процессора, характерный, например, для циклической вычислительной задачи, ПО в составе второй ВМ, также решая периодическую задачу, загружает (в среднем) 50% процессорного времени за первую половину интервала мониторинга и простаивает во второй половине. Функционирование третьей ВМ носит не периодический характер, что характерно для интерактивной рабочей нагрузки.

В случае рассмотрения результатов мониторинга на уровне всего ВЦОД можно говорить о формировании шаблонов специфического вида временного ряда – тепловой карты (heatmap) – метода его цветовой визуализации, при котором числовые значения оцениваемого показателя кодируются набором цветов. На рисунке 1.13 представлен пример тепловой карты, отображающей требования к ресурсам ВЦОД со стороны множества ВМ исследуемого ВЦОД [37] за определенные временные интервалы.

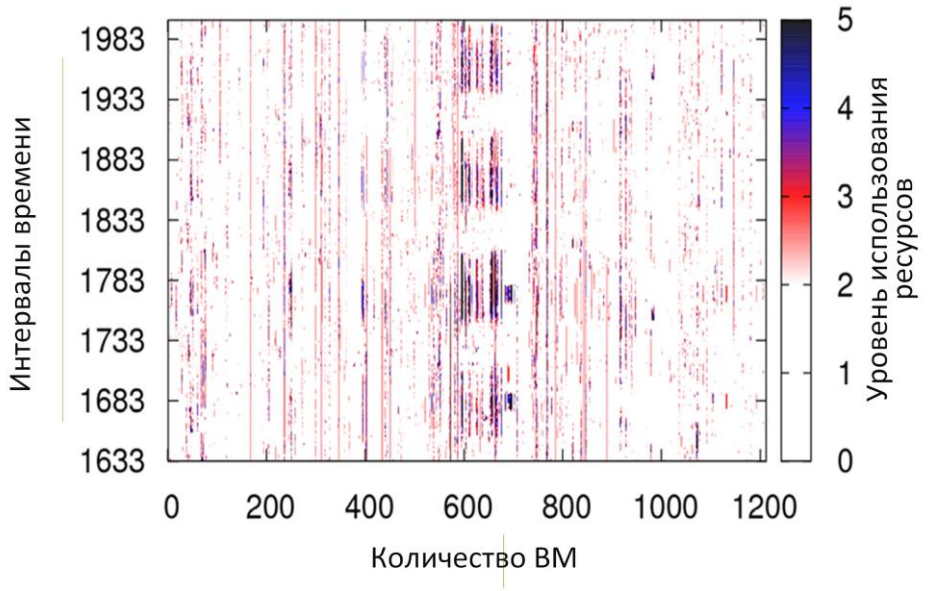


Рис. 1.13 Пример представления временного ряда рабочей нагрузки ВЦОД в виде тепловой карты

Важной особенностью функционирования подсистемы мониторинга рабочей нагрузки современных ВЦОД является сохранение полученных значений временных рядов выбранных показателей рабочей нагрузки функционирующих объектов ВЦОД (ФМ, ВМ, контейнеров) в специализированной базе данных, именуемой базой ретроспективных данных рабочей нагрузки (workload historical data database), являющейся частью СХД ВЦОД, используемой службой администрирования. В большинстве случаев ретроспективные данные подобных баз большинства ВЦОД имеют конфиденциальный статус. Известными базами ретроспективных данных рабочей нагрузки, находящихся в открытом доступе и доступными для исследователей, являются: база ВЦОД Google Cluster [38] и база ВЦОД Alibaba Cluster [39]. В рамках проводимого исследования использовались временные ряды ретроспективных данных рабочей нагрузки базы ВЦОД Google Cluster. Пример временных рядов разной масштабности (ежедневный и ежеминутный) из этой базы – трассировки показателя CPU Usage Rate – коэффициент загрузки процессора, процент времени, в течение которого он занят обработкой задач, рассчитываемый путём деления времени работы процессора на общее время за заданный период, представлены на рисунке 1.14.

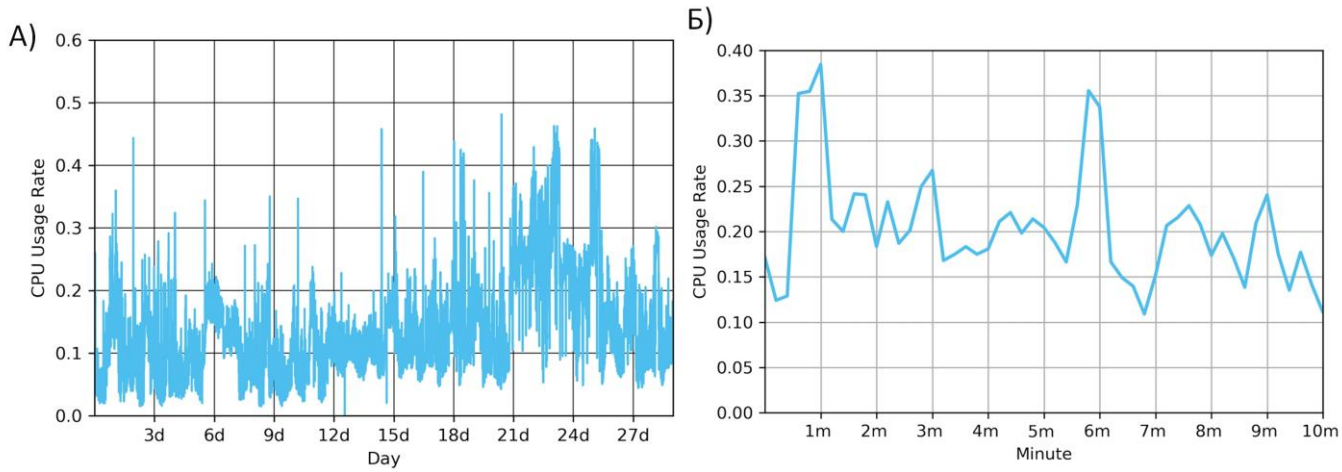


Рис. 1.14 Пример временных рядов показателя CPU Usage Rate: а) ежедневной за период 30 дней, б) ежеминутной за период 10 минут, из базы ретроспективных данных рабочей нагрузки ВЦОД Google Cluster

1.1.5. Анализ исследований задачи прогнозирования рабочей нагрузки ВЦОД на основе временных рядов ее ретроспективных данных

Наличие во временных рядах показателей рабочей нагрузки ВЦОД ее достаточно устойчивых профилей (шаблонов), а также возможность сохранения множества полученных временных рядов в базе ретроспективных данных рабочей нагрузки, позволяет сформулировать и реализовать задачу прогнозирования рабочей нагрузки ВЦОД (workload prediction, workload forecasting).

Исследования решения задачи прогнозирования рабочей нагрузки ВЦОД представлены в [5, 37, 40-42]. В частности, в [40] дается обширное исследование подходов к решению задачи прогнозирования рабочей нагрузки и формулируется ее постановка.

В общем случае задача прогнозирования рабочей нагрузки формулируется, как получение значений временных рядов выбранных показателей рабочей нагрузки в будущие моменты времени на основе ее шаблонов временных рядов в предшествующие моменты времени (рисунок 1.15).

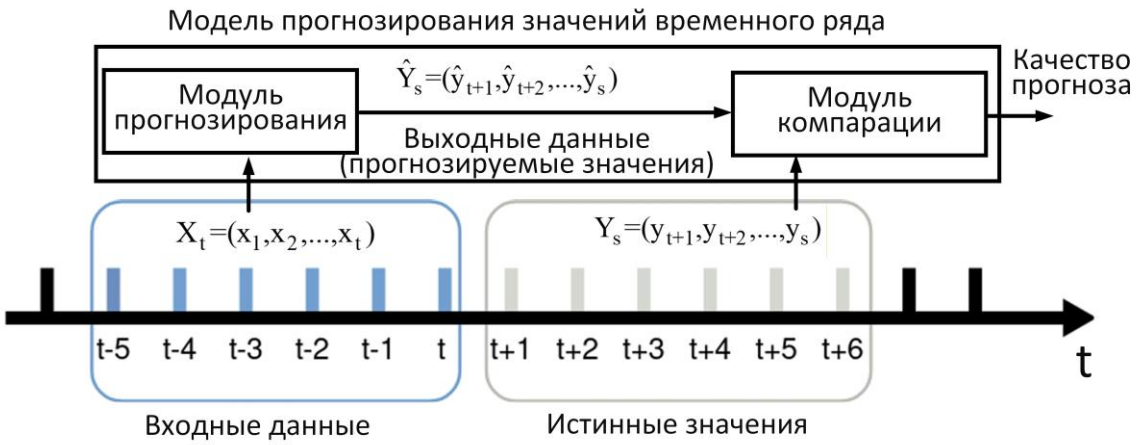


Рис. 1.15 Входные и выходные данные задачи прогнозирования временного ряда

Формально указанную задачу можно представить следующим образом. Пусть выборка показателя рабочей нагрузки из базы ее ретроспективных данных представлена множеством значений (истинных значений):

$$X_t=(x_1,x_2,...,x_t), \quad (1)$$

где x_t - значение показателя, полученное в t-й момент времени.

Обозначим множество значений показателя в будущий интервал времени $(t+1,t+2,...,s)$ – истинных значений, как:

$$Y_s=(y_{t+1},y_{t+2},...,y_s) \quad (2)$$

Обозначим множество ожидаемых (прогнозируемых) значений показателя в будущий интервал времени $(t+1,t+2,...,s)$, как:

$$\hat{Y}_s=(\hat{y}_{t+1},\hat{y}_{t+2},...,\hat{y}_s) \quad (3)$$

Целевая функция задачи прогнозирования может быть выражена следующим образом:

$$\hat{Y}_s=Z_{predict}(X_t) \quad (4)$$

Мощность множеств Y_s и $\hat{Y}_s$ является одинаковой. Тогда видом целевой функцией задачи прогнозирования рабочей нагрузки $Z_{predict}$ является минимизация расхождения между значениями множеств Y_s и:

$$Z_{predict} = \min Y_s \setminus \hat{Y}_s \quad (5)$$

То есть, фактически, целевой функцией задачи $Z_{predict}$ является минимизация разности множеств Y_s и $\hat{Y}_s$.

1.1.6. Метрики оценивания эффективности процесса прогнозирования рабочей нагрузки ВЦОД

Важной исследовательской задачей, связанной с решением проблемы прогнозирования рабочей нагрузки ВЦОД, является обоснованный выбор метрик оценивания эффективности процесса прогнозирования.

Анализ источников, посвященных решению задачи прогнозирования временных рядов в различных предметных областях, в том числа в предметной области управления рабочей нагрузкой ВЦОД [43-45], показал, что для оценивания эффективности разрабатываемых моделей и алгоритмов прогнозирования, в общем случае, используют следующие метрики, отражающие различные аспекты точности и погрешности процесса прогнозирования. К ним относят:

1. Среднеквадратическая ошибка (MSE).
2. Коэффициент детерминации (R-квадрат, R^2) [45].
3. Среднеквадратическая логарифмическая ошибка (RMSLE).
4. Средняя абсолютная ошибка (MAE).

Как было указано в п. 1.1.5 качестве показателей используются:

– y_n – истинное значение элемента временного ряда;

– $\hat{y}_n$ – прогнозируемое значение элемента временного ряда, где n – количество выборок.

Рассмотрим указанные метрики подробнее:

Метрика MSE – это средняя сумма квадратов разностей между истинными и предсказанными значениями, и диапазон ее значений составляет $[0, +\infty]$.

$$MSE=\frac{1}{N}\sum_{n=1}^{N}\left(\hat{y}_n - y_n\right)^2 \quad (6)$$

Когда предсказанное значение совпадает с фактическим, значение метрики MSE равно 0. Кроме того, чем больше ошибка прогнозирования, тем больше значение метрики MSE.

Метрика R^2 используется для оценивания степени соответствия прогнозируемых данных фактическим:

$$R^2=1-\frac{\sum_{n=1}^{N}\left(\hat{y}_n-y_n\right)^2}{\sum_{n=1}^{N}\left(\hat{y}_n-\bar{y}\right)^2}, \quad (7)$$

где, $\bar{y}$ обозначает среднее значение всех истинных значений.

В идеальном случае, когда все прогнозируемые значения равны фактическим, значение метрики R2 равно 1.

Диапазон значений этой метрики составляет $[-\infty,1]$.

Метрика RMSLE используется для обработки больших выбросов в данных. Она применяется, когда имеется предположение об искажениях входных данных (истинных значений – выражение 1) и, соответственно значения метрики MSE завышены. Поэтому данные обрабатываются с помощью логарифмической операции, и обработанные данные используются для вычисления метрики RMSLE следующим образом:

$$RMSLE=\sqrt{\frac{\sum_{n=1}^{N}\left(\log\left(y_n+1\right)-\log\left(\hat{y}_n+1\right)\right)^2}{n}} \quad (8)$$

Метрика MAE определяет среднее арифметическое абсолютных значений ошибок между прогнозируемыми и фактическими значениями и определяется следующим образом:

$$MAE=\frac{1}{N}\sum_{n=1}^{N}\left|\hat{y}_n - y_n\right| \quad (9)$$

1.1.7. Проблемы зашумления временных рядов ретроспективных данных рабочей нагрузки ВЦОД

Одной из наиболее значимых проблем, связанных с решением задач анализа временных рядов ретроспективных данных рабочей нагрузки ВЦОД, в частности, задачи прогнозирования рабочей нагрузки (п. 1.1.5) является влияние на значения временных рядов ее показателей факторов «зашумления», искажающих эти значения. Такими факторами могут выступать, как сами приложения или сервисы, функционирующие в составе ВМ и контейнеров ВЦОД, так и приложения и сервисы в составе других ВМ и контейнеров, оказывающие взаимное влияние на использование вычислительных и коммуникационных ресурсов.

Анализ исследований, связанных с факторами зашумления значений временных рядов в смежных предметных областях [46, 47] позволил выделить два наиболее значимых фактора:

1. Фактор зашумления, связанный с проблемой «шумных соседей» (Noisy Neighbours);

2. Фактор зашумления, связанный с проблемой «старения» программного обеспечения (Software Aging).

Формулировка проблемы «шумных соседей» применительно к зашумлению значений временного ряда показателей рабочей нагрузки ВЦОД, в частности, показателей, связанных с использованием ресурса процессорного времени, представлена в [48], и конкретизирована в рамках исследования в [49]. Основой возникновения проблемы является особенность архитектуры многоядерных процессоров и многопроцессорных систем, связанная с предоставлением процессам и/или потокам конкретных ядер процессоров ФМ ВЦОД. Указанная особенность реализуется технологией «CPU Affinity» [48] – закреплением и откреплением процесса или потока к/от конкретному(ого) ядру(а) процессора, процессору или множеству процессоров многопроцессорной (многоядерной) вычислительной системы, так что процесс или поток будут выполняться только на указанном ядре,

процессоре или процессорах, а не на любом процессоре. Использование технологии «CPU Affinity» обеспечивает реализацию принципа локальности обрабатываемых данных, размещаемых процессом/потоком в кэш-памяти процессорного ядра, за которым он закреплен и, в основном, ориентирован на статический вычислительный процесс. Пример назначения потоку конкретного ядра процессора представлен на рисунке 1.16.

В условиях динамической реконфигурации вычислительных ресурсов, характерных для ВЦОД, технология «CPU Affinity» порождает эффекты взаимного влияния потоков (как одного процесса, так и межпроцессных потоков). Это связано с необходимостью получения доступа к кэш-памяти закрепленного процессорного ядра, даже если поток на нем уже не выполняется, но, в силу закрепления, сохранил там свои данные.

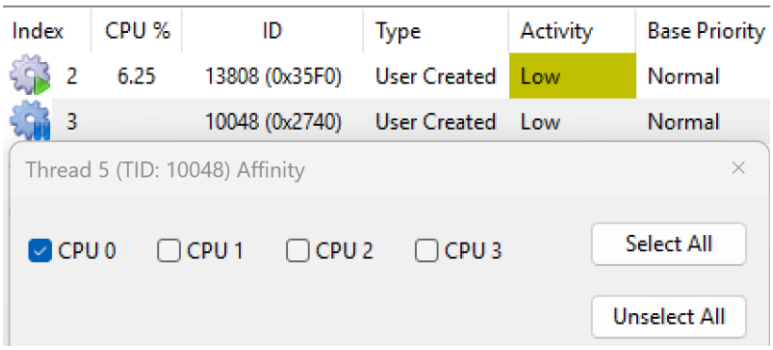


Рис. 1.16 Пример закрепления за потоком (TID=10048) первого процессорного ядра (CPU0) четырёхъядерного процессора

На рисунке 1.17 рассматривается схема синтезированного в ходе исследования эффекта «шумных соседей» многопоточного приложения CPUStress (рассмотрен в [49]).

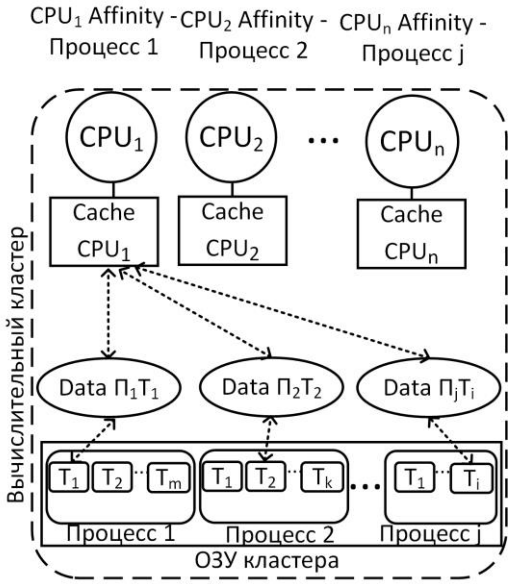


Рис. 1.17 Пример возникновения проблемы «шумные соседи» за счет технологии «CPU Affinity»

Из рисунка 1.17 видно, что в момент времени t закрепления трех многопоточных процессов $П_1$, $П_2$ и $П_j$ за процессорами (процессорными ядрами) 1, 2 и n соответственно. При этом поток T_1 размещает свои данные $Data(П_1T_1)$ в кэше процессора CPU1. Однако, в предыдущие моменты времени t-1 и t-2 процессор CPU1 был закреплен за процессами $П_2$ и $П_j$ соответственно, и их потоки разместили в его кэш-памяти данные $Data(П_2T_2)$ и $Data(П_jT_i)$. Будучи перезакрепленными в момент времени t за CPU2 и CPUn соответственно, процессы $П_2$ и $П_j$, тем не менее, требуют получения своих данных из кэш-памяти процессора CPU1, что приводит к необходимости приостановки процесса $П_1$. В зависимости от частоты таких обращений к кэшу производительность процесса $П_1$ будет снижаться, что отразится на временном ряду значений метрики «% использования CPU» системы мониторинга. При этом процессы $П_2$ и $П_j$ выступают в роли «шумных соседей» процесса $П_1$, зашумляя своими значениями временной ряд его значений (рисунок 1.18). Практическая реализация рассмотренного выше синтезированого эффекта «шумных соседей» представлена на рисунке 1.19 (использовался монитор «Microsoft Performance monitor», интервал мониторинга 0,01 секунды).

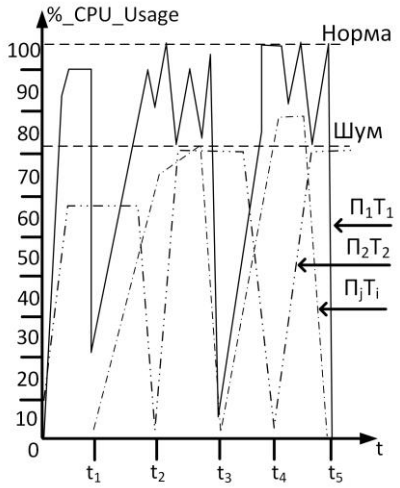


Рис. 1.18 Зашумление временного ряда показателя «% использования процессора» потока $П_1Т_1$ потоками $П_2Т_2$ и $П_jТ_i$

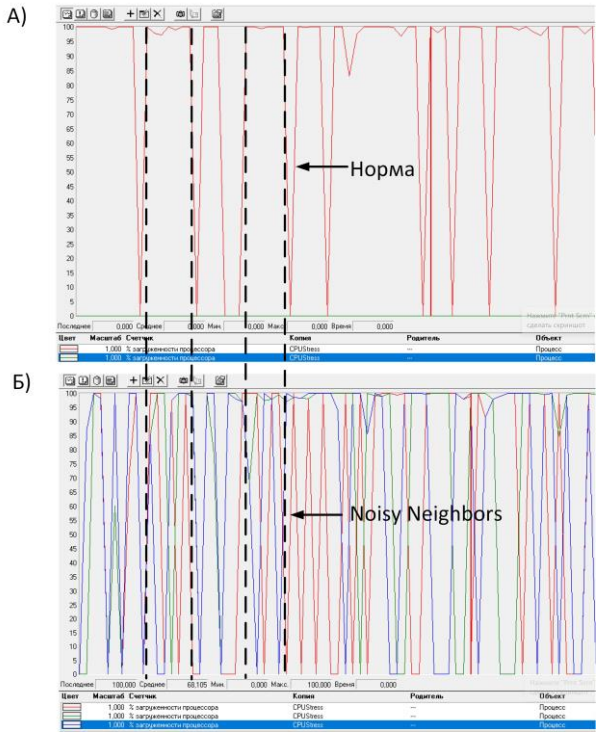


Рис. 1.19 Визуализация проблемы Noisy Neighbors для трех потоков (показатель «%загруженности процессора»)

Пример возникновения проблемы «шумных соседей» в процессе функционирования реального ВЦОД рассмотрен в [50]. Эффект «шумного соседа» наблюдался в арендуемой вычислительной платформе «Titus» (сервис потокового видео Netflix). При этом «шумный сосед» представляет собой контейнер или

системный сервис, который интенсивно использует серверные ресурсы, что приводит к падению производительности близких к нему контейнеров. Поиск проблемы осуществлялся путем наблюдения за временем, которое процессы проводят в очередях до получения процессорного времени. Слишком длительное ожидание процесса в этой очереди может свидетельствовать о проблемах с производительностью, особенно в тех случаях, когда контейнер не использует выделенные ему ресурсы CPU в полном объёме. На рисунке 1.20 представлены временные ряды исследуемого процесса до и после возникновения фактора зашумления «шумные соседи».

Рис. 1.20 Пример возникновения проблемы «шумные соседи» при выполнении двух контейнеров, развернутых в ВЦОД «Titus». А) Нормальная производительность контейнера container1, Б) Снижение производительности контейнера *container1* за счет захвата ресурса времени процессора контейнером *container2*

На рисунке 1.20 б) график зеленого цвета отображает метрику *sched.switch.out,* которая демонстрирует, что причиной проблемы было усиление вытеснения процессов контейнера *container1* «шумными соседями» – системными процессами, запущенные контейнером *container2*, и потребляющими все доступные процессорные мощности.

К не менее значимым фактором зашумления временных рядов показателей рабочей нагрузки ВЦОД исследования [51-54] относят проблему «старения» ПО (software aging).

В [51] под старением ПО подразумевают явление, заключающееся в нарастающей деградации внутреннего состояния ПО в течение его жизненного цикла. Проблема старения ПО носит кумулятивный характер и, соответственно, происходит интенсивнее в вычислительных системах, работающих в непрерывно или в течение длительного времени, что характерно для иерархии ПО ВЦОД.

На рисунке 1.21 представлены фазы старения ПО, рассмотренные в [52].

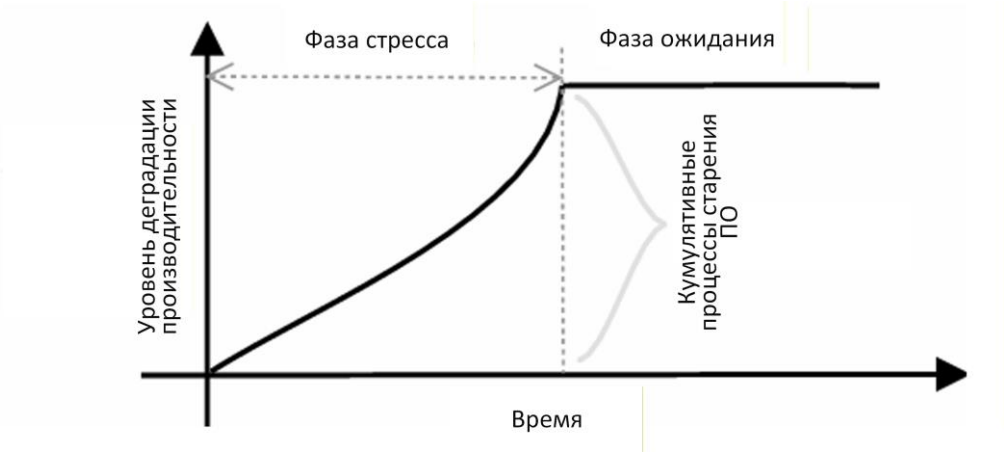


Рис. 1.21 Фазы старения программного обеспечения

Из рисунка видно, что базовой фазой является фаза стресса, в течение которой уровень деградации производительности ПО повышается по некоторому закону (чаще всего зависимость нелинейная). Эта фаза отражает процесс кумулятивного накопления множества факторов, определяющих старение ПО. Следующей фазой является фаза ожидания, в течение которой ведется наблюдение за показателем уровня деградации производительности. Если в течение длительного времени

уровень деградации производительности не снижается, то принимается решение о факте старения исследуемого ПО.

Таким образом, в отличие от проблемы «шумных соседей», проблема старения ПО связана с недостаточно эффективной логикой его функционирования, заложенной при его проектировании, что может приводить к:

– несвоевременному освобождению процессом/потоком кэш-памяти процессорного ядра после завершения работы на нем;

– порождению несанкционированных дочерних процессов и потоков, требующих использования процессорного ядра;

– появлению незавершаемых процессов (состояние zombie), требующих использования процессорного времени и оперативной памяти;

– загрузке в кэш-память процессора несанкционированных внешних данных, что требует приостановки выполнения на ядре текущего процесса/потока и т.д.

На рисунке 1.22 [53] представлена трассировка показателя свободной оперативной памяти UNIX-системы, демонстрирующая деградацию объема этого ресурса при выполнении множества процессов.

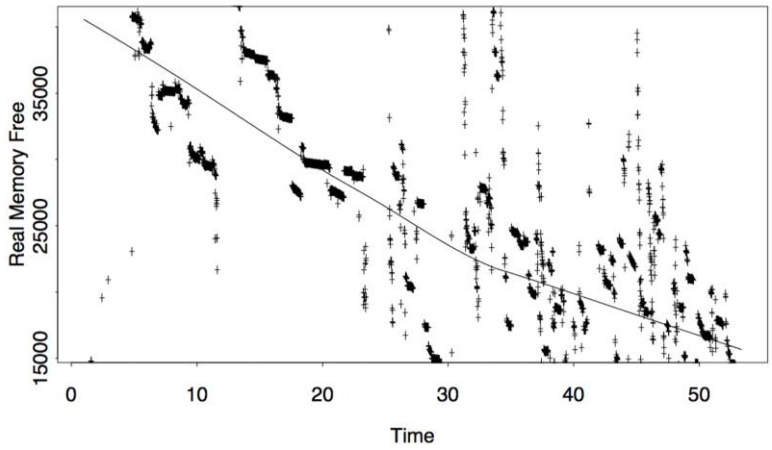


Рис. 1.22 Пример деградации ресурса свободной оперативной памяти, связанной с фактором старения ПО

В [54] проблема старения ПО рассматривается на нескольких уровнях ВЦОД, а также выделяются метрики - индикаторы оценивания этого процесса для каждого из уровней (рисунок 1.23).

Рис. 1.23 Многоуровневое представление метрик – индикаторов процесса старения ПО

Рассмотренные выше факторы могут оказывать влияние на значения временного ряда данных, как по-отдельности, так и комбинировано, порождая различные по характеру картины зашумления.

Подобные «зашумленные» значения временных рядов ретроспективных данных не могут быть использованы в качестве обучающей выборки системы прогнозирования рабочей нагрузки, поскольку не отражают объективное представление производительности вычислительной системы в конкретный момент времени. В связи с этим существует множество исследований, связанных со снижением влияния шумовой компоненты. Используемые в них методы обобщаются в решение задачи «очистки данных».

1.1.8. Постановка задачи исследования

Исходя из рассмотренной в п. 1.1.5 формализации задачи прогнозирования временных рядов, а также рассмотренных в п. 1.1.6 метрик оценивания эффективности процесса прогнозирования и, рассмотренных в п. 1.1.7 факторов зашумления временных рядов, можно выдвинуть следующие предположения:

– рабочая нагрузка ВЦОД может быть представлена множеством временных рядов значений ее показателей (в частности, использования вычислительных ресурсов), составляющих базу ретропективных данных рабочей нагрузки;

– временные ряды значений показателей рабочей нагрузки в различные моменты времени отражают шаблоны (профили) ее реализации, требующие от службы администрирования ВЦОД реконфигурации его инфраструктуры, с целью оптимизации использования вычислительных и коммуникационных ресурсов;

– прогнозирование временных рядов рабочей нагрузки на основе ее ретроспективных данных, в частности, участков временных рядов, соответствующих требуемым шаблонам рабочей нагрузки, позволит повысить эффективность процесса функционирования ВЦОД;

– в силу особенностей реализации вычислительных процессов ВЦОД, связанных с развертыванием, эксплуатацией и реконфигурацией множества ВМ по множеству ФМ ВЦОД, возникают эффекты зашумления временных рядов показателей рабочей нагрузки, способные исказить результаты прогнозирования.

Таким образом, обобщенно задачу исследования можно представить следующей схемой (рисунок 1.24).

Формально, задачу исследования можно представить следующим образом:

Дано:

$X_t=(x_1,x_2,...,x_t)$ - исходный временной ряд показателя рабочей нагрузки, подверженный факторам зашумления.

$Y_s=(y_{t+1},y_{t+2},...,y_s)$ - временной ряд прогнозируемых (истинных значений рабочей нагрузки.

Требуется:

1. Разработать модель, реализующую функцию $\hat{X}_t=f_{denoise}(X_t)$ преобразования элементов множества X_t в элементы множества $\hat{X}_t$, не содержащие факторы зашумления.

2. Разработать модель, реализующую функцию $\hat{Y}_s=f_{denoise}(\hat{X}_t)$ прогнозирования рабочей нагрузки, представленной элементами множества $\hat{X}_t$ для

получения прогнозных значений временного ряда $\hat{Y}_s=(\hat{y}_{t+1},\hat{y}_{t+2},...,\hat{y}_s)$, такую что выполняется условие (критерий пригодности):

$$M_{predict}^{research} \leq M_{predict}^{real}, \quad (10)$$

где, $M_{predict}^{real}$ – выбранная метрика или группа метрик точности прогнозирования, рассмотренных в п. 1.1.6 существующей подсистемы прогнозирования рабочей нагрузки ВЦОД, а $M_{predict}^{research}$ – указанная метрика (группа метрик) предлагаемого в исследовании решения.

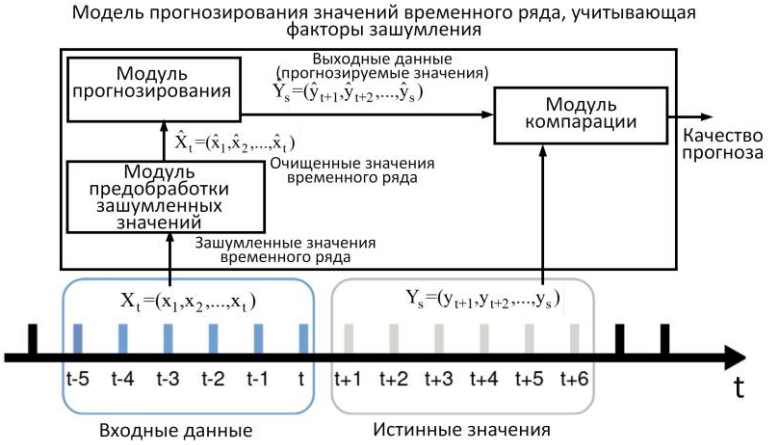


Рис. 1.24 Схема представления задачи исследования

Исходя из представленной постановки задачи исследования можно сформулировать **обобщенную цель исследования**:

Разработка модели представления временного ряда рабочей нагрузки и алгоритма, снижающего факторы ее зашумления, а также модели, алгоритма и специального программного обеспечения для получения прогнозных значений рабочей нагрузки, удовлетворяющих условию точности прогнозирования (выражение 10).

1.2. Моделирование временного ряда рабочей нагрузки виртуализированного центра обработки данных в условиях воздействия факторов его зашумления

Как было рассмотрено в п. 1.1.7, сохраняемые службой администрирования в базе ретроспективных данных рабочей нагрузки ВЦОД временные ряды показателей использования вычислительных и коммуникационных ресурсов в общем случае искажены относительно их истинных значений за счет влияния ряда факторов зашумления. Чтобы использовать их в качестве входных данных разрабатываемой системы прогнозирования рабочей нагрузки ВЦОД, требуется разработка способа, снижающего влияние факторов зашумления на значения временного ряда. Как было указано в п. 1.1.5, подходы к решению подобной задачи в различных предметных областях именуются «очисткой данных» (data clearing, noise removal).

Таким образом, важной исследовательской задачей является разработка моделирование временного ряда рабочей нагрузки ВЦОД, обеспечивающего снижение влияния факторов зашумления и, как следствие, формирование варианта временного ряда, который может быть использован в качестве входных данных модуля прогнозирования рабочей нагрузки. Функционально решение этой задачи выполняется модулем предварительной обработки временного ряда (рисунок 1.24).

1.2.1. Исследование подходов к снижению уровня влияния факторов зашумления временных рядов

Проблема зашумления значений временных рядов данных имеет место в различных предметных областях, связанная с мониторингом некоторых целевых показателей.

Исследование источников, посвященных этой проблеме [55-59], что, наряду с вопросами прогнозирования рабочей нагрузки ВЦОД, высокой степенью актуальности эта проблема обладает в следующих предметных областях:

– анализ показателей временных рядов электрокардиограмм и электроэнцефалограмм [55-56];

– анализ показателей временных рядов сейсмограмм в ходе исследования сейсмической активности [57];

– анализ показателей временных рядов, аппроксимирующих вид береговой линии в арктических и антарктических зонах [58];

– анализ показателей временных рядов, отражающих вибрацию деталей (например, подшипников) [59].

Следует отметить, что большое количество исследований, связанных с устранением факторов зашумления имеется в предметной области анализа речевых сигналов. Однако, в силу нелинейности и нестационарности распределения значений временного ряда, а также в силу различного временного масштаба формирования значений временного ряда в этой предметной области (доли секунды) и в предметной области анализа рабочей нагрузки ВЦОД (минуты-дни), рассмотрение методов, применяемых в очистке речевых сигналов, например, на основе преобразования Фурье, является нецелесообразным.

В общем виде проблема снижения влияния факторов зашумления значений временного ряда представлена в [55] и схематично выражается следующий схемой (рисунок 1.25).

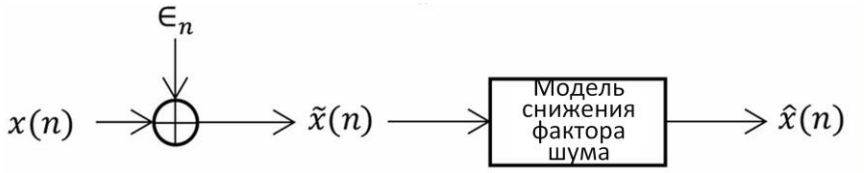


Рис. 1.25 Схема представления проблемы снижения влияния факторов зашумления значений временного ряда

Из рисунка видно, что зашумленные значения $\tilde{x}(n)$ формируются путем воздействия некоторого фактора (факторов) зашумления $\in_n$ на исходные значения $x(n)$. Функцией модели снижения фактора шума является преобразование зашумленных значений $\tilde{x}(n)$ в некоторый вид $\hat{x}(n)$, который максимально коррелирует со значениями $x(n)$.

Для формирования модели снижения факторов шума в анализируемых исследованиях рассматриваются следующие подходы (рисунок 1.26).

Рис. 1.26 Подходы к снижению влияния факторов зашумления значений временных рядов

В исследованиях, связанных с шумоподавлением в значениях временных рядов основой является подход, представляющий временной ряд – непрерывным сигналом. Исходя из такого представления рассматриваются подходы на основе традиционной [60] и адаптивной фильтрации [61]. К традиционным подходам стоит отнести шумоподавление на основе нелокальных средних (Non-local means – NLM), а также решение задачи наименьших квадратов с регуляризацией полной вариации [62]. Подходы на основе адаптивной фильтрации базируются на моделях статистической фильтрации, таких как расширенный фильтр Калмана (Extended Kalman Filter – EKF), расширенный сглаживатель Калмана (Extended Kalman Smoother - EKS) и «незагрязненный» фильтр Калмана (Unscented Kalman Filter – UKF) [63]. С некоторыми ограничениями рассматриваются подходы на основе вейвлет преобразования, применяемого для разложения сигнала, определения типа пороговой обработки и последующей реконструкции сигнала. Подходы на основе моделей глубокого обучения ориентированы на особенности использования автоэнкодеров (deep-learning based autoencoder models – DAE), которые направлены

на восстановление сигнала из его искаженной версии путем оптимизации целевой функции [64].

Также широкое применение в решении задачи снижения зашумленности сигнальных конструкций нашли подходы на основе преобразования Гильберта-Хуанга [65]. К особенностям использование этих подходов следует отнести:

– их доказанную эффективность в снижении влияния факторов зашумления для сигналов нелинейной и нестационарной природы;

– возможность получения высокого качестве снижения уровня шума для временных рядов крупной масштабности;

– относительно низкая вычислительная и временная сложность реализующих их алгоритмов.

В рамках исследования было приято решение об использовании в модуле предобработки зашумленных значений временного ряда показателей рабочей нагрузки ВЦОД подходов на основе преобразования Гильберта-Хуанга.

1.2.2. Исследование методов модовой декомпозиции значений временного ряда на основе преобразования Гильберта-Хуанга

Впервые Хуанг с группой других исследователей представил подход к разложению сигналов, именуемый декомпозиция на эмпирические моды (ДЭМ) (Empirical Mode Decomposition – EMD), в [66]. Поскольку в основе разложения лежит характеристическая шкала времени данных, оно наиболее эффективно для анализа нелинейных и нестационарных процессов. Эта особенность позволила применить разложение Гильберта-Хуанга для анализа временных рядов.

В основе метода ДЭМ лежит разложение сигнала на множество модовых вариаций (мод), именуемых функции внутреннего режима или колебательные модовые функции (КМФ) (Intrinsic Mode Function – IMF) вместе с трендом. Функции КМФ, фактически, представляют различные частотные компоненты временного ряда, представленного сигнальной конструкцией и образуют его полную и практически ортогональную основу.

В [66] функция КМФ определяется, как удовлетворяющая условиям:

– число экстремумов в них, а также точек пересечения нуля не отличаются более чем на одно значение;

– усредненное значение по верхней и нижней огибающим равно нулю.

Таким образом КМФ являются функциями более общего вида в отличие от метода Фурье, в котором разложение сигнала выполняется в базисе тригонометрических функций, обладающих заданными частотой и фазой. Каждый элемент множества КМФ фактически является амплитудно-частотной модуляцией в заданной узкой полосе частот, что позволяет связать его с определенным процессом. Далее, для получения значений мгновенной частоты к множеству КМФ применяется гильбертов спектральный анализ (HSA) [67]. Поскольку разложение сигнала происходит во временной, а не частотной области, а длина каждого элемента множества КМФ совпадает с длиной исходного сигнала, преобразование Гильберта-Хуанга сохраняет характеристики изменяющейся частоты сигнала, что делает его близким к Фурье преобразованию, применяемому для гармонических колебаний.

Пример спектрального анализа Гильберта применительно к смешанному сигналу, содержащему синусоидальные волны с различными значениями амплитуды и частоты, представлен на рисунке 1.27.

Из рисунка видно, что спектр Гильберта представляет собой разреженный график, отражающий мгновенный частотный спектр каждого компонента, полученный путем разложения исходного смешанного сигнала. На графике видно получение трех КМФ функций, а также красным пунктиром выделено изменение частоты сигнала на 1 секунде.

Особенностью КМФ является их получение в ходе выполнения процесса (эмпирически), что в отличие от классического анализа гармонических сигналов позволяет выявлять области локальности. Так, первый элемент множества КМФ – $КМФ_1$ (базовая КМФ) несет высокочастотные компоненты, характерные, например, для шума, то ее можно отклонить, тем самым снижая влияние факторов зашумления.

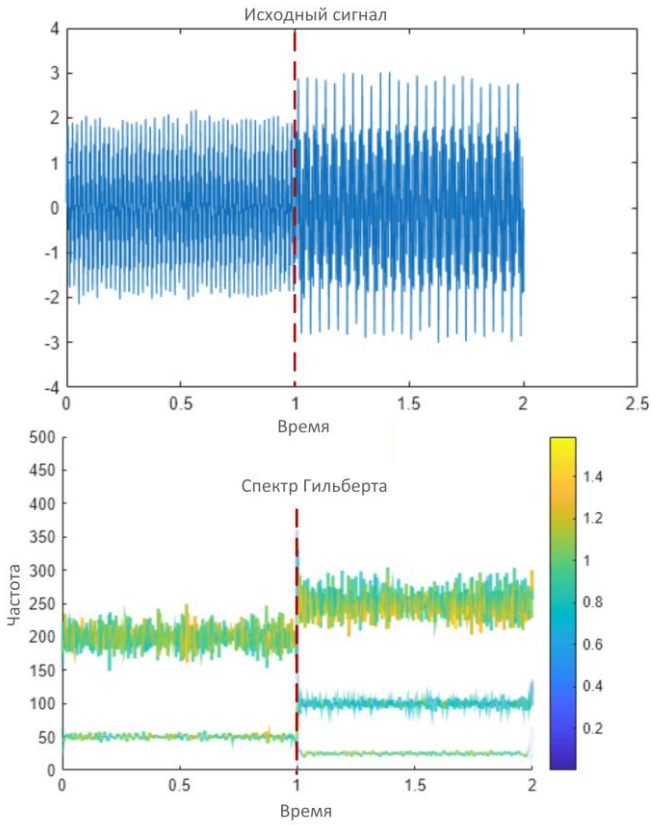


Рис. 1.27 Пример спектрального анализа Гильберта для смешанного сигнала

На рисунке 1.28 представлено разложение методом ДЭМ исходного сигнала (рисунок 1.27) на шесть КМФ функций и остаток R.

Базовый подход, реализованный в методе ДЭМ, не является оптимальным, поскольку, в зависимости от степени нестационарности сигнала, допускает смешивание мод (КМФ функций), что существенно усложняет их интерпретацию. Кроме того, метод показывает низкую точность разложения при резких изменениях масштаба временного ряда.

Соответственно, были проведены исследования, по оптимизации базового метода ДЭМ. Так в [68] был представлен метод декомпозиции на вариационные моды (ДВМ) (Variation Mode Decomposition – VMD), а в [69] – метод ансамблевой эмпирической модовой декомпозиции (АДЭМ, англ. EEDM).

В их основе лежит добавление гауссовоского шума к входному сигналу и выполнение нескольких итераций ДЭМ для удаления наложения спектров в элементах множества КМФ. Общим недостатком ВДМ и АДЭМ является формирование дополнительных мод – элементов множества КМФ, что несколько

усложняет последующий анализ. Решение этой проблемы было предложено в [70] и получило название полная (комплементарная) ансамблевая модовая декомпозиция с адаптивным шумом (КДЭМАШ, англ. CEEMDAN). Особенностью метода КДЭМАШ является адаптивное добавление гауссовского шума на каждом уровне декомпозиции и его учет в каждой получаемой КМФ функции.

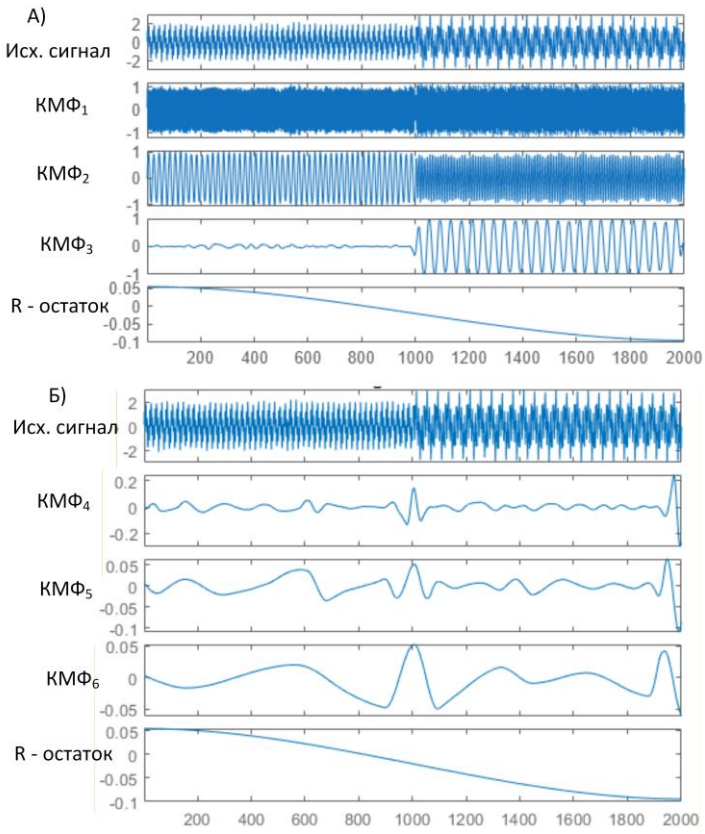


Рис. 1.28 Пример разложения смешанного исходного сигнала на шесть КМФ функций и остаток. А) КМФ$_1$-КМФ$_3$ – высокочастотные функции, Б) КМФ$_4$-КМФ$_6$ – низкочастотные функции

В настоящее время вариации метода КДЭМАШ нашли свое применение в сейсмографии, оценивании скорости перемещения воздушных масс [71], в электрокардиографии [72].

Рассмотренные выше особенности методов на основе преобразования Гильберта-Хуанга, в частности, их доказанная эффективность применительно к анализу нестационарных вариантов временных рядов разной масштабности позволяет принять решение об их выборе в качестве методологической базы модуля

предварительной обработки зашумленных значений временных рядов показателей рабочей нагрузки ВЦОД.

В общем виде схем применения указанного подхода представлена на рисунке 1.29.

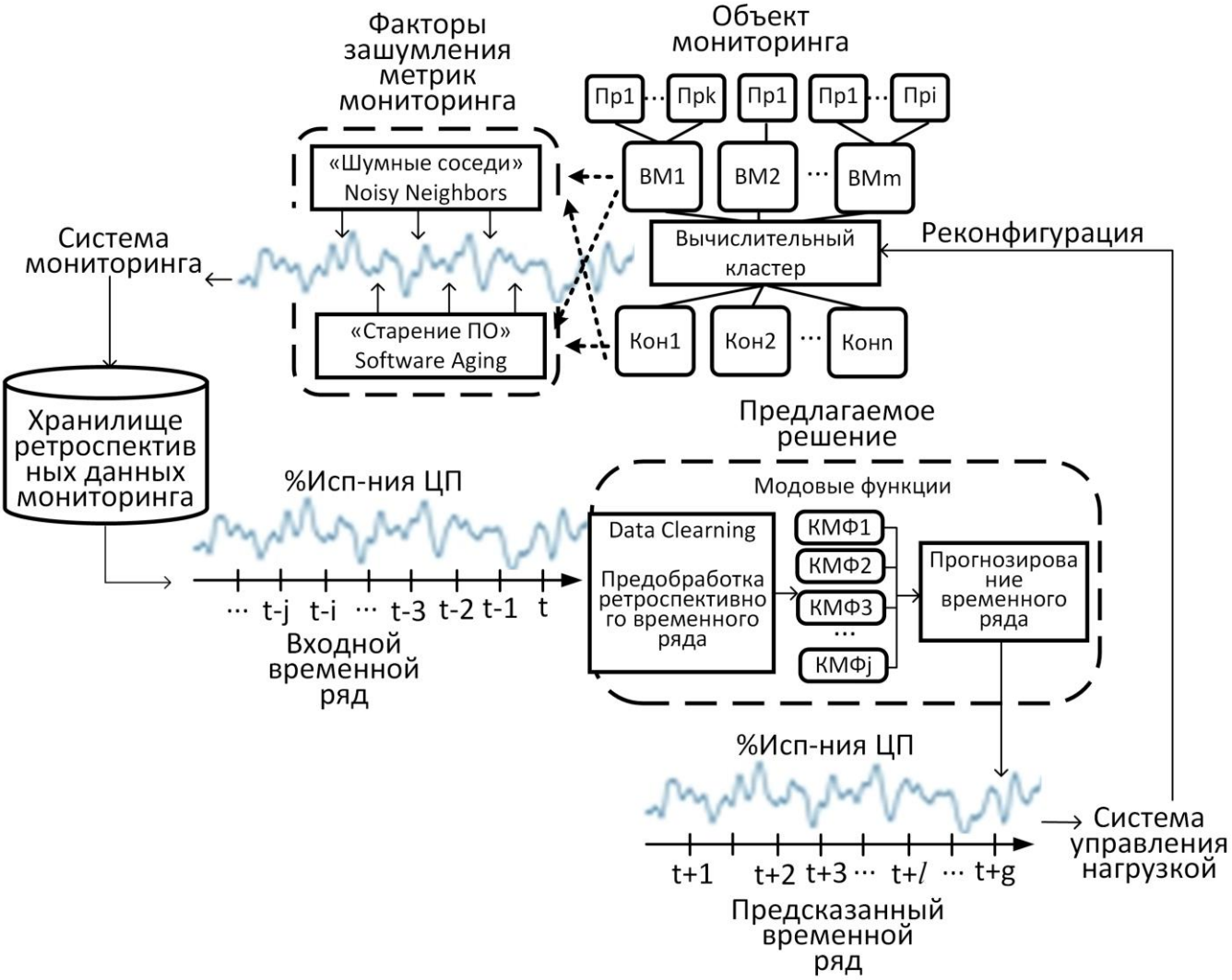


Рис. 1.29 Представление подхода к прогнозированию рабочей нагрузки ВЦОД на основе предварительной обработки зашумленного временного ряда ретроспективных данных нагрузки за счет его разложения на множество колебательных модовых функций

1.2.3. Моделирование модовой декомпозиции временного ряда показателей ретроспективных данных рабочей нагрузки ВЦОД методами КДЭМАШ и ДВМ

Рассмотрим применение метода КДЭМАШ к разложению временного ряда данных ретроспективных значений метрик использования процессорных ядер, определяющих рабочую нагрузку вычислительного процесса $w(t)$.

Выполним объединение анализируемого временного ряда $w(t)$ с гауссовым белым шумом $n_i(t)$ для создания N_e зашумленных реализаций сигнала $D_i(t)$:

$$D_i(t)=w(t)+en_i(t), \qquad (11)$$

где $i=1,2,...,N_e$, а значение e определяет амплитуду $n_i(t)$ с нормальным гауссовским распределением $N(\mu,\sigma^2)$, где коэффициент сдвига $\mu=0$.

Далее, согласно этапам классического алгоритма ДЭМ, выделим из сигнала $D_i(t)$ первую КМФ:

$$\overline{КМФ_1}(t)=\frac{1}{N_e}\sum_{i=1}^{N_e} КМФ_1^i(t) \qquad (12)$$

Поскольку гауссовский шум $n_i(t)$ имеет распределение $N(\mu,\sigma^2)$ с нулевым коэффициентом сдвига μ, то можно предположить, что благодаря усреднению по нескольким зашумленным реализациям сигнала $D_i(t)$ шумовая компонента будет существенно снижена, а сами значения временного ряда $w(t)$ не изменятся.

Вычислим значение первого остатка $r_1(t)$:

$$r_1(t)=w(t)-\overline{КМФ_1}(t) \qquad (13)$$

Повторим рассмотренные выше операции (выражения 12 и 13) для вычисления значение последующей $(k+1)$ функции КМФ:

$$\overline{КМФ_{k+1}}(t)=\frac{1}{N_e}\sum_{i=1}^{N_e} E_1(r_k(t)+\varepsilon_k E_k(n_k(t))), \qquad (14)$$

где E_k – процедура получения k-й КМФ, включающая:

– определение ее локальных экстремумов;

– построение огибающих;

– получение средней огибающей;

– вычитание ее из исходного сигнала.

При этом значение остатка $r_k(t)$ обновляется на каждой итерации:

$$r_k(t)=r_{k-1}(t)-\overline{КМФ}_k(t) \qquad (14)$$

Очевидно, что конечное значение остатка будет получено при условии останова – невозможности выполнения следующей итерации КДЭМАШ (выражение 16):

$$R(t)=w(t)-\sum_{k=1}^{K}\overline{КМФ}_k(t) \qquad (16)$$

После останова итераций КДЭМАШ исходный временной ряд $w(t)$ будет представлен суммой множества КМФ и конечного остатка $R(t)$:

$$w(t)=\sum_{k=1}^{K}\overline{КМФ}_k(t)+R(t) \qquad (17)$$

При этом доказано [66, 70], что компоненты разложения (КМФ) в достаточной мере сохраняют статистические свойства данных временного ряда $w(t)$ и не теряют информацию о долгосрочных временных тенденциях.

Как было указано выше метод ДЭМ (в том числе и его вариация КДЭМАШ) обладает рядом недостатков, к которым, в первую очередь относятся функционирование в амплитудно-частотной области сигнала (временного ряда), что снижает эффективность разделения частотных компонентов.

Однако, наибольшую проблему создает итеративность алгоритма получения множества функций КМФ, что может приводить:

– к существенным временным затратам при высокой нестационарности данных исходного временного ряда $w(t)$ в силу плохой сходимости к получению остатка $R(t)$;

– формированию избыточного количества элементов множества КМФ, имеющих высокую степень корреляции и затрудняющие последующий процесс прогнозирования.

Одним из вариантов решения указанных проблем является применение метода декомпозиции на вариационные моды (ДВМ).

Исходя из [68], метод ДВМ устраняет основной недостаток методов на основе ДЭМ – эмпирический, интерационный способ получения элементов $\overline{КМФ_k}(t)$.

Фактически, метод ДВМ выполняет декомпозицию временного ряда $w(t)$ на множество k функций КМФ, каждый элемент которого $КМФ_k$, имеет ограниченную полосу пропускания:

$$КМФ_k = A_K(t)\cos(\varphi(t)) \qquad (18)$$

где, $A_K(t)$ является значением мгновенной амплитуды функции $КМФ_k(t)$ – далее обозначаемой, как $f_k(t)$. Значение ширины полосы пропускания $f_k(t)$ может быть оценено с помощью сглаживания методом численного интегрирования Гаусса.

В основе метода ДВМ лежит вариационная модель вида:

$$\min_{(f_k),(c_k)} \begin{cases} \sum_{k=1}^{K} \left\| \partial_t \left[\sigma(t) + \frac{i}{\pi t} f_k(t) \right] e^{-jc_k t} \right\|_2^2, \\ \text{при условии } \sum_{k=1}^{K} f_k(t) = w(t) \end{cases} \qquad (19)$$

где, ∂_t – частная производная t, а элементы множества $c_K = c_1, c_2, ..., c_k$ являются значениями центральных частот каждого элемента множества функций КМФ.

Для получения оптимального решения этой вариационной задачи применяется метод множителей Лагранжа с квадратичным перемножением штрафной функции:

$$L(f_k,c_k,\lambda)=\alpha\sum_{k=1}^{K}\left\|\partial_t\left[\sigma(t)+\frac{i}{\pi t}f_k(t)\right]e^{-jc_kt}\right\|_2^2 + \left\|w(t)-\sum_{k=1}^{K}f_k(t)\right\|_2^2+\left\langle\lambda(t),w(t)-\sum_{k=1}^{K}f_k(t)\right\rangle, \quad (20)$$

где λ – множитель Лагранжа, а α – коэффициент штрафной функции.

В дальнейшем получается экстремальное решение для частот каждого элемента множества КМФ и их центральных частот c_k.

$$f_k^{n+1}(c)=\frac{\hat{w}(c)-\sum_{k=1,k\neq K}^{k}f_k(c)+\frac{\hat{\lambda}(c)}{2}}{1+2\alpha(c-c_k)^2} \quad (21)$$

$$c_k^{n+1}=\frac{\int_0^\infty c\left|f_k(c)^2\right|dc}{\left|f_k(c)^2\right|dc} \quad (22)$$

Для получения оптимального решения выражения 21 для всех элементов множества КМФ используется известный алгоритм ADMM (Alternating Direction Method of Multipliers) [74], который декомпозирует сложную вариационную задачу на частные подзадачи. Его этапами являются:

1. Установка нулевых значений параметров $f_k(t)$, c_k, λ_k^1 и n.

2. Обновление значений $f_k^{n+1}(t)$ и c_k^{n+1} (выражения 12 и 13).

3. Обновление значения оператора множителя Лагранжа λ^{n+1}:

$$\hat{\lambda}^{n+1}(c)=\hat{\lambda}^{n+1}(c)+\tau\left(\hat{f}(c)-\sum_{k=1}^{K}\hat{f}_k^{n+1}(c)\right) \quad (23)$$

4. Повтор этапов 2-3 производится, пока не будет получено значение $f_k^{n+1}(c)$ при выполнении следующего условия $\sum_{k=1}^{k}\frac{\left\|f_k^{n+1}-f_k^{n}\right\|_2^2}{\left\|f_k^{n}\right\|_2^2}<\varepsilon$.

Метод ДВМ, работая в частотной области сигнала, является вычислительной процедурой – решением варианта вариационной задачи. Таким образом, применение метода ДВМ сохраняет основные частотные характеристики временного ряда, минимизируя шумовую компоненту, и при этом полученное множество функций КМФ является полностью реконструктивным (обеспечивающим восстановление значений исходного временного ряда). При этом, однако, этот метод обладает высокой вычислительной сложностью и требует предварительного определения мощности множества функций КМФ.

1.2.4. Комплексный подход к моделированию модовой декомпозиции временного ряда показателей ретроспективных данных рабочей нагрузки ЦОД методами КДЭМАШ и ДВМ

Как было рассмотрено в п. 1.2.2, моделирование временного ряда значений показателей рабочей нагрузки с использованием методов модовой декомпозиции приводит к получению множества функция КМФ, в рамках которого высокочастотные элементы, чаще всего связанные с влиянием факторов зашумления, консолидируются в базовой функции КМФ, в то время, как остальные элементы множества потенциально содержат информацию о шаблонах рабочей нагрузки. Следовательно, используя это подмножество функций КМФ в качестве входных данных модуля прогнозирования рабочей нагрузки (рисунок 1.27) можно получить более точные значения прогноза.

При этом, как методы декомпозиции на эмпирические моды (в частности, метод КДЭМАШ), так и методы декомпозиции на вариационные моды (ДВМ) обладают определенными недостатками, усложняющими процесс предварительной обработки исходного временного ряда.

В связи с этим в исследовании предлагается ряд подходов, снижающих недостатки используемых методов модовой декомпозиции.

Одним из способов решения проблемы избыточного количества элементов множества КМФ функций, а также проблемы их смешивания, присущих методам ДЭМ (в том числе, в определенной мере, и КДЭМАШ) является их редукция на основе некоторого критерий отсеивания. Наиболее известным подходом является

комбинирование оценивания сложности и нерегулярности полученных КМФ функций, а также их кластеризация по этим показателям.

Для оценивания сложности и нерегулярности значений полученных функций $\overline{КМФ_k}(t)$ рассчитаем выборочную энтропию каждого значения каждой функции.

Выборочная энтропия (SampEn) – это метрика, используемая для количественной оценки нерегулярности значений временного ряда [73].

Анализируя выборочную энтропию значения функции $\overline{КМФ_k}(t)$, можно идентифицировать компоненты значений временного ряда $w(t)$ со схожими уровнями сложности. Более высокие значения выборочной энтропии указывают на большую сложность в исходной последовательности данных $w(t)$, что означает усложнение процесса прогнозирования будущих значений $\hat{y}_t$.

Анализ выборочной энтропии позволяет различать элементы $\overline{КМФ_k}(t)$, содержащие шаблоны рабочей нагрузки, что способствует более точному и оптимальному решению задачи ее прогнозирования за счет объединения элементов $\overline{КМФ_k}(t)$ с близкими значениями.

Рассмотрим этапы процесса вычисления выборочной энтропии:

1. Выборка одномерной последовательности с равномерными временными интервалами, что приводит к временному ряду с N точками данных, представленному, как $w(1),w(2),...,w(i)...,w(N)$.

2. Преобразование полученного временного ряда в m-мерный формат вида $W(1),W(2),...,W(i),...,W(N-m+1)$. Обозначим i-й элемент ряда, представленный m-мерным векторным пространством, как:

$$W_m(i)=\left[w(i),w(i+1),w(i+2),...,w(i+m-1)\right] \qquad (24)$$

3. Оценивание сходства между i-m и j-м векторами путем вычисления расстояния $D\left[W_m(i),W_m(j)\right]$ – максимальной абсолютной разницы между их элементами:

$$D\left[W_m(i),W_m(j)\right]=\max\left(\left|w(i+k)-w(j+k)\right|\right), \quad (25)$$

где $1\le i\le N-m+1$, $i\ne j$, а k находится в диапазоне от 0 до m-1.

4. Вычисление среднего значения векторов, удовлетворяющих пороговому условию $N-m$ - $C^m(l)$, определяющего неравенство $l\ge D\left[W_m(i),W_m(j)\right]$, где l – условный порог путем подсчета числа:

$$C^m(l)=\frac{1}{N-m+1}\sum_{i=1}^{N-m+1}C_i^m(l) \quad (26)$$

5. Итерации расчета выполняются для получения среднего значения $C^{m+1}(l)$.

6. После получения значения $C^{m+1}(l)$ выборочная энтропия для m измерений с условным порогом l может быть представлена как:

$$\mathrm{SampEn}(m,l)=\lim_{N\to\infty}\left[-\ln\left(\frac{C^{m+1}(l)}{C^m(l)}\right)\right] \quad (27)$$

В силу конечности значения N, это выражение можно аппроксимировать до вида:

$$\mathrm{SampEn}(m,l,N)=-\ln\left(\frac{C^{m+1}(l)}{C^m(l)}\right) \quad (28)$$

Анализ выборочной энтропии формирует основу для сравнения сходства КМФ функций и позволяет в дальнейшем сгруппировать их в кластеры по уровню частоты в категории высокой, средней и низкой частоты.

В качестве способа такой кластеризации предлагается использование метода К-средних (K-means), как наиболее эффективного для данных высокой размерности, предоставляя значимые сведения из согласованной структуры данных [75]. Таким образом, метод К-средних позволяет сегментировать КМФ функции в соответствии с одинаковыми частотными характеристиками, тем самым уменьшая вычислительную нагрузку и повышая точность последующего прогнозирования.

Процесс кластеризации методом К-средних представляется, как функция минимизации:

$$\min_{M} \sum_{i=1}^{k} \sum_{\overline{КМФ(t)} \in M_i} \left\| \overline{КМФ}(t) - \mu_i \right\|^2, \tag{29}$$

где, M_i – элемент множества кластеров M, k – определяет количество кластеров, а μ_i – центроид i-го кластера.

Результат кластеризации подмножества КМФ функций можно для удобства выбора функций различной частотности можно представить в виде графика осыпи [76].

Другим способом решения проблемы избыточного количества элементов множества КМФ функций, а также проблемы их смешивания является последовательное использование методов декомпозиции ДЭМ и ДВМ. В силу особенностей реализации (решение вариационной задачи) и ориентации на частотную составляющую сигнала, метод ДВМ обеспечивает получение более оптимального с точки зрения частотного разделения множества КМФ функций. Таким образом, после реализации метода ДЭМ, в результате которого сформировалось неоптимальное (например, в силу высокой корреляции) множество КМФ функций, базовую функцию $КМФ_1$ можно использовать в качестве входного сигнала для метода ДВМ.

В обобщенном виде такой комплексный подход к организации процесса предобработки временных рядов ретроспективных данных рабочей нагрузки представлен на рисунке 1.30.

В силу эмпирического характера метода КДЭМАШ, оценивание качества разработанной модели модовой декомпозиции временного ряда ретроспективных данных рабочей нагрузки ВЦОД будет выполнено на этапе разработки алгоритмических решений.

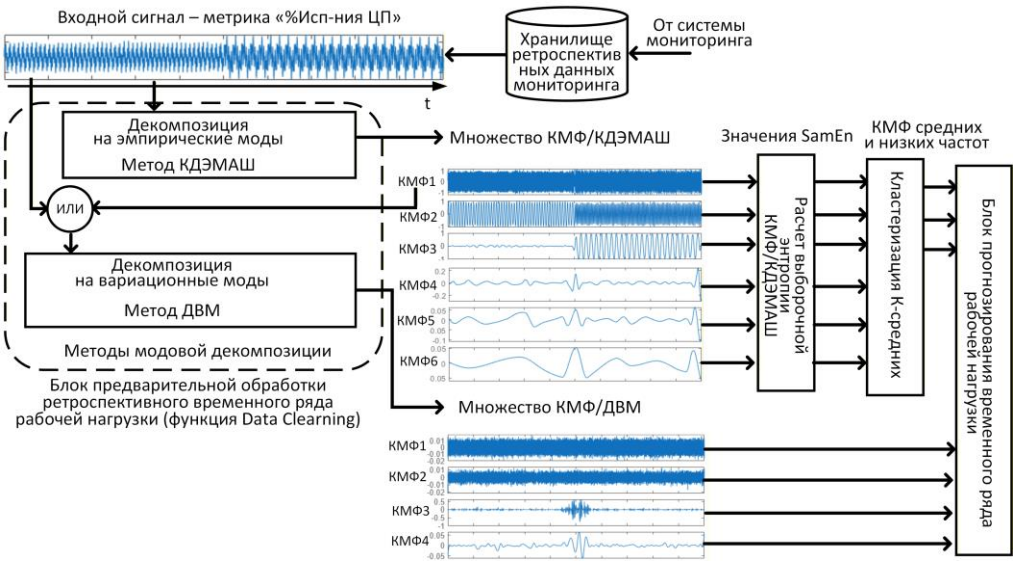


Рис. 1.30 Схема комплексного подхода к моделированию модовой декомпозиции временного ряда ретроспективных данных рабочей нагрузки ВЦОД

1.3. Выводы по главе

Таким образом, в данной главе на основании анализа исследований, посвященных организации и функционированию службы администрирования ВЦОД:

– рассмотрена актуальность задач реактивного и проактивного управления рабочей нагрузкой ВЦОД;

– определена структурная схема подсистемы мониторинга вычислительных ресурсов, отображающих рабочую нагрузку, методы и средства их анализа;

– рассмотрен подход к сохранению ретроспективных данных рабочей нагрузки в виде временных рядов значений заданных показателей утилизации вычислительных ресурсов;

– обобщенно сформулирована задача прогнозирования рабочей нагрузки ВЦОД на основе анализа временных рядов ее показателей; выявлены проблемы их искажения за счет внешних и внутренних факторов зашумления;

– рассмотрены особенности проблем «шумные соседи» (Noisy Neighbours) и «старение» программного обеспечения (Software Aging), как факторов зашумления, наиболее влияющих на значения временного ряда рабочей нагрузки;

– сформулировано противоречие между необходимостью получения приемлемых (заданных) прогнозных значений рабочей нагрузки ВЦОД и отсутствием в существующих службах их администрирования методов и алгоритмов, учитывающих факторы зашумления значений временных рядов заданных показателей производительности вычислительных ресурсов;

– сделана постановка научной задачи по разработке моделей, алгоритмов и архитектуры системы прогнозирования рабочей нагрузки ВЦОД в условиях зашумления значений временных рядов ее ретроспективных данных.

– проведено исследование использования модели модовой декомпозиции для анализа временных рядов сигналов различной природы и ее применение для снижения влияния факторов зашумления за счет дифференцированного анализа элементов множества КМФ;

– исследованы теоретические основы модовой декомпозиции сигналов, в частности, преобразование Гильберта-Хуанга, а также методы ее реализации. Выделяются: семейство методов декомпозиции на эмпирические моды (ДЭМ) и методы декомпозиции на вариационные моды (ДВМ). Рассматриваются их особенности, достоинства и недостатки

– разработана комплексная модель модовой декомпозиции временного ряда, которая обеспечивает снижение влияния факторов зашумления на значения временного ряда за счет двухэтапной процедуры его разложения на множество колебательных модовых функций (КМФ) – амплитудно-частотных модуляций в заданных узких полосах частот, что позволяет связать их с определенным процессом и выявлять области локальности.

Глава 2. Разработка комплексного алгоритма предварительной обработки временного ряда рабочей нагрузки

Как было рассмотрено в п. 1.2.4 предварительная обработка значений показателей временного ряда рабочей нагрузки ВЦОД предназначена для снижения влияния факторов зашумления, связанных с особенностями эксплуатации систем виртуализации и контейнеризации ВЦОД (п. 1.1.7).

В силу того, что влияние факторов зашумления приводит к формированию погрешностей значений временного ряда в высокочастотной области, в п. 1.2 было выдвинуто предположение о возможности использования методов модовой декомпозиции сигнала, основанных на преобразовании Гильберта-Хуанга, в качестве способа снижения влияния факторов зашумления. Поскольку в процессе модовой декомпозиции формируется множество КМФ функций, часть из них описывает высокочастотные составляющие сигнала (временного ряда), которые с большой вероятностью относятся к шумовым составляющим, есть возможность выполнить их селекцию с целью редукции элементов множества КМФ функций, поступающих на вход модуля прогнозирования рабочей нагрузки (рисунок 1.24).

В результате анализа особенностей и недостатков методов декомпозиции на эмпирические моды (ДЭМ) и на вариационные моды (ДВМ) было предложено комплексное решение, основанное на последовательном использовании методов КДЭМАШ (вариант ДЭМ) и ДВМ. При этом формируется два подмножества КМФ функций, условно обозначаемых КМФ/КДЭМАШ и КМФ/ДВМ, которые являются входными данными (вариантами временного ряда) для модуля прогнозирования рабочей нагрузки.

2.1. Разработка схемы комплексного алгоритма декомпозиции временного ряда рабочей нагрузки

Несмотря на то, что методы КДЭМАШ и ДВМ в своей основе имеют эмпирический характер получения КМФ функций, в исследованиях для различных предметных областей, посвященных их применению для снижения факторов

зашумления, рассматривается их аналитическое представление. При этом, для программной реализации модуля предварительной обработки зашумленных значений временного ряда (рисунок 1.24) требуется разработка схем алгоритмов указанных методов, а также процедур, обеспечивающих оптимизацию полученных в рамках их использования множеств КМФ функций.

В связи с этим в главе рассматриваются подходы к разработке схем алгоритмов функциональных блоков модуля предварительной обработки временного ряда, представленных на рисунке 1.30.

В общем виде схема алгоритма реализации предлагаемой комплексной модели модовой декомпозиции значений показателей временного ряда рабочей нагрузки ВЦОД может быть представлена следующими предопределенными процедурами (рисунок 2.1).

Из рисунка видно, что базовой предопределенной процедурой предлагаемого комплексного алгоритма предварительной обработки значений параметров временного ряда рабочей нагрузки ВЦОД является алгоритм КДЭМАШ, входными значениями которого являются:

– исходный временной ряд X(n) одного из показателей рабочей нагрузки ВЦОД, полученный из баз ретроспективных данных рабочей нагрузки, где n – значений временного ряда;

– K – пороговое значение числа получаемых в ходе декомпозиции КМФ функций. Выбор значения K выполняется эмпирически и определяет допустимый максимальный предел числа КМФ функций, превышение которого является признаком неэффективной процедуры декомпозиции, допускающей корреляцию данных между соседними КМФ функциями.

Алгоритм КДЭМАШ реализует вариант декомпозиции $X(n) \to \{\text{КМФ}_1, \text{КМФ}_2, ..., \text{КМФ}_k\}$, где $\{\text{КМФ}_1, \text{КМФ}_2, ..., \text{КМФ}_k\}$ - множество КМФ/КДЭМАШ функций мощностью k.

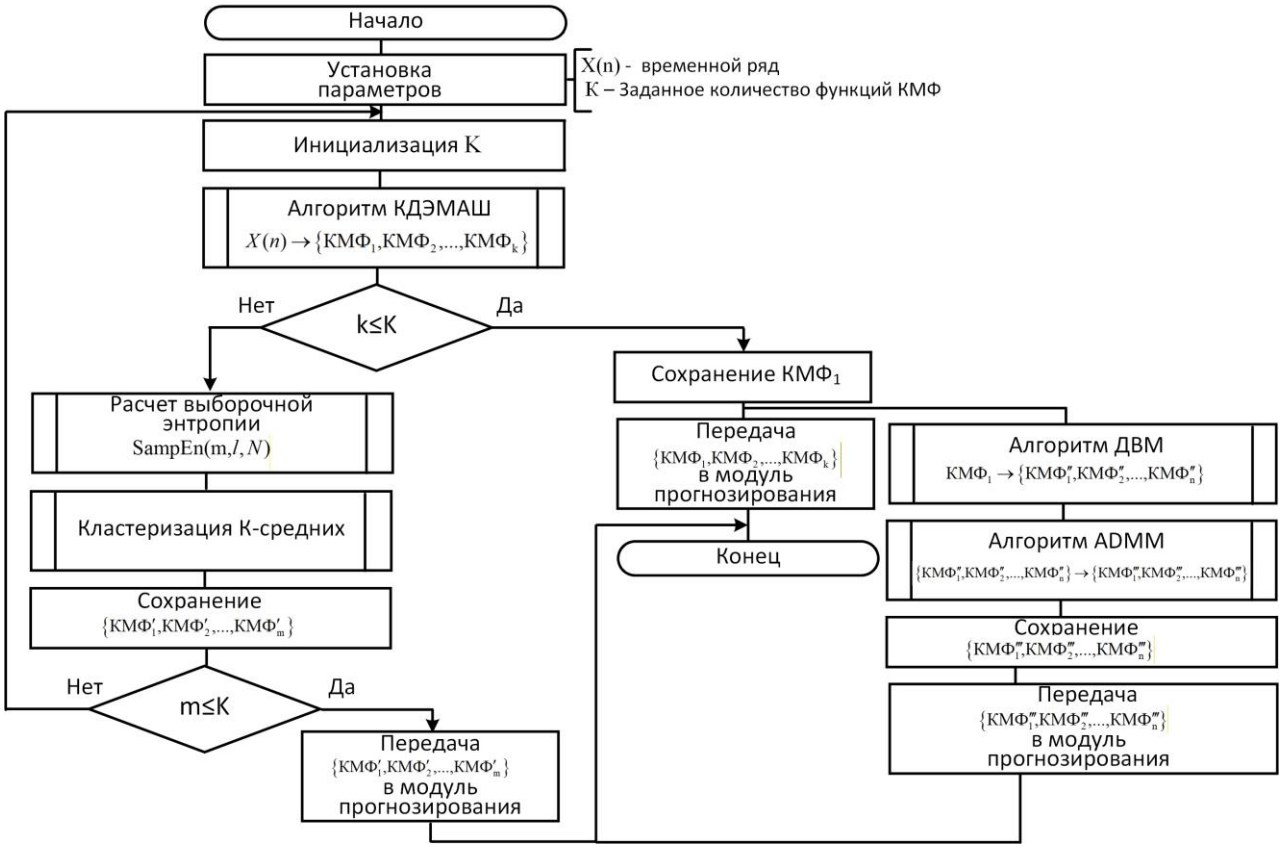


Рис. 2.1 Схема комплексного алгоритма предварительной обработки значений параметров временного ряда рабочей нагрузки ВЦОД

Дальнейшие действия комплексного алгоритма зависят о проверки условия превышения порогового значения $k \le K$. В случае выполнения этого условия, базовая КМФ функция ($КМФ_1$) сохраняется для использования в качестве входных данных предопределенной процедура алгоритма ДВМ, а все множество $\{КМФ_1, КМФ_2, ..., КМФ_k\}$ передается на вход модуля прогнозирования рабочей нагрузки.

В противном случае выполняется оптимизация числа элементов множества $\{КМФ_1, КМФ_2, ..., КМФ_k\}$ путем последовательного выполнения процедур:

– расчета выборочной энтропии полученного множества, упорядочивающего его элементы по уровню значимости;

– кластеризации элементов упорядоченного множества методом К-средних.

Результатом выполненной оптимизации является подмножество $\{КМФ'_1, КМФ'_2, ..., КМФ'_m\}$ мощностью m.

В случае положительного исхода проверки условия $m \le K$ редуцированной множество $\{КМФ'_1, КМФ'_2, ..., КМФ'_m\}$ передается на вход модуля прогнозирования рабочей нагрузки.

Предопределенная процедура, реализующая алгоритм ДВМ выполняется параллельно процедурам оптимизации числа элементов множества $\{КМФ_1, КМФ_2, ..., КМФ_k\}$. Как было указано выше, в качестве входных данных она использует базовую функцию $КМФ_1$, полученную в результате выполнения процедуры алгоритма КДЭМАШ. По определению эта функция практически полностью повторяет исходный временной ряд X(n) за исключением ряда высокочастотных компонентов. Таким образом, процедура алгоритма ДВМ выполняет преобразование $КМФ_1 \to \{КМФ''_1, КМФ''_2, ..., КМФ''_n\}$, где $\{КМФ''_1, КМФ''_2, ..., КМФ''_n\}$ множество функций КМФ/ДВМ мощностью n. Поскольку метод ДВМ реализует численное решение вариационной задачи, проверку условия на превышение порога числа КМФ функций $n \le K$ выполнять не требуется. В силу вычислительной сложности вариационной задачи алгоритма ДВМ, для оптимизации его процесса выполняется процедура алгоритма чередующихся направлений множителей (Alternating Direction Method of Multipliers – ADMM) [77]. Целью использования алгоритма ADMM является разделение сложной оптимизационной задачи, к которой относится вариационная задача алгоритма ДВМ на несколько относительно простых подзадач, и их итерационное выполнение таким образом, что выходные данные подзадач предшествующих итераций используются как входные для подзадач последующих итераций. Сходимость общего решения при объединении решений частных подзадач обеспечивается использованием в алгоритме ADMM множителей Лагранжа.

Использование алгоритма ADMM при получении множества функции КМФ/ДВМ с одной стороны превращает этот процесс в итерационный, что сказывается на времени его выполнения, однако, благодаря декомпозиции вариационной задачи на подзадачи, в целом, вычислительный процесс становится менее трудоемким.

Алгоритм ADMM реализует преобразование $\{КМФ''_1,КМФ''_2,...,КМФ''_n\} \rightarrow \{КМФ'''_1,КМФ'''_2,...,КМФ'''_n\}$, где $\{КМФ'''_1,КМФ'''_2,...,КМФ'''_n\}$ множества функций КМФ/ДВМ мощностью m.

Таким образом, на вход модуля прогнозирования рабочей нагрузки поступает два подмножества КМФ функций, полученных в процессе выполнения комплексного алгоритма предварительной обработки сигнала (рисунок 2.2). Из них одно подмножество выбирается динамически, по результатам выполнения алгоритма КДЭМАШ (оптимизированное или не оптимизированное подмножество).

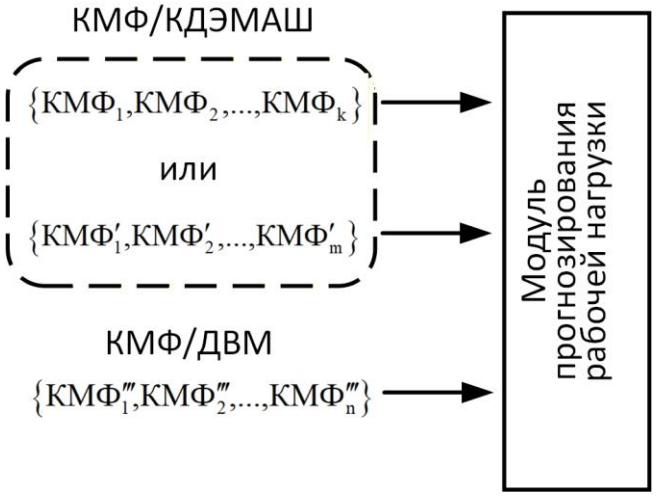


Рис. 2.2 Варианты подмножеств КМФ функций, поступающих на вход модуля прогнозирования рабочей нагрузки ВЦОД

2.2. Разработка алгоритма декомпозиции на эмпирические моды временного ряда рабочей нагрузки

На рисунке 2.3 представлена разработанная схема предопределенной процедуры алгоритма КДМЭАШ, являющейся частью комплексного алгоритма предварительной обработки данных временного ряда рабочей нагрузки.

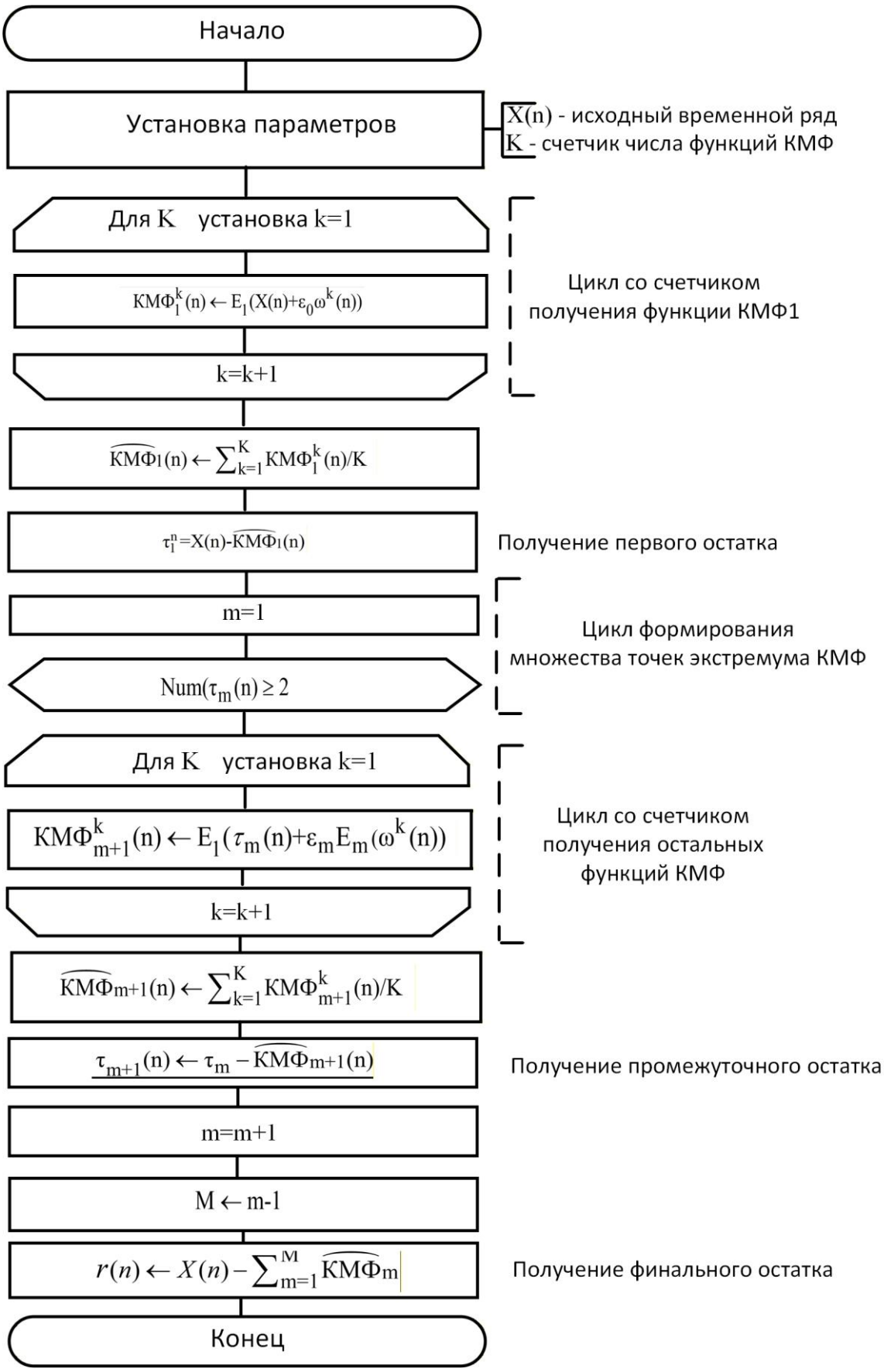


Рис. 2.3 Схема алгоритма предопределенной процедуры КДЭМАШ

Из рисунка видно, что для входного временного ряда $X(n)$ получение функции $КМФ_1(n)$ достигается путем реализации:

– k-проходного цикла добавления гауссового шума ω_k с распределением $N(0,1)$ с стандартным отклонением ε_k, получением среднего значения по каждой из K реализаций;

– получения первого остатка $\tau_1^n = X(n) - КМФ_1(n)$ и определения параметров цикла получения оставшихся реализаций КМФ функций. Для этого в алгоритме определяются функции: E_m - режима получения остатков и $Num(\tau_m(n))$ – формирования множества экстремальных точек.

Далее, представленные выше k-проходный цикл добавления гауссового шума и получение остатка циклично выполняется для оставшегося подмножества КМФ мощностью m. Критерием останова этого цикла, а, следовательно, и останова алгоритма, является получение остатка r_n, удовлетворяющего условию $X(n) = \sum_{m=1}^{M} КМФ_n - r(n)$.

Таким образом, алгоритм КДМЭАШ реализует итеративную процедуру получения множества КМФ, каждый элемент которого находится эмпирически на основе цикличного поиска экстремальных точек предварительно зашумленных входных значений. При этом функция $КМФ_1(n)$, являясь основой получения значений остатков для последующих итераций, фактически совпадает с входным временным рядом $X(n)$, представляя усредненное значение его зашумленного варианта. В дальнейшем она используется в качестве исходных данных предопределенной процедуры алгоритма ДВМ.

Оценивание качества разработанного алгоритма проводилось путем его программной реализации и выполнения декомпозиции тестового временного ряда рабочей нагрузки. Вариант временного ряда выбирался из базы ретроспективных данных ВЦОД Google Cluster, находящейся в открытом доступе для проведения исследований [38]. Выбранный временной ряд предварительно преобразуется в совокупность векторов-скользящих окон фиксированного размера.

Дополнительно данные каждого из векторов нормализуются с целью приведения их характеристик к единому масштабу. В качестве метода нормализации предложено использовать минимаксное масштабирование:

$$X_{норм} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (30)$$

В качестве показателя нормализованных значений временного ряда был выбран показатель CPU_Usage Rate – коэффициент загрузки процессора, процент времени, в течение которого он занят обработкой задач, рассчитываемый путём деления времени работы процессора на общее время за заданный период. Размер вектора-скользящего окна обрабатываемых данных представлен 500 условными отсчетами. Выбранный вариант временного ряда представлен на рисунке 2.4.

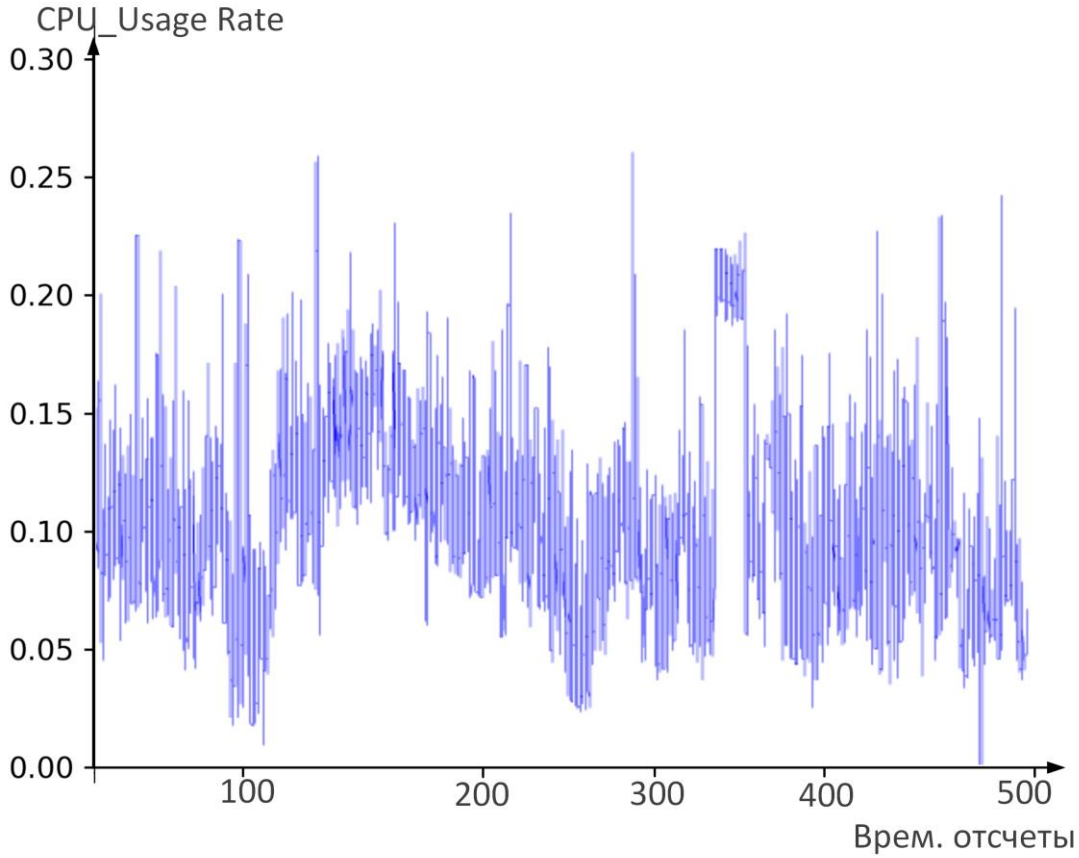


Рис. 2.4 Вариант реализации тестового временного ряда

Реализация разработанного алгоритма КДЭМАШ выполнялась в фреймворке MatLab Digital Signal Processing Toolbox [78], обеспечивающем формирование этапов алгоритма, запуск его выполнения с тестовым временным рядом и визуализацию полученного решения.

В результате реализации разработанного алгоритма КДЭМАШ было получено следующее множество КМФ функций (рисунок 2.5).

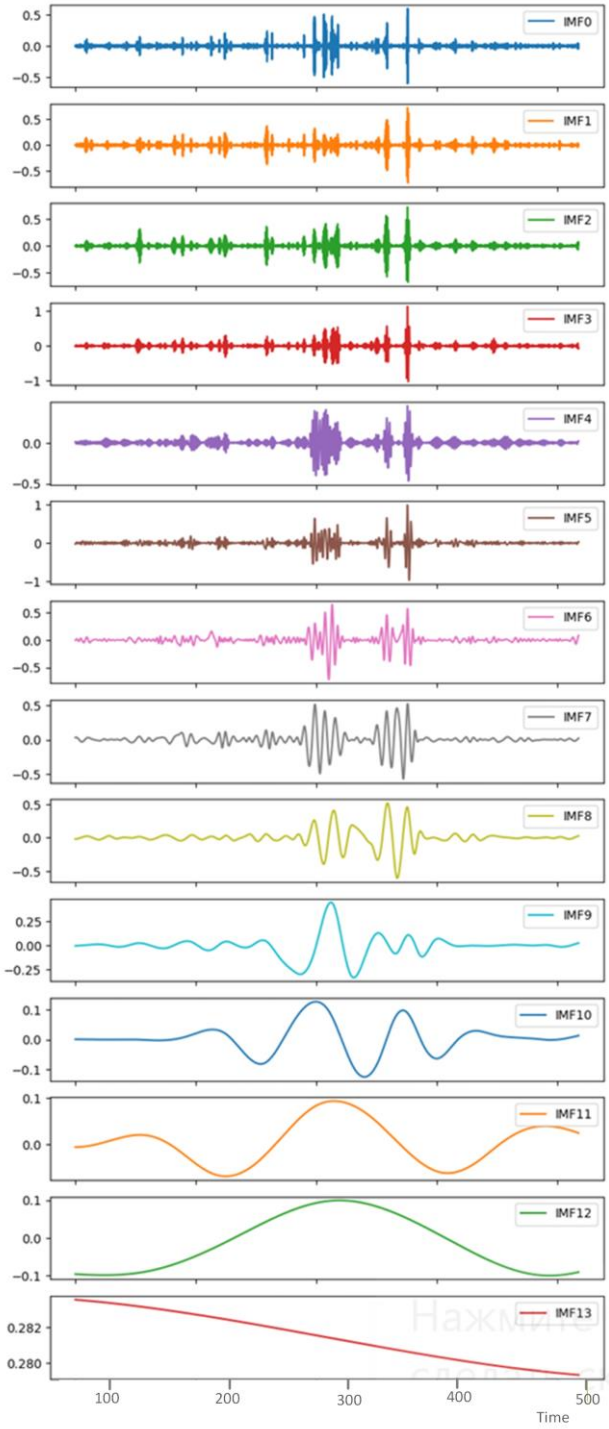


Рис. 2.5 Множество КМФ/КДЭМАШ функций

Из рисунков 2.4 и 2.5 видно, что в результате модовой декомпозиции алгоритмом КДЭМАШ было получено 13 КМФ/КДЭМАШ функций. При этом базовая функция $КМФ_1$ (на рисунке 2.5 – IMF0) в высокочастотной области идентична исходному временному ряду (отсчеты с 300 по 500).

Также видно, что функции $КМФ_2$-$КМФ_5$ (на рисунке 2.5 – IMF1-IMF4) сильно коррелированы, что вносит избыточность в мощность полученного множества функций КМФ/КДЭМАШ. Указанные особенности соответствуют рассмотренным в ходе разработки модели модовой декомпозиции (п. 1.2.3) и требуют выполнения процедуры оптимизации.

2.2.1. Разработка алгоритмов оптимизации множества КМФ функций декомпозиции на эмпирические моды

Как рассмотрено в п. 1.2.4, а также в ходе разработки комплексного алгоритма предварительной обработки временного ряда рабочей нагрузки (рисунок 2.1) множество КМФ функций, получаемых в процессе модовой декомпозиции на эмпирические моды, в частности с использованием предложенного алгоритма КДЭМАШ, не всегда является оптимальным. Эмпирических характер получения КМФ функций в большинстве случае приводит к формированию избыточных элементов множества КМФ, часть из которых имеет сильную корреляцию. Это потенциально усложнить процесс прогнозирования рабочей нагрузки.

Для оптимизации полученного множества функций КМФ/КДЭМАШ при разработке модели модовой декомпозиции временного ряда (п. 1.2) было предложено использование в качестве уровня значимости каждой из КМФ функций значения ее выборочной энтропии (SampEn) с последующей кластеризацией полученного множества этих значений. В качестве подхода к кластеризации было предложено использование метода К-средних.

Алгоритмически метод расчета выборочной энтропии и кластеризация методом К-средних являются известными вычислительными процедурами.

На рисунке 2.6 представлен алгоритм расчета выборочной энтропии для двух соседних значений элементов временного ряда.

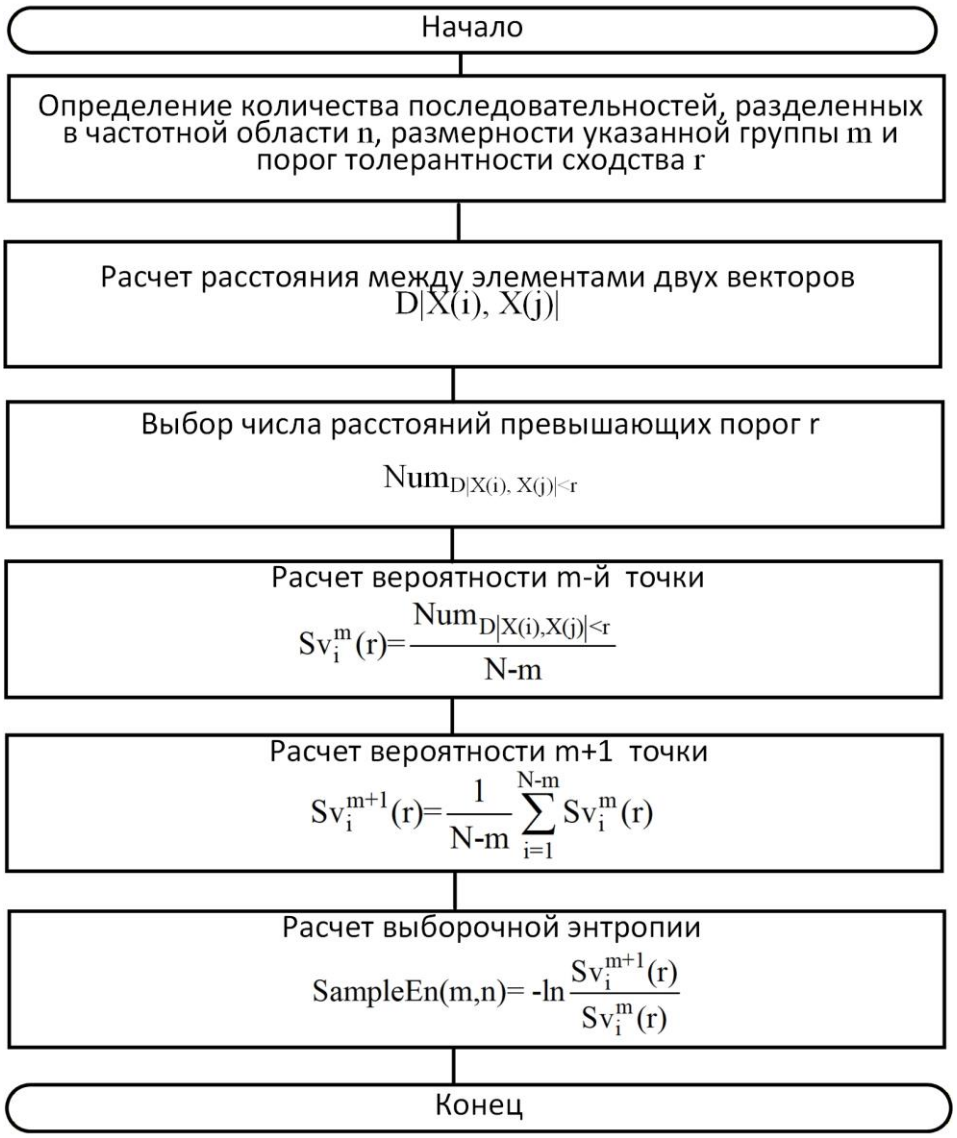


Рис. 2.6 Схема алгоритма расчета выборочной энтропии для соседних элементов временного ряда

Итерационное выполнение представленной выше предопределенной процедуры, применительно к значениям временных рядов полученного множества функций КМФ/КДЭМАШ относительно среднего значения векторов, удовлетворяющих пороговому условию (выражение 26) приводит к формированию обобщенного значения выборочной энтропии КМФ функции, находящегося в диапазоне от 0 (низкая значимость) до 1 (высокая значимость).

Итерации алгоритма расчета выборочной энтропии применительно к полученному множеству функций КМФ/КДЭМАШ для тестового временного ряда рабочей нагрузки (рисунок 2.5) были реализованы в фреймворке MatLab Statistics

and Machine Learning Toolbox [79]. Полученные значения SampEn представлены на рисунке 2.7.

Рис. 2.7 Расчетные значения выборочной энтропии для полученного множества функций КМФ/КДЭМАШ

Из рисунка 2.7 следует, что значения выборочной энтропии подмножества функций $КМФ_2$-$КМФ_4$ (на рисунке IMF1-IMF3) превышают порог 0,5 и в общем случае являются наиболее значимыми для использования в качестве входных данных модуля прогнозирования рабочей нагрузки.

Для уточнения этого предположения выполним кластеризацию полученных значений методом К-средних, используя следующий алгоритм кластеризации (рисунок 2.8).

В результате выполнения итераций алгоритма было получено 6 кластеров значений выборочной энтропии. Для визуально представления их распределения в пространстве состояний был использован подход на основе построения графика

осыпи [76], формируемого на основе расчета сумм квадратов расстояний между центроидами полученных кластеров. Полученный график представлен на рисунке 2.9.

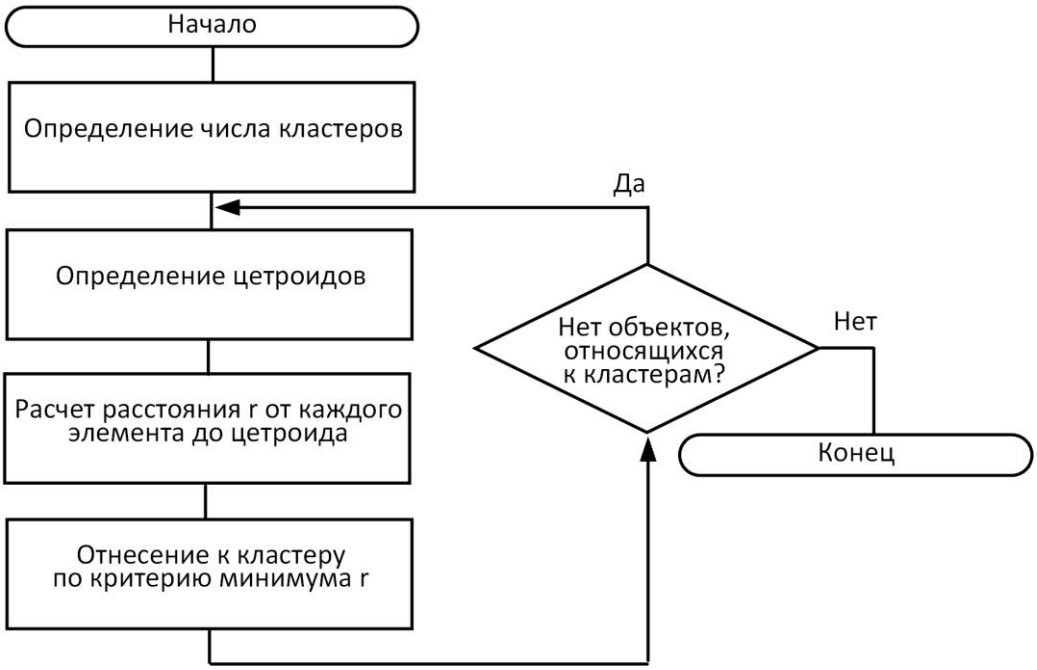


Рис. 2.8 Обобщенная схема алгоритма кластеризации методом К-средних

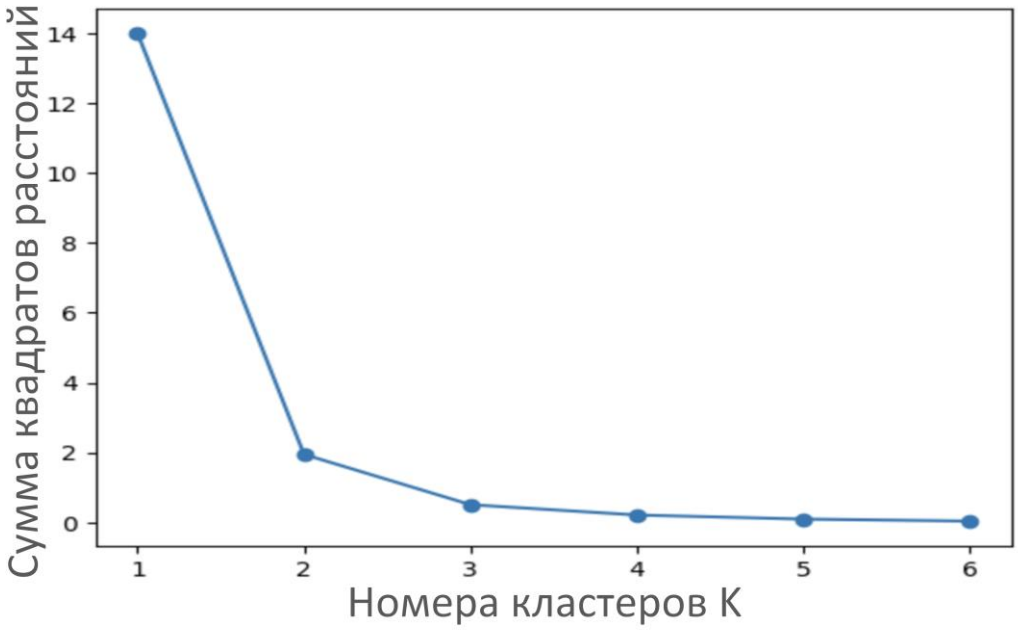


Рис. 2.9 График осыпи для полученного множества кластеров

Из рисунка 2.9 следует, что подмножество функций КМФ/КДЭМАШ - $КМФ_2$-$КМФ_4$, отнесенное к кластеру 1 является наиболее значимым для использования при прогнозировании рабочей нагрузки. То есть, согласно схеме алгоритма (рисунок 2.1) это подмножество относится к подмножеству $\{КМФ'_1, КМФ'_2, ..., КМФ'_n\}$.

2.3. Разработка алгоритма декомпозиции на вариационные моды временного ряда рабочей нагрузки

Выбор этапа разложения временного ряда рабочей нагрузки на вариационные моды (множество функций КМФ/ДВМ) связан с основным достоинством этого метода декомпозиции: получения подмножества достаточно малой мощности при сохранении наиболее значимых величин выбранных показателей рабочей нагрузки.

Это достигается решением сложной вариационной задачи декомпозиции (выражение 19), являющейся вычислительно сложной процедурой.

В его основе лежит нерекурсивный метод разложения сигналов на множество КМФ мощностью K, при котором u_k мода изменяется в пределах ее средней частоты ω_k.

Чтобы решить указанную задачу алгоритмически, используется штрафной параметр α в сочетании с множителем Лагранжа $\lambda(t)$. При этом сложная вариационная задача с ограничениями преобразуется в простую без ограничений, представленную следующими выражениями:

$$\omega_k^{n+1} \leftarrow \arg_{u_k} \min L(\{u_{i<k}^{n+1}\}, \{u_{i\geq k}^{n}\}, \{\omega_i^n\}, \lambda^n) \qquad (31)$$

$$\omega_k^{n+1} \leftarrow \arg_{\omega_k} \min L(\{u_i^{n+1}\}, \{u_{i<k}^{n}\}, \{\omega_{i\geq k}^n\}, \lambda^n) \qquad (32)$$

$$\lambda^{n+1}(\omega) = \lambda^n + \tau(f - \sum_k u_k^{n+1}), \qquad (33)$$

где τ - показатель устойчивости к сходимости.

Исходя из этого была разработана следующая схема алгоритма ДВМ (рисунок 2.10).

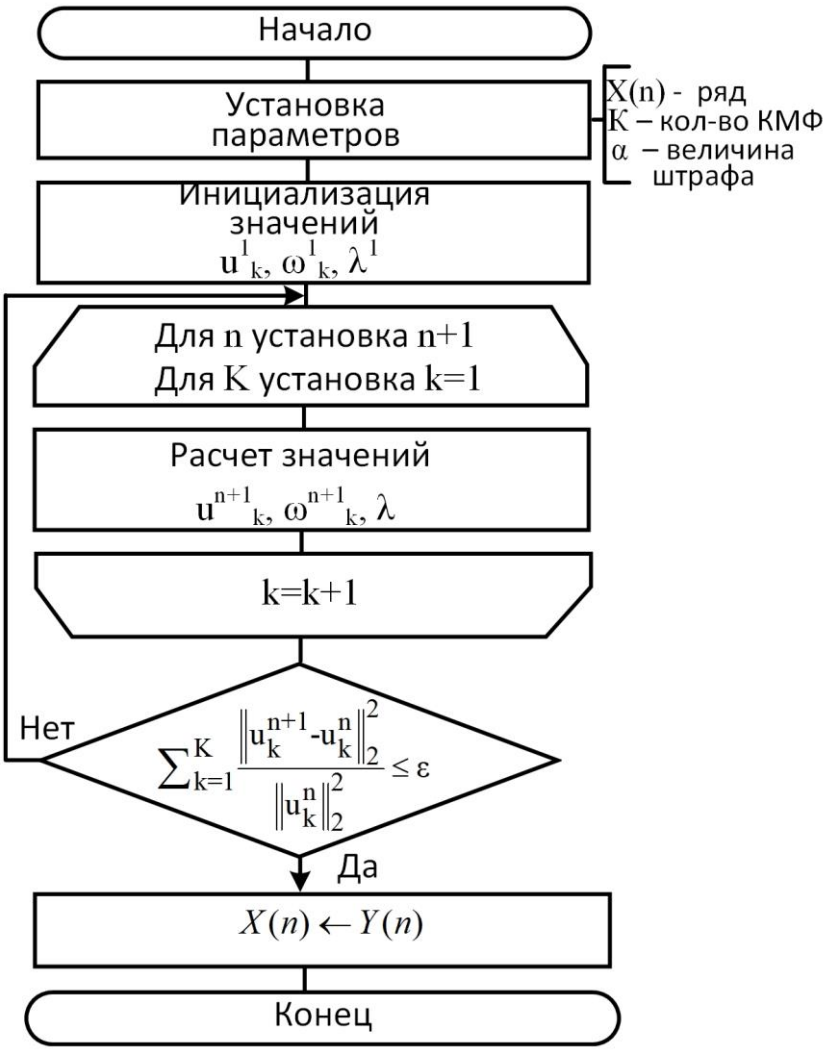


Рис. 2.10 Схема алгоритма ДВМ для получения множества функций КМФ/ДВМ

Из рисунка 2.10 видно, что важным фактором в реализации алгоритма ДВМ является первоначальное определение значений K и штрафного параметра α, существенно влияющих на последующие этапы декомпозиции. Включение в алгоритм цикла подбора этих параметров существенно усложняет его и увеличивает и без того высокую вычислительную сложность. Поэтому наиболее оптимальным способом является подбор этих параметров, основанный на эмпирическом опыте администратора системы прогнозирования, хорошо знакомого с особенностями рабочей нагрузки конкретного ВЦОД.

В качестве входных данных алгоритм ДВМ может использовать, как непосредственные значения временного ряда X(n), так и результат получения

функции КМФ$_1$(n) алгоритмом КДМЭАШ, что позволяет получить более точную декомпозицию за счет усреднения входных значений.

Для программной реализации разработанного алгоритма ДВМ, обеспечивающей снижение вычислительных затрат на получение каждого элемента множества КМФ/ДВМ в п. 1.2.4 было предложено использование алгоритма оптимизации ADMM, обеспечивающего итерационное распараллеливание получения функций КМФ.

2.3.1. Разработка алгоритма чередующихся направлений множителей (ADMM)

В основе алгоритма ADMM глобальной оптимизации, решаемой вариационной задачей (выражение 19) на итерационное выполнение частных оптимизационных задач. При этом на каждой итерации происходит обновление информации о граничных условиях, обновление глобальных и локальных переменных , а также штрафного параметра α.

После разбиения глобальная оптимизация преобразуется во внутреннюю оптимизацию нескольких подобластей. Для решения частной оптимизационной задачи используется расширенный метод Лагранжа (выражение 34):

$$\begin{cases} F_{ADMM}=f_{u_k}+f_{\omega_k}+\sum\left\{\begin{matrix}\sigma(u_{k+1}^n-u_k^n)+\mu(\omega_{k+1}^n-\omega_k^n)\\ +\frac{\alpha}{2}\left[\sigma(u_{k+1}^n-u_k^n)^2+\mu(\omega_{k+1}^n-\omega_k^n)^2\right]\end{matrix}\right\}, \\ \text{при} \\ g_u=0, h_u\geq 0 \\ g_\omega=0, h_\omega\geq 0 \end{cases} \quad (34)$$

где: F_{ADMM} - расширенная форма целевой функции по Лагранжу, f_{u_k} и f_{ω_k} - целевые функции мод и частот; u_k^n и ω_k^n - мода и частота соответственно, выполняющие роль глобальных переменных, обновляемых на каждой итерации; σ и

μ - расширенные множители Лагранжа, выполняющие роль вторичных переменных двойными переменными; k - номер итерации; g и h — ограничения типа равенства и ограничения типа неравенства для каждой моды и частоты соответственно.

Обновление u_k^n и ω_k^n (глобальных переменных) и σ, μ (вторичных переменных) выполняется согласно выражениям 35 и 36.

$$\begin{cases} u_{k+1}^n = \dfrac{(u_k^{n-1} + u_k^{n+1})}{2} \\ \omega_{k+1}^n = \dfrac{(\omega_k^{n-1} + \omega_k^{n+1})}{2} \end{cases} \tag{35}$$

$$\begin{cases} \sigma_{k+1}^u = \sigma_k^u + \alpha^u (u_{k+1}^n - u_k^n) \\ \mu_{k+1}^u = \mu_k^u + \alpha^u (u_{k+1}^n - u_k^n) \\ \sigma_{k+1}^\omega = \sigma_k^\omega + \alpha^\omega (\omega_{k+1}^n - \omega_k^n) \\ \mu_{k+1}^\omega = \mu_k^\omega + \alpha^\omega (\omega_{k+1}^n - \omega_k^n) \end{cases} \tag{36}$$

Таким образом, вычисление выполняется для каждой пары смежных мод и частот. Расчет соответствующих первичных и вторичных остатков r_{k+1}^n, s_{k+1}^n выполняется на основе выражения 36:

$$\begin{cases} r_{k+1}^n = \sqrt{\sum (u_{k+1}^n - u_k^n)^2 + (\omega_{k+1}^n - \omega_k^n)^2} \\ s_{k+1}^n = \sqrt{\sum (u_k^n - u_{k-1}^n)^2 + (\omega_k^n - \omega_{k-1}^n)^2} \end{cases} \tag{37}$$

Поскольку неверный выбор штрафного параметра α может оказать влияние на сходимость алгоритма, то в процессе выполнения итераций производится его корректировка согласно выражению 38:

$$\alpha_{k+1}^{n}=\begin{cases}\dfrac{\alpha_k^n}{1+\tau}, & r_{k+1}^{n} \le \delta s_{k+1}^{n} \\ (1+\tau)\alpha_k^n, & s_{k+1}^{n} \le \delta r_{k+1}^{n}\end{cases} \quad (38)$$

где, значения коэффициентов $\tau>0$ и $\delta\in(0,1)$ соответственно.

Итерации выполняются, пока не выполнится условие сходимость (выражение 39):

$$\left\| \begin{matrix} r_k^n \\ s_k^n \end{matrix} \right\| \le \varepsilon \quad (39)$$

Схема алгоритма ADMM представлена на рисунке 2.11.

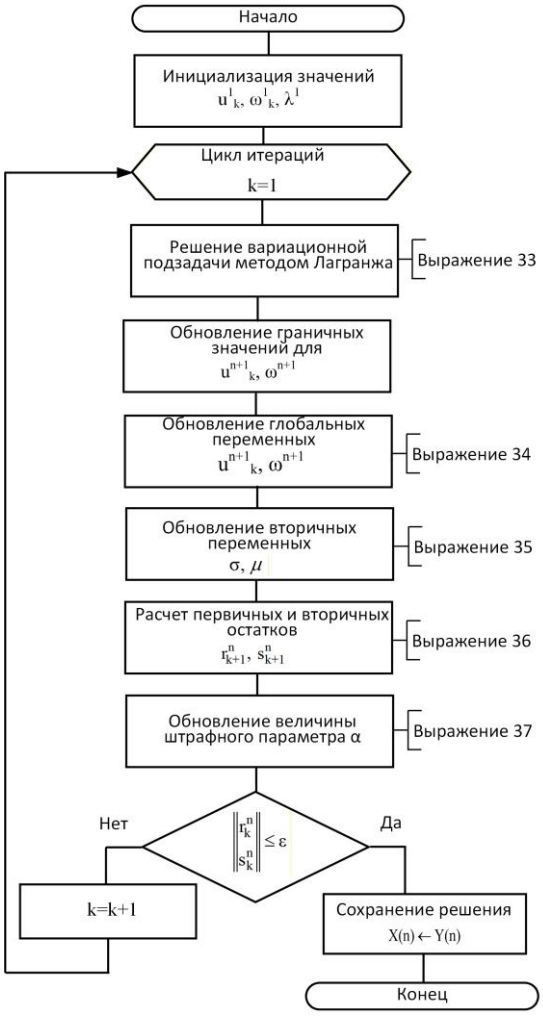


Рис. 2.11 Схема алгоритма ADMM

Обобщенная реализация алгоритма ADMM имеется в составе фреймворка MatLab Statistics and Machine Learning Toolbox. В ходе исследования она была модифицирована для решения задачи декомпозиции методом ДВМ.

В результате его выполнения с входными данным в виде функции КМФ$_1$ множества КМФ/КДЭМАШ (IMF0 на рисунке 2.5) было получено следующее множество функций КМФ/ДВМ (рисунок 2.12).

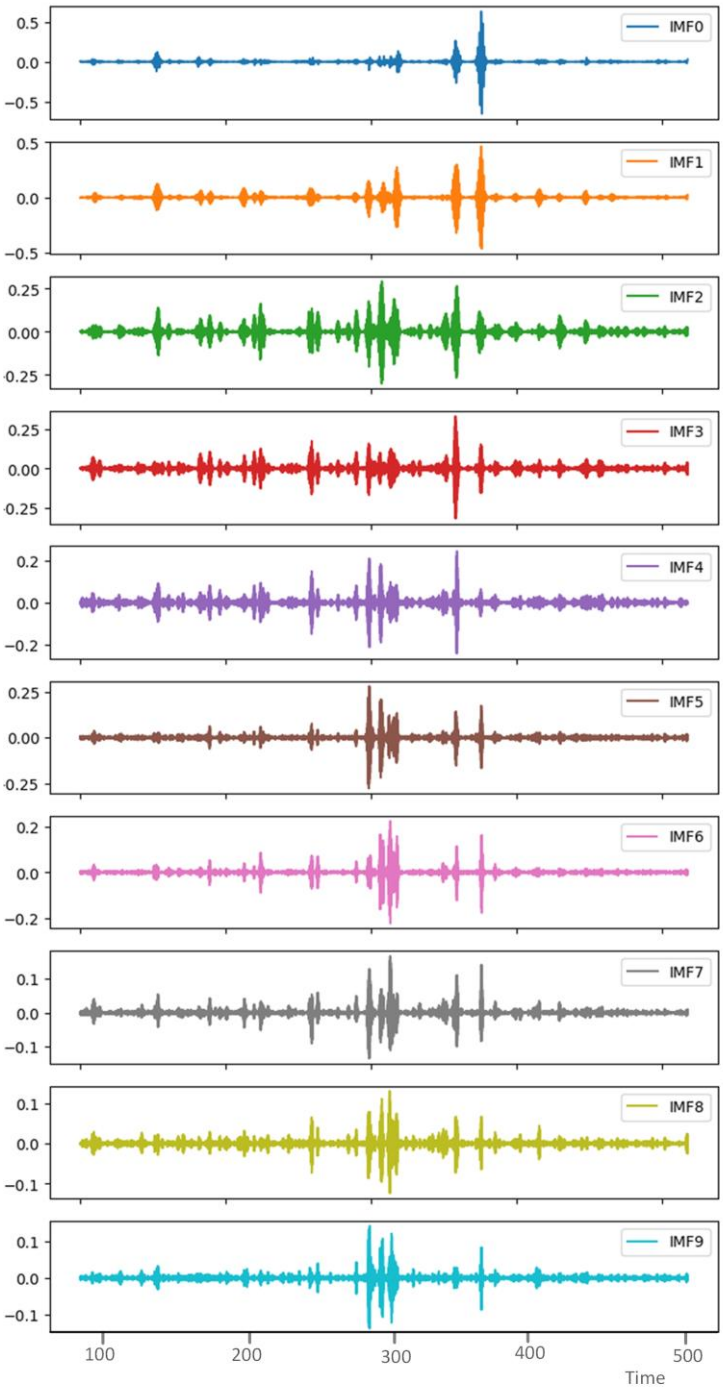


Рис. 2.12 Множество КМФ/ДВМ функций

Из рисунка 2.12 видно, что благодаря численному характеру метода ДВМ получено редуцированное в 1,5 раза множество функций КМФ/ДВМ относительно множества КМФ/КДЭМАШ.

При этом каждая из полученный функций КМФ несет в себе информативную составляющую, что не требует решения задачи выбора наиболее значимых КМФ функций, характерной для метода КДЭМАШ (п. 2.1.1).

Таким образом, полученное множество функций КМФ/ДВМ можно отнести к подмножеству $\{КМФ'''_1, КМФ'''_2, ..., КМФ'''_n\}$ (рисунок 2.1) разработанного комплексного алгоритма предварительной обработки временного ряда рабочей нагрузки.

2.4. Выводы по главе

В главе рассмотрен разработка комплексного алгоритма модуля предварительной обработки временных рядов показателей рабочей нагрузки ВЦОД. На основе разработанной модели модовой декомпозиции временного ряда, обеспечивающей формирование множества колебательных модовых функций (КМФ), использование которых позволяет отсеять составляющие, характерные для факторов зашумления, был разработан ряд предопределенных процедур комплексного алгоритма, выходом которого являются подмножества КМФ, используемые, вместо исходного временного ряда, в качестве входных данных модуля прогнозирования рабочей нагрузки ВЦОД.

В качестве основы комплексного алгоритма была разработана/преопределенная процедура, реализующая алгоритм варианта декомпозиции на эмпирические моды, а именно комплементарой декомпозиции на эмпирические моды с адаптивным шумом (КДЭМАШ). С целью устранения основного недостатка этого алгоритма – избыточной мощности множества КМФ/КДЭМАШ, а также возможной корреляции значений соседних КМФ функций, были разработаны процедуры оптимизации этого множества функций, включая расчет выборочной энтропии каждой КМФ функции, отображающей уровень

ее значимости в составе исходного временного ряда, а также алгоритм кластеризации значений выборочной энтропии на основе метода К-средних.

Также, с целью повышения качества входных данных модуля прогнозирования рабочей нагрузки, были разработаны процедуры декомпозиции на вариационные моды (ДВМ), в частности, вариант не итерационного алгоритма ДВМ, а также алгоритм его итерационной оптимизации, основанный на известном алгоритме ADMM. В качестве входных данных разработанный алгоритм ДВМ использует базовую функцию $КМФ_1$, полученную в процессе декомпозиции исходного временного ряда на эмпирические моды.

Для оценивания качества разработанных алгоритмов была выполнена декомпозиция тестового временного ряда рабочей нагрузки базы ретроспективных данных ВЦОД Google Cluster.

Таким образом, разработан комплексный алгоритм предварительной обработки временного ряда рабочей нагрузки, отличающийся наличием этапа вторичной вариационной модовой декомпозиции базовой колебательной модовой функции, полученной методом эмпирической модовой декомпозиции, которая обеспечивает формирование множеств обучающей и тестовой выборок для системы прогнозирования элементов временного ряда.

Глава 3. Разработка гибридного алгоритма прогнозирования рабочей нагрузки виртуализированного центра обработки данных

Как было рассмотрено в главе 1, а также в [80, 81], управление рабочей нагрузкой ВЦОД реализуется службой его администрирования. Оно основано на решении двух классов задач:

1. Мониторинга текущего состояния вычислительных ресурсов по показателям их производительности, надежности и т.д.

2. Реконфигурации виртуализированной инфраструктуры ВЦОД путем миграции и/или приостановки функционирования виртуальных машин/контейнеров с целью повышения качества потребительского обслуживания.

Очевидно, что указанные задачи взаимосвязаны. Так для оперативного управления задача текущей реконфигурации инфраструктуры ВЦОД базируется на анализе результатов текущего мониторинга его вычислительных ресурсов.

При этом одной из особенностей ВЦОД является сохранение результатов мониторинга вычислительных ресурсов за предыдущие периоды его функционирования в специализированном СХД. Такие данные именуются ретроспективными данными рабочей нагрузки (workload historical data) [82]. Анализ таких данных позволяет реализовывать, как функции текущего (оперативного, реактивного) управления рабочей нагрузкой, так и ее упреждающего (проактивного) управления.

Функция проактивного управления, ориентирована на решении класса задач, именуемых прогнозирование рабочей нагрузки (workload prediction). Обобщенной целью этого класса задач является получение прогноза ее шаблонов в будущие моменты времени функционирования ВЦОД.

3.1. Анализ методов прогнозирования временных рядов рабочей нагрузки ВЦОД

Из п. 1.14 следует, что подсистема мониторинга службы администрирования ВЦОД сохраняет ретроспективные данные рабочей нагрузки в виде временного ряда (time series) выбранных показателей. Таким образом, задача прогнозирования рабочей нагрузки относится к классу задач анализа временных рядов [83].

В общем виде задача прогнозирования временных рядов представлена в [84]. На рисунке 3.1 рассматриваются основные компоненты указанной задачи.

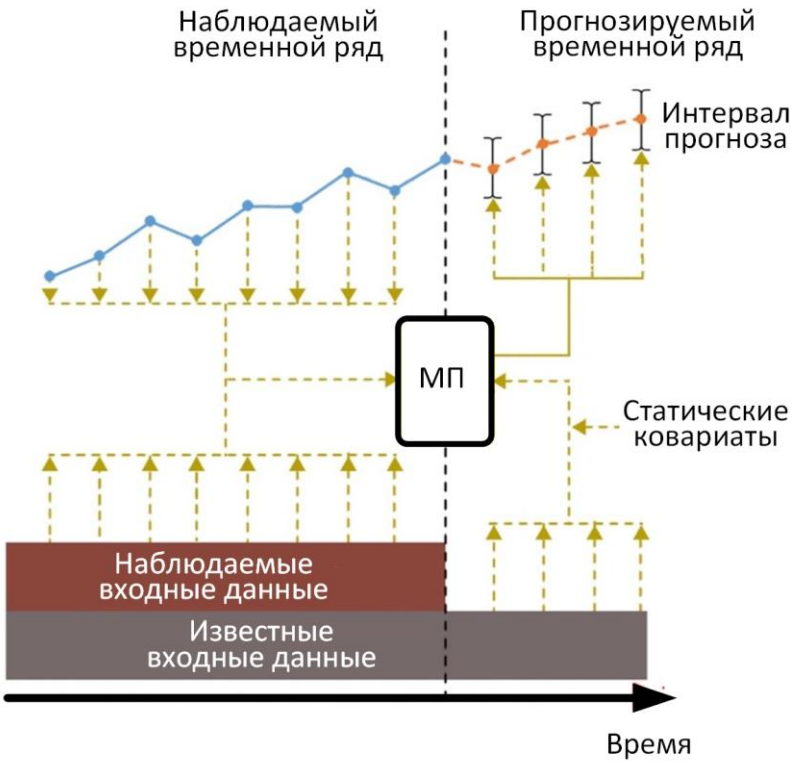


Рис. 3.1 Обобщенное представление задачи прогнозирования временных рядов

Из рисунка видно, что основой решения задачи является модель прогнозирования (МП), которая объединяет:

– наблюдаемый (исходный) временной ряд, значения которого являются наблюдаемыми (ретроспективными) входными данными;

– известные входные данные, получаемые от предыдущих решений задачи прогнозирования и, включающие ретроспективные данные и прогнозные данные;

– статические ковариаты – константы, определяющие закономерности трендов, сезонности и колебаний данных и не меняющиеся в течение прогнозируемого периода;

– прогнозируемый (целевой) временной ряд, являющийся выходными данными, значения которого могут колебаться в пределах некоторого допустимого интервала прогноза, который описывает степень неопределенности оценок.

Рассмотрение исследований в предметной области анализа временных рядов различной природы [85, 86] показало, что выбор МП существенно зависит:

– от характеристик используемых исходных временных рядов;

– имеющейся набора временных рядов, представляющих релевантную выборку для данной предметной области.

В силу этого, методы и алгоритмы, реализуемые в МП, существенно зависят от особенностей предметной области, в рамках которой решается задача прогнозирования, а также возможностей по сбору, предварительному анализу и обработке получаемого множества исходных временных рядов.

Так, в ряде случаев, получение временного ряда выбранных показателей является сложной или единичной задачей, что затрудняет использование некоторых методов, основанных на обучении МП и требует применение статистического подхода к получения целевого временного ряда.

В [87] дается подход к обобщению методов прогнозирования и рассматривается место методов прогнозирования временных рядов. В графическом виде он представлен на рисунке 3.2.

Из рисунка видно, что решение задачи прогнозирования временных рядов обобщает следующие количественные методы:

– основанные на экспоненциальном сглаживании (ES – Exponential Smoothing);

– основанные на авторегрессии (AR - AutoRegressive);

– основанные на регрессионных моделях (RM – Regressive Model);

– основанные на моделях глубокого обучения (МГО, DLM – Deep Learning Model).

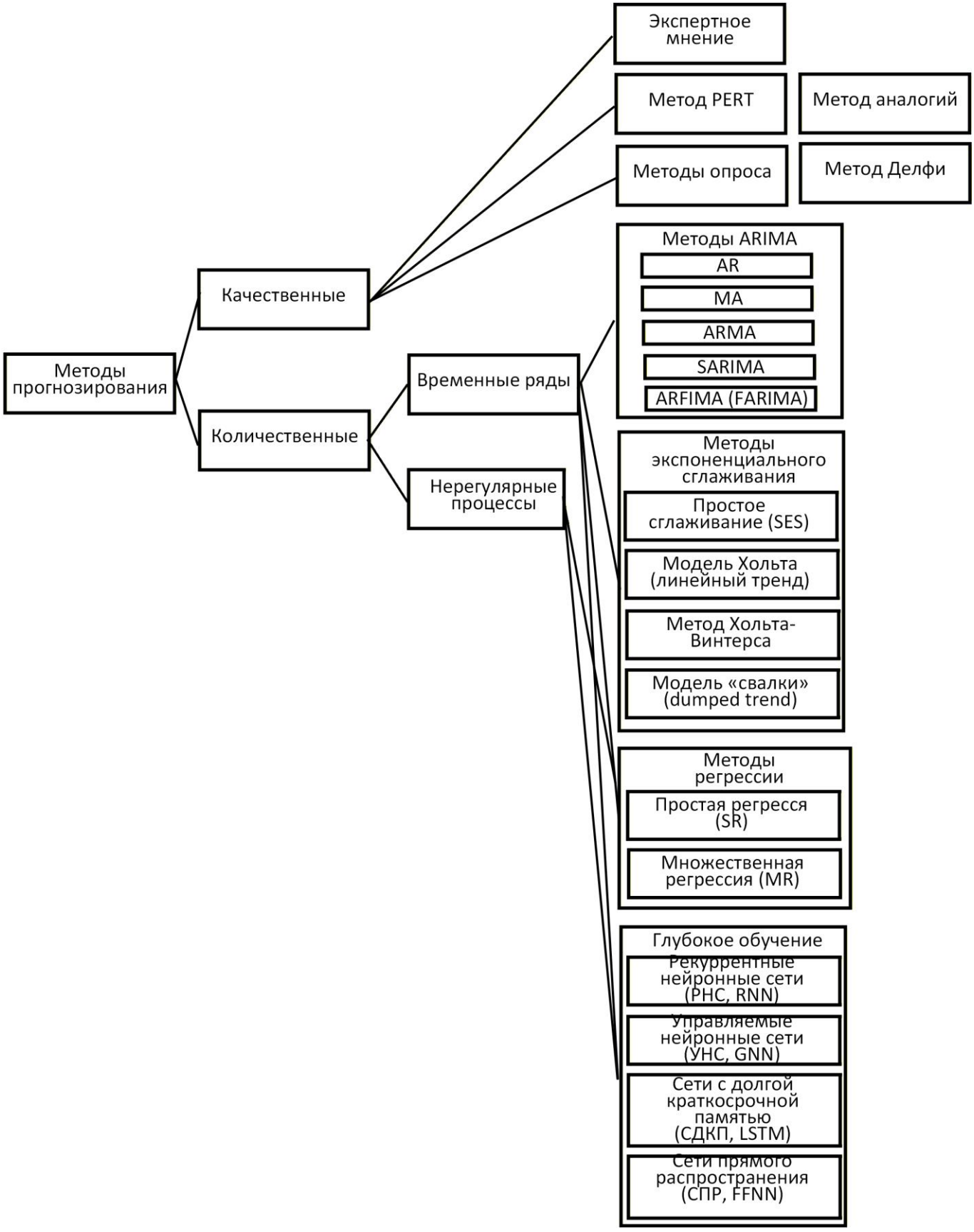


Рис. 3.2 Таксономия методов прогнозирования временных рядов

В целом рассмотренные классы методов прогнозирования временных рядов можно сгруппировать в две категории:

1. Методы, основанные на статистическом вероятностном моделировании (СВМ). К ним относятся методы экспоненциального сглаживания, авторегрессионные методы, регрессионные модели.

2. Методы, основанные на глубоком обучении (МГО).

Одним из наиболее известных методов СВМ является ARIMA [87, 88], который состоит из трёх основных компонентов:

– модели авторегрессиии (AR);

– метода скользящего среднего (MA);

– метода интегрироваия (I).

Авторегрессионная модель оценивает насколько текущее значение данных зависит от прошлых значений данных, метод скользящего среднего прогнозирует возникновение ошибки, по их анализу за предыдущий период. Метод интегрирования обеспечивает стационарность данных, путём удаления долгосрочных трендов. В базовом варианте ARIMA подходит для наборов данных, не демонстрирующих регулярных (сезонных) закономерностей, а для их анализа разработан его вариант SARIMA (Season ARIMA) [89].

Для случая временных рядов с выраженными долгосрочными зависимостями (значения в отдалённом будущем в значительной степени определяются значениями в отдалённом прошлом) разработана модификация ARFIMA (другой вариант – FARIMA), базирующаяся на фрактальной теоретической [90].

В общем виде модель ARIMA(p,d,q) для некоторого нестационарного временного ряда X_t представляется выражением 40:

$$\Delta^d X_t = c + \sum_{i=1}^{p} a_i \Delta^d X_{t-i} + \sum_{j=1}^{q} b_j \varepsilon_{t-j} + \varepsilon_t, \quad (40)$$

где: ε_t - стационарный временной ряд; p и q — целые числа, задающие порядок модели ARMA; c, a_i, bj — параметры модели ARIMA; Δ^d - оператор

разности временного ряда порядка d (коэффициент дифференцирования). В модели ADIMA коэффициент d является целочисленным, в то время как в модели FARIMA – дробным.

Методы экспоненциального сглаживания находят широкое применение также при прогнозировании относительно стационарных временных рядов. Так простое экспоненциальное сглаживание (модель Брауна) реализует присвоение больших весов самым последним данным, что подходит для временных рядов без выраженных трендов или явлений сезонности [91].

Известными модификациями метода экспоненциального сглаживания, обеспечивающими учет во временных рядах трендов, являются: модель линейного тренда Хольта, а также метод Хольта-Уинтерса, поддерживающий прогнозирование временных рядов с сезонными закономерностями [92].

В общем виде модель Хольта (являющаяся, по сути, расширением модели Брауна) для прогнозирования целевого временного ряда на основе исходного ряда $y_r,...,y_t$, $y_r \in R$ представляется выражением 41:

$$\hat{y}_{t+d}=a_t+db_t, \quad (41)$$

Где a_t – прогноз на основе простого экспоненциального сглаживания (модель Брауна) (выражение 42), а b_t – параметр линейного тренда (выражение 43).

$$a_t=\alpha_1 y_t+(1-\alpha_1)(a_{t-1}-b_{t-1}) \quad (42)$$

$$b_t=\alpha_2(a_t-a_{t-1})+(1-\alpha_2)b_{t-1} \quad (43)$$

Для временных рядов с постепенно затухающих трендом разработана соответствующая модификация модель Хольта [93].

Очевидно, что применение рассмотренных методов и моделей СВМ ориентировано только на формирование прогнозных значений исходного временного ряда и не обеспечивают решение задачи распознавания значений показателей рабочей нагрузки, связанных с ее шаблонами (профилями поведения).

С целью решения указанной задачи, в реальных реализациях модулей прогнозирования рабочей нагрузки ВЦОД, модели СВМ дополняются моделями машинного обучения (ММО), основанными на статистических, метрических и вероятностных методах классификации. Так, в качестве классификатора последовательных данных, к которым можно отнести временные ряды, хорошо себя зарекомендовали скрытые марковские модели (СММ, HMM – Hidden Markov Model) [94]. Также для выделения шаблонов рабочей нагрузки нашли свое применение машины опорных векторов (МОВ, SVM – Support Vector Machine) [95]. Например, в оптимизаторе рабочей нагрузки ВЦОД Google Cluster применяется подход на основе комбинации классификатора МОВ и моделей прогнозирования ARIMA [96].

В обобщенном виде подобных подход к прогнозированию временных рядов можно представить, как статистические вероятностные модели обучения (СВМО).

Рассмотренные методы прогнозирования наиболее эффективны для прогнозирования временных рядов с относительно большими участками стационарности, поскольку ориентируются на поиск статистических зависимостей.

Для нелинейных временных рядов в последнее время эффективность доказали метода на основе МГО, базирующиеся на искусственных нейронных сетях разной архитектуры.

Наиболее простым подходом является использование архитектур прямого распространения, таких, например, как сверточные нейронные сети (СНН, CNN) [97].

Однако наибольшую эффективность для решения задачи прогнозирования временных рядов показали нейронные сети с элементами памяти и управления, такие как для рекуррентные нейронные сети (РНС, RNN) [98], управляемые рекуррентные сети [99], а также варианты сетей с долгой краткосрочной памятью (СДКП, LSTM) [100].

В общем виде этапы использования рассмотренных СВМО и МГО в процессе решения задачи прогнозирования временных рядов в сравнительном виде представлены на рисунке 3.3.

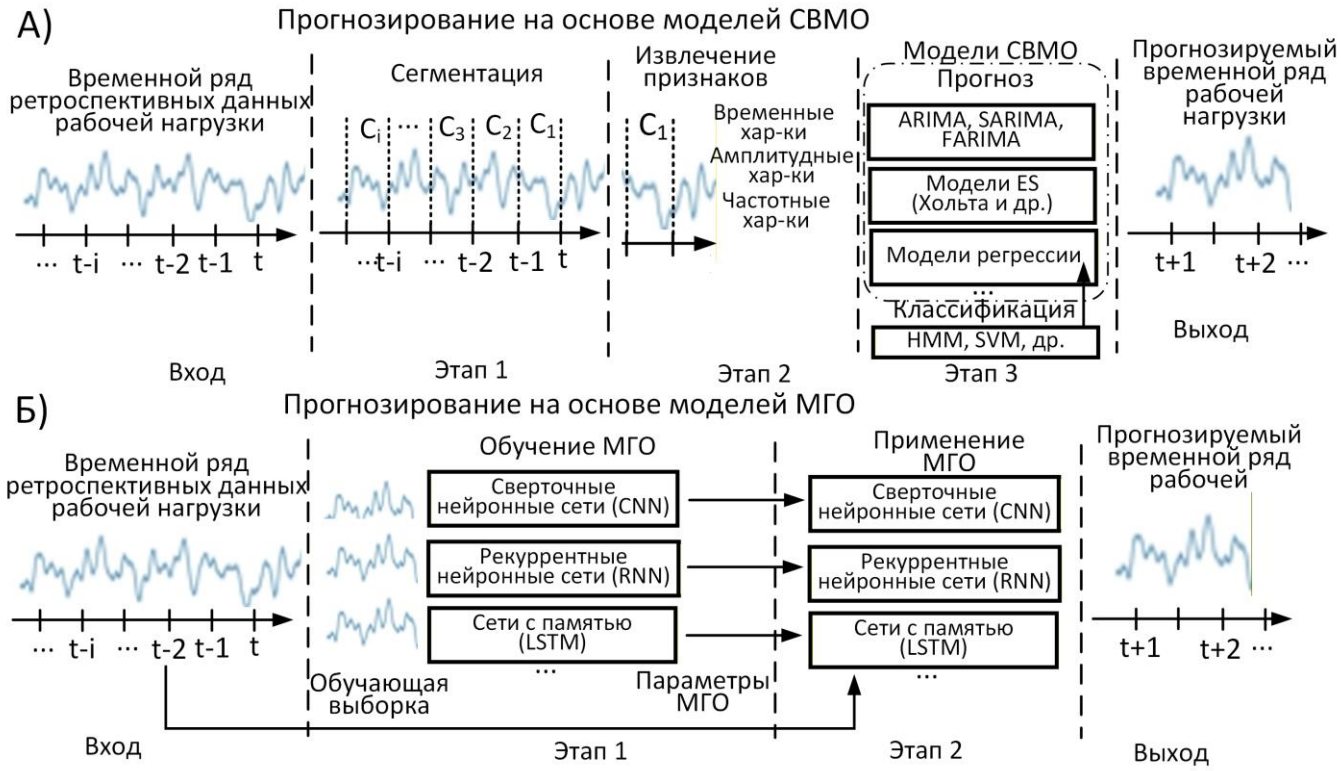


Рис. 3.3 Сравнительное представление этапов функционирования статистических вероятностных моделей и моделей глубокого обучения при решении задачи прогнозирования временных рядов

Из рисунка 3.3 видно, что основой моделей СВМО являются модели и методы статистического прогнозирования (ARIMA, ES и др.), дополненные хорошо зарекомендовавшие себя в задачах кластеризации и классификации методами МО, такие как скрытые марковские модели (HMM), машины опорных векторов (SVM) и др.

Особенностью использования моделей СВМО является необходимость предварительной обработки входных данных. Применительно к временным рядам – это: сегментация значений временного ряда, нормализация данных в полученных сегментах, а также выделение в них значимых признаков, например, частотной и иных характеристик. Процесс обучения при этом может быть реализован, как по технологии «с учителем», так и «без учителя».

В отличие от моделей СВМО, модели МГО базируются на нейронных сетях различного типа и, благодаря многократной реализации в ходе их обучения

процесса обратного распространения ошибки, минимизирующего функцию потерь, они способны на выявление во входных данных значимых с точки зрения исследователя признаков, являющихся их выходными значениями.

В рамках исследования разработка алгоритмов модуля прогнозирования рабочей нагрузки производилась на основе моделей МГО.

3.2. Разработка структуры модели глубокого обучения для решения задачи прогнозирования рабочей нагрузки ВЦОД

Как было рассмотрено в п. 3.1 модели МГО, в силу своих структурно-параметрических и функциональных особенностей обеспечивают, как возможность поиска зависимостей значений показателей рабочей нагрузки ВЦОД, так и возможность формирования прогнозных значений ее временного ряда.

При этом достаточно большое разнообразие архитектур МГО требует решения задачи выбора таких из них, которые наиболее эффективно реализуют решение частных задач прогнозирования:

– выделения в исходных временных ряда признаков, определяющих шаблоны, характерные для различных условий реализации рабочей нагрузки ВЦОД;

– формирование целевого временного ряда рабочей нагрузки ВЦОД, содержащего ее прогнозные значения относительно найденных шаблонов.

В общем виде подход к решению задачи прогнозирования с использованием МГО представлен на рисунке 3.4.

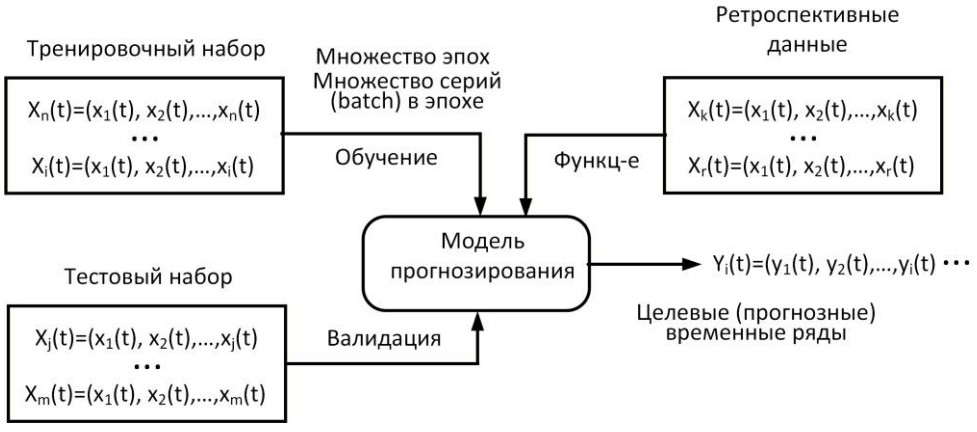


Рис. 3.4 Этапы использования МГО в качестве модели прогнозирования временных рядов

Таким образом, представленная на рисунке 3.4 МГО модель прогнозирования, фактически должна реализовывать две функции: выделение в значениях исходного временного ряда зависимостей, характерных для типовых и не типовых (аномальных) шаблонов рабочей нагрузки; формировать прогнозные значения целевого временного ряда, с учетом указанных зависимостей.

Анализ литературы, посвященной использованию МГО для прогнозирования временных рядов в различных предметных областях [101, 102], показал, что в рамках одной архитектуры МГО решение указанных задач невозможно. Наиболее приемлемым подходом является комбинация МГО, последовательно решающих указанные задачи (гибридный подход).

Соответственно, важными исследовательскими задачами являются:

– выбор архитектуры МГО для распознавания шаблонных зависимостей временного ряда;

– выбор архитектуры МГО, обеспечивающих формирование целевого (прогнозного) временного ряда;

– выбор методов и средств интеграции указанных архитектур МГО.

3.2.1. Разработка структуры модели одномерной сверточной нейронной сети и алгоритма ее обучения для решения задачи распознавания значимых признаков показателей временного ряда

Выбор модели МГО для распознавания значимых признаков показателей временного ряда рабочей нагрузки проводился на основе анализа моделей МГО, используемых для решения задачи распознавания образов. Анализ источников этой предметной области [103, 104] показал, что наибольшей эффективностью в решении подобной задачи обладают архитектуры сверточных нейронных сетей (далее СНН).

Также анализ источников, посвященных использованию СНН для распознавания объектов [105] позволил определить, что применительно к распознаванию заданных и/или аномальных значений временных рядов используется специальная архитектура СНН, именуемая одномерная СНН (СНН-ОМ, англ. 1D-CNN).

Структура СНН-ОМ специализирована для получения входных данных в виде одномерного массива, к варианту которого относятся ретроспективные данные рабочей нагрузки, представленные временными рядами заданных параметров производительности вычислительных ресурсов ВЦОД.

При этом, как и в случае традиционных двумерных вариантов СНН (СНН-ДМ, 2D-CNN), адаптированных, например, для решения задач классификации изображений, СНН-ОМ содержит чередование сверточных слоев и слоев субдискретизации (pooling-слоев), которые формируют входной вектор признаков для полносвязного слоя, формирующего выход СНН-ОМ. В отличие от двумерных СНН, в которых сканирующее ядро (С – от core) – фильтр, представлен двумерным массивом, ядро СНН-ОМ является одномерным, что обеспечивает его скольжение по вектору входных значений временного ряда $[x_1,x_2,...,x_n]$ для нахождения требуемых признаков, характеризующих те или иные шаблоны рабочей нагрузки.

Отличительные особенности сверточного слоя и вида ядра в СНН-ДМ и СНН-ОМ представлены на рисунке 3.5.

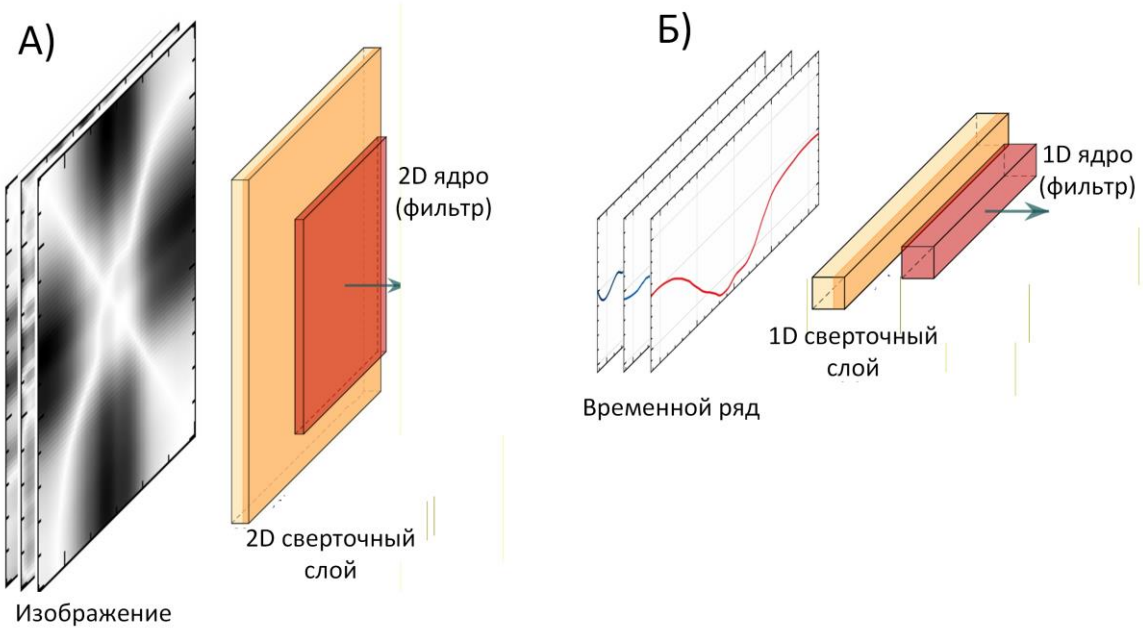


Рис. 3.5 Отличия сверточного слоя и ядра (фильтра) в двумерной и одномерной сверточных нейронный сетях

Обобщенно архитектура СНН-ОМ для распознавания значимых признаков показателей временного ряда представлена на рисунке 3.6.

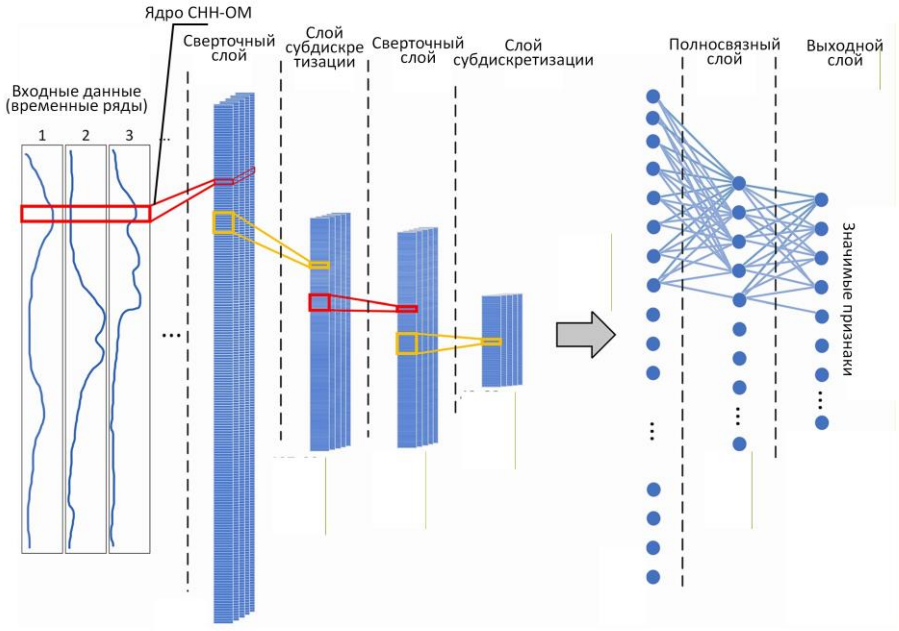


Рис. 3.6 Обобщенная архитектура одномерной сверточной нейронной сети (СНН-ОМ) для распознавания значимых признаков показателей временных рядов

Этапы разработки архитектуры СНН-ОМ в рамках проводимого исследования представлены в [80]. При этом операция свертки вектора входных значений временного ряда и ядраc может быть представлена выражением:

$$(\mathrm{x}*\mathrm{C})(t) = \sum_{\mathrm{i}=0}^{\mathrm{r}-1} x(\mathrm{t}+\mathrm{i}) \cdot \mathrm{C}(i), \qquad (44)$$

где произведение $(\mathrm{x}*\mathrm{C})(t)$ - свёртка значений x и Cв t -й позиции ряда, r - размерность ядра,$x(\mathrm{t}+\mathrm{i})$ — элемент входной последовательности в $\mathrm{t}+\mathrm{i}$позиции ряда, $C(i)$ - элемент ядра в i -й позиции ряда.

Очевидно, что от размерности ядра C зависит масштаб охвата значений временного ряда, в разных масштабах выборки его значений.

Прямое распространение (Forward Propagation) в СНН-ОМ включает в себя прохождение входных данных через один или несколько сверточных слоев, слои субдискретизации и полносвязные слои, так что карта признаков Z_c задается как:

$$Z_c = f_c(x * w_c + b_c), \quad (45)$$

где x - входные данные, w_c - веса ядра C, b_c- смещение, f_c – функция активации свёртки.

Пространственная размерность карты признаков Z_c уменьшается путём агрегирования информации из соседних значений за счет операции субдискретизации (polling), которая определяется как:

$$A_p = \mathrm{P}(Z_c). \quad (46)$$

Далее полносвязный слой объединяет признаки, полученные в результате операций свертки и субдискретизации, и для получения выходных данных используется итоговая функция активации Y.

На рисунке 3.7 представлена схема трех последовательных слоев k -го нейрона предлагаемой структуры СНН-ОМ.

Из рисунка видно, что процесс прямого распространения от предыдущего слоя $l-1$ для создания входа k -го нейрона следующего слоя l можно выразить как:

$$x_k^l = \mathrm{b}_k^l + \sum_{\mathrm{i}=1}^{N_{l-1}} \mathrm{conv1D}(C_{\mathrm{ik}}^{l\text{-}1}, \mathrm{s}_i^{l\text{-}1}), \quad (47)$$

где x_k^l— вход, b_k^l — смещение k -го нейрона в слое l, $s_i^{l\text{-}1}$— выход i -го нейрона в слое $l-1$, $C_{\mathrm{ik}}^{l\text{-}1}$ — одномерное ядро от i -го нейрона в слое $l-1$ до k -го нейрона в слое l.

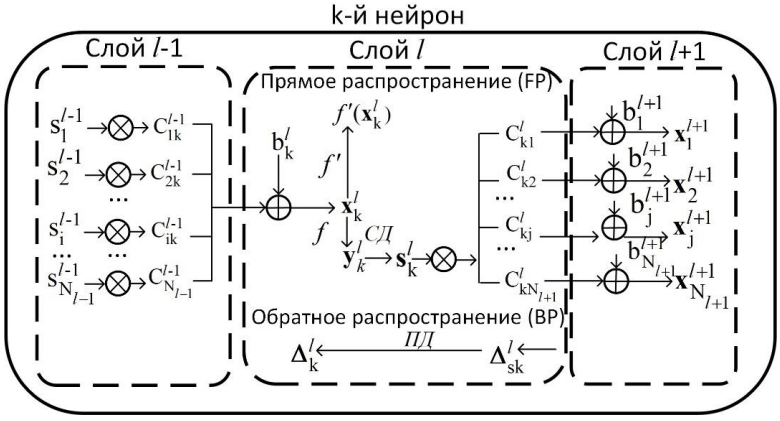


Рис. 3.7 Структурная схема взаимодействия слоев k-го нейрона одномерной сверточной нейронной сети

Для возможности определения произвольного количества скрытых слоев, предлагается рассматривать подход адаптивной СНН [106]. В ней коэффициент субдискретизации СД (рисунок 3.7) выходного слоя назначается адаптивно в зависимости от размеров его карты признаков Z_c. То есть, размерность входа карты признаков текущего слоя уменьшается на $r-1$, где r - размерность ядра $C_{\text{ik}}^{l\text{-}1}$.

Рассмотрим процесс обучения СНН-ОМ методом обратного распространения ошибки (Back Propagation). Определим $l=1$ и $l=L$ как входной и выходной слои СНН-ОМ соответственно.

Среднеквадратическая ошибка (MSE) в выходном слое может быть выражена как:

$$\text{MSE}\left(y_1^L,\ldots,y_{N_L}^L\right)=\text{E}=\sum_{\text{i}=1}^{N_L}(y_i^L-\text{t}_i)^2. \qquad (48)$$

Очевидно, что для вектора входных значений временного ряда $[x_1,x_2,\ldots,x_n]$и соответствующего ему выходного вектора $[y_1^L,\ldots,y_{N_L}^L]$ требуется найти производную MSE по каждому индивидуальному весу $C_{\text{ik}}^{l\text{-}1}$, связанному с k -м нейроном, и смещению b_k^l для последующего применения метода градиентного спуска, обеспечивающего минимизацию MSE. Так, Δ_k^l - дельта k -го нейрона в слое l будет использоваться для обновления значения b_k^l этого нейрона и всех весов нейронов в предыдущем слое, как:

$$\frac{\partial E}{\partial \text{C}_{\text{ik}}^{l\text{-}1}}=\Delta_k^l y_i^{l\text{-}1}, \qquad (49)$$

$$\frac{\partial E}{\partial b_{\text{k}}^l}=\Delta_k^l$$

Таким образом, от входного полносвязного слоя до выходного слоя СНН-ОМ скалярная функция обратного распространения Δs_k^l задается как:

$$\frac{\partial E}{\partial s_k^l}=\Delta\text{s}_k^l=\sum_{\text{i}=1}^{N_{l+1}}\frac{\partial E}{\partial x_i^{l+1}}\cdot\frac{\partial x_i^{l+1}}{\partial s_k^l}=\sum_{\text{i}=1}^{N_{l+1}}\Delta_i^{l+1}\,\text{C}_{\text{ki}}^l. \qquad (50)$$

После того, как функция Δs_k^l выполнена от слоя $l+1$ до слоя l, она распространяется на вход Δ_k^l (рисунок 3). Определим это преобразование, как повышение дискретизации (ПД), как $\text{ПД}(s_k^l)$. Тогда Δ_k^l определяется как:

$$\Delta_k^l = \frac{\partial E}{\partial y_k^l} \cdot \frac{\partial y_k^l}{\partial x_k^l} = \frac{\partial E}{\partial \text{ПД}_k^l} \cdot \frac{\partial \text{ПД}_k^l}{\partial y_k^l} f(x_k^l) = \text{ПД}(\Delta s_k^l) \beta f'(x_k^l), \quad (51)$$

где $\beta = (\text{СД})^{-1}$ – операция, обратная субдискретизации, поскольку каждый элемент s_k^l был получен путем усреднения количества элементов СД промежуточного выхода y_k^l.

Тогда Δs_k^l - дельта ошибки при выполнении функции обратного распространения между слоями может быть выражена, как:

$$\Delta s_k^l = \sum_{i=1}^{N_{l+1}} \text{conv1Dz}(\Delta_k^{l+1}, \text{rev}(C_{ki}^l)), \quad (52)$$

где $\text{rev}(C_{ki}^l)$ - операция реверса массива весов C_{ki}^l, а $\text{conv1Dz}()$- операция свертки в одномерном пространстве с добавлением $r-1$ нулей. При этом чувствительность к весу и смещению определяются выражениями 10 и 11 соответственно:

$$\frac{\partial E}{\partial C_{ki}^l} = \text{convD1}(s_k^l, \Delta_i^{l+1}) \quad (53)$$

$$\frac{\partial E}{\partial b_k^l} = \sum \Delta_k^l (n) \quad (54)$$

Таким образом, процесс обучения СНН-ОМ в целом соответствует таковому для двумерных вариантов СНН, однако, существенно зависит от размерности r ядра сверточного слоя. Эта особенность позволяет реализовать обучение варианта СНН-ОМ для поиска требуемых закономерностей временного ряда в определенном временном масштабе.

Для решения задачи охвата различных временных масштабов предлагается использование ансамбля из нескольких СНН-ОМ, отличающихся размерностью r ядра С. Размерность ядер в сверточных слоях СНН-ОМ, входящих в ансамбль, условно представлена, как «малая» - min, «средняя» - mid и «большая» - max.

Размерность ядра min применяется для идентификации шаблонов рабочей нагрузки с быстро меняющимися параметрами.

Такое ядро может применяется для захвата краткосрочных зависимостей в данных рабочей нагрузки, критических для быстрого обнаружения и адаптации быстрых изменений. Примерами таких изменений являются быстрые всплески рабочей нагрузки из-за начала выполнения новых задач или резкие снижения нагрузки после завершения задач.

Размерность ядра mid используется для эффективного захвата временных шаблонов средней размерности. Такое ядро пригодно для обнаружения переходных состояний, таких, где рабочая нагрузка увеличивается или уменьшается умеренно, прежде чем достичь устойчивой фазы.

Размерность ядра max пригодна для захвата обширных временных закономерностей в последовательностях данных. Такое ядро наиболее эффективно фиксирует периодические явления в данных, такие как повторяющиеся пики и спады в рабочей нагрузке с ежедневными или еженедельными рабочими циклами.

Процесс свертки для ядер указанной размерности можно представить выражениями:

$$(y*c_{min})[i]=\sum_{j=0}^{r_{min}-1} y[i+j]\times c_{min}[j] \quad (55)$$

$$(y*c_{mid})[i]=\sum_{j=0}^{r_{mid}-1} y[i+j]\times c_{mid}[j] \quad (56)$$

$$(y*c_{max})[i]=\sum_{j=0}^{r_{larg}-1} y[i+j]\times c_{max}[j], \quad (57)$$

где y - входной вектор признаков, включающий ряд точек данных длиной n, i отмечает конкретную позицию вектора y, где применяется ядро, c_{min}, c_{mid} и c_{max} - представляют малое, среднее и большое сверточные ядра соответственно, характеризующиеся размерностями r_{min}, r_{mid} и r_{max}.

Выходом каждой СНН-ОМ ансамбля являются вероятности шаблонов рабочей нагрузки на анализируемых участках ее временного ряда. Процесс агрегации подразумевает их конкатенацию для формирования входной

последовательности, используемой для решения задачи прогнозирования (рисунок 3.8).

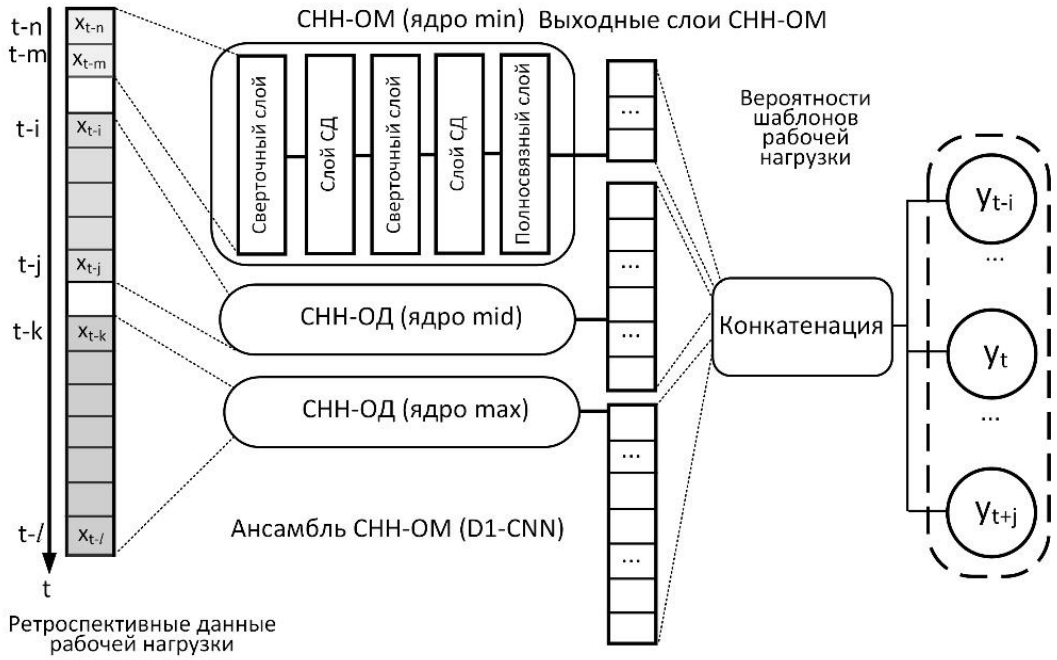


Рис. 3.8 Обобщенная схема предлагаемого ансамбля одномерных сверточных нейронных сетей для решения задачи выявления значимых признаков шаблонов ретроспективных данных рабочей нагрузки

Важной исследовательской задачей является разработка алгоритма обучения предложенных вариантов СНН-ОМ. Поскольку выходные данные ансамбля СНН-ОМ используются в качестве входа МГО прогнозирования, то от качества их обучения зависит насколько эффективно будут выделены значимые признаки шаблонов рабочей нагрузки.

На рисунке 3.9 представлен алгоритм обучения (прямое распространение данных, обратное распространение ошибки) сети СНН-ОМ, входящей в ансамбль СНН-ОМ.

Из рисунка видно, что важным этапом является определение первоначальных параметров обучения, а именно:

– количество M итераций прямого распространения (ПР) данных в слоях СНН-ОМ;

– количество J анализируемых выборок данных (batch);

– количество L сверточных слоев сети;

– количество N нейронов в сверточных слоях сети.

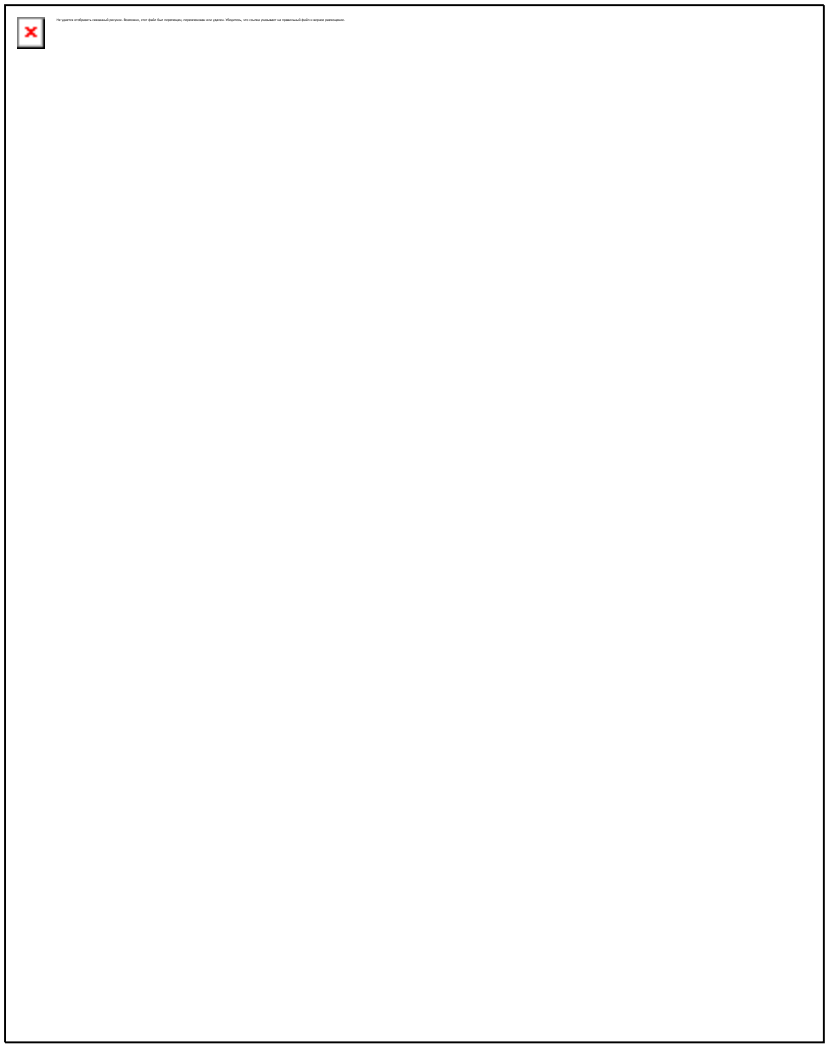


Рис. 3.9 Схема алгоритма обучения сети СНН-ОМ

Обычно указанные параметры подбираются эмпирически.

На этапе ПР реализуются вложенные циклы со счетчиками M и J, обеспечивающие комплексный анализ путем расчета в каждом нейроне каждого сверточного слоя выходных значений y_l^i и значений дельта ошибки Δ_l^k в полносвязном слое для реализации процесса обратного распространения ошибки (ОР).

После реализации процесса ОР выполняется перерасчет весовых коэффициентов $w_{ik}^{l\text{-}1}(t+1)$ и смещений $b_k^l(t+1)$ с целью обновления гиперпараметров сети.

3.2.2. Разработка структуры модели двунаправленной нейронной сети с долгой краткосрочной памятью для решения задачи прогнозирования значений временного ряда

Как было отмечено в п. 3.1, модели МГО в отличие от моделей СВМО (рисунок 3.3), глубокие нейронные сети позволяют анализировать сложные и нелинейные связи, фиксируя иерархические особенности [107]. Например, в [108] экспериментально доказано, что метод ARIMA не в полной мере может охватить нелинейную динамику рабочих нагрузок ВЦОД.

Использование других вариантов СВМО также не лишено недостатков. Так в [109] для прогнозирования рабочей нагрузки в сетях Echo State Networks (ESN) в качестве модуля прогнозирования предложено использование автоэнкодера – нейронной сети специального типа для обучения без учителя. Однако сложность выбора начальных весов делает такой подход мало применимым на практике.

Среди архитектур моделей МГО специальное место занимает класс рекуррентных сетей, предназначенных для моделирования последовательных данных. К ним относятся: собственно рекуррентные нейронные сети (РНС, RNN) и их модификации, сети с долгой краткосрочной памятью (СДКП, LSTN) и их модификации, и сети с управляемым рекуррентным блоком (СУРБ, GRU).

В отличие от нейронных сетей прямого распространения (например, СНН) архитектура рекуррентной сети отличается наличием обратных связей, которые позволяют ей сохранять и использовать данные из предыдущих шагов обработки.

Классическая архитектур РНС содержит трех слоя и механизм циклических связей. Слоями РНС являются:

– входной слой, который получает данные для обработки на каждом временном шаге;

– скрытый слой (слои), который содержит рекуррентные ячейки и реализует функцию памяти. На каждом шаге он на два входа: текущий ввод x(t) и скрытое состояние (активацию) текущего состояния h(t+1) с предыдущего временного шага h(t);

– выходной слой, который генерирует результат (прогноз) для текущего временного шага y(t) на основе текущего скрытого состояния h(t+1).

Динамика функционирования указанных слоев РНС представлена на рисунке 3.10.

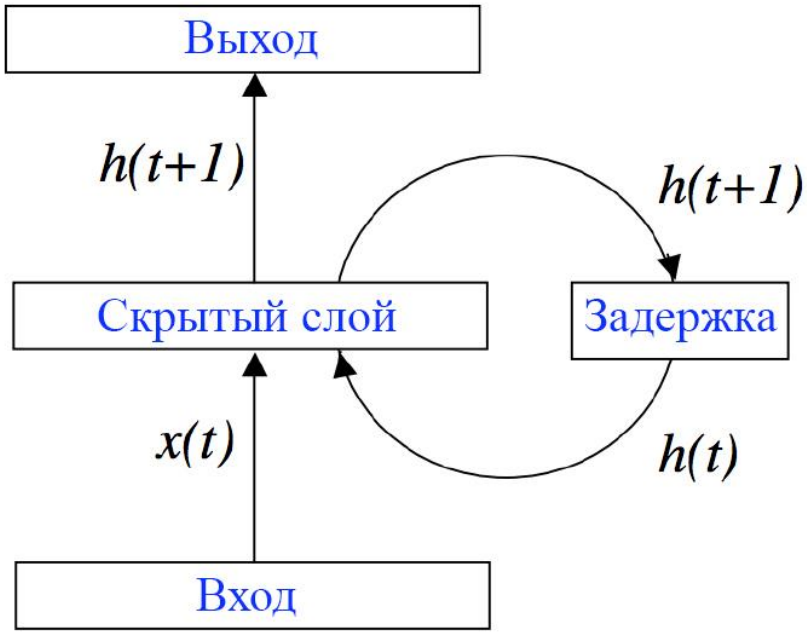


Рис. 3.10 Диаграмма состояний рекуррентной нейронной сети

В общем случае реализация функции памяти в РНС распространяется на небольшие участки последовательности. Для преодоления этого недостатка были разработаны модификации РНС, обеспечивающие решение задачи прогнозирования на длительных участках последовательностей.

Сети СДКП содержат специальные ячейки памяти (memory cells) и три типа вентилей (gates) – входной, забывающий и выходной – которые регулируют поток данных и позволяют запоминать или «забывать» их на длительный срок.

Сети СУРБ являются упрощенным вариантом СДКП с двумя вентилями: обновления и сброса.

Структура рекуррентных блоков указанных видов сетей представлена на рисунке 3.11.

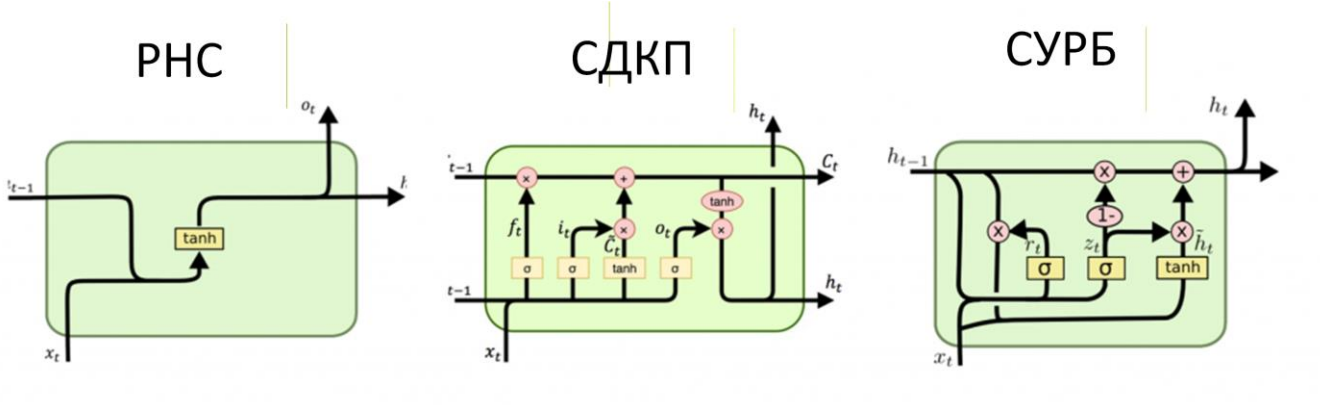


Рис. 3.11 Структура рекуррентного блока сетей РНС, СДКП и СУБР

В рамках исследования для реализации функции прогнозирования была выбрана архитектура сетей СДКП. Особенности ее реализации и принципы процесса обучения представлены в [110]. На рисунке 3.12 представлена взаимосвязь рекуррентных блоков СДКП для получения будущих значений последовательсти на основе ее предыдущих значений.

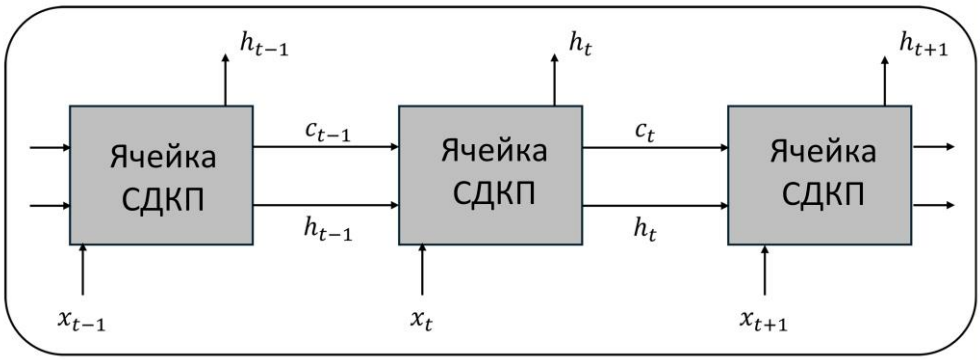


Рис. 3.12 Взаимосвязь рекуррентных блоков в сети СДКП

Анализ источников в разных предметных областях [111-113] показал, что в рамках прогнозирования значений временных рядов используется двунаправленная модификация СДКП – СДКП-ДН (англ. Bi-LSTM).

Сеть СДКП-ДН является модифицированным вариантом классической СДКП сети. В сети СДКП каждая ячейка получает входные данные, зависящие

от вычислений, выполненных в ячейке (ячейках) на предыдущих временных шагах. В отличие от СДКП архитектура сети СДКП-ДН характеризуется двунаправленным потоком информации, фактически агрегируя две СДКП, каждая из которых обрабатывает данные об одном из направлений их распространения. При этом в СДКП-ДН выходы обеих сетей объединяются на выходном слое.

Обобщенно это представляется выражением:

$$\begin{cases} h_f = \text{LSTM}(x_i, \text{h}_{\text{f-1}}) \\ h_b = \text{LSTM}(x_b, \text{h}_{\text{b-1}}), \\ h_t = \text{w}_t h_f + \text{v}_t h_b + \text{b}_t \end{cases} \quad (58)$$

где x_i – входные данные, h_f– состояние скрытого слоя при прямом проходе, h_b – состояние скрытого слоя при обратном проходе, h_t – состояние скрытого слоя в -й позиции ряда, w_t – выходной вес скрытого слоя при прямом проходе, v_t – выходной вес скрытого слоя при обратном проходе, b_t – величина ошибки.

Выходом сети СДКП-ДН является вектор $[y_{\text{t-i}},\ldots,y_t,\ldots,y_{\text{t+j}}]$, определяющий представление значений временного ряда в моменты времени, предшествующие и последующие значению в i-й позиции ряда. При этом последующие значения $(y_{\text{t+1}},\ldots,y_{\text{t+j}})$ являются прогнозными.

В общем виде архитектура СДКП-ДН представлена на рисунке 3.13.

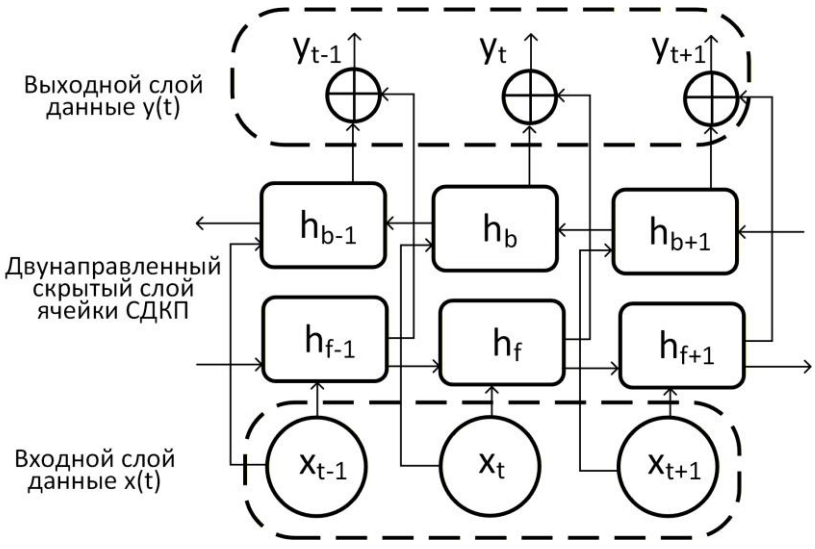


Рис. 3.13 Обобщенная архитектура сети СДКП-ДН

3.3. Разработка гибридной архитектуры модели глубокого обучения и алгоритма прогнозирования рабочей нагрузки ВЦОД

В обобщенном виде предлагаемая гибридная модель МГО представлена на рисунке 3.14.

На рисунке представлен ансамбль трех СНН-ОМ, отличающихся размерами ядер (c1,c2,c3), выход которого – реализация временного ряда $X'=\{x'_1,x'_2,...,x'_n\}$ является входом сети СДКП-ДН, которая содержит: два скрытых двунаправленных слоя ячеек, attention-слой, объединяющий найденные СНН-ОМ зависимости и выбирающий критические важные из них путем перераспределения весовых коэффициентов h_n, и полносвязный слой, формирующий выходные данные прогноза – ряд значений $Y=\{y_1,y_2,...,y_n\}$.

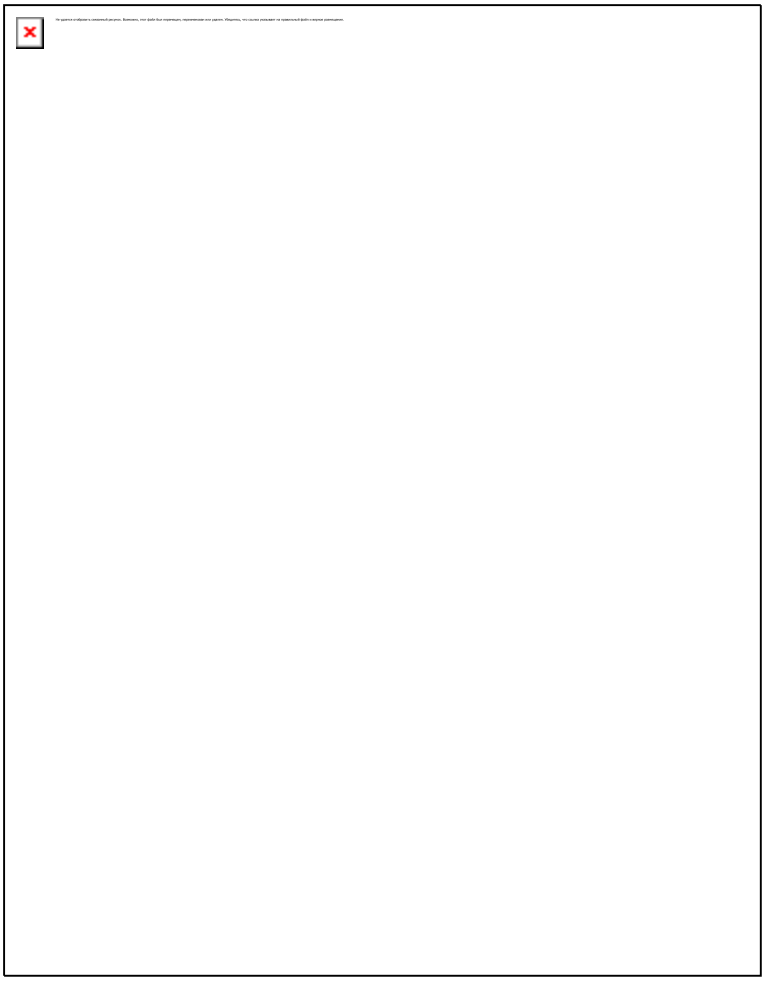


Рис 3.14 Обобщенная схема гибридной (СНН-ОМ и СДКП-ДН) модели глубокого обучения для прогнозирования рабочей нагрузки ВЦОД

Использование выходных данных ансамбля сетей СНН-ОМ (рисунок 3.8), содержащих найденные зависимости временного ряда ретроспективных данных, определяющие те или иные шаблоны рабочей нагрузки в качестве входной последовательности сети СДКП-ДН, позволяет определять вероятности появления этих зависимостей на участках временного ряда t,t+1,...,t+j. Глубина получения прогнозных значений зависит от структуры скрытых слоев сети СДКП-ДН (рисунок 3.13).

Таким образом, для решения задачи прогнозирования рабочей нагрузки ВЦОД предлагается использовать гибридный вариант МГО, в состав которого входит ансамбль СНН-ОМ, каждая сеть в котором отличается размерностью ядра, а также сеть СДКП-ДН, обобщающая результат конкатенации выходов ансамбля СНН-ОМ. Вариантом такой гибридной модели может выступать каскад сетей СНН-ОМ и СДКП-ДН, каждый уровень в котором настроен на получение прогноза заданного вида зависимостей рабочей нагрузки, которая характерна, например, в ВЦОД общего назначения, обеспечивающих поддержку широкого спектра пользовательских запросов.

Пример реализации такой каскадной гибридной модели представлен на рисунке 3.15. Из рисунка видно, что каскадными уровнями являются:

– взаимодействующие по выходам ансамбли сетей СНН-ОМ и СНН-ОМ’;

– взаимодействующие по выходу сети СДКП-ДН и СДКП-ДН’, входом которых является конкатенируемый выход каскада СНН-ОМ.

При этом различная размерность ядра в каждой из сетей СНН-ОМ обеспечивает выделение значимых признаков на определенном масштабе временного ряда, а конкатенация выходов этих сетей обеспечивает обобщение одних и тех же зависимостей, найденных разными сетями и выделение зависимостей, специфичных для временного масштаба каждой сети.

Эта особенность позволяет варьировать глубину прогноза в каскаде сетей СДКП-ДН и СДКП-ДН’, которые отличаются количеством скрытых слоев, что обеспечивает различный временной охват прогноза. Естественно, что, в зависимости от задач прогнозирования рабочей нагрузки, количество сетей СДКП-ДН в каскаде

может быть больше двух. Однако следует учитывать, что простое масштабирование каскада сетей СДКП-ДН не ведет к увеличению глубины прогноза. Дополнительным условием является подбор соответствующих весов w_t.

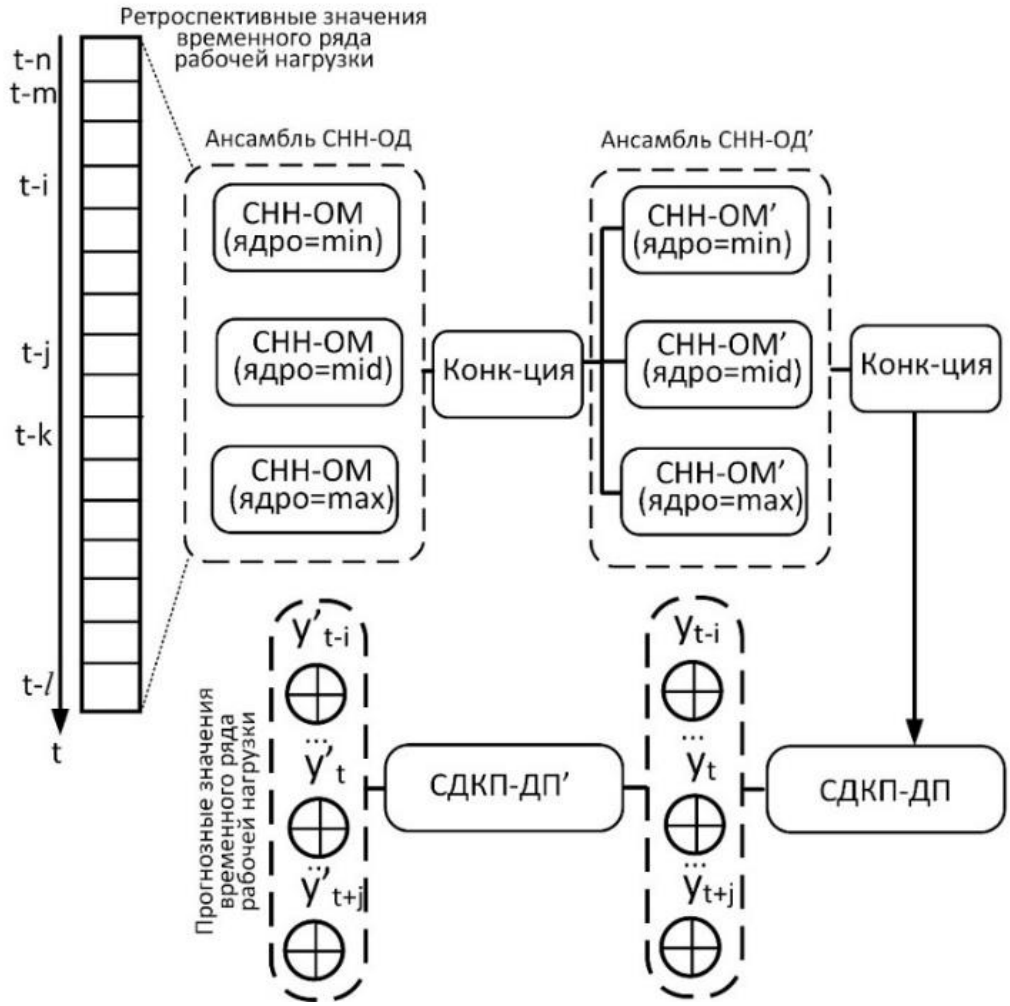


Рис. 3.15 Схема предлагаемой каскадной гибридной модели глубокого обучения для прогнозирования рабочей нагрузки ВЦОД на основе ее ретроспективных данных

3.3.1. Разработка схемы гибридного алгоритма прогнозирования рабочей нагрузки

Поскольку в предложенном варианте гибридной МГО анализ каждой из КМФ функций выполняется ансамблем СНН-ОМ, каждый экземпляр которого отличается размером ядра c_n, в разрабатываемом обобщенном алгоритме гибридной МГО

предлагается этап параллельной обработки $СНН_n^{вых} \leftarrow СНН_n(X,c_n)$ данных X, включенный во вложенные циклы со счетчиками:

– K – количества эпох обучения;

– M – количества обучающих выборок (batch).

Конкатенированный результат $СНН^{вых}$ функционирования ансамбля СНН-ОМ – $СДКП_2^{вых} \leftarrow СДКП_1^{вых} \leftarrow СНН^{вых}$ поступает в каскад двух сетей СДКП-ДН, отличающихся числом скрытых слоев для решения задачи прогнозирования в разных временных масштабах.

Схема алгоритма функционирования предложенной гибридной МГО представлена на рисунке 3.16.

Из рисунка видно, что сложность разработанного алгоритма обусловлена наличием этапа параллельной обработки данных ансамблем сетей СНН-ОМ. В силу того, что этот этап включен во вложенные циклы общего назначения систем глубокого обучения (перебор значений числа эпох обучения и размера блоков обучающей (тестовой) выборок), его особенность – возможность возникновения эффекта гонок отдельных параллельных процедур, может оказывать влияние на общее время выполнения алгоритма. Вариантом решения указанной проблемы является включение в этап параллельной обработки механизмов синхронизации по времени. В простейшем случае возможно использование последовательного буфера, обеспечивающего заполнение выходными данными $СНН_n^{вых}$ по мере их появления.

Такое решение возможно, поскольку последующая операция конкатенации $СНН^{вых}=cat(СНН_1^{вых},...,СНН_n^{вых})$ не предполагает определенного порядка следования выходных данных $СНН_n^{вых}$.

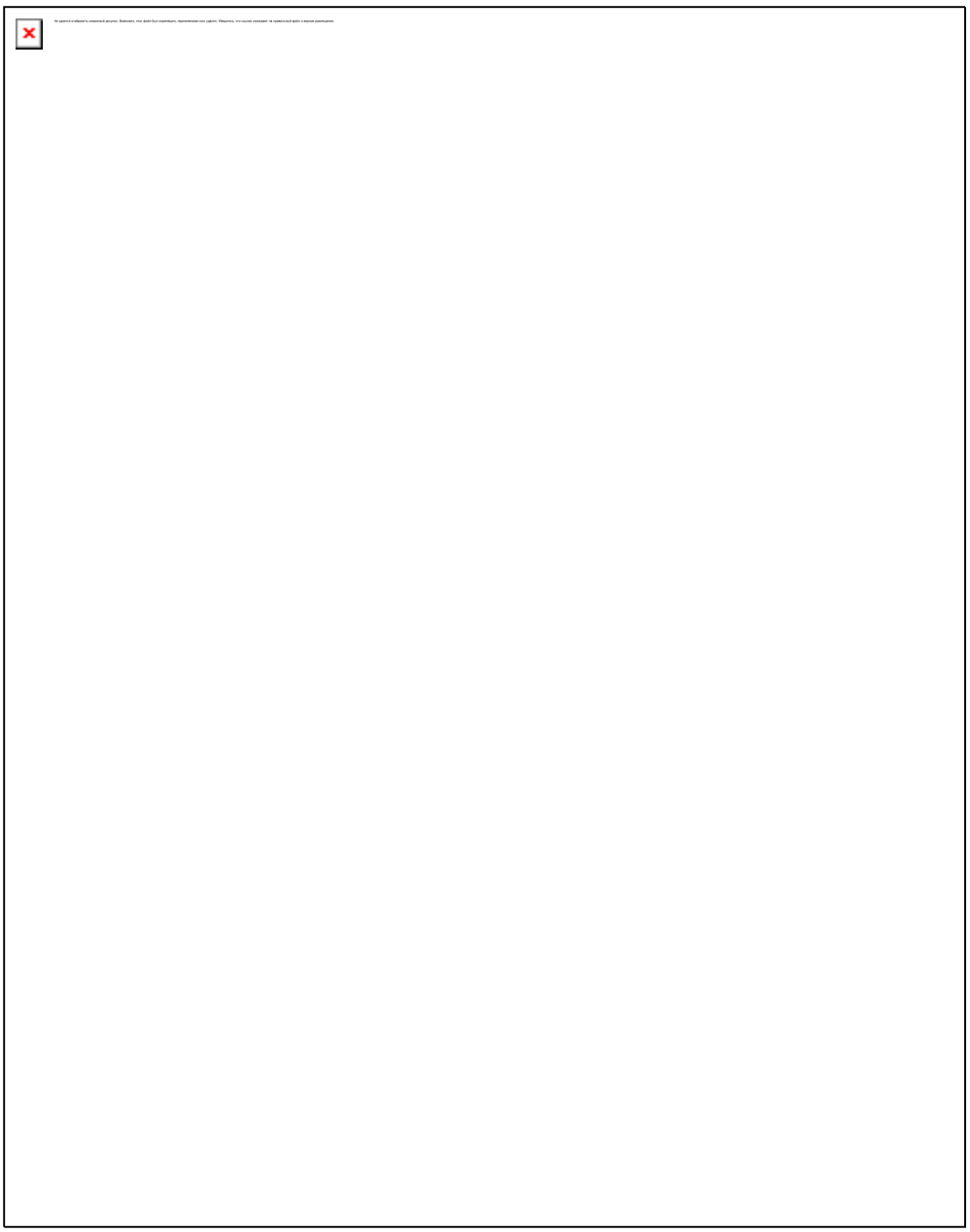


Рис. 3.16 Схема алгоритма гибридной модели глубокого обучения для прогнозирования рабочей нагрузки ВЦОД

Предложенный алгоритм гибридной МГО является одним из вариантов, поскольку решение вопроса о количестве СНН-ОМ в ансамбле сетей СНН, а также количество СДКП-ДН в каскаде сетей СДКП существенно зависит от характеристик обрабатываемого временного ряда ретроспективных данных рабочей нагрузки, которые, в свою очередь зависят от особенностей организации ВЦОД, а также особенностей их потребительской нагрузки.

Таким образом, предложенный алгоритм (рисунок 3.16) носит обобщенный характер и требует детализации в процессе разработки конкретного экземпляра системы прогнозирования рабочей нагрузки.

3.4. Выводы по главе

В главе представлен разработанный гибридный алгоритм прогнозирования рабочей нагрузки ВЦОД.

Выполнен анализ моделей и методов прогнозирования временных рядов, среди которых выделены статистические вероятностные методы и модели, а также модели глубокого обучения.

Рассмотрены особенности процесса прогнозирования таких статистических методов, как ARIMA и его вариантов и методов экспоненциального сглаживания. Определены их ограничения, в частности, применительно к формированию прогноза для нелинейных временных рядов с высокой нестационарностью, к которым относятся временные ряды показателей рабочей нагрузки ВЦОД.

Исследованы схемы организации однонаправленной сверточной нейронной сети для реализации процесса распознавания шаблонов рабочей нагрузки в вариантах временных рядов ее показателей, а также двунаправленной сети с долгой краткосрочной памятью для реализации функции прогнозирования значений временного ряда

Предложены: ансамблевая структура однонаправленной сверточной нейронной сети с ядрами разной размерности и адаптивным назначением коэффициента субдискретизации, каскадная схема двунаправленной сети с долгой краткосрочной памятью, отличающаяся разным числом скрытых слоев к вкаждом элементе каскада, а также подход к их интеграции по данным. Для предложенной схемы разработана алгоритмическая реализация.

Таким образом, разработан гибридный алгоритм прогнозирования временного ряда рабочей нагрузки для системы глубокого обучения, который обеспечивает получение разномасштабных прогнозных значений временного ряда рабочей нагрузки.

Глава 4. Разработка архитектуры системы прогнозирования рабочей нагрузки виртуализированного центра обработки данных на основе ретроспективной информации о загрузке его вычислительных ресурсов с учетом ее зашумления

4.1. Структура программного комплекса системы прогнозирования рабочей нагрузки виртуализированного центра обработки данных

Предложенный в главе 1 подход к решению задачи прогнозирования рабочей нагрузки ВЦОД, основан на использовании временных рядов ее показателей из базы ретроспективных данных системы мониторинга, а также разработанные в глава 2 и 3 алгоритмы:

– снижения присущих временным рядам рабочей нагрузки факторов зашумления за счет решения задачи комбинированной декомпозиции временного ряда на множество эмпирических и вариационных мод (КМФ функций);

– выделения значимых признаков шаблонов рабочей нагрузки в декомпозированном временно ряде ее показателей за счет применения ансамбля одномерных сверточных нейронных сетей (СНН-ОМ) с различной размерностью ядер;

– получения прогнозных значений полученного варианта декомпозированного временного ряда с использованием каскада двунаправленных сетей с долгой краткосрочной памятью (СДКП-ДН).

Важной исследовательской задачей является оценивание качества полученных решений и эффективности процесса прогнозирования рабочей нагрузки ВЦОД.

Для решения этой задачи в рамках исследования был разработан вариант архитектуры системы прогнозирования рабочей нагрузки, основанный на разработанных алгоритмических решениях.

В обобщенном виде предлагаемая архитектура системы прогнозирования представлена на рисунке 4.1, который отражает базовые компоненты архитектуры, а также входные и выходные данные этих компонентов.

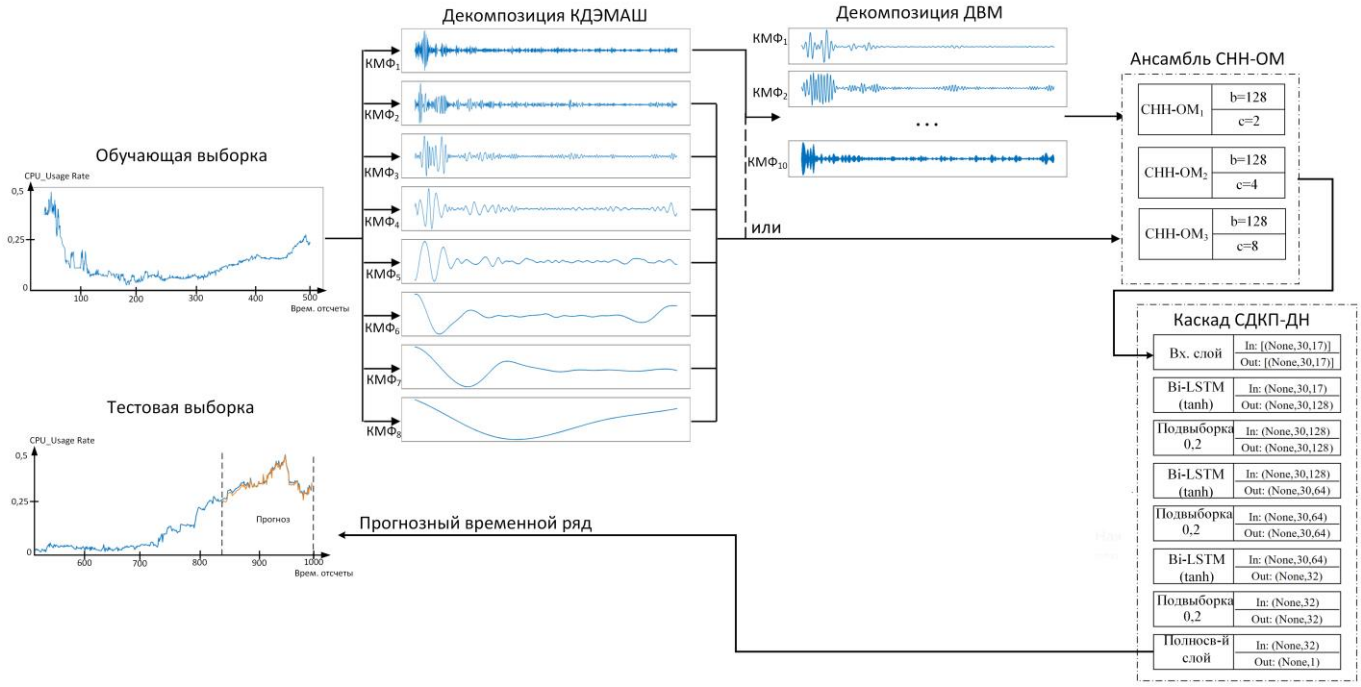


Рис. 4.1 Обобщенная архитектура системы прогнозирования рабочей нагрузки ВЦОД

Из рисунка видно, что в целом предлагаемая архитектура системы прогнозирования рабочей нагрузки соответствует множеству разработанных в главах 2 и 3 частных алгоритмов ее функционирования и состоит из двух основных компонентов:

1. Подсистемы декомпозиции временного ряда рабочей нагрузи на комбинированное множество КМФ функций, а именно: КМФ/КДЭМАШ и КМФ/ДВМ, что соответствует разработанной модели модовой декомпозиции временного ряда рабочей нагрузки (рисунок 1.30) и предложенному комплексному алгоритму предварительной обработки значений параметров временного ряда рабочей нагрузки ВЦОД (рисунок 3.1).

2. Подсистемы прогнозирования рабочей нагрузки на основе ансамбля сетей СНН-ОМ, выделяющего значимые компоненты рабочей нагрузки и каскада сетей СДКП-ДН для формирования прогнозных значений целевого временного ряда рабочей нагрузки ВЦОД.

Как было рассмотрено в п. 3.2 (рисунок 3.4) использование в предлагаемом решении моделей МГО требует реализации процессов:

– обучения моделей на основе некоторого подмножества временных рядов, являющегося обучающей выборкой;

– валидации структурно-параметрических характеристик моделей на основе некоторого подмножества временных рядов, являющегося тестовой выборкой.

Таким образом, входом предлагаемой архитектуры систем прогнозирования на этапе ее обучения является обучающая выборка временных рядов рабочей нагрузки, на этапе ее валидации – тестовая выборка временных рядов рабочей нагрузки, а на этапе эксплуатации – исходные временные ряды, подаваемые из базы ретроспективных данных рабочей нагрузки ВЦОД.

Предлагаемая архитектура системы прогнозирования требует программной конкретизации, а следовательно предварительного решения задач выбора программных фреймворков ее подсистем.

4.2. Выбор программных фреймворков системы прогнозирования рабочей нагрузки ВЦОД

Анализ источников, посвященных программной реализации моделей модовой декомпозиции временных рядов временных рядов [114-117] позволил выделить два используемых исследователями подхода:

– разработка частных программных решений, обеспечивающих процесс декомпозиции на эмпирические или вариационные моды;

– использование существующих фреймворков декомпозиции общего назначения.

Очевидно, что частные программные решения реализуют специализированный, применительно к исследованиям их разработчиков процесс декомпозиции. Кроме того, в предлагаемом комплексном алгоритме декомпозиции (рисунок 3.1) процедуры декомпозиции на эмпирические и вариационные моды взаимодействуют по входам, что для частных программных решений требует разработки дополнительных программных модулей предобразования данных.

Таким образом, в качестве программной основы подсистемы предварительной обработки временного ряда предлагается использование существующих программных фреймворков общего назначения.

4.2.1. Выбор фреймворка для модовой декомпозиции временного ряда рабочей нагрузки ВЦОД

Исследование функциональных возможностей таких систем, как MathCAD, Wolfram Mathematica и MatLab выявлил следующее:

– в составе систем MathCAD и Wolfram Mathematica отсутствуют модули модовой декомпозиции сигналов, хотя существующие в них специализированные скриптовые языки позволяют реализовать численные подходы методов КДЭМАШ и ДВМ;

– в составе системы MatLab имеется подсистема MatLab Signal Processing [117], обеспечивающая не только реализацию методов КДЭМАШ и ДВМ, но также, благодаря наличию специализированных API-функций, взаимодействие этих методов по данным.

На рисунке 4.2 представлен внешний интерфейса подсистемы MatLab Signal Processing.

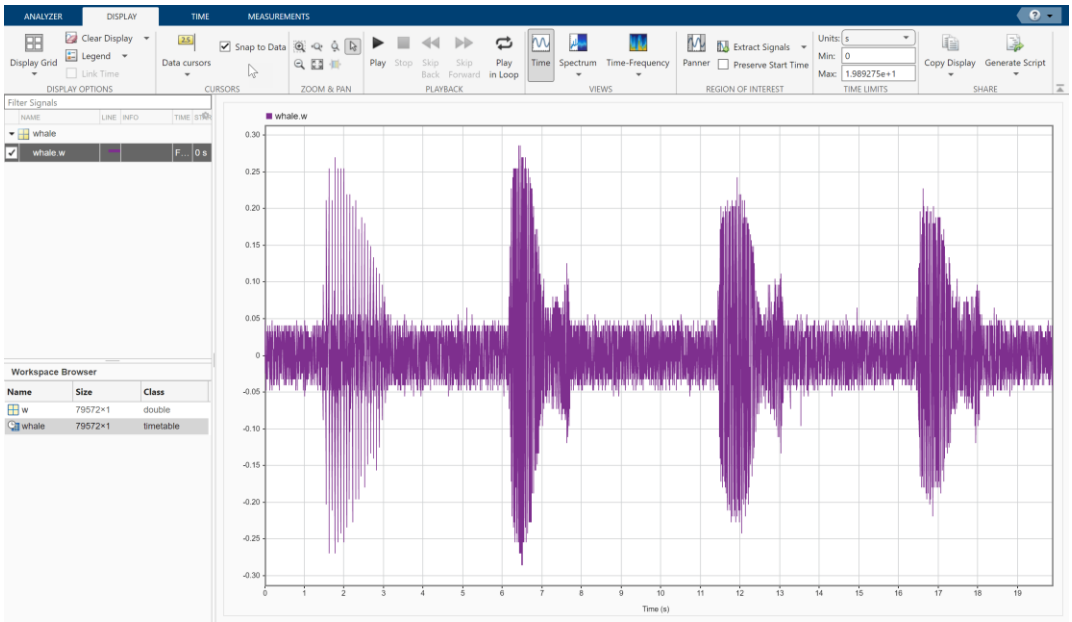


Рис. 4.2 Интерфейс подсистемы MatLab Signal Processing

Пример реализации метода КДЭМАШ в подсистеме MatLab Signal Processing представлен на рисунке 4.3.

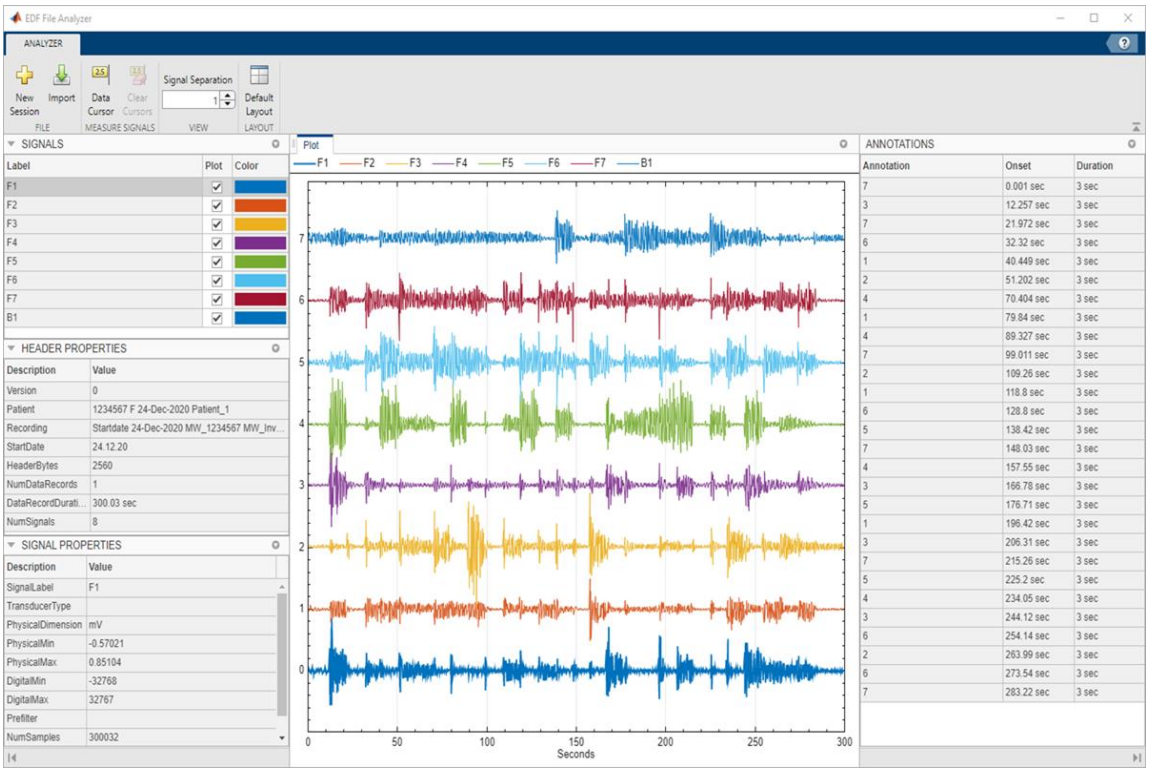


Рис. 4.3 Результат декомпозиции КДЭМАШ в подсистеме MatLab Signal Processing

Для заданного разработанным комплексным алгоритмов модовой декомпозиции взаимодействия предопределенных процедур была выбрана среда Simulink [118] – среда блок-диаграмм для многодоменного моделирования, поддерживающая системное проектирование, моделирование, автоматическую генерацию кода. Среда Simulink предоставляет графический редактор, настраиваемые библиотеки блоков и компонентов для моделирования и формирования динамических систем. Она интегрирована с MatLab, что позволяет внедрять алгоритмы, реализованные в MatLab в модели Simulink и выполнять обратный экспортирт результатов моделирования в MatLab. Интерфейс среды Simulink представлен на рисунке 4.4.

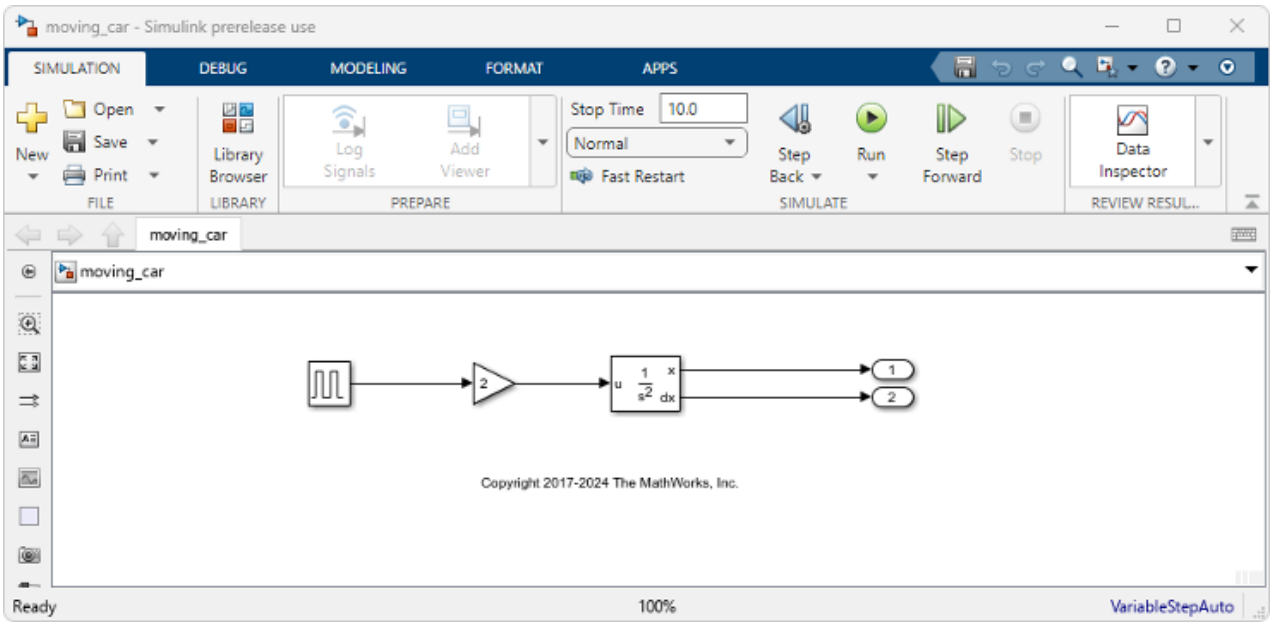


Рис. 4.4 Интерфейс среды Simulink

4.2.2. Выбор фреймворка для разработки моделей глубокого обучения системы прогнозирования рабочей нагрузки ВЦОД

В отличие от подсистемы предварительной обработки временных рядов рабочей нагрузки с целью снижения влияния факторов их зашумления, для формирования моделей глубокого обучения существует достаточно много фреймворков, как коммерческого уровня, так и проектов с открытым исходным кодом.

Анализ источников, посвященных программной реализации сетей СНН-ОМ, а также СДКП-ДН показал, что в качестве основных разработчики используют следующие фреймворки:

– Tensorflow [119] – среда разработки с развитой базой библиотек моделей МГО, а также языком программирования на основе Python и интерфейсами API, позволяющими экспортировать полученное решение для различного применения;

– Keras [120] – фреймворк разработки МГО, ориентированный на быстроту создания, отладки и интеграции моделей МГО;

– PyTorch [121] – фреймворк машинного обучения с открытым исходным кодом, разработанный исследовательской лабораторией FAIR, и обеспечивающий разработку и быстрое прототипирование моделей МГО;

– Skilit Learn [122] – развитый фреймворк машинного обучения поддерживающий не только разработку моделей МГО, но и разработку моделей СВМО. В составе Skilit Learn имеются библиотеки поддержки таких методов классификации, кластеризации и регрессии, как машина опорных векторов, К-средних, ближайших соседей, случайного леса, логистической регрессии и др.

Рассмотренные фреймворки обладают достаточной функциональностью для реализации разработанного в ходе исследования гибридного алгоритма гибридной модели глубокого обучения для прогнозирования рабочей нагрузки ВЦОД, однако наличие в предлагаемом решении модуля предварительной обработки временных рядов рабочей нагрузки требует дополнительной реализации интерфейса его сопряжения с разрабатываемым модулем прогнозирования рабочей нагрузки, что усложняет процесс разработки и может привести к ошибкам в данных в ходе их преобразования.

В связи с этим было приято решение разработку представленных в главе 3 моделей и алгоритмов глубокого обучения выполнить в среде MatLab, а именно в ее специализированном фреймворке MatLab Deep Learninig Toolbox [123]. Эта среда предоставляет функции, приложения для проектирования, реализации и моделирования нейронных сетей различной архитектуры и обеспечивает основу для их создания, использования и визуализации результатов функционирования. Также, как и среда MatLab Digital Signal Processing Toolbox, эта среда совместима с функциями конструктора MatLab Simulink, что обеспечивает интеграцию по данным и командам управления. Пример интерфейса редактора МГО MatLab Deep Network Designer app из этой среды приведен на рисунке 4.5.

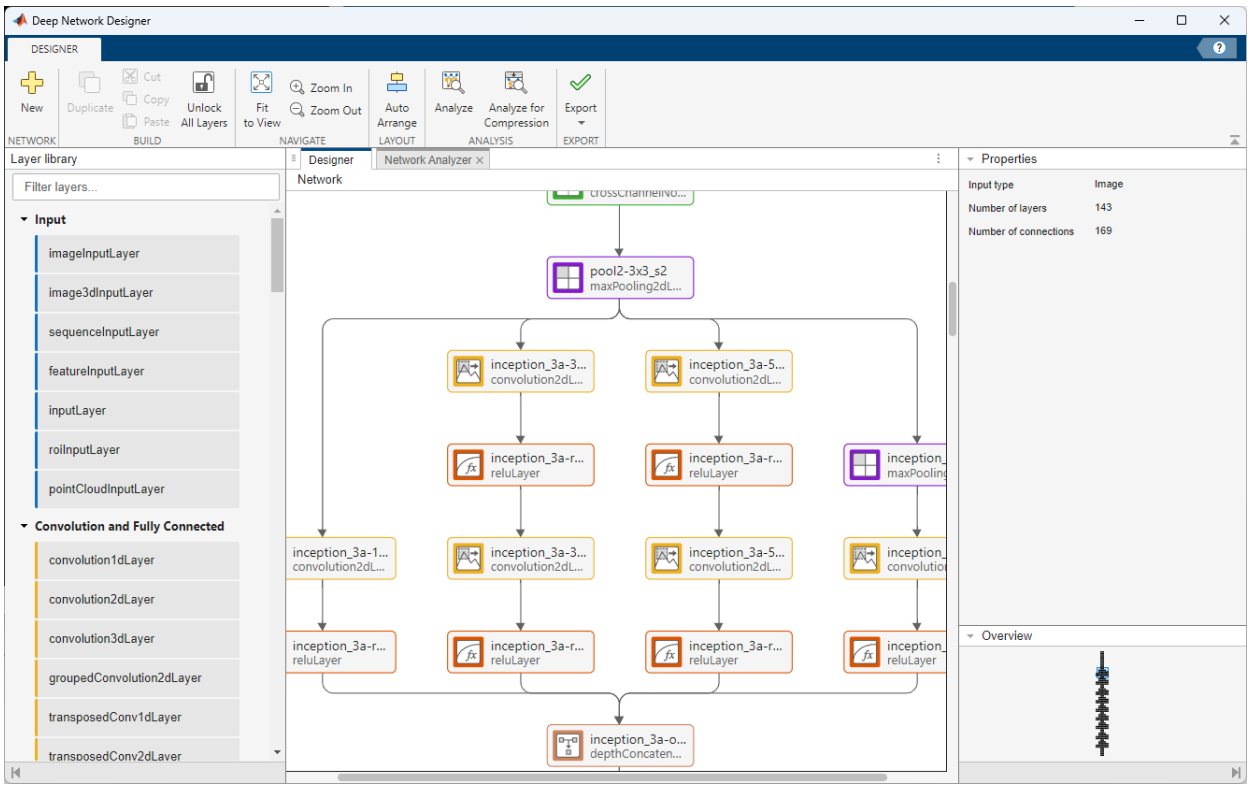


Рис. 4.5 Интерфейс модуля Deep Network Designer среды MatLab Deep Learninig Toolbox

4.3. Разработка структуры программного комплекса системы прогнозирования рабочей нагрузки ВЦОД

Выбранные в п. 4.1 фреймворки подсистем предварительной обработки временных рядов, а также прогнозирования рабочей нагрузки на основе разработанных в главах 2 и 3 соответствующих алгоритмов позволил разработать структуру программного комплекса, поддерживающего функции указанных подсистем. Его обобщенная структурная схема представлена на рисунке 4.6.

Из рисунка 4.6 видно, что предлагаемая структура прогнозирования рабочей нагрузки ВЦОД реализована в интегрированной среде MatLab. Ее составными компонентами являются:

– блок декомпозиции (предварительной обработки временных рядов), реализованный на базе Digital Signal Processing Toolbox и выполняющий функции формирования множества КМФ функций методами КДЭМАШ и ДВМ;

– блок моделей глубокого обучения (прогнозирования), реализованный на базе Deep Learning Toolbox и выполняющий функции обучения и тестирование ансамбля сетей СНН-ОМ, каскада сетей СДКП-ДН, а также их взаимодействия по входным и выходным данным;

– блок интеграции на базе MatLab Simulink, обеспечивающий: интерфейс сопряжения блоков декомпозиции и МГО, компарацию получаемых данных и визуализацию их сравнительной оценки, а также реализующий общую схему выполнения компонентов разработанной схемы в соответствии с разработанными алгоритмами;

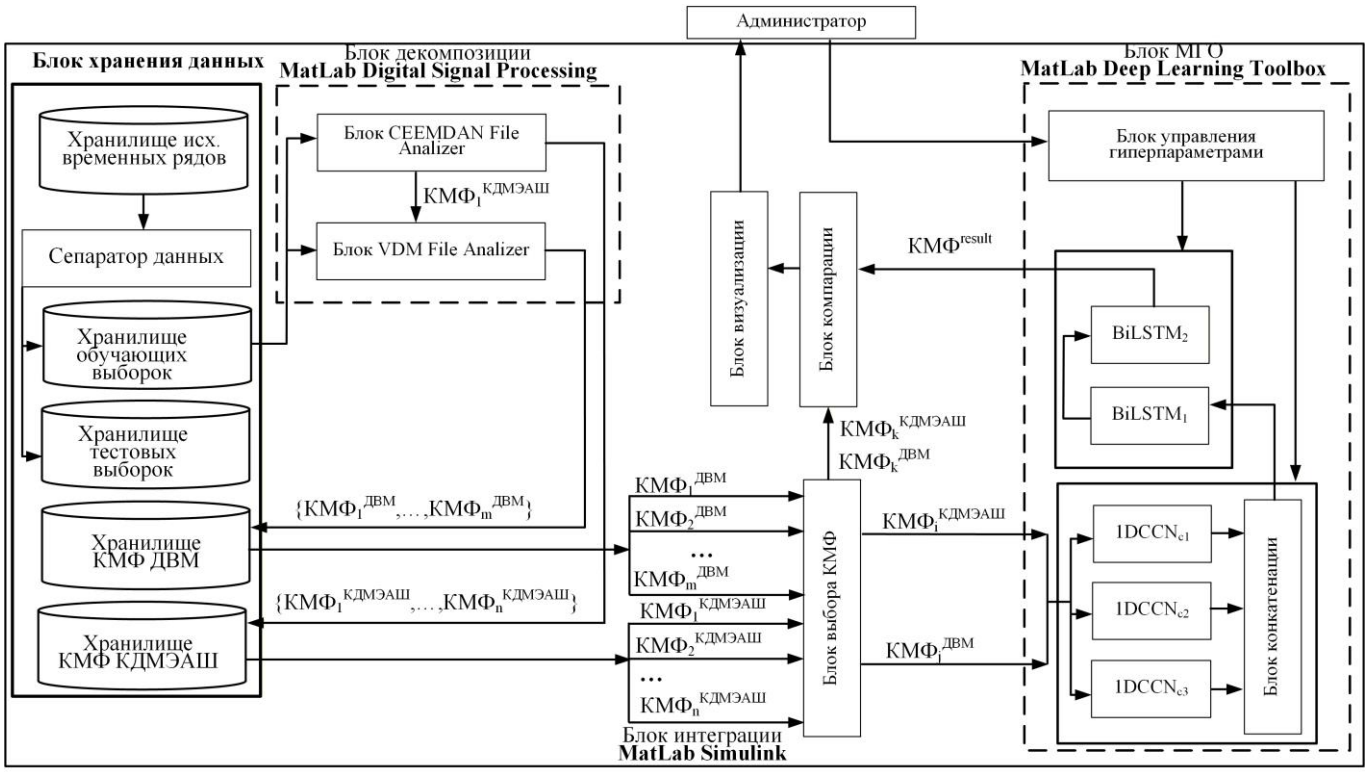


Рис. 4.6 Структура программного комплекса системы прогнозирования рабочей нагрузки ВЦОД

– блок хранения данных на базе файловой системы, реализующий следующие типы хранилищ: хранилище исходных временных рядов, получаемых из базы ретроспективных данных рабочей нагрузки ВЦОД; хранилище обучающих и тестовых выборок, содержащие нормализованные варианты временных рядов, предназначенных для декомпозиции на моды и использования в процессе обучения и валидации моделей МГО блока МГО в соответствии с предложенной ниже схемой

кросс-валидации; временные хранилища КМФ функций, получаемых с выходов модулей КДЭМАШ (CEEMDAN File Analyzer) и ДВМ (VDM File Analyzer) для их последующей нормализации и преобразования в обучающие и тестовые наборы данных.

Достоинством предлагаемой структуры программного комплекса системы прогнозирования рабочей нагрузки ВЦОД является реализация его компонентов в рамках единой интегрированной среды MatLab, что обеспечивает единый процесс разработки и отладки его функциональных блоков по данным и управляющим сигналам.

0.4. Экспериментальное оценивание предложенного решения

Разработанные модель, алгоритмы и архитектура системы прогнозирования рабочей нагрузки ВЦОД в условиях зашумления значений ее ретроспективных данных требует оценивания их качества, а также эффективности реализуемых ими процессов снижения влияния факторов зашумления значений исходного временного ряда и получения прогнозных значений целевого временного ряда.

В силу особенностей реализации современных ВЦОД, функционирующих в непрерывном режиме, подход, основанный на натурном экспериментальном оценивании, является нереализуемым. Таким образом, вариантом экспериментального оценивания является проведение сравнительного полунатурного эксперимента на стенде, максимально близко имитирующим функциональность подсистем реального ВЦОД, выполняющих функции:

– сохранения и управления базой ретроспективных данных рабочей нагрузки;

– прогнозирования рабочей нагрузки существующим в настоящий момент способами.

Для эффективного решения подобной задачи оценивания предшествующее ей решение следующих частных задач:

– выбор существующей системы управления рабочей нагрузкой, поддерживающей развитую базу ретроспективных данных, для использования в качестве альтернативного решения;

– разработка экспериментального стенда для проведения сравнительного эксперимента;

– формирование на данных указанной базы обучающей (тренировочной) и тестовой (валидационной) выборок временных рядов для обучения существующей и исследуемой моделей прогнозирования рабочей нагрузки.

4.4.1. Разработка схемы экспериментального стенда системы прогнозирования рабочей нагрузки ВЦОД

Анализ источников, посвященных реализациям подсистем прогнозирования рабочей нагрузки [124-127] позволил определить, что базы ретроспективных данных рабочей нагрузки и соответствующие алгоритмы прогнозирования, имеющиеся в открытом доступе, поддерживаются двумя системами прогнозирования реальных ВЦОД:

– Google Cluster Workload Traces [124, 125];

– Alibaba Workload Miner (AWM) [126, 127].

Исследование источников, посвященных методам прогнозирования указанных подсистем позволил выделить следующее:

– методами прогнозирования, реализованным в действующих вариантах подсистем и доступными для изучения и использования, являются ARIMA и FARIMA в Google Cluster, и ARIMA и SARIMA в Alibaba Workload Miner. При этом в Google Cluster реализована комбинированная схема с предварительным выделением шаблонов рабочей нагрузки на основе метода классификации Машина опорных векторов (SVM);

– в рамках исследовательской деятельности в этих подсистемах тестируются перспективные методы и алгоритмы прогнозирования рабочей нагрузки, основанные на моделях МГО, а также ансамблевых методах (Random Forest, Gradient Boosting).

Таким образом, в силу функциональной близости предлагаемому в данном исследовании решению, в качестве альтернативного решения при проведении сравнительного эксперимента была выдрана подсистема прогнозирования рабочей нагрузки Google Cluster, базирующаяся на комбинированном метода SVM-ARIMA (SVM-FARIMA).

На рисунке 4.7 представлена схема предлагаемого экспериментального стенда, детализирующая разработанную архитектуру системы прогнозирования рабочей нагрузки (рисунок 4.1) применительно к проведению сравнительного эксперимента.

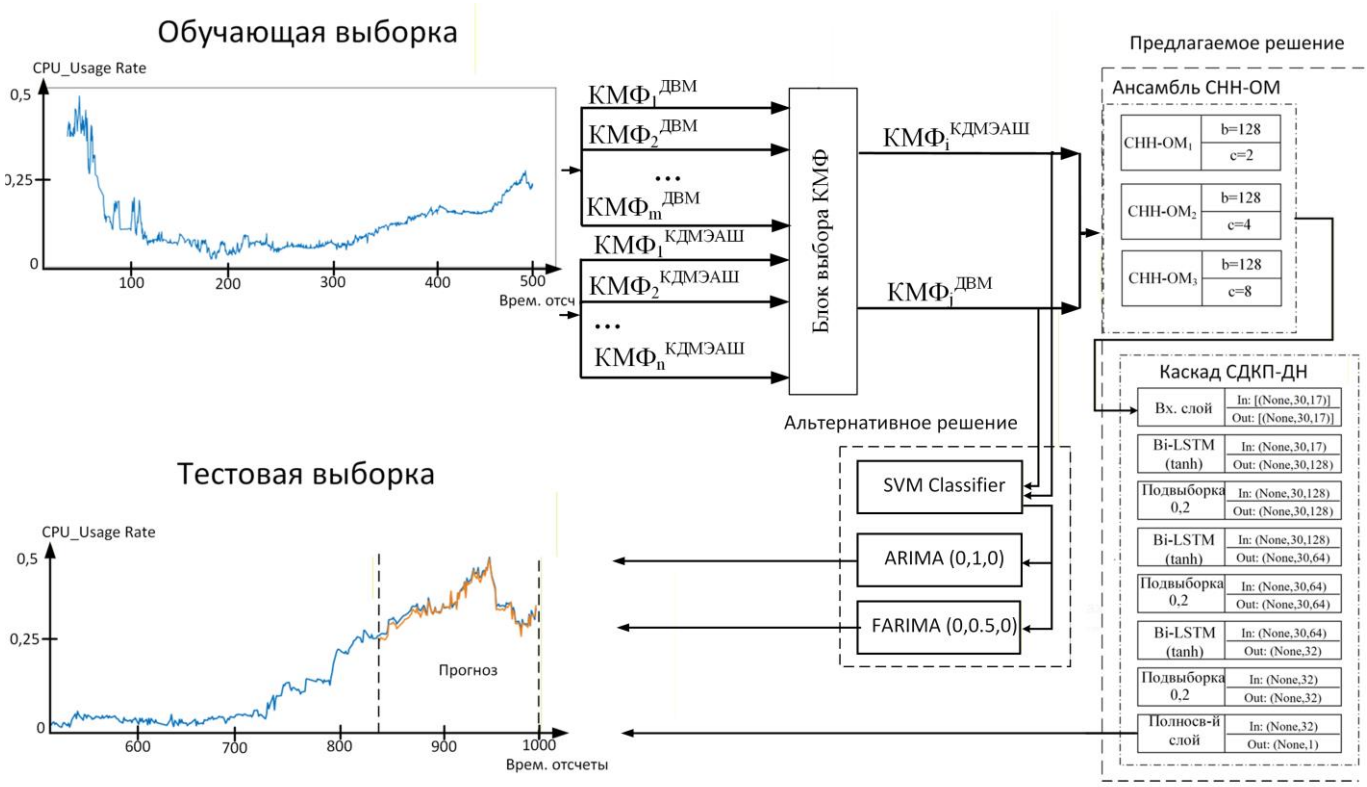


Рис. 4.7 Структура экспериментального стенда системы прогнозирования рабочей нагрузки ВЦОД

4.4.2. Подход к формированию обучающей и тестовой выборок данных для сравнительного эксперимента по оцениванию эффективности системы прогнозирования рабочей нагрузки ВЦОД

Традиционно подходы к прогнозированию временных рядов на основе моделей СВМО и МГО включают:

– обучение модели на подмножестве данных (обучающий (тренировочный) набор данных);

– прогнозирование временного ряда на масштабе (горизонте) фиксированного размера;

– расчет эффективности функционирования модели по выбранным метрикам эффективности на тестовом (валидационном) наборе данных.

Очевидно, что для временных рядов, особенно с большим количеством значений, такой подход требует доработки, поскольку априори предполагает, что распределение значений временного ряда не меняется с течением времени (период стационарности). Очевидно, что временные ряды показателей рабочей нагрузки не соответствуют этому условию, поскольку на разных масштабах временного ряда значения показателей могут формировать участки нестационарности. К таким масштабам относятся как сезонные или цикличные (межсезонные) участки временного ряда, так и участки небольшой масштабности, отражающие возникновение нерегулярных (случайных) событий. В целом указанные особенности временного ряда рабочей нагрузки влияют на его результирующую составляющую и отражаются показателем тренда (рисунок 4.8).

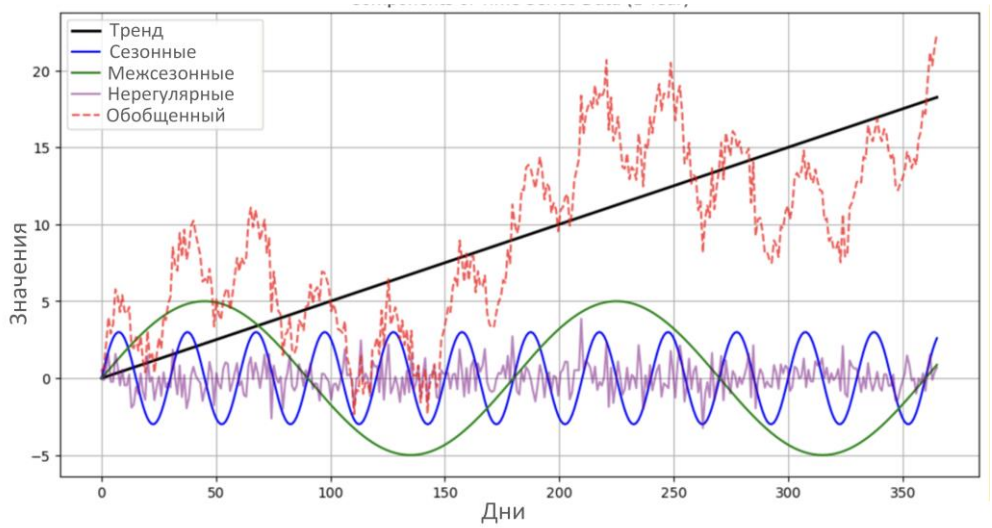


Рис. 4.8 Участки стационарности и нестационарности условного временного ряда продолжительностью 1 год

Таким образом, при формировании обучающей и тестовой выборок временного ряда, необходимо учитывать наличие этой особенности.

Для учета различной масштабности значений временного ряда при обучении моделей СВМО и МГО используется подход, основанные на специальном распределении объемов тестовой и обучающей выборок, именуемый кросс-валидационное распределение [128]. В общем виде идея этого подхода представлена на рисунке 4.9.

Рис. 4.9 Особенность метода кросс-валидационного распределения выборки данных

Из рисунка видно, что основными особенностями кросс-валидационного распределения данных при реализации процесса МО являются:

– наличие нескольких этапов МО с разным объемом обучающей выборки;

– линейный рост используемых для МО данных от 50% до 100% с заданным исследователем шагом (обычно используется шаг 25%);

– фиксированный размер тестовой выборки.

Таким образом, в процессе обучения МО реализуется характерный для анализа временных рядов гиперпараметр, именуемый «период ретроспективного анализа» (Loopback Period) [129]. Из рисунка 4.9 видно, что для периода ретроспективного анализа в N временных шагов модель использует N прошлых наблюдений в качестве входных признаков для прогнозирования (N+1)-го наблюдения (значения данных тестовой выборки перекрывают значения данных обучающей выборки.

4.4.3. Формирование параметров модулей предварительной обработки и прогнозирования временного ряда и плана сравнительного эксперимента

В соответствии с разработанным в главах 2 и 3 алгоритмами функционирования модулей предварительной обработки значений временного ряда (снижение уровня влияния факторов зашумления) и прогнозирования временного ряда рабочей нагрузки важной задачей является выбор параметров, инициализирующих указанные алгоритмы.

К параметрам модуля предварительной обработки временного ряда следует отнести:

– ε – отношение сигнал/шум (Noise Ratio) для алгоритма КДЭМАШ;

– m – число измерений (для алгоритма выборочной энтропии);

– ρ – пороговое значение (для алгоритма выборочной энтропии);

– K – количество формируемых КМФ функций (мод) для алгоритма ДВМ.

К параметрам модуля прогнозирования, основанного на гибридной модели глубокого обучения, следует отнести:

– X – размерность входного слоя;

– Y – размерность выходного слоя;

– α - коэффициент скорости (уровня) обучения;

– b – размер блока данных (Batch Size);

– epoch – количество эпох обучения;

Для модели СНН-ОМ:

– C – размер ядра каждой модели в ансамбле;

Для модели СДКП-ДН:

– h – количество скрытых слоев каждой модели в каскаде.

В рамках планирования сравнительного эксперимента для оценивания качества разработанного решения по критерию пригодности были выбраны следующие значения рассмотренных параметров (таблицы 4.1-4.2).

Таблица 4.1

Значения параметров алгоритмов модуля предварительной обработки временного ряда

Параметр	Значение
ε (КДЭМАШ)	0,005
m (выб. энтропия)	2
ρ (выб. энтропия)	0,2
K (ДВМ)	10

Таблица 4.2

Значения параметров алгоритмов модуля прогнозирования временного ряда

Параметр	Значение
X (гибридный алгоритм)	50
Y (гибридный алгоритм)	1
α (гибридный алгоритм)	0,01
b (гибридный алгоритм)	128
epoch (гибридный алгоритм)	100
C (min, mid, max) (СНН-ОМ)	2, 4, 8
h (СДКП-ДН)	32, 64, 128

Следует отметить, что в большинстве своем выбор значений параметров был выполнен для настроек алгоритмов по умолчанию (default mode) и в процессе эксперимента может подвергаться коррекции.

Кроме выбора параметров инициализации алгоритма в рамках исследования был сформирован план сравнительного эксперимента по оцениванию качества предложенного решения. Разработанный план включает следующие этапы:

1. Выбор размера экспериментального временного ряда (общая выборка – рисунок 4.9.

2. Нормализация значений выбранного временного ряда по оси времени.

3. Распределение выбранного временного ряда на тестовые и обучающие выборки в соответствии с методом кросс-валидации (25% тестовая выборка; 25%, 50%, 75% обучающая выборка).

4. Декомпозиция значений полученных тестовых и обучающих выборок на эмпирические и вариационные моды – функции КМФ/КЛЭМАШ и КМФ/ДВМ соответственно.

5. Предварительное обучение и валидация гибридного алгоритма на множестве функций КМФ/КЛЭМАШ и КМФ/ДВМ.

6. Выбор КМФ функции с наименьшей функцией потерь гибридного алгоритма.

7. Выполнение эпох обучения на выбранных в п. 3 обучающих выборках.

8. Валидация полученных параметров модели МГО для каждой из эпох обучения.

9. Выполнение прогнозирования на выбранной тестовой выборке альтернативного варианта: моделей SVM-ARIMA и SVM-FARIMA.

10. Расчет значений метрик оценивания эффективности процесса прогнозирования (п. 1.1.6) для каждой из эпох обучения модели МГО, а также моделей альтернативного варианта.

11. Визуализация результатов сравнительного оценивания.

4.4.4. Результаты сравнительного эксперимента по оценке эффективности процесса прогнозирования рабочей нагрузки

При выполнении сравнительного эксперимента в качестве исходного временного ряда рабочей нагрузки был выбран один из временных рядов базы ретроспективных данных рабочей нагрузки Google Cluster, сохраненных за 2019 год [124]. В качестве показателя значений временного ряда рабочей нагрузки использовался показатель CPU_Usage Rate – коэффициент загрузки процессора: процента времени, в течение которого процессор занят обработкой задач. Этот показатель рассчитывается путем деления времени работы процессора на общее время мониторинга за заданный период. Длительность временного ряда выбиралась исходя из необходимости наличия в нем сезонных и межсезонных и нерегулярных колебаний рабочей нагрузки (рисунок 4.8), и составила два календарных месяца.

Следующим экспериментальным этапом являлась нормализация значений выбранного временного ряда. Необходимость нормализации обусловлена особенностями функционирования ВЦОД Google Cluster, которым характерна высокая динами обработки потребительских запросов, что приводит к значительным изменениям значений показателя CPU_Usage Rate в течение выбранного временного отрезка.

Поскольку алгоритмы МГО, реализованные в СНН-ОМ и СДКП-ДН ориентированы на вычисление расстояния между признаками, высокие из менения показателя CPU_Usage Rate приводят к его увеличению, что ведет к низкой сходимости алгоритмов.

В качестве метода нормализации был выбран метод масштабирования Min-Max (Min-Max Scaler), который широко применяется при обучении МГО [130].

Получение нормализованных значений X временного ряда при этом выполняется согласно выражению:

$$X_{норм} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (59)$$

После выполнения нормализации был получен исходный временной ряд длительностью 1000 условных отсчетов.

Поскольку, предположительно значения полученного исходного временного ряда подвержены влиянию факторов зашумления в эксперименте использовались предложенные в главе 2 алгоритмы декомпозиции КДЭМАШ и ДВМ для получения значимого подмножества КМФ функций, используемых в качестве входных данных модуля прогнозирования рабочей нагрузки и параметрами, представленными в таблице 4.1.

Декомпозиция временного ряда КДЭМАШ в совокупности с алгоритмами расчета выборочной энтропии и К-средних (п. 1.2.4) позволили выделить тестовой подмножество функций $КМФ_{test}=\{КМФ_2,КМФ_3,КМФ_4\}$. Второй этап декомпозиции функции $КМФ_1$, фактически совпадающей по высокочастотным составляющим с исходным временным рядом, позволил редуцировать мощность множества $КМФ_{test}$ до двух значений $КМФ_{test}=\{КМФ_3,КМФ_4\}$. Полученные функции представлены на рисунке 4.10.

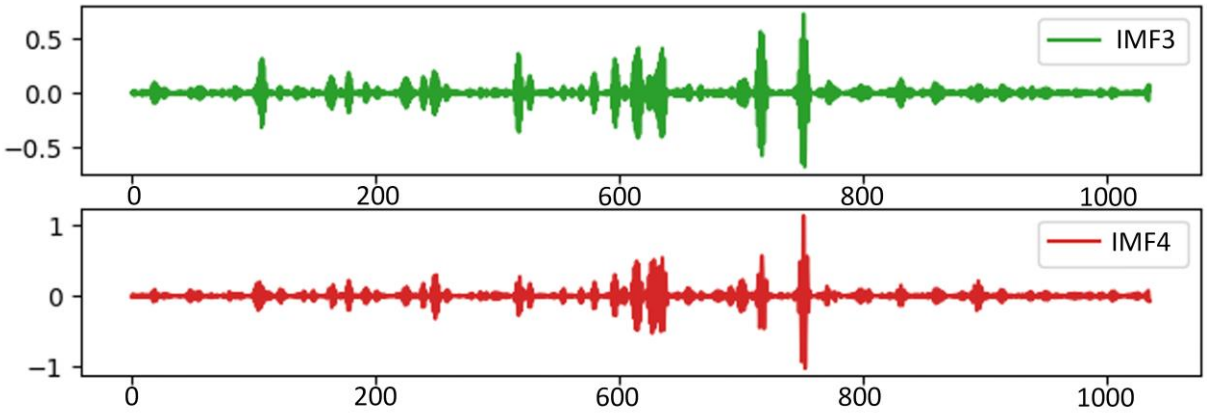


Рис. 4.10 Функции КМФ, полученные в результате функционирования модуля предварительной обработки сигнала

Полученные КМФ функции были последовательно использованы в качестве тестовой обучающей выборки для алгоритмов модели МГО с параметрами, представленными в таблице 4.2 с целью получения функции потерь (loss function),

которая количественно оценивает ошибку между прогнозными значениями МГО и фактическими значениями временного ряда, представленного КМФ функциями. В качестве функции потерь использовалась метрика MSE (п. 1.1.6).

Графики функции потерь, полученные в процессе тестового обучения МГО, представлены на рисунке 4.11.

Из рисунка видно, что, несмотря на достаточно близкие значения функции потерь при обучении МГО (значение MSE), в процессе валидации временной ряд, представленный функцией КМФ$_3$, показывает низкие значения функции потерь (высокие значения Val_MSE – ошибка валидации) с ростом числа эпох обучения/валидации. Это означает при использовании функции КМФ3 в качестве входных данных гибридной модели МГО, она показывает низкую способность к обучению.

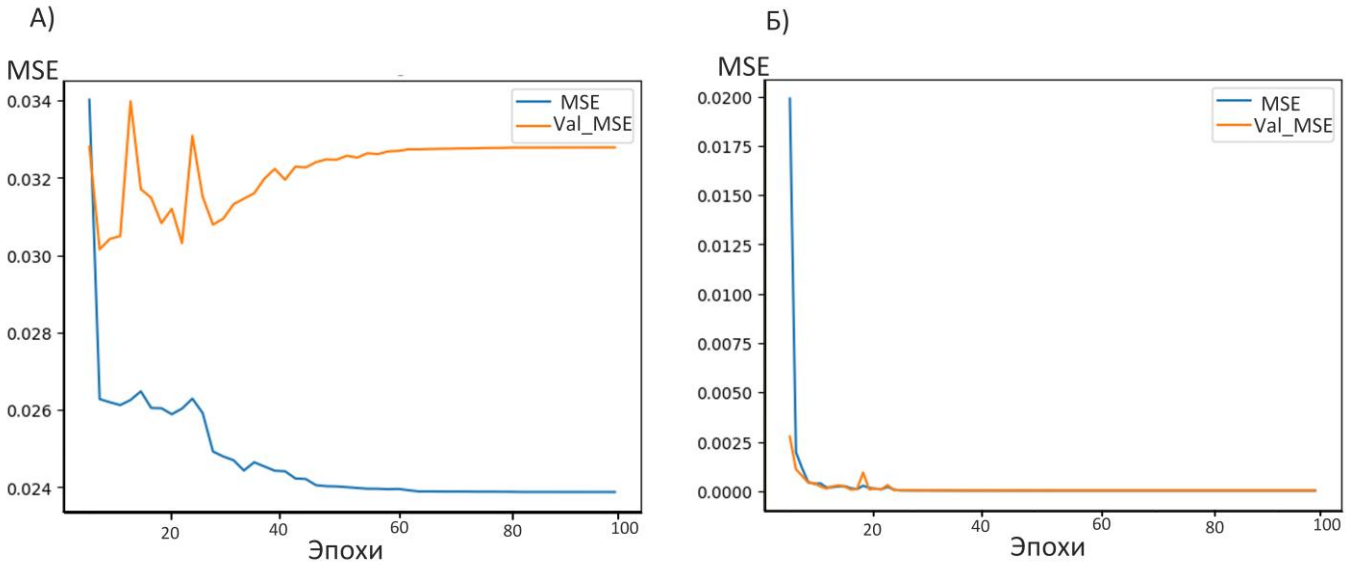


Рис. 4.11 Тестовые (MSE) и валидационные (Val_MSE) функции потерь при обучении на значениях а) функции КМФ$_3$, б) функции КМФ$_4$

Таким образом, для выполнения экспериментальных серий был выбран вариант временного ряда, представленный функцией КМФ$_4$.

Экспериментальные серии проводились на тестовых и валидационных выборках разделенного временного ряда (рисунок 4.9) условно обозначенными следующим образом:

– Training1 – обучающая выборка = 25% исходного временного ряда;

– Training2 – обучающая выборка = 50% исходного временного ряда;

– Training3 – обучающая выборка = 75% исходного временного ряда.

Валидационная выборка представлена отсчетами 500-1000 исходного временного ряда из которых в качестве прогнозных (не используемых в процессе обучения) рассматривались отсчеты 750-1000 (25% исходного временного ряда).

Результаты экспериментальных серий по выбранным в п. 1.1.6 метрикам представлены в таблице 4.3.

Таблица 4.3

Результаты экспериментальных серий оценивания качества предлагаемой гибридной модели глубокого обучения

Метрики эффективности прогнозирования	MAE	MSE	RMSLE	R^2
Traning1	0,020021	0,001491	0,0315	0,9790
Traning2	0,017927	0,000903	0,0321	0,9782
Traning3	0,009739	0,000822	0,0313	0,9792

Визуализация прогнозных значений тестового участка целевого временного ряда (восстановлен из функции КМФ до исходного состояния) для указанных серий представлена на рисунке 4.12.

Из рисунка 4.12 видно, что 25% увеличение размера обучающей выборки в каждой серии (в частности, обучения модели СДКП-ДН), асимптотически увеличивает точность прогноза при нелинейном росте времени, затрачиваемом на обучение.

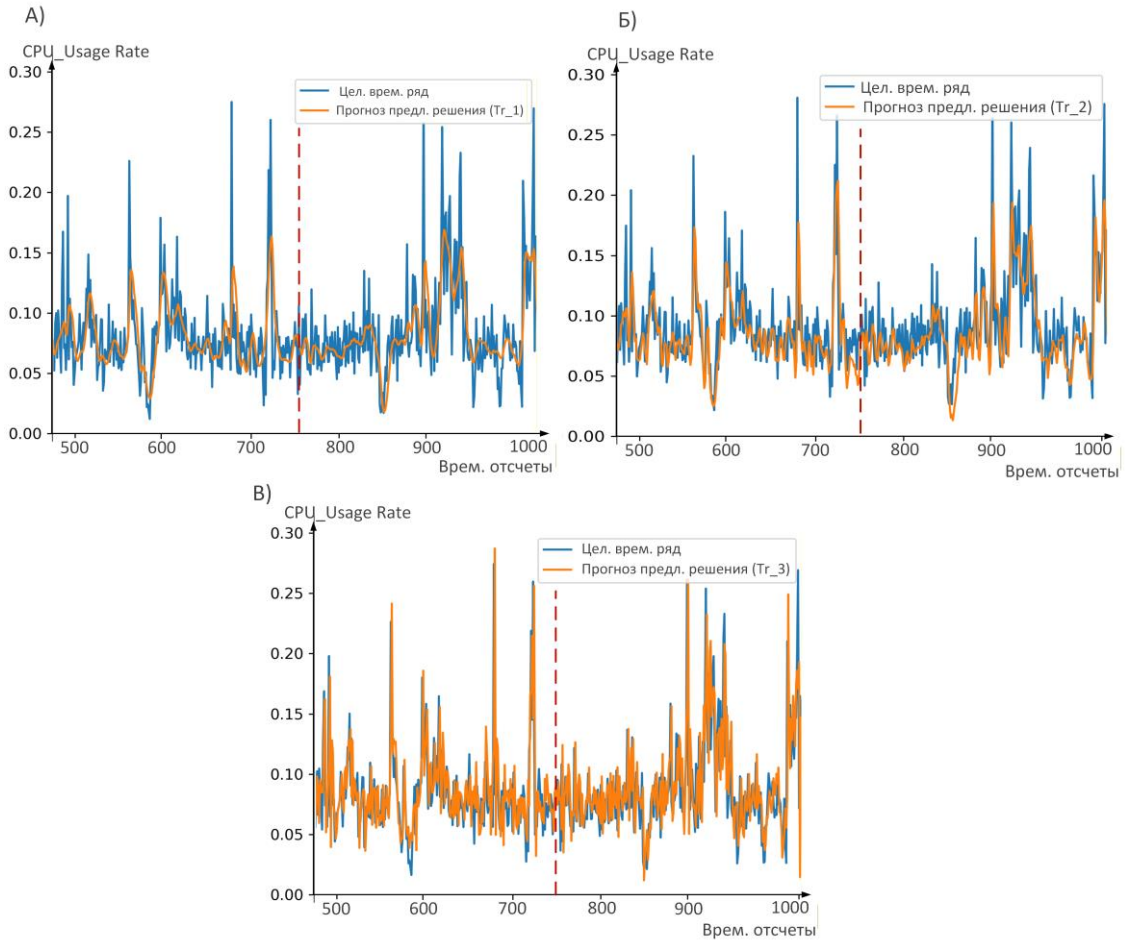


Рис 4.12 Сравнение экспериментальных серий оценки эффективности процесса прогнозирования временного ряда предлагаемой каскадной гибридной модели глубокого обучения: а) серия Training1, б) серия Training2 в) серия Training3

Используя исходный временной ряд в качестве входных данных моделей SVM-ARIMA(p,d,q), где d – коэффициент дифференцирования – является целочисленным и равен 1 (линейный тренд значений временного ряда). Дополнительно, с целью получения картины прогнозного тренда на разных масштабах рассмотрения, использовался вариант модели SVM-FARIMA(p,d,q) (фрактальный авторегрессионный интегральный процесс скользящего среднего), применяемой для долгосрочных прогнозов, где коэффициент d является дробным в диапазоне d∈("-0,5;0,5"). Указанные методы применяются в качестве способа прогнозирования рабочей нагрузки по умолчанию в системе администрирования ВЦОД Google Cluster. С их настройками по умолчанию были выполнены три тестовые серии прогнозирования с глубиной прогноза 250 временных отсчетов.

Результаты экспериментальных серий по выбранным в п. 1.1.6 метрикам представлены в таблице 4.4.

Таблица 4.4

Результаты экспериментальных серий оценивания качества предлагаемой моделей SVM-ARIMA и SVM-FARIMA

SVM-ARIMA				
Метрики эффективности прогнозирования	MAE	MSE	RMSLE	R^2
Predict1	0,056202	0,003656	0,0406	0,9566
Predict2	0,056628	0,003858	0,0426	0,9594
Predict3	0,049455	0,003655	0,0391	0,9519
$Predict_{Avr}$	0,054095	0,003372	0,0407	0,9559
SVM-FARIMA				
Метрики эффективности прогнозирования	MAE	MSE	RMSLE	R^2
Predict1	0,031782	0,002061	0,0393	0,9611
Predict2	0,031548	0,002355	0,0367	0,9721
Predict3	0,032303	0,002805	0,0388	0,9598
$Predict_{Avr}$	0,031877	0,00259	0,0382	0,9643

Визуализация прогнозных значений тестового участка целевого временного ряда для указанных серий представлена на рисунке 4.13.

Из рисунка 4.13 видно, что классификатор SVM достаточно точно выделил шаблоны рабочей нагрузки в обеих альтернативных моделях, а модель прогнозирования ARIMA (рисунок 4.13 а) достаточно точно спрогнозировала линейный тренд на участках стационарности. Его расхождение с трендом исходного временного ряда на 0,1 коэффициента CPU_Usage Rate связано с тем, что процесс сглаживания начался не на участке стационарности, а на первом выделенном методом SVM шаблоне рабочей нагрузки. В целом, это расхождение не оказывает влияния на принятие решения о реконфигурации ВЦОД. Модель прогнозирования FARIMA (рисунок 4.13 б) продемонстрировала более высокую точность прогноза,

в силу ее ориентации на тренд рабочей нагрузки в долгосрочной перспективе (каковым является тестовый временной ряд). При этом, ей, как и модели ARIMA, присуще смещение прогнозных оценок на участках стационарности, правда, в меньшей степени: 0,05 против 0,1 у ARIMA. Также особенностью модели прогнозирования FARIMA является более высокая вычислительная сложность, в силу не целочисленного значения коэффициента d.

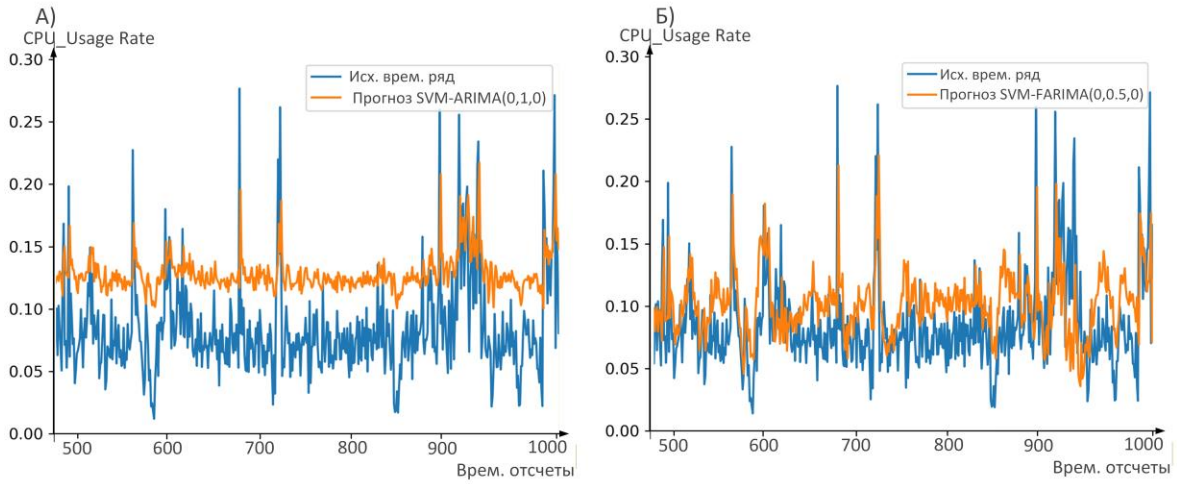


Рис 4.13 Сравнение экспериментальных серий оценки эффективности процесса прогнозирования временного ряда альтернативных моделей: а) SVM-ARIMA б) SVM-FARIMA

Результаты сравнения полученных экспериментальных серий предлагаемого решения и альтернативного решения (таблицы 4.2 и 4.3, рисунки 4.12 и 4.13) показывают, что в анализируемом окне временных отсчетов (750-1000) в общем случае предложенное решение дает прогнозные значения, в среднем отклоняющиеся от значений исходного временного ряда на участках стационарности менее чем на 2 %, в то время как прогнозные значения альтернативного способа имеют отклонение от 5 % (FARIMA) до 15 % (ARIMA). Причины такого расхождения рассмотрены выше. В общем случае способ на основе SVM-ARIMA (или вариантов ARIMA) предпочтительно использовать для получения быстрого результата о тренде рабочей нагрузки, в то время как результаты, полученные предлагаемым решением, целесообразно использовать для развернутого анализа проблемных

участков временного ряда с целью более тонкой оптимизации процесса реконфигурации ВЦОД.

4.5. Выводы по главе

В главе предложена архитектура системы прогнозирования рабочей нагрузки, отличающаяся интеграцией модуля предварительной обработки временного ряда рабочей нагрузки и гибридной моделью прогнозирования рабочей нагрузки.

В отличие от существующих подсистем прогнозирования рабочей нагрузки ВЦОД (на примере ВЦОД Google Cluster) предложенное решение базируется на гибридной модели глубокого обучения, интегрирующей ансамбль одномерных сверточных нейронных сетей с разным размером ядер (фильтров), а также каскада двунаправленных сетей с долгой краткосрочной памятью с разным размером скрытых слоев.

В главе обоснован выбор фреймворков для программной реализации разработанной архитектуры и подробно описана разработанная структура программного комплекса на их основе.

В качестве оценки эффективности предлагаемых решений представлены результаты сравнительного имитационного эксперимента по оцениванию эффективности процесса прогнозирования рабочей нагрузки с помощью предлагаемого решения и существующих статистических вероятностных моделей.

Таким образом, показана возможность применения предложенных модели и алгоритмов в составе программного обеспечения систем прогнозирования рабочей нагрузки на основе временных рядов ее ретроспективных данных в условиях воздействия факторов зашумления. Результаты сравнительного эксперимента подтверждают достижение цели исследования. На реализованные элементы разработанного ПО специальных модулей локального и глобального классификаторов, получено свидетельство о регистрации программы для ЭВМ в реестре ФИПС.

ЗАКЛЮЧЕНИЕ

Диссертационная исследование посвящено разработке моделей и алгоритмов прогнозирования рабочей нагрузки виртуализированного центра обработки данных на основе временных рядов ее показателей в условиях воздействия на них факторов зашумления присущих процессу функционирования центра. Указанные временные ряды получают из базы ретроспективных данных рабочей нагрузки системы мониторинга центра.

Научная задача, решенная в диссертации, может быть классифицирована как задача применения известных научных методов в новой предметной области.

Достоверность и обоснованность полученных результатов подтверждается научно организованным сравнительным экспериментом, корректным применением известных методов исследования, адекватных природе изучаемых процессов и явлений, непротиворечивостью и воспроизводимостью результатов, полученных в процессе проведения серий экспериментов.

В процессе выполнения диссертационного исследования получены следующие основные результаты:

1. Проведен анализ исследований, посвященных организации и функционированию службы администрирования виртуализированного центра обработки данных: рассмотрена актуальность задач реактивного и проактивного управления рабочей нагрузкой; определена структурная схема подсистемы мониторинга вычислительных ресурсов, отображающих рабочую нагрузку, методы и средства их анализа; рассмотрен подход к сохранению ретроспективных данных рабочей нагрузки в виде временных рядов значений заданных показателей утилизации вычислительных ресурсов; обобщенно сформулирована задача прогнозирования рабочей нагрузки на основе анализа временных рядов ее показателей; выявлены проблемы их искажения за счет внешних и внутренних факторов зашумления

2. На основе представления зашумленного временного ряда рабочей нагрузки в виде сигнальной конструкции, теоретических подходов к разложению сигнала на

колебательные модовые функции и двухэтапного использования методов эмпирической и вариационной модовой декомпозиции разработана модель модовой декомпозиции временного ряда рада рабочей нагрузки, обеспечивающая снижение влияния факторов зашумления на значения временного ряда.

3. Разработан комплексный алгоритм предварительной обработки временного ряда рабочей нагрузки, отличающийся наличием этапа вторичной вариационной модовой декомпозиции базовой колебательной модовой функции, полученной методом эмпирической модовой декомпозиции, который обеспечивает формирование множеств обучающей и тестовой выборок для системы прогнозирования элементов временного ряда.

4. Разработан гибридный алгоритм прогнозирования временного ряда рабочей нагрузки для системы глубокого обучения, который обеспечивает получение разномасштабных прогнозных значений временного ряда рабочей нагрузки.

5. Разработана архитектура системы прогнозирования рабочей нагрузки, базирующаяся на предложенных алгоритмических решениях. На программную реализацию разработанной архитектуры получено свидетельство о регистрации программы для ЭВМ в реестре ФИПС № 2026613698 от 09.02.2026.

6. Проведены сравнительные эксперименты по оцениванию эффективности процесса прогнозирования рабочей нагрузки полученным решением и существующими подходами на основе статистических вероятностных моделей. Результаты экспериментов показывают для полученного решения 2% расхождение в точности прогноза на участках стационарности тестового временного ряда и 5-15 % расхождения для существующих решений.

СПИСОК ТЕРМИНОВ, СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

В настоящей работе применяются следующие сокращения:

АДЭМ – ансамблевая эмпирическая модовая декомпозиция
ВМ – виртуальная машина
ВЦОД – виртуализированный центр обработки данных
ДВМ – декомпозиция на вариационные моды
ДЭМ – декомпозиция на эмпирические моды
ИИ – искусственный интеллект
КДЭМАШ – комплементарная декомпозиция на эмпирические моды с адаптивным шумом
КМФ – колебательная модовая функция
ММО – модель машинного обучения
МП – модель прогнозирования
ОР – обратное распространение
ПД – повышение дискретизации
ПО – программное обеспечение
ПР – прямое распространение
РНС – рекуррентная нейронная сеть
СВМО – статистическая вероятностная модель обучения
СД – субдискретизация
СДКП-ДН – сеть с долгой краткосрочной памятью двунаправленная
СНН-ДМ – сверточная нейронная сеть двумерная
СНН-ОМ – сверточная нейронная сеть одномерная
СУРБ – сеть с управляемым рекуррентным блоком
СХД – система хранения данных
ФМ – физическая машина
ЦОД – центр обработки данных

ADMM – Alternating Direction Method of Multipliers
AR – Auto Regression
ARIMA – Autoregressive Integrated Moving Average
ARFIMA – Autoregressive Integrated Fractal Moving Average
CEEMDAN – Complete Ensemble Empirical Mode Decomposition with Adaptive Noise
CNN – Convolutional Neural Network
DAE – Deep-learning based AutoEncoder
DLM – Deep Learning Model
EKF – Extended Kalman Filter
EKS – Extended Kalman Smoother
EMD – Empirical Mode Decomposition
EEMD – Ensemble Empirical Mode Decomposition ES
HMM – Hidden Markov Model

HSA – Hilbert Spectrum Analysis
GRU – Gate Recurrent Unit
IMF – Intrinsic Mode Function
LSTM – Long Short-Term Memory
MAE – Mean Absolute Error
MSE - Mean Squared Error
RM – Regression Model
RMSLE – Root Mean Squared Logarithmic Error
RNN – Recurrent Neural Network
SARIMA – Season Autoregressive Integrated Moving Average
SLA – Service Level Agreement
SVM – Support Vector Machine
UKF – Unscented Kalman Filter
VDM – Variation Mode Decomposition

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Armbrust, M., Fox, A., Gri th, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M. A View of Cloud Computing // Communications of the ACM. 53(4), pp. 50-58. (2010).
2. Buyya, R., Yeo, C., Venugopal, S., Broberg, J., Brandic, I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility // Future Generation Computer Systems. 25(6), pp. 599-616. (2009).
3. Gani, A., Qi, H., Shiraz, M., Liu, J. Data center network architecture in cloud computing: review, taxonomy, and open research issues // Journal of Zhejiang University, SCIENCE C, 15(9). pp. 776-793. (2014).
4. Bouman, J., Trienekens, J., Zwan, M. Specification of service level agreements, clarifying concepts on the basis of practical research // In Proc. of the Software Technology and Engineering Practice Conference, pp. 169-175, 1999.
5. Rayan, A., Nah, Y. Resource Prediction for Big Data Processing in a Cloud Data Center: A Machine Learning Approach // In: IEIE Transactions on Smart Processing and Computing, vol. 7, no. 6:478-490, December 2018.
6. Cloud Services Reach $130B, Dwarfs Data Center Spending // [Электронный ресурс]. – Режим доступа: https://www.crn.com/news/data-center/cloud-services-reach-130b-dwarfs-data-center-spending/ – Дата доступа: 15.02.2023.
7. Cloud Data Center Market Set for Strong Growth Amid Cloud Adoption Trends // [Электронный ресурс]. – Режим доступа: https://www.precedenceresearch.com/cloud-data-center-market/ – Дата доступа: 17.02.2023.
8. Azmandian, F., Mo e, M., Dy, J., Aslam, J., Kaeli, D. Workload Characterization at the Virtualization Layer // In: Proc. of the 19th Int. Symp. on Modeling, Analysis Simulation of Computer and Telecommunication Systems- MASCOTS11, pp. 63-72. (2011).
9. Chen, X., Lu, C., Pattabiraman, K. Failure Analysis of Jobs in Compute Clouds: A Google Cluster Case Study // In: Proc. of the 25th Int. Symp. on Software Reliability Engineering- ISSRE14, pp. 167-177. (2014).

10. Bi, J., Zhu, Z., Tian, R., Wang, Q. Dynamic Provisioning Modeling for Virtualized Multi-tier Applications in Cloud Data Center. In: Proc. of the 3rd Int. Conf. on Cloud Computing- CLOUD10, pp. 370-377. (2010).
11. Schad, J., Dittrich, J., Quiane-Ruiz, J.A. Runtime Measurements in the Cloud: Observing, Analyzing, and Reducing Variance. Proceedings of the VLDB Endowment 3(1-2), pp. 460-471 (2010).
12. Daneshyar, S. Evaluation of Data Processing Using MapReduce Framework in Cloud and Stand - Alone Computing // In: International Journal of Distributed and Parallel systems 3(6):51-63 (2012).
13. Aliyu, M., Gital, A., Souley, B., Kabir, R. A Multi-Tier Architecture for the Management of Supply Chain of Cloud Resources in a Virtualized Cloud Environment // International Journal of Information Systems and Supply Chain Management, 14(3):1-17 (2021).
14. Bruneo, D., Distefano, S., Longo, F., Pulia to, A., Scarpa, M. Workload-Based Software Rejuvenation in Cloud Systems // IEEE Transactions on Computers 62(6), pp. 1072-1085. (2013).
15. Azmandian, F., Mo e, M., Dy, J., Aslam, J., Kaeli, D. Workload Characterization at the Virtualization Layer // In: Proc. of the 19th Int. Symp. on Modeling, Analysis Simulation of Computer and Telecommunication Systems- MASCOTS11, pp. 63-72. (2011).
16. Atikoglu, B., Xu, Y., Frachtenberg, E., Jiang, S., Paleczny, M. Workload Analysis of a Large-scale Key-value Store // In: Proc. of the 12th ACM SIGMETRICS/PERFORMANCE Joint Int. Conf. on Measurement and Modeling of Computer Systems, pp. 53-64. (2012).
17. Solis Moreno, I., Garraghan, P., Townend, P., Xu, J. Analysis, Modeling and Simulation of Workload Patterns in a Large-Scale Utility Cloud // IEEE Transactions on Cloud Computing 2(2), 208-221. (2014).
18. Juan, D.C., Li, L., Peng, H.K., Marculescu, D., Faloutsos, C. Beyond Poisson: Modeling Inter-Arrival Time of Requests in a Datacenter // In: Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, vol. 8444, pp. 198-209. (2014).

19. Tai, J., Zhang, J., Li, J., Meleis, W., Mi, N. Adaptive Resource Allocation for Cloud Computing Environments under Bursty Workloads. In: Proc. of the 30th Int. Conf. on Performance Computing and Communications- IPCCC11, pp. 18. (2011).
20. Chen, S., Ghorbani, M., Wang, Y., Bogdan, P., Pedram, M. Trace-Based Analysis and Prediction of Cloud Computing User Behavior Using the Fractal Modeling Technique // In: Proc. of the 7th Int. Congress on Big Data, pp. 733-739. (2014).
21. Ghorbani, M., Wang, Y., Xue, Y., Pedram, M., Bogdan, P. Prediction and Control of Bursty Cloud Workloads: A Fractal Framework // In: Proc. of the Int. Conf. on Hardware/Software Codesign and System Synthesis- CODES14, vol. 12. pp. 1-9. (2014).
22. Yin, J., Lu, X., Chen, H., Zhao, X., Xiong, N. System resource utilization analysis and prediction for cloud based applications under bursty workloads // Information Sciences, no. 279, pp. 338-357. (2014).
23. Pacheco-Sanchez, S., Casale, G., Scotney, B., McClean, S., Parr, G., Dawson, S. Markovian Workload Characterization for QoS Prediction in the Cloud // In: 4th Int. Conf. on Cloud Computing- CLOUD, vol. 11, pp. 147-154. (2011).
24. Tickoo, O., Iyer, R., Illikkal, R., Newell, D. Modeling Virtual Machine Performance: Challenges and Approaches // SIGMETRICS Performance Evaluation Review, 37(3). Pp. 55-60 (2010).
25. Du, J., Sehrawat, N., Zwaenepoel, W. Performance Proling of Virtual Machines // ACM SIGPLAN Notices, 46(7). pp. 3-14. (2011).
26. Ren, G., Tune, E., Moseley, T., Shi, Y., Rus, S., Hundt, R. Google-wide Proling: A Continuous Proling Infrastructure for Data Centers // IEEE Micro, 30(4). pp. 65-79. (2010).
27. Birke, R., Chen, L., Smirni, E. Multi-Resource Characterization and their (In)dependencies in Production Datacenters // In: Network Operations and Management Symposium-NOMS'14, vol. 1. pp. 1-17. (2014).
28. Alhamazani, K., Ranjan, R., Mitra, K., Rabhi, F., Jayaraman, P., Khan, S., Guabtni, A., Bhatnagar, V. An overview of the commercial cloud monitoring tools: research dimensions, design issues, and state-of-the-art // Computing, 97(4). pp. 357-377. (2015).

29. Methods for measuring physical CPU utilization in a cloud computing infrastructure // URL: https://patents.google.com/patent/US9749207B2/en // (дата обращения 13.05.2024).

30. Shen, L., Qian, S., Zhai, T., Li, L., Li, Z. Research on cloud computing high-density data center infrastructure and environment matching technology // MATEC Web of Conferences, 336, 02028. pp. 3-14. (2021).

31. The Ultimate Guide to Cloud Resource Monitoring: Optimizing Costs and Efficiency // URL: https://diversedaily.com/the-ultimate-guide-to-cloud-resource-monitoring-optimizing-costs-and-efficiency // (дата обращения 14.05.2024).

32. Метрики Yandex Compute Cloud. // URL: https://yandex.cloud/ru/docs/monitoring/metricsref/computeref?utm_referrer=https%3A%2F%2Fyandex.ru%2F/ (дата обращения 15.05.2024).

33. The Industry Standard in Open Source IT Monitoring Tools. // URL: https://www.nagios.org // (дата обращения 16.05.2024).

34. Meng, S., Liu, L. Enhanced Monitoring-as-a-Service for Eective Cloud Management // IEEE Transactions on Computers 62(9), pp. 1705-1720. (2013).

35. Mueller, J., Palma, D., Landi, G., Soares, J., Parreira, B., Metsch, T., Gray, P., Georgiev, A., Al-Hazmi, Y., Magedanz, T., Simoes, P. // Monitoring as a Service for Cloud Environments // In: 5th Int. Conf. on Communications and Electronics-ICCE14, pp. 174-179. (2014).

36. Du, J., Sehrawat, N., Zwaenepoel, W. Performance Pro ling of Virtual Machines // ACM SIGPLAN Notices, 46(7) pp. 3-14. (2011).

37. Khan, A., Yan, X., Tao, S., Anerousis, N. Workload characterization and prediction in the cloud: A multiple time series approach // In: 2012 IEEE Network Operations and Management Symposium, vol. 1. pp. 1-8. (2012).

38. Google/cluster-data (Public) // URL: https://github.com/google/cluster-data // (дата обращения 21.05.2024).

39. Alibaba/clusterdata (Public) // URL: https://github.com/alibaba/clusterdata // (дата обращения 21.05.2024).

40. Amiri, M., Mohammad-Khanli, L. Survey on prediction models of applications for resources provisioning in cloud // Journal of Network and Computer Applications, vol. 82, pp. 93–113. (2017).
41. Li, S.,Wang, Y., Qiu, X., Wang, D., Wang, L. A workload prediction-based multi-vm provisioning mechanism in cloud computing // In: 15th Asia Pacific Network Operations and Management Symposium (APNOMS), vol. 1. pp. 1–6. (2013).
42. Lu, Y., Panneerselvam, J., Liu, L., Wu, Y. RVLBPNN: A Workload Forecasting Model for Smart Cloud Computing // Scientific Programming, no. 4, pp. 1–9. (2016).
43. Yadav, M., Yadav, D. Workload Prediction over Cloud Server using Time Series Data // Part of the book series: Advances in Intelligent Systems and Computing ((AISC,vol. 1393)). pp. 447-459. (2021).
44. Lackinger, A., Morichetta, A., Dustdar, S. Time Series Predictions for Cloud Workloads: A Comprehensive Evaluation // In: IEEE International Conference on Service-Oriented System Engineering (SOSE), vol. 1. pp. 36-46. (2024).
45. Felix, E., Lee, S. Integrated approach to software defect prediction // In: IEEE Access, vol. 5. pp. 21524–21547. (2017).
46. Tugnait, J. A Data-Cleaning Approach to Robust Multisensor Detection of Improper Signals. // IEEE Access (99). pp. 1-12. (2019).
47. Velayudhan, A., Peter, S. Noise Analysis and Different Denoising Techniques of ECG Signal - A Survey. // IOSR Journal of Electronics and Communication Engineering, no. 2. pp. 40-44. (2016).
48. Bouatteour, H., Slimen, Y., Mechteri, M., Biallach, H. Root Cause Analysis of Noisy Neighbors in a Virtualized Infrastructure. // E Wireless Communications and Networking Conference (WCNC), no. 8-9. pp. 203-218. (2020).
49. Мартыненков Б.В., Кравец О.Я., Белецкая С.Ю., Скурихин А.А. Подход к предварительной обработке зашумленных ретроспективных данных об уровне загрузки вычислительных ресурсов для системы прогнозирования рабочей нагрузки виртуализированного центра обработки данных // Системы управления и информационные технологии, №3(101), 2025. С. 41-50.

50. Обнаружение «шумных соседей» с помощью eBPF // URL: https://habr.com/ru/companies/wunderfund/articles/859978 // (дата обращения 24.05.2024).
51. Melo1, C., Araujo, J., Alves, V., Maciel, P. Investigation of Software Aging Effects on the OpenStack Cloud Computing Platform // Journal of Software, vol. 12, no. 2. pp. 125-138. (2017).
52. Torquato, M., Araujo, J., Umesh, I., Maciel, P. SWARE: A Methodology for Software Aging and Rejuvenation Experiments // Journal of Information Systems Engineering & Management, no. 3(2). pp. 1-13. (2018).
53. Cotroneo, D., Natella, R., Pietrantuono, Russo, S. ASurvey of Software Aging and Rejuvenation Studies // ACM Journal on Emerging Technologies in Computing Systems, vol. 1, no. 1. pp. 1-35. (2019).
54. Parashivamurthy, S., Cholli, N. Software aging prediction – a new approach // International Journal of Electrical and Computer Engineering. vol. 13, no. 2. pp. 1773-1781. (2023).
55. Chatterjee, S. , Thakur, R., Yadav, R., Gupta, L., K, Raghuvanshi. Review of noise removal techniques in ECG signals // IET Signal Process, vol. 14, iss. 9. pp. 569-590 (2020).
56. Han, G., Lin, B., Xu, Z. Electrocardiogram signal denoising based on empirical mode decomposition technique: an overview // J. Instrum., 12(3). pp. 300–0310 (2017).
57. Liu, L., Hao, T., Guo, Y., Lü, C., Wang, S. A method for denoising active source seismic data via Fourier transform and spectrum reconstruction // Geophysics, 89(6). pp. 1-40 (2024).
58. Cavalieri, D., Parkinson, C., Gloersen, P., Comiso, J., Zwally, H. Deriving long-term time series of sea ice cover from satellite passive-microwave multi sensor data sets // Journal of Geophysical Research: Oceans, 104(C7). pp. 15803-15814 (1999).
59. Gao, J., Wang, R. A novel manifold learning denoising method on bearing vibration signals // Journal of Vibroengineering, no. 18(1). pp. 175-189. (2016).
60. Joshi, S., Vatti, R., Tornekar, R. A survey on ECG signal denoising techniques // In: Communication Systems and Network Technologies, vol.1. pp. 60–64. (2013).

61. Sameni, R., Shamsollahi, M.B., Jutten, C. A nonlinear Bayesian filtering framework for ECG denoising // IEEE Trans. Biomed, no. 54(12). pp. 2172–2185. (2007).
62. Condat, L. A direct algorithm for 1-D total variation denoising // IEEE Signal Process. Lett., 20(11). pp. 1054–1057. (2013).
63. Van Alsté, J., Schilder, T. Removal of base-line wander and power-line interference from the ECG by an efficient FIR filter with a reduced number of taps // IEEE Trans. Biomed. Eng., 32 (12). pp. 1052–1060. (1985).
64. Chiang, H., Hsieh, Y., Fu, S.nNoise reduction in ECG signals using fully convolutional denoising autoencoders // IEEE Access, no. 7, pp. 60806–60813. (2019).
65. Wu, Z., Huang, N. On the filtering properties of the empirical mode decomposition // Adv. Adapt. Data Anal., 2(4). pp. 397–414. (2010).
66. Huang, N., Shen, Z., Long, S. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis // A Mathematical and Physics of Engineering Science, no. 454. pp. 903–995. (1998).
67. Sandoval, S., De Leon, P. Theory of the Hilbert Spectrum // URL: https://www.researchgate.net/publication/275670067_Theory_of_the_Hilbert_Spectrum // (дата обращения 07.06.2024).
68. Dragomiretskiy, K. Zosso, D. Variational Mode Decomposition // In: IEEE Transactions on Signal Processing, vol. 62, no. 3. pp. 531-544. (2014).
69. Wu, Z., Huang, N. Ensemble empirical mode decomposition: a noise-assisted data analysis method // Advanced Adaptaptive Data Analize, 1(01). pp. 1–41. (2009).
70. Torres, M., Colominas, M., Schlotthauer, G., Flandrin, P. A complete ensemble empirical mode decomposition with adaptive noise // In: IEEE international conference on acoustics, speech and signal processing (ICASSP-2011), vol. 1. pp 4144–4147. (2011).
71. Hu, C., Zhao, Y., Jiang, H., Jiang, M., You, F., Liu, Q. Prediction of ultra-short-term wind power based on CEEMDAN-LSTM-TCN // Energy Reports, vol. 8. pp. 483–492 (2022).
72. Xu, Y., Luo, M., Li, T., Song, G. ECG Signal De-noising and Baseline Wander Correction Based on CEEMDAN and Wavelet Threshold // Sensors, vol. 17, no. 12. p. 27-54 (2017).

73. Bhavsar, R., Helian, N. Efficient Methods for Calculating Sample Entropy in Time Series Data Analysis // Procedia Computer Science, no. 145. pp. 97-104. (2018).
74. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. // Foundations and Trends in Machine Learning, vol. 3, no. 1. pp. 1–122 (2010).
75. Kobylin, O. Time Series Clustering Based on the K-Means Algorithm // Journal La Multiapp, no. 1(3). pp. 1-7. (2020).
76. Humaira, H., Rasyidah, R. Determining The Appropiate Cluster Number Using Elbow Method for K-Means Algorithm // In: 2nd Workshop on Multidisciplinary and Applications (WMA), vol. 1. pp. 24-25. (2018).
77. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers // Machine Learning, vol. 3, no. 1. pp. 1–122 (2010).
78. Bunkheila, G. Signal Processing with MATLAB: System Simulation and Real Time Implementation // URL: https://www.mathworks.com/content/dam/mathworks/mathworks-dot-com/solutions/automotive/files/uk-expo-2012/master-class-signal-processing-with-matlab-system-simulation-and-real-time-implementation.pdf // (дата обращения 15.06.2024).
79. Statistics and Machine Learning Toolbox: Analyze and model data using statistics and machine learning // URL: https://www.mathworks.com/products/statistics.html // (дата обращения 18.06.2024).
80. Мартыненков Б.В. Разработка гибридной модели глубокого обучения для прогнозирования рабочей нагрузки в виртуализированных центрах обработки данных // Моделирование, оптимизация и информационные технологии, том 13, № 4, 2025. Doi
81. Control the workload prediction of a virtualized data center for noisy retrospective data / Martynenkov B.V., Tcvetkov A.V. // Modern informatization problems: Proc. of the XXXI-th Int. Open Science Conf. - Yelm, WA, USA: Science Book Publishing House, 2026.

82. Fuenzalida E. Effect of Workload History on Task Performance. The Journal of the Human Factors and Ergonomics Society. 2007. 49(2):277-292.
83. Shumway, R., Stoffer, D. Time Series Analysis and Its Applications. Fourth edition // Springer, P. 568. 2016.
84. Lim, B., Arık, S., Loeff, N., Pfister, T. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting // International Journal Forecast, vol. 37, no. 4, pp. 1748–1764. (2021).
85. Le, L. Time series analysis and applications in data analysis, forecasting and prediction // HPU2 Journal of Science: Natural Sciences and Technology, vol. 3, no. 1. pp. 20-30. (2023).
86. Sanchez-Espigares, J., Argueta, L. Lecture Notes on Forecasting Time Series // URL: https://upcommons.upc.edu/server/api/core/bitstreams/e414fd60-9617-41f8-9e76-b1452ab56e51/content // (дата обращения 12.07.2024).
87. Ho, M., Darman, H., Musa, S. Stock Price Prediction Using ARIMA, Neural Network and LSTM Models // Journal Physics Conference Series, vol. 1988, no. 1, pp. 12-41. (2021).
88. Ning, Y., Kazemi, H., Tahmasebi, P. A comparative machine learning study for time series oil production forecasting: ARIMA, LSTM, and Prophet // Computer Geoscience, vol. 164, pp. 105-126. (2022).
89. Parasyris, A., Alexandrakis, G., Kozyrakis, G., Spanoudaki, K., Kampanis, N. Predicting Meteorological Variables on Local Level with SARIMA, LSTM and Hybrid Techniques // Atmosphere (Basel), vol. 13, no. 6, pp. 8-78. (2022).
90. Mahmad Azan, A., Mohd Zulkifly Mototo, N., Mah, P. The Comparison between ARIMA and ARFIMA Model to Forecast Kijang Emas (Gold) Prices in Malaysia using MAE, RMSE and MAPE // Journal of Computing Research and Innovation, vol. 6, no. 3, pp. 22–33. (2021).
91. Abdelati, M., Abdelwali, H. Optimizing Simple Exponential Smoothing for Time Series Forecasting in Supply Chain Management // Indonesian Journal of Innovation and Applied Sciences (IJIAS), vol. 4, no. 3, pp. 247-256. (2024).

92. Abdullah, M. Using the Single-Exponential-Smoothing Time Series Model under the Additive Holt-Winters Algorithm with Decomposition and Residual Analysis to Forecast the Reinsurance-Revenues Dataset // Pakistan Journal of Statistics and Operation Research, no. 1, pp. 311–340. (2024).
93. Corberán-Vallet, A., Vercher, E., Segura, J., Bermúdez, J. A new approach to portfolio selection based on forecasting // Expert System Application, vol. 215, p. 119-370. (2023).
94. Garcia, H., Navarrete, T., Orozco, C. Workload Hidden Markov Model for anomaly detection // In: International Conference on Security and Cryptography (SECRYPT 2006), vol. 1, pp. 125-137. (2006).
95. Sun, Q., Tan, Z., Zhou, X. Workload prediction of cloud computing based on SVM and BP neural networks // Journal of Intelligent & Fuzzy Systems, no. 39(3), pp. 2861-2867. (2020).
96. Islam, A., Rahman, M. Workload Prediction on Google Cluster Trace // International Journal of Grid and High Performance Computing, no. 6(3), pp. 34-52. (2014).
97. Warsito, B., Santoso, R., Yasin, H. Cascade Forward Neural Network for Time Series Prediction // Journal Physics Conference Series, vol. 1025, p. 12-97. (2018).
98. Af'idah, D., Handayani, S. Comparative Analysis of Deep Learning Models for Retrieval-Based Tourism Information Chatbots // Jurnal Ilmiah Teknik Elektro Komputer dan Informatika, vol. 11, no. 1, pp. 53–67. (2025).
99. Riyadi, W. Comparative Analysis of Optimizer Effectiveness in GRU and CNN-GRU Models for Airport Traffic Prediction // Jurnal Ilmiah Teknik Elektro Komputer dan Informatika, vol. 10, no. 3, pp. 580–593. (2024).
100. Hossain, I., Islam, M., Martin, H. Potential Applications and Limitations of Artificial Intelligence in Remote Sensing Data Interpretation: A Case Study // Control Systems and Optimization Letters, vol. 2, no. 3, pp. 295–302. (2024).
101. Naji, H. , Xue, Q., Li, T. ADistributed VMD-BiLSTM Model for Taxi Demand Forecasting with GPS Sensor Data // Sensors, no. 24, pp. 66-83. (2024).
102. Lu, W., Rui, Y., Yi, Z., Ran, B., Gu, Y. A Hybrid Model for Lane-Level Traffic Flow Forecasting Based on Complete Ensemble Empirical Mode Decomposition and Extreme Gradient Boosting // IEEE Access Conference, vol. 1, pp. 1-13. (2020).

103. Goel, R., Sharma, A., Kapoor, R. Object Recognition Using Deep Learning // Journal of Computational and Theoretical Nanoscience, no. 16(9), pp. 4044-4052. (2019).
104. Saleem, M., Afsar, S., Mateen, A., Zaheer, A., Tariq, M., Raza, M. An Analysis on Object Recognition Using Convolutional Neural Networks // International Journal of Advanced Trends in Computer Science and Engineering, vol. 10, no. 3, pp. 1928-1936. (2021).
105. Escottá, A., Beccaro, W., Ramírez, M. Evaluation of 1D and 2D Deep Convolutional Neural Networks for Driving Event Recognition // Sensors, no. 22, pp. 1-21. (2022).
106. Ban Y., Zhang D., He Q., Shen Q. APSO-CNN-SE: An Adaptive Convolutional Neural Network Approach for IoT Intrusion Detection // Computers, Materials and Continua. vol. 81, issue 1, pp. 567-601 (2024).
107. Lu, Y., Panneerselvam, J., Liu, L., Wu, Y. RVLBPNN: A Workload Forecasting Model for Smart Cloud Computing // Scientific Programming, no. 2, pp. 1–9. (2016).
108. Janardhanan, D., Barrett, E. CPU workload forecasting of machines in data centers using LSTM recurrent neural networks and ARIMA models // In: 12th international conference for internet technology and secure transactions (ICITST), vol. 1, pp 55–60 (2017).
109. Yang, Q., Zhou, Y., Yu, Y., Yuan, J., Xing, X., Du, S. Multi-step-ahead host load prediction using autoencoder and echo state networks in cloud computing // The Journal of Supercomputing, vol. 71, no. 8, pp. 3037–3053. (2015).
110. Krichen, M., Mihoub, A. Long Short-Term Memory Networks: A Comprehensive Survey // AI, no. 6(9), pp. 1-21. (2025).
111. Gupta S, Dinesh, D. Resource usage prediction of cloud workloads using deep bidirectional long short term memory networks // In: International conference on advanced networks and telecommunications systems (ANTS), vol. 1, pp 1–6. (2017).
112. Prasetya Wibawa1, A., Fanny Fadhilla1, A., Khansa'a Iffat Paramarta, A., Putra Pertama Triono, A. and all. Bidirectional Long Short-Term Memory (Bi-LSTM) Hourly Energy Forecasting // In: International Conference on Computer Science Electronics and Information (ICCSEI 2023), vol. 501, pp. 1-7. (2023).

113. Guo, Q., Zhang, H., Zhang, Y., Jiang, H. Prediction of sea ice area based on the CEEMDAN-SO-BiLSTM model // PeerJ, vol. 1, pp. 1-15. (2023).
114. Wang, C., Li, H., Zhao, D. A Preconditioning Framework for the Empirical Mode Decomposition Method // Circuits, Systems, and Signal Processing, vol. 37, Issue 12, pp. 5417-5440. (2018).
115. EMD (Empirical Mode decomposition) light weight library // URL: https://github.com/emdforsale/emd // (дата обращения 07.09.2024).
116. About variational mode decomposition and its variants // URL: https://github.com/XinweiJiang/VMD // (дата обращения 12.09.2024).
117. MatLab Signal Processing Toolbox // URL: https://www.mathworks.com/help/signal/index.html?s_tid=CRUX_lftnav // (дата обращения 20.09.2024).
118. Simulink. Моделирование и проектирование на основе моделей // URL: https://www.mathworks.com/help/simulink/index.html?s_tid=CRUX_lftnav // (дата обращения 24.09.2024).
119. Tensorflow: An end-to-end platform for machine learning // URL: https://www.tensorflow.org/ // (дата обращения 5.10.2024).
120. Keas: A superpower for ML developers // URL: https://keras.io/ // (дата обращения 12.10.2024).
121. PyTorch // URL: https://pytorch.org/ // (дата обращения 18.10.2024).
122. Scikit-Learn: Machine Learning in Python // URL: https://scikit-learn.org/stable/ (дата обращения 27.10.2024).
123. MatLab Deep Learning Toolbox: design, train, analyze, and simulate deep learning networks // URL: https://www.mathworks.com/help/deeplearning/index.html?s_tid=CRUX_lftnav (дата обращения 7.11.2024).
124. Google cluster workload traces 2019 // URL: https://research.google/tools/datasets/google-cluster workload-traces-2019/ // (дата обращения 12.11.2024).

125. Liu, Z.. Cho, S. Characterizing machines and workloads on a google cluster // In: 41st International Conference on Parallel Processing Workshops, vol. 1, pp. 397–403. (2012).
126. MANSI-MEHTA / Workload-Prediction-of-Alibaba-Cluster-dataset // URL: https://github.com/MANSI-MEHTA/Workload-Prediction-of-Alibaba-Cluster-dataset/blob/master/AR_model_with_sliding_window.ipynb // (дата обращения 24.11.2024).
127. Alibaba Cloud // URL: https://www.alibabacloud.com/product/databases // (дата обращения 3.12.2024).
128. Dabhi, Z., Jain, M. A Study in Time Series Forecasting Model // Journal of Network Security and Data Mining,vol. 7, Issue 1, pp. 8-29. (2024).
129. Koparanov, K., Georgiev, K., Shterev, V. Lookback Period, Epochs and Hidden States Effect on Time Series Prediction Using a LSTM based Neural Network // 28th National Conference with International Participation (TELECOM), vol. 1, pp. 1-17. (2020).
130. Feature Scaling: MinMax, Standard and Robust Scaler – Machine Learning Geek [Электронный ресурс] // URL: https://machinelearninggeek.com/feature-scaling-minmax-standard-and-robust-scaler/ // (дата обращения 15.12.2024).