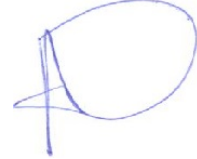


На правах рукописи



ЮРТИН Алексей Артемьевич

**НЕЙРОСЕТЕВЫЕ МЕТОДЫ ВОССТАНОВЛЕНИЯ  
ПОТОКОВЫХ ДАННЫХ**

2.3.5. — Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей

Автореферат  
диссертации на соискание ученой степени  
кандидата физико-математических наук

Челябинск — 2026

Работа выполнена на кафедре системного программирования  
ФГАОУ ВО «Южно-Уральский государственный университет  
(национальный исследовательский университет)»

**Научный  
руководитель:** ЦЫМБЛЕР Михаил Леонидович,  
доктор физ.-мат. наук, доцент,  
заместитель директора Научно-образовательного  
центра «Искусственный интеллект и квантовые  
технологии», ФГАОУ ВО «Южно-Уральский  
государственный университет (национальный  
исследовательский университет)» (г. Челябинск)

**Официальные  
оппоненты:** ГРОМОВ Василий Александрович,  
доктор физ.-мат. наук, профессор,  
заместитель руководителя департамента анализа  
данных и искусственного интеллекта  
ФГАОУ ВО «Национальный исследовательский  
университет «Высшая школа экономики» (г. Москва)

ВОЕВОДИН Вадим Владимирович,  
кандидат физ.-мат. наук,  
заведующий Лабораторией анализа  
суперкомпьютерных систем и приложений  
Научно-исследовательского вычислительного центра  
ФГБОУ ВО «Московский государственный  
университет имени М.В. Ломоносова» (г. Москва)

**Ведущая  
организация:** ФГАОУ ВО «Казанский (Приволжский)  
федеральный университет» (г. Казань)

Защита состоится 20 мая 2026 г. в 12:00 часов на заседании диссертационного совета  
24.2.437.10 при Южно-Уральском государственном университете по адресу: 454080,  
г. Челябинск, пр. Ленина, 76, ауд. 1001.

С диссертацией можно ознакомиться в библиотеке Южно-Уральского государ-  
ственного университета и на сайте: <https://www.susu.ru/ru/dissertation/24243710-d-21229818/yurtin-aleksey-artemevich>

Автореферат разослан « \_\_\_\_\_ » \_\_\_\_\_ 2026 г.

Ученый секретарь дис. совета



Я.А. Краева

## Общая характеристика работы

**Актуальность темы** диссертационного исследования обусловлена следующими основными факторами: широкое распространение потоковых данных в прикладных задачах и высокая практическая значимость задачи их восстановления; интенсивные научные исследования, направленные на повышение точности методов глубокого обучения, применяемых для восстановления потоковых данных; актуальность задачи повышения эффективности жизненного цикла нейросетевых моделей восстановления потоковых данных.

*Широкое распространение потоковых данных.* В условиях современной информационной среды многочисленные сенсоры, журналы событий, сетевые источники и другие средства мониторинга формируют непрерывные потоки данных. Одной из наиболее распространенных форм представления таких данных являются *временные ряды* — упорядоченные последовательности наблюдений, фиксируемых через равные интервалы времени и отражающих динамику исследуемых процессов. В связи с ростом объемов потоковых данных методы их интеллектуального анализа находят все более широкое применение в различных предметных областях, включая здравоохранение, суперкомпьютерные системы, моделирование климата, интернет вещей и др.

*Важность задачи восстановления пропусков.* Пропущенные значения в потоковых данных, возникающие из-за технических сбоев, ошибок сбора данных или ограничений измерительных систем, нарушают целостность данных и приводят к искажению аналитических выводов и снижению точности последующего анализа. Существующие подходы к восстановлению подразделяются на *офлайн* методы, опирающиеся на полный доступ ко всем полученным данным и обеспечивающие более точное восстановление за счет учета контекста до и после пропуска, и *онлайн* методы, выполняющие восстановление при поступлении данных последовательно и не располагающие информацией о будущих значениях. Во многих прикладных областях потоковые данные характеризуются наличием устойчиво повторяющихся паттернов, соответствующих активностям или поведенческим сценариям исследуемой системы. В рамках таких активностей нередко наблюдаются *поведенческие шаблоны* — непрерывные участки потоковых данных, демонстрирующие устойчивое, воспроизводимое поведение.

*Развитие нейросетевых методов восстановления данных.* В последнее десятилетие для решения задачи восстановления потоковых данных активно применяются нейросетевые модели различных архитектур. Методы, использующие нейросетевые модели, обеспечивают высокую точность восстановления за счет способности нейросетей моделировать временные и пространственные зависимости и адаптироваться к разнообразным шаблонам пропусков. Существенное влияние на точность восстановления оказывает выбор функции потерь, способной оценивать поведенческое сходство восстановлен-

ных и исходных потоковых данных. Если это сходство не учитывается при обучении, нейросетевая модель будет генерировать значения, не соответствующие реальному поведению объекта и обладающие меньшей практической ценностью.

*Важность задачи прогнозирования ошибки и времени обучения нейросетевых моделей.* Архитектура модели существенно влияет на точность восстановления потоковых данных. В условиях, когда перебор всех возможных конфигураций становится вычислительно затратным, возникает необходимость в разработке методов предварительного прогнозирования точности нейросетевых моделей. Подобные методы позволяют на основе характеристик данных, структуры пропусков и архитектуры нейросетевой модели прогнозировать ее ожидаемую точность восстановления, тем самым существенно снижая потребность в ресурсоемких экспериментах.

*Жизненный цикл нейросетевых моделей восстановления потоковых данных.* В современных системах машинного обучения восстановление потоковых данных можно рассматривать как *жизненный цикл* нейросетевых моделей — замкнутую последовательность взаимосвязанных этапов, каждый из которых выполняет специализированную функцию и обеспечивает корректность последующей обработки данных. В рамках такого подхода ключевые этапы (сбор и предварительная обработка данных, выбор, обучение и применение нейросетевой модели) рассматриваются не как изолированные задачи, а как часть единого процесса, обеспечивающего максимизацию точности восстановления. Ошибки на любом из этапов могут привести к каскадному ухудшению качества результатов работы нейросетевых моделей.

**Степень разработанности темы.** Као (Cao) и др. предложили модель BRITS (Bidirectional Recurrent Imputation for Time Series), представляющую собой двунаправленную рекуррентную нейросеть, которая последовательно восстанавливает пропуски временного ряда. GP-VAE (Gaussian Process Variational Autoencoder), предложенная Фортъен (Fortuin) и др., использует сверточные нейронные сети для формирования скрытого представления подпоследовательности временного ряда и гауссовский процесс для моделирования зависимостей между точками данных в получившемся пространстве. Ду (Du) и др. предложили модель SAITS (Self-Attention-based Imputation for Time Series), которая использует механизм внимания для восстановления пропущенных значений во временном ряде. Несмотря на высокую точность восстановления, современные методы, как правило, не используют поведенческие шаблоны, которые могли бы повысить точность восстановления и интерпретируемость результатов, выступая в качестве эталона ожидаемого поведения.

Существенным ограничением применения нейросетевых моделей является высокая вариативность их архитектур, затрудняющая проектирование решений, адаптированных к особенностям конкретного временного ряда. В связи с этим в настоящее время активно развиваются методы поиска архитектур

нейронных сетей (Neural Architecture Search, NAS), применяемые в рамках AutoML и MLOps для автоматического проектирования и повышения точности нейросетевых моделей.

Исходя из рассмотренного выше, можно заключить, что на сегодняшний день задача разработки эффективных методов и алгоритмов восстановления пропущенных значений временных рядов остается в фокусе интенсивных научных исследований и практических разработок.

**Объект и предмет исследования.** *Объектом исследования* являются системы обработки и анализа потоковых данных. *Предметом исследования* выступают модели, методы и алгоритмы восстановления пропущенных значений в потоковых данных.

**Цель и задачи исследования.** *Цель* данной работы состояла в разработке новых и повышении эффективности существующих методов восстановления потоковых данных, представленных в форме многомерных временных рядов, на основе совместного использования нейросетевых моделей и поведенческих шаблонов. Данная цель предполагает решение следующих *задач*.

1. Разработать и исследовать комплекс нейросетевых моделей, методов и алгоритмов, предназначенных для восстановления потоковых данных, представленных в форме временных рядов, и включающий в себя следующие компоненты:
  - нейросетевые методы восстановления многомерных временных рядов в офлайн и онлайн режимах;
  - функцию потерь для обучения нейросетевых моделей восстановления временных рядов, учитывающую поведенческий контекст;
  - метод прогнозирования ошибки и времени обучения нейросетевых моделей восстановления временных рядов.
2. Провести вычислительные эксперименты с синтетическими и реальными временными рядами, исследующие эффективность предложенных нейросетевых моделей, методов и алгоритмов.

**Научная новизна** работы заключается в следующем:

1. Разработаны новые нейросетевые методы восстановления потоковых данных, представленных в форме временных рядов, в режимах офлайн и онлайн, впервые использующие концепцию поведенческих шаблонов, которая обеспечивает повышение точности восстановления.
2. Предложена новая функция потерь, обеспечивающая повышение точности восстановления потоковых данных, представленных в форме временных рядов, нейросетевыми моделями за счет учета поведенческого сходства подпоследовательностей во время обучения. Разработан эффективный параллельный алгоритм, который позволяет интегрировать предложенную функцию потерь в современные фреймворки глубокого обучения.

3. Разработан новый метод прогнозирования ошибки и времени обучения нейросетевых моделей восстановления потоковых данных, представленных в форме временных рядов, основанный на представлении обучаемых нейросетевых моделей в виде направленного ациклического графа и обеспечивающий возможность предварительной оценки точности нейросетевых моделей без необходимости их обучения.

**Теоретическая значимость** диссертационной работы состоит в следующем. В разработанных методах восстановления потоковых данных предложены оригинальные архитектуры нейросетевых моделей, позволяющие использовать в качестве дополнительного источника информации шаблоны, извлекаемые из исходных данных. Разработанная функция потерь нейросетевых моделей восстановления потоковых данных, в отличие от аналогов, обеспечивающих геометрическую интерпретацию точек, позволяет учитывать во время обучения поведенческую схожесть исходных данных. Предложена нейросетевая модель, позволяющая прогнозировать ошибку и время обучения моделей восстановления потоковых данных, представленных в форме многомерных временных рядов, без необходимости их обучения.

**Практическая значимость** диссертационной работы заключается в том, что разработанные нейросетевые модели, методы и алгоритмы могут быть использованы для решения задач, связанных с восстановлением временных рядов в интеллектуальных системах обработки потоковых данных. Использование предложенных методов и моделей в качестве инструментальных средств разработки цифровых продуктов позволит автоматизировать подготовку и обработку потоковых данных, содержащих пропуски, минимизируя участие человека за счет автоматизированного восстановления. Полученные методы могут быть использованы в системах AutoML, MLOps и иных интеллектуальных системах машинного обучения в качестве инструмента для повышения устойчивости и точности анализа потоковых данных, представленных в форме временных рядов.

**Методология и методы исследования.** Методологической основой диссертационной работы являются методы глубокого обучения, интеллектуального анализа данных и теория временных рядов. Для программной реализации разработанных подходов применялись методы объектно-ориентированного проектирования и разработки нейросетевых моделей с использованием фреймворка PyTorch.

**Положения, выносимые на защиту.** На защиту выносятся следующие новые научные результаты:

1. Разработан нейросетевой метод SANNI для восстановления потоковых данных, представленных в форме временных рядов, в режиме онлайн, который для повышения точности восстановления использует поведенче-

ские шаблоны, извлекаемые из репрезентативного фрагмента исходных данных.

2. Разработан нейросетевой метод SAETI для восстановления потоковых данных, представленных в форме временных рядов, в режиме офлайн, который для повышения точности восстановления использует поведенческие шаблоны, извлекаемые из репрезентативного фрагмента исходных данных.
3. Разработана функция потерь MPDE, предназначенная для обучения нейросетевых моделей восстановления потоковых данных, представленных в форме временных рядов. Предложенная функция представляет собой меру поведенческой схожести временных рядов, в отличие от традиционных функций потерь, основанных на геометрической интерпретации ошибок. Реализован параллельный алгоритм вычисления данной функции, обеспечивающий возможность ее использования в современных фреймворках глубокого обучения.
4. Разработан нейросетевой метод tsGAP для прогнозирования ошибки и времени обучения нейросетевых моделей восстановления потоковых данных, представленных в форме временных рядов, который для повышения точности прогноза использует графовое представление нейросетевых моделей восстановления.

**Степень достоверности результатов.** Эффективность разработанных методов и алгоритмов подтверждена результатами вычислительных экспериментов, проведенных на реальных и синтетических данных в соответствии с общепринятыми стандартами. Репродуцируемость полученных результатов обеспечивается размещением исходного кода, реализующего предложенные нейросетевые модели, методы и алгоритмы, в свободно доступных репозиториях в сети Интернет.

**Апробация работы.** Основные положения диссертационной работы, разработанные методы, алгоритмы и результаты вычислительных экспериментов докладывались на следующих международных и всероссийских научных конференциях:

- ПаВТ'2025: Всероссийская научная конференция с международным участием «Параллельные вычислительные технологии 2025», 8–10 апреля 2025 г., Москва;
- DAMDID'2024: International Conference on Data Analytics and Management in Data Intensive Domains, 23–25 October 2024, Nizhny Novgorod, Russia;
- ПаВТ'2024: Всероссийская научная конференция с международным участием «Параллельные вычислительные технологии 2024», 2–4 апреля 2024 г., Челябинск (*диплом I степени в конкурсе докладов молодых ученых*);

- ЦИСП'2023: Всероссийская научная конференция с международным участием «Цифровая индустрия: состояние и перспективы развития 2023», 21–23 ноября 2023 г., Челябинск;
- ПаВТ'2023: Всероссийская научная конференция с международным участием «Параллельные вычислительные технологии 2023», 28–30 марта 2023 г., Санкт-Петербург (*диплом II степени в конкурсе докладов молодых ученых*).

**Публикации.** Основные результаты диссертации опубликованы в *пяти научных работах* в журналах категорий К1 и К2 Перечня ВАК, в том числе 4 статьи опубликованы в журналах, входящих в Ядро РИНЦ (где одна из статей опубликована в журнале, который также входит в квартиль Q2 библиографической базы данных Scopus).

**Личный вклад.** В статьях [1,5] научному руководителю М.Л. Цымблеру принадлежит постановка задачи, А.А. Юртину — все полученные результаты.

**Соответствие диссертации паспорту научной специальности.** Тематика, содержание и полученные результаты диссертационного исследования соответствуют следующим направлениям исследований паспорта специальности 2.3.5 «Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей»: 4. Интеллектуальные системы машинного обучения, управления базами данных и знаний, инструментальные средства разработки цифровых продуктов.

**Структура и объем работы** Диссертация состоит из введения, четырех глав, заключения, списка литературы и трех приложений. Объем диссертации составляет 193 страницы, объем списка литературы — 168 наименований.

## Содержание работы

**Во введении** приводится обоснование актуальности темы и степень ее разработанности; формулируются цели и задачи исследования; раскрываются новизна, теоретическая и практическая значимость полученных результатов; формулируется методологическая основа диссертационного исследования; дается обзор содержания диссертации.

**В первой главе, «Современные методы восстановления потоковых данных, представленных в форме временных рядов»**, дается обзор современных нейросетевых моделей, методов и алгоритмов восстановления потоковых данных, представленных в форме временных рядов. Представлены два основных подхода к восстановлению: аналитические и нейросетевые методы. Для каждой группы приводится классификация с описанием типичных представителей. Даны формальные определения и введена нотация, используемая в дальнейшем для описания методов восстановления потоковых

данных, представленных в форме временных рядов. Проведен обзор публикаций, наиболее близких к тематике диссертационной работы, в котором рассматриваются нейросетевые методы восстановления, различные функции потерь, а также существующие подходы к предварительной оценке точности нейросетевых моделей восстановления потоковых данных, представленных в форме временных рядов.

Одномерный временной ряд  $T = \{t_i\}_{i=1}^n$ ,  $t_i \in \mathbb{R}$ , длины  $n = |T|$ , рассматривается как хронологически упорядоченная последовательность значений. Его подпоследовательность длины  $m$ , начиная с  $i$ -го элемента, обозначается  $T_{i,m} = \{t_k\}_{k=i}^{i+m-1}$ , а множество всех подпоследовательностей длины  $m$  обозначается как  $S_T^m$ ,  $|S_T^m| = n - m + 1$ . Подпоследовательность длины  $\ell < m$  внутри  $T_{i,m}$  называется *окном*, мощность множества окон равна  $c = m - \ell + 1$ . Многомерный временной ряд  $\mathbf{T} = [\{T^{(k)}\}_{k=1}^d]^\top$  состоит из  $d$  синхронизированных одномерных рядов. Его подпоследовательности и точки обозначаются  $\mathbf{T}_{i,m} = [\{T_{i,m}^{(k)}\}_{k=1}^d]^\top$  и  $\mathbf{t} = [\{t^{(k)}\}_{k=1}^d]^\top$ , соответственно. Подпоследовательности  $\mathbf{T}_{i,m}$  делятся на *полные*  $\mathbf{S}_T^m$  (не содержащие пропусков) и *неполные*  $\overset{\circ}{\mathbf{S}}_T^m$  (существует хотя бы один пропуск  $t^{(k)} = \text{NaN}$ ). Восстановленные подпоследовательности  $\overset{\bullet}{\mathbf{T}}_{i,m}$  формируются из неполных подпоследовательностей путем замены всех пропусков на синтетические значения; множество всех восстановленных подпоследовательностей обозначается  $\overset{\bullet}{\mathbf{S}}_T^m$ .

Поведенческие шаблоны реализованы на базе концепции снippetов, предложенной Кеогом и др. Снippetы представляют собой подпоследовательности временного ряда длины  $m$ . Для временного ряда  $T$  формируется набор не перекрывающихся сегментов  $Seg_T^m = \{Seg_i\}_{i=1}^{n/m}$ ,  $Seg_i = T_{m \cdot (i-1) + 1, m}$ , из которых выбираются  $K$  снippetов  $C_T^m = \{C_i\}_{i=1}^K$ ,  $C_i \in Seg_T^m$ . Каждый снippet характеризуется следующими атрибутами: индекс сегмента, указывающий на первую точку снippetа в ряде,  $C_i.index$ , множество ближайших соседей  $C_i.NN$ , включающее близкие по функции расстояния MPdist подпоследовательности, и покрытие  $C_i.frac$  — доля подпоследовательностей ряда, относящихся к данному снippetу. Для многомерного ряда  $\mathbf{T}$  формируется словарь снippetов  $\mathbf{C}_T^m = \bigcup_{k=1}^d C_{T^{(k)}}^m$ , объединяющий снippetы по всем измерениям. В дальнейшем для упрощения изложения под шаблоном будет подразумеваться снippet.

В данной главе выделены ключевые инструментальные средства реализации каждого этапа жизненного цикла нейросетевых моделей восстановления потоковых данных, представленных в форме временных рядов.

1. Методы восстановления временных рядов в режиме офлайн, применяемые на этапе предварительной обработки данных для корректного заполнения пропусков в исходных временных рядах.

2. Методы восстановления временных рядов в режиме онлайн, применяемые для заполнения пропусков в потоковых данных непосредственно перед подачей на вход другой нейросетевой модели.
3. Методы предварительной оценки, которые позволяют оценить точность архитектур целевой нейросетевой модели без полного ее обучения.
4. Функции потерь, используемые для обучения нейросетевых моделей восстановления временных рядов.

Указанные инструментальные средства решающим образом влияют на точность восстановления временных рядов и качество последующей обработки потоковых данных.

**Во второй главе, «Нейросетевые методы и модели восстановления потоковых данных, представленных в форме временных рядов»,** представлены новые методы восстановления потоковых данных, представленных в форме временных рядов, в режимах онлайн и офлайн, получившие названия SANNI (Snippet and Neural Network based Imputation) и SAETI (Snippet based Autoencoder for Timeseries Imputation). Подробно описаны основные этапы обработки входных данных, архитектура нейросетевых моделей и методы их обучения. Описан способ применения поведенческих шаблонов совместно с нейросетевыми подходами для повышения точности восстановления пропущенных значений в различных сценариях восстановления.

Метод SANNI предназначен для онлайн-восстановления потоковых данных, представленных в форме временных рядов. На вход поступает репрезентативный фрагмент  $\mathbf{T}$  длины  $n$  с  $d$  измерениями, и длина подпоследовательности  $m$ . Предполагается, что репрезентативный фрагмент  $\mathbf{T}$  не содержит пропущенных значений и включает наблюдения, относящиеся ко всем основным активностям объекта, деятельность которого описывает данный временной ряд. Метод состоит из двух фаз: предобработки и восстановления. На этапе предобработки Препроцессор формирует обучающие выборки и словарь шаблонов  $\mathcal{C}_T^m$ . На основе выборок выполняется обучение нейросетевых моделей: Распознавателя, классифицирующего активности, и Реконструктора, восстанавливающего пропущенные значения. Фаза восстановления выполняется последовательным применением обученных моделей к подпоследовательностям с пропусками.

Каждый шаблон  $C_i$  обобщает характерный тип поведения в рамках одной активности и ассоциирован с группой схожих подпоследовательностей. На основе близости подпоследовательностей к шаблонам формируется естественное разбиение временного ряда на классы, формальное множество которых задается как  $\Psi = \{\psi\}_{\psi=1}^K$ , где  $\psi$  — номер класса. Функция *классификации подпоследовательности*  $f_{\text{class}}$  сопоставляет каждой одномерной подпоследовательности  $T_{i,m}^{(k)}$  метку класса  $\psi \in \Psi$ , определяемую по принадлеж-

ности к множеству ближайших соседей  $C_{\psi}^{(k)}.NN$  соответствующего шаблона. Для многомерного случая каждая подпоследовательность представляется набором одномерных подпоследовательностей, для которых вычисляются индивидуальные метки. Совокупность этих меток образует *вектор меток*  $\psi = \{\psi_i\}_{i=1}^d$ , а множество таких векторов обозначается как  $\Psi$ . Функция *многомерной классификации*  $f_{\text{class}}$  сопоставляет каждой многомерной подпоследовательности ее вектор меток, реализуя обобщение процедуры одномерной классификации на многомерный случай:

$$\begin{aligned} f_{\text{class}} : S_T^m &\rightarrow \Psi, f_{\text{class}}(T_{i,m}^{(k)}) = \psi_i^{(k)}, T_{i,m}^{(k)} \in C_{\psi}^{(k)}.NN, \\ f_{\text{class}} : \mathbf{S}_T^m &\rightarrow \Psi, f_{\text{class}}(\mathbf{T}_{i,m}) = \psi_i. \end{aligned} \quad (1)$$

Поскольку истинные метки могут быть определены только для подпоследовательностей без пропусков, функция  $f_{\text{class}}$  применима исключительно к полным подпоследовательностям. Для обобщения ее действия на множество подпоследовательностей  $\overset{\circ}{S}_T^m$ , включающее последовательности с пропусками, вводится расширенная функция  $\overset{\circ}{f}_{\text{class}}$ , формирующая вектор прогнозных меток  $\overset{\circ}{\psi} \in \overset{\circ}{\Psi}$ . Нейросетевая модель Распознаватель аппроксимирует функцию  $\overset{\circ}{f}_{\text{class}}$  и формирует вектор прогнозных меток  $\overset{\circ}{\psi} \in \overset{\circ}{\Psi}$  для каждой входной подпоследовательности. Нейросетевая модель принимает на вход многомерную подпоследовательность  $\mathbf{T}_{i,m}$  с пропусками во всех значениях последней точки. Перед обработкой входных данных пропущенные значения заменяются на значение  $\text{NIL} = -1$ . Входная подпоследовательность последовательно обрабатывается тремя сверточными слоями, одним рекуррентным слоем и двумя полносвязными слоями. Каждый сверточный слой содержит 256 фильтров с размером ядра 5, после которых применяется Max-pooling с окном 2. Рекуррентный слой реализован с помощью GRU-слоя с размером скрытого состояния равным 32. Первый полносвязный слой содержит  $32 \cdot m$  нейронов. Второй слой, состоящий из  $d \cdot K$  нейронов, формирует выходную матрицу вероятностей  $\mathbf{P} \in \mathbb{R}^{d \times K}$ . Вектор прогнозных меток  $\overset{\circ}{\psi}$  формируется путем применения операции  $\arg \max$  по столбцам к каждой строке матрицы вероятностей  $\mathbf{P}$ , выбирающей наиболее вероятный класс для соответствующего измерения. На основе  $\overset{\circ}{\psi}$  формируется шаблонная подпоследовательность  $\mathbf{F} \in \mathbb{R}^{m,d}$ . Каждое измерение шаблонной подпоследовательности соответствует шаблону, извлеченному из словаря  $\mathbf{C}_T^m$ . Шаблон выбирается на основе предсказанной метки класса для соответствующего измерения входной подпоследовательности:

$$\mathbf{F}^{(k)} = C_{\psi_k}^{\bullet}, C_{\psi_k}^{\bullet} \in C_{T^{(k)}}^m, C_{T^{(k)}}^m \in \mathbf{C}_T^m, \psi_i = \arg \max_{1 \leq j \leq K} P_i^j, 1 \leq k \leq d. \quad (2)$$

На вход нейросетевой модели Реконструктор поступает многомерная подпоследовательность временного ряда  $\mathbf{T}_{i,m}$  с пропусками и шаблонная подпоследовательность  $\mathbf{F}$ . Во время обработки входных данных формируется набор  $\mathbf{M}$ , состоящий из  $d$  подпоследовательностей-макетов  $\mathbf{M}_k \in \mathbb{R}^{m \times 2}$ . Первое и второе измерение  $k$ -й подпоследовательности-макета содержат  $k$ -е измерения  $\mathbf{T}_{i,m}$  и  $\mathbf{F}$  соответственно. Набор  $\mathbf{M}$  обрабатывается набором из  $k$  независимых последовательностей слоев. Каждая последовательность слоев, преобразующая одну подпоследовательность-макет в векторное представление, включает три сверточных слоя, рекуррентный слой и полносвязный слой. Сверточные слои содержат 64, 128 и 256 фильтров с размером ядра, равным 5. Рекуррентный слой реализуется с помощью GRU-слоя с размером скрытого состояния, равным 200. Полносвязный слой с  $m$  нейронами формирует векторное представление подпоследовательности-макета. Векторные представления, полученные с каждой последовательности слоев, объединяются в матрицу векторных представлений размером  $m \times d$ . Эта матрица поступает на вход рекуррентного слоя на основе GRU с размером скрытого состояния, равным 200. Выход рекуррентного слоя передается на вход полносвязного слоя, состоящему из  $d$  нейронов, который формирует восстановленную многомерную точку  $\hat{\mathbf{t}}$ .

Для обучения модели Распознаватель формируется обучающая выборка  $D_{\text{Recognizer}}$ , определяемая следующим образом:

$$D_{\text{Recognizer}} = \{ \langle \mathbf{X}, \mathbf{Y} \rangle \mid X^{(k)} = T_{i,m-1}^{(k)} \cdot \text{NIL}, \mathbf{Y} = \boldsymbol{\psi}_i, \boldsymbol{\psi}_i = \mathbf{f}_{\text{class}}(\mathbf{T}_{i,m}), \quad (3) \\ 1 \leq i \leq n - m + 1, 1 \leq k \leq d, 1 \leq s \leq K \},$$

где  $\mathbf{X}$  и  $\mathbf{Y}$  — входные и выходные данные модели, « $\cdot$ » обозначает операцию конкатенации. Аналогично, для обучения модели Реконструктор формируется обучающая выборка, которая имеет следующее определение:

$$D_{\text{Reconstructor}} = \{ \langle \mathbf{X}, \mathbf{Y} \rangle \mid X^{(k)} = T_{i,m-1}^{(k)} \cdot \text{NIL}, Y^{(k)} = t_{i+m}^{(k)}, \quad (4) \\ 1 \leq i \leq n - m + 1, 1 \leq k \leq d \}.$$

Во время обучения нейросетевой модели Распознаватель в качестве функции потерь используется кросс-энтропия, оценивающая соответствие предсказанных векторов меток и истинных меток многомерных подпоследовательностей. Для нейросетевой модели Реконструктор применяется среднеквадратичная ошибка, которая измеряет отклонение предсказанных значений восстановленной многомерной точки от эталонных значений.

Метод SAETI является инструментальным средством разработки, решающим задачу офлайн-восстановления потоковых данных, представленных в форме временных рядов. На вход метода поступает репрезентативный фрагмент многомерного временного ряда  $\mathbf{T}$  длины  $n$ , содержащего  $d$  измерений.

Репрезентативный фрагмент может содержать ограниченное число непересекающихся подпоследовательностей с пропущенными значениями, количество которых не превышает порог  $\alpha \cdot (n - m + 1)$ , где  $0 < \alpha < 1$ , типичное значение  $\alpha = 0.5$ . Метод, аналогично SANNI, включает две фазы: обучение и восстановление. На фазе обучения из репрезентативного фрагмента формируются обучающие выборки, которые используются для обучения нейросетевых моделей. На фазе восстановления каждая подпоследовательность с пропущенными значениями в любом измерении любой многомерной точки последовательно обрабатывается нейросетевыми моделями. В результате формируется подпоследовательность, пропуски в которой были заменены на синтетические значения.

Распознаватель метода SAETI принимает на вход многомерную подпоследовательность временного ряда, в которой пропущенные значения заменены на нули. На выходе моделей формируется матрица вероятностей  $\mathbf{P}$ , которая является основой для формирования шаблонной подпоследовательностей  $\mathbf{F}$  (см. формулу 2). В методе SAETI сверточные слои Распознавателя имеют меньшее количество фильтров при сохранении размера ядра: 128, 128 и 64 соответственно. Первый полносвязный слой состоит из 32 нейронов и принимает на вход скрытое состояние, соответствующее последнему временному шагу последовательности. На основе шаблонной и входной подпоследовательности формируется вход нейросетевой модели Реконструктор — макет  $\mathbf{M}$ . Реконструктор в данном методе реализуется с помощью модифицированного Автоэнкодера. С целью повышения качества кодирования подпоследовательности и повышения точности восстановления на вход Энкодера дополнительно подается шаблонная подпоследовательность. При этом на выходе Декодера формируется только восстановленная подпоследовательность без шаблонов.

Энкодер состоит из  $d$  последовательностей слоев, каждая из которых включает три сверточных слоя с 128, 64 и 32 фильтрами размером ядра 5. Выходы сверточных слоев объединяются в матрицу размером  $32d \times m$ , подаваемую на двунаправленный GRU-слой с размером скрытого состояния  $m$ . Рекуррентный слой формирует векторное представление подпоследовательности. Полученное векторное представление поступает на вход Декодера, структура которого зеркально повторяет Энкодер с транспонированными сверточными слоями вместо обычных.

Обучающая выборка для нейросетевой модели Распознаватель формируется следующим образом. Входными данными полагаются подпоследовательности без пропусков, в которые пропуски были внесены искусственно. Выходными данными полагается вектор классов, соответствующий истинным активностям в каждом измерении исходной подпоследовательности. Входными данными для нейросетевой модели Реконструктор служат подпоследовательности, в которых фактические пропуски заменены нулевыми значениями и дополнительно внесены искусственные пропуски. Выходными дан-

ными являются исходные подпоследовательности без каких-либо модификаций. Функция потерь Реконструктора вычисляется как среднеквадратичная ошибка между восстановленными и истинными значениями, при этом учитываются только те элементы подпоследовательности, для которых доступны исходные данные.

**Третья глава, «Оценка точности нейросетевых моделей восстановления потоковых данных, представленных в форме временных рядов»**, посвящена новым методам анализа нейросетевых моделей восстановления потоковых данных, представленных в форме временных рядов. В главе предложена новая функция потерь MPDE (Mean Profile Distance Error), повышающая точность нейросетевых моделей восстановления за счет учета поведенческих особенностей подпоследовательностей в процессе обучения. Описан новый метод tsGAP (time-series Graph-Attention Performance Prediction model), предназначенный для прогнозирования ошибки и времени обучения нейросетевых моделей восстановления потоковых данных, представленных в форме временных рядов.

Функция MPDE определяется как среднее значение расстояний между парами окон сравниваемых подпоследовательностей. Каждое окно имеет длину  $\ell$ , где  $\ell < m$ , и предполагается, что длина окна пропорциональна длине подпоследовательности  $m$ . Эта пропорция задается параметром  $\gamma$ , удовлетворяющим условию  $0 < \gamma < 1$ , так что  $\ell = \lfloor \gamma m \rfloor$ . Общее число окон обозначим  $c = m - \ell + 1$ . При этом сравнение проводится между окнами с одинаковыми индексами. Для вычисления расстояний между окнами подпоследовательностей используются евклидово расстояние и его z-нормализованная версия, обозначаемые как  $ED(\cdot, \cdot)$  и  $\widehat{ED}(\cdot, \cdot)$  соответственно. Для вычисления евклидовых расстояний под корень вносится машинный эпсилон  $\varepsilon > 0$ .

$$ED(X, Y) = \sqrt{\sum_{i=1}^{\ell} (x_i - y_i)^2 + \varepsilon}, \quad \widehat{ED}(X, Y) = \sqrt{\sum_{i=1}^{\ell} (\hat{x}_i - \hat{y}_i)^2 + \varepsilon}, \quad (5)$$

где z-нормализация представляет собой следующее:

$$\hat{x}_i = \frac{x_i - \mu_x}{\sigma_x}, \quad \mu_x = \frac{1}{\ell} \sum_{i=1}^{\ell} x_i, \quad \sigma_x = \sqrt{\frac{1}{\ell - 1} \sum_{i=1}^{\ell} (x_i - \mu_x)^2 + \varepsilon}. \quad (6)$$

Веса указанных слагаемых,  $\alpha$  и  $\beta$  соответственно, представляют собой вещественные числа из отрезка  $[0, 1]$  и являются параметрами функции потерь.

Функция потерь MPDE вычисляется следующим образом:

$$\text{MPDE}(X, Y) = \frac{1}{2 \cdot c} \sum_{i=1}^c \left( \alpha \text{ED}(X_{i, i+\ell}, Y_{i, i+\ell}) + \beta \widehat{\text{ED}}(X_{i, i+\ell}, Y_{i, i+\ell}) \right), \quad (7)$$

$$0 \leq \alpha, \beta \leq 1.$$

Полагая, что  $X, Y \in \mathbb{R}^m$ ,  $0 < \alpha, \beta < 1$ ,  $\ell < m$ ,  $c = m - \ell + 1$ ,  $\varepsilon = 10^{-6}$  в диссертационном исследовании доказаны следующие утверждения о свойствах функции потерь  $\text{MPDE}(\cdot, \cdot)$ .

**Утверждение 1** Функция  $\text{MPDE}(X, Y)$  положительна.

**Утверждение 2** Глобальный минимум функции  $\text{MPDE}(\cdot, \cdot)$  достигается при  $X = Y$ .

**Утверждение 3** Рассмотрим нейросетевую модель, порождающую выходной вектор

$$X = F(Z, \theta) \in \mathbb{R}^m, \quad (8)$$

где  $\theta$  обозначает совокупность всех весов нейросетевой модели, а  $Z$  входной вектор модели. Пусть целевой вектор  $Y \in \mathbb{R}^m$  фиксирован, а функция потерь имеет вид

$$L(X, Y) = \text{MPDE}(X, Y), \quad (9)$$

где  $\ell < m$ ,  $c = m - \ell + 1$ , а  $\varepsilon > 0$ ,  $0 < \alpha, \beta < 1$ .

Градиент функции потерь MPDE по выходам последнего слоя нейросети вычисляется как сумма вкладов всех окон, содержащих соответствующую компоненту. Для компоненты  $x_k$  выходного вектора  $X$  получаем

$$\begin{aligned} \frac{\partial \text{MPDE}}{\partial x_k} &= \frac{1}{2c} \sum_{i=i_{\min}}^{i_{\max}} \left( \alpha \frac{x_k - y_k}{\text{ED}(X_{i, \ell}, Y_{i, \ell})} + \right. \\ &\quad \left. + \beta \frac{1}{\widehat{\text{ED}}(X_{i, \ell}, Y_{i, \ell})} \sum_{j=1}^{\ell} (\hat{x}_{i+j} - \hat{y}_{i+j}) \frac{\partial \hat{x}_{i+j}}{\partial x_k} \right). \end{aligned} \quad (10)$$

$$i_{\min} = \max(1, k - \ell), i_{\max} = \min(k, m - \ell + 1).$$

где частная производная нормализации по элементу окна  $x_k$  равна

$$\frac{\partial \hat{x}_j^{(i)}}{\partial x_k} = \frac{\delta_{jk} - \frac{1}{\ell}}{\sigma_{X_{i, \ell}}} - \frac{x_j^{(i)} - \mu_{X_{i, \ell}}}{(\sigma_{X_{i, \ell}})^2} \cdot \frac{x_k^{(i)} - \mu_{X_{i, \ell}}}{(\ell - 1) \sigma_{X_{i, \ell}}}. \quad (11)$$

**Утверждение 4** Вычислительная сложность функции  $\text{MPDE}(X, Y)$   $O(m^2)$ .

Частная производная функции потерь по переменной  $x_i$  может быть представлена как сумма трех компонент:

$$\frac{\partial \text{MPDE}(X, Y)}{\partial x_i} = A + B + C, \quad (12)$$

где  $A$ ,  $B$  и  $C$  вычисляются следующим образом:

$$\begin{aligned} A &= \frac{\alpha}{2 \cdot c} \cdot \sum_{j_{\min}}^{j_{\max}} \frac{x_i - y_i}{\text{ED}(X_{j, j+l}, Y_{j, j+l})}, \quad B = \frac{\beta}{2 \cdot c} \cdot \sum_{j_{\min}}^{j_{\max}} \frac{\hat{x}_i - \hat{y}_i}{\text{ED}(X_{j, j+l}, Y_{j, j+l})}, \\ C &= \frac{\beta}{2 \cdot \ell \cdot c} \cdot \sum_{j_{\min}}^{j_{\max}} \frac{\partial \text{ED}(X_{j, j+l}, Y_{j, j+l})}{\partial \mu_j} \cdot \frac{\partial \mu_j}{\partial x_i} + \\ &+ \frac{\beta}{2 \cdot \sigma \cdot c} \cdot \sum_{j_{\min}}^{j_{\max}} \frac{\partial \text{ED}(X_{j, j+l}, Y_{j, j+l})}{\partial \sigma_j} \cdot \frac{\partial \sigma_j}{\partial x_i}. \end{aligned} \quad (13)$$

Слагаемое  $A$  вычисляется как сумма разностей между точками восстановленного и истинного временного ряда, деленная на евклидово расстояние соответствующих окон. Данное слагаемое минимизирует отклонения между отдельными точками, обеспечивая корректное воспроизведение масштаба и среднего значения исходного временного ряда. Для воспроизведения поведенческого сходства подпоследовательностей в слагаемом  $B$  используются  $z$ -нормализованные окна. Нормализация устраняет влияние масштаба и среднего значения и делает сопоставимыми подпоследовательности с различной амплитудой и смещением. В результате сравниваются относительные изменения точек внутри окна, включая локальные тренды, пики, впадины и др. Слагаемое  $B$  стимулирует нейросетевую модель воспроизводить поведенческие особенности подпоследовательностей. Слагаемое  $C$  учитывает косвенное влияние каждой точки на нормализацию окна подпоследовательности: изменение одного значения смещает параметры нормализации ( $\mu$  и  $\sigma$ ) и, соответственно, нормализованные ошибки остальных точек, создавая каскадный эффект.

В случае многомерного ряда значение функции потерь вычисляется как среднее арифметическое значений функции MPDE по всем измерениям между соответствующими одномерными подпоследовательностями. Для обеспечения практической применимости функции потерь MPDE в современных фреймворках глубокого обучения был разработан параллельный алгоритм ее вычисления. Алгоритм принимает на вход два батча по  $b$  многомерных подпоследовательностей длины  $m$  с  $d$  измерениями (восстановленные и истинные) и выполняет следующие шаги. Из каждой подпоследовательности извлекаются все скользящие окна длины  $\ell$ . Для каждого окна вычисляются параметры нормализации. Затем каждое окно нормализуется по собственному средне-

му и дисперсии. На следующем шаге для исходных и нормализованных окон вычисляются поэлементные евклидовы расстояния между восстановленными и истинными значениями. На завершающем этапе формируется взвешенная сумма расстояний с коэффициентами, регулирующими вклад каждой компоненты, которая усредняется по всем элементам батча.

---

**Алг. 1.** TorchMPDE(IN:  $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{b \times m \times d}$ ,  $\alpha, \beta$ ; OUT: MPDE)

---

- 1:  $\mathbf{X}_{\text{slice}} \leftarrow \text{unfold}(\mathbf{X}, \ell, 1)$ ;  $\mathbf{Y}_{\text{slice}} \leftarrow \text{unfold}(\mathbf{Y}, \ell, 1)$
  - 2:  $\mu_x \leftarrow \text{mean}(\mathbf{X}_{\text{slice}}, 2)$ ;  $\mu_y \leftarrow \text{mean}(\mathbf{Y}_{\text{slice}}, 2)$
  - 3:  $\sigma_x \leftarrow \text{sqrt}(\text{var}(\mathbf{X}_{\text{slice}}, 2) + \varepsilon)$ ;  $\sigma_y \leftarrow \text{sqrt}(\text{var}(\mathbf{Y}_{\text{slice}}, 2) + \varepsilon)$
  - 4:  $\mathbf{X}_{\text{norm}} \leftarrow (\mathbf{X}_{\text{slice}} - \mu_x)/\sigma_x$ ;  $\mathbf{Y}_{\text{norm}} \leftarrow (\mathbf{Y}_{\text{slice}} - \mu_y)/\sigma_y$
  - 5:  $\text{ED2}_{\text{norm}} \leftarrow \text{sum}((\mathbf{X}_{\text{norm}} - \mathbf{Y}_{\text{norm}})^2, 3)$ ;  $\text{ED2} \leftarrow \text{sum}((\mathbf{X}_{\text{slice}} - \mathbf{Y}_{\text{slice}})^2, 3)$
  - 6:  $\text{ED}_{\text{norm}} \leftarrow \text{sqrt}(\text{ED2}_{\text{norm}} + \varepsilon)$ ;  $\text{ED} \leftarrow \text{sqrt}(\text{ED2} + \varepsilon)$
  - 7:  $\text{MPDE} \leftarrow \text{mean}(\alpha \cdot \text{ED} + \beta \cdot \text{ED}_{\text{norm}})/2$
  - 8: **return** MPDE
- 

Метод tsGAP предназначен для прогнозирования ошибки и времени обучения нейросетевых моделей восстановления потоковых данных, представленных в форме временных рядов. Анализируемая на этапе выбора нейросетевая модель, называемая *целевой*, может быть представлена как ориентированный ациклический граф: вершины графа соответствуют слоям модели, а дуги — связям между ними. *Набор типов слоев*  $\mathcal{C} = \{C_i\}_{i=1}^c, C_i \in \mathbb{Z}$ , представляет упорядоченный набор из  $c$  целочисленных кодов, каждому из которых соответствует допустимый тип нейросетевого слоя или операции. *Набор функций активации*  $\mathcal{A} = \{A_i\}_{i=1}^a, A_i \in \mathbb{Z}$  представляет упорядоченный набор из  $a$  целочисленных кодов, каждый из которых соответствует допустимой функции активации, используемой в слоях нейросетевой модели.

Формально целевая нейросетевая модель может быть представлена парой объектов: упорядоченный набор слоев  $L$  и матрица связей  $R \in \mathbb{R}^{\lambda \times 2}$ . Набор слоев  $L$  содержит  $\ell$  четырехэлементных кортежей, каждый из которых описывает один слой нейронной сети:

$$L_k = (C_i, p_1, p_2, A_j), \quad 1 \leq i \leq c, \quad (14)$$

$$1 \leq j \leq a, p_1, p_2 \in \mathbb{R}, 1 \leq k \leq \ell, C_i \in \mathcal{C}, A_j \in \mathcal{A},$$

где  $C_i$  — код слоя,  $p_1$  и  $p_2$  — числовые параметры слоя,  $A_j$  — код функции активации. Связи между слоями нейронной модели задаются матрицей  $R$ , строки которой соответствуют дугам ориентированного графа. Каждая дуга задает направление передачи данных от одного слоя к другому и определяется парой индексов, указывающих на исходный и целевой слой:

$$R(i, \cdot) = (r_j, r_k), \quad 1 \leq i \leq \lambda, \quad 1 \leq r_j, r_k \leq \ell, \quad i < j, \quad (15)$$

где  $r_j$  — индекс слоя-источника,  $r_k$  — индекс целевого слоя,  $\lambda$  — общее количество связей в нейросетевой модели.

Метод tsGAP включает следующие компоненты, последовательно обрабатывающие входные данные: Автоэнкодер целевой нейросетевой модели, Энкодер параметров обучения и Агрегатор признаков. Автоэнкодер формирует векторное представление целевой нейросетевой модели, обозначаемое как  $Z \in \mathbb{R}^z$ . Энкодер формирует векторное представление параметров обучения целевой нейросетевой модели. Векторные представления передаются на вход Агрегатору, который на выходе продуцирует прогноз в виде вектора из двух значений: ошибки нейросетевой модели и времени обучения.

Каждый слой из набора слоев  $L$  проходит предварительную обработку, включающую следующие этапы: заполнение, нормализация параметров и кодирование. В результате предварительной обработки набор слоев  $L$  преобразуется в матрицу слоев  $\hat{L} \in \mathbb{R}^{\ell \times (c+a+2)}$ . На этапе заполнения входной набор слоев  $L$  приводится к фиксированной длине  $\ell$  путем добавления специальных фантомных слоев. Числовые параметры слоя  $p_1$  и  $p_2$  перед подачей на вход нейронной сети подвергаются минимаксной нормализации. Коды слоя  $C_i$  и функции активации  $A_j$  преобразуются с использованием one-hot кодирования. С помощью one-hot кодирования индексов матрица связей  $R$  преобразуется в нормализованную матрицу связей  $\hat{R} \in \mathbb{R}^{\lambda \times 2\ell}$ .

Для формирования векторного представления целевой модели, на вход Автоэнкодера подается нормализованная матрица слоев  $\hat{L}$ , исходная матрица связей  $R$  и ее нормализованная версия  $\hat{R}$ . Автоэнкодер включает четыре подсети. Первые две подсети представляют собой Энкодер слоев и Энкодер связей, которые отвечают за построение векторного представления целевой нейросетевой модели  $Z \in \mathbb{R}^z$ . Оставшиеся две подсети представляют собой Декодер слоев и Декодер связей. Декодеры выполняют восстановление графового представления нейросетевой модели из векторного.

Энкодер связей принимает на вход нормализованную матрицу связей  $\hat{R}$  и преобразует ее в векторное представление размерности  $z$ . Преобразование выполняется последовательно тремя полносвязными слоями: первые два содержат по  $2\lambda\ell$  нейронов, а последний слой включает  $z$  нейронов. Энкодер слоев принимает на вход нормализованную матрицу слоев и матрицу связей. На выходе данной подсети формируется векторное представление слоев. Данная нейросетевая модель включает пять GATConv слоев и один полносвязный слой. Первые четыре слоя GATConv имеют размерностью скрытого представления, равную 64, тогда как последний слой имеет размерность 32. Последний полносвязный слой состоит из  $z$  нейронов. Векторные представления слоев и связей подаются на вход полносвязного слоя, содержащего  $z$  нейронов. Данный слой формирует итоговое векторное представление модели, обозначаемое как  $Z$ .

Для декодирования целевой модели векторное представление  $Z$  поступает на вход одного полносвязного слоя, содержащего  $z$  нейронов. Выход данного слоя поступает на вход Декодера слоев и Декодера связей. Декодер связей с помощью трех последовательно применяемых полносвязных слоев формирует декодированную матрицу связей  $R^* \in \mathbb{R}^{\lambda \times 2}$ . Каждый из этих слоев содержит  $2\lambda\ell$  нейронов. Декодер слоев преобразует векторное представление и матрицу связей  $R^*$  с помощью полносвязного слоя с  $32\ell$  нейронами и пяти слоев GATConv, формирующих итоговую декодированную матрицу слоев  $\widehat{L}^*$ . Первые четыре слоя GATConv имеют размер скрытого представления 64, последний слой —  $c + a + 2$ .

На вход Энкодеру параметров обучения поступает вектор  $G \in \mathbb{R}^{10}$ , включающий гиперпараметры обучения и характеристики вычислительного устройства. Архитектура Энкодера состоит из слоя с механизмом внимания со скрытым представлением размерности 10 и двух полносвязных слоев по 10 нейронов, формирующих векторное представление параметров обучения. Агрегатор принимает векторные представления целевой нейросетевой модели и параметров ее обучения. Агрегатор обрабатывает входные данные с помощью последовательности нейросетевых слоев, включающую один слой с механизмом внимания и пять полносвязных слоев. Скрытое состояние слоя внимания имеет размерность  $z + 10$ . Первые четыре полносвязных слоя содержат по 200 нейронов, последний слой состоит из двух нейронов, формирующих выходной вектор  $V \in \mathbb{R}^2$ . Первый элемент выходного вектора соответствует ошибке модели, второй — прогнозу времени обучения.

Для обучения описанной выше модели используется упорядоченный набор из  $n$  четырехэлементных кортежей  $M = \{(\widehat{L}_i, R_i, G_i, V_i)\}_{i=1}^n$ , где  $\widehat{L}_i$  — матрица слоев,  $R_i$  — матрица связей между слоями,  $G_i$  — вектор параметров обучения,  $V_i$  — вектор значений качества модели. Входные данные представляют собой кортеж, включающий графовое представление целевой модели и вектор параметров обучения  $G_i$ . Описание целевой модели передается в виде нормализованной матрицы слоев  $\widehat{L}_i$  и матрицы связей  $R_i$ . Выходными данными полагаются кортеж, содержащий подаваемую на вход целевую модель, и вектор параметров качества  $V_i$ .

Ошибка нейросетевой модели представлена взвешенной суммой компонент трех типов: вероятностных (прогноз наличия связей между слоями, бинарная кросс-энтропия), классификационных (определение типа слоя и функции активации, кросс-энтропия) и регрессионных (прогноз параметров слоев, оценка точности предсказания ошибки и времени обучения, функция потерь Хубера).

**Четвертая глава, «Вычислительные эксперименты»**, посвящена вычислительным экспериментам, исследующим эффективность предложенных методов, нейросетевых моделей и алгоритмов. В главе резюмированы наборы данных, которые использовались в экспериментах, их категории, сце-

нарии формирования пропусков и метрики для оценки качества восстановления потоковых данных, представленных в форме временных рядов. Глава содержит сравнительный анализ предложенных методов с передовыми аналогами.

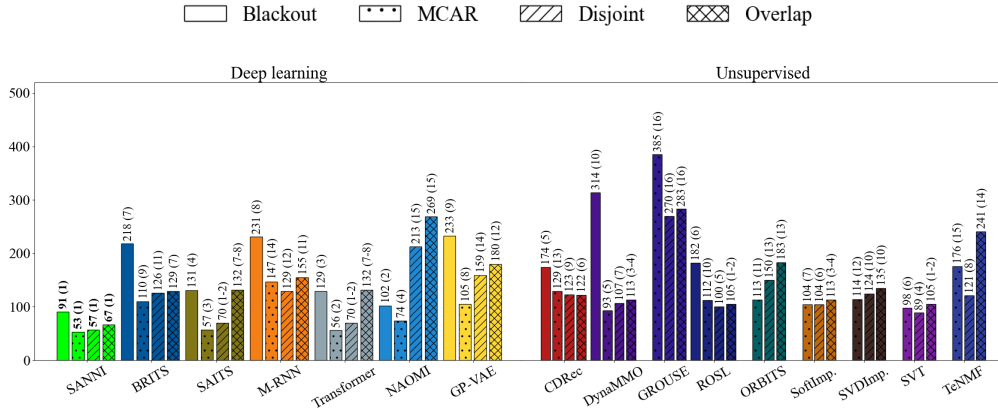


Рис. 1: Сравнение ошибки восстановления метода SANNI с аналогами на рядах с активностями,  $RMSE \times 10^{-3}$

Метод SANNI достигает наивысшей средней точности (см. рис. 1), превышая средний уровень конкурентов на 56%, а по сравнению с наилучшим аналогом — на 17%. В сценарии Blackout, имитирующем условия онлайн-режима, SANNI успешно восстанавливает пропущенные данные, используя только исторические значения текущего измерения. Анализ вычислительной эффективности подтверждает применимость метода для использования в большинстве прикладных задач онлайн-восстановления.

Табл. 1: Сравнение ошибки восстановления метода SAETI с аналогами,  $RMSE \times 10^{-3}$

Методы		Наборы данных											Ср. ошибка
Тип	Название	Категория А				Ср. ошибка	Категория Б					Ср. ошибка	
		Electricity	Madrid	Soccer	WalkRun		BAFU	Climate	MAREL	MeteoSwiss	Saalecane		
Нейросетевые	BRITS	<b>57(1)</b>	35(2)	20(5)	68(4)	45(2)	17(4)	52(2)	<b>77(1)</b>	52(2)	<b>54(1)</b>	<b>50(1)</b>	48(2)
	GP-VAE	74(4)	73(7)	142(14)	123(5)	103(7)	46(7)	<b>15(1)</b>	172(9)	129(14)	92(8)	91(6)	96.3(7)
	M-RNN	180(16)	108(15)	164(16)	227(16)	170(15)	88(14)	138(7)	161(7)	169(15)	140(15)	139(14)	153(15)
	NAOMI	75(5)	49(5)	9(2)	134(7)	67(5)	40(6)	76(6)	128(5)	88(12)	107.1(13)	88(5)	78(5)
	SAITS	73(3)	45(3-4)	14(3)	49(2-3)	45.1(3)	15.2(2)	57.8(5)	79(2-3)	65(3-4)	57(2-3)	54.4(4)	50(3)
	Transformer	81(6)	45(3-4)	20(4)	49(2-3)	49(4)	<b>15(1)</b>	57(4)	79(2-3)	65(3-4)	57(2-3)	54(3)	52(4)
	<b>SAETI</b>	60(2)	<b>31(1)</b>	<b>6(1)</b>	<b>42(1)</b>	<b>35(1)</b>	16(3)	53(3)	88(4)	<b>52(1)</b>	58(4)	53(2)	<b>45(1)</b>

Метод SAETI демонстрирует превосходство на рядах с различными активностями (см. табл. 1): средняя точность выше конкурентов на 63%, по сравнению с наилучшим аналогом — на 22%. Благодаря своей архитектуре метод SAETI может эффективно применяться для офлайн-восстановления временных рядов.

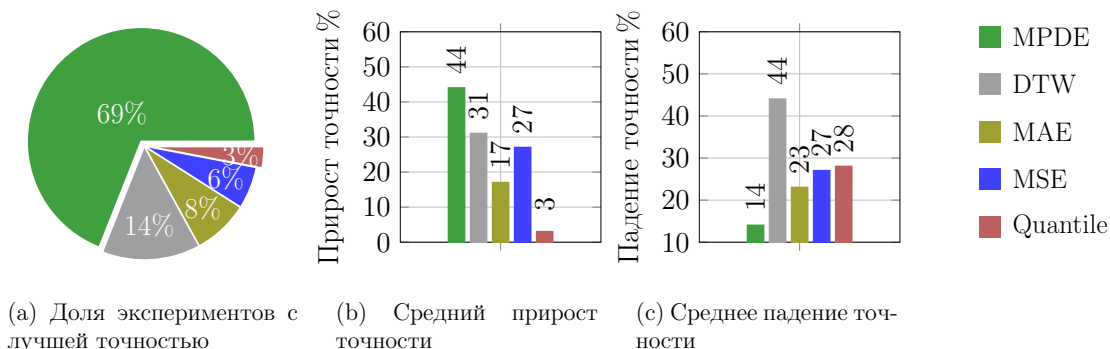


Рис. 2: Сравнение влияния функций потерь на точность нейросетевых моделей восстановления

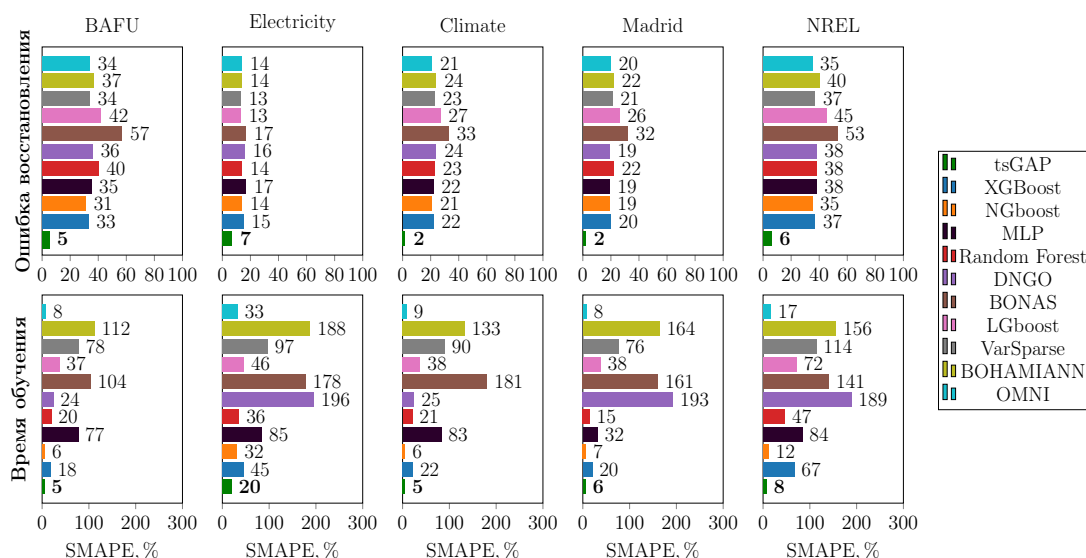


Рис. 3: Сравнение метода tsGAP с конкурентами по точности прогнозирования, SMAPE

Нейросетевые модели, обученные с помощью функции потерь MPDE, демонстрируют более высокую точность восстановления по сравнению с традиционными функциями потерь (см. рис. 2): в 69% случаев точность повышалась в среднем на 44%. Оценка производительности показала, что MPDE требует меньше вычислительных ресурсов, чем специализированные методы для временных рядов, например DTW. Разработанный параллельный алгоритм обеспечивает скорость обучения, сопоставимую с классическими функциями, что позволяет эффективно интегрировать MPDE в жизненный цикл нейросетевых моделей.

Для вычислительных экспериментов было выполнено более 12 000 запусков обучения целевой нейросетевой модели с различными комбинациями гиперпараметров, в которых исследовалось пространство поиска, состоящее из 200 уникальных архитектур моделей восстановления. Эксперименты на реальных и синтетических данных из различных предметных областей продемонстрировали высокую точность метода tsGAP (см. рис. 3): в среднем он

превосходит передовые аналоги на 21% по точности прогноза ошибки целевой модели и на 56.7% по точности прогноза времени обучения. Предложенный метод превосходит лучшего конкурента в среднем на 19.6% по прогнозированию ошибки и 4.4% по прогнозированию времени.

**В заключении** в краткой форме излагаются итоги выполненного диссертационного исследования; представляются отличия данной работы от ранее выполненных схожих работ других авторов; формулируются основные результаты диссертационной работы, выносимые на защиту; приводятся данные о публикациях и апробациях автора по теме диссертации; рассматриваются направления дальнейших исследований в данной области.

## Заключение

*Основным результатом исследования* является комплекс нейросетевых моделей, методов и алгоритмов, предназначенный для использования в рамках жизненного цикла нейросетевых моделей восстановления потоковых данных и включающий следующие разработки. *Метод SANNI* для восстановления потоковых данных, представленных в форме временных рядов, в режиме онлайн, основанный на совместном использовании поведенческих шаблонов и рекуррентных нейросетей. *Метод SAETI* для восстановления потоковых данных, представленных в форме временных рядов, в режиме офлайн, основанный на совместном использовании поведенческих шаблонов и автоэнкодеров. *Функция потерь MPDE*, предназначенная для обучения нейросетевых моделей восстановления потоковых данных, представленных в форме временных рядов, и оценивающая поведенческую схожесть подпоследовательностей. *Метод tsGAP*, обеспечивающий прогнозирование ошибки и времени обучения нейросетевых моделей восстановления потоковых данных, представленных в форме временных рядов, за счет использования представления нейросетевой модели восстановления в виде ориентированного ациклического графа.

*Результаты исследования могут быть использованы* в качестве инструментальных средств разработки различных этапов жизненного цикла нейросетевых моделей для повышения качества восстановления пропущенных значений потоковых данных, содержащих активности. Разработанные методы SANNI и SAETI обеспечивают восстановление потоковых данных, представленных в форме временных рядов, в режимах онлайн и офлайн с высокой точностью. Функция потерь MPDE обеспечивает повышение точности восстановления потоковых данных, представленных в форме временных рядов, нейросетевыми моделями. Метод tsGAP позволяет прогнозировать ошибки и оценивать время обучения нейросетевых моделей восстановления потоковых данных без необходимости полного обучения.

Основные результаты, достигнутые в ходе настоящего диссертационного исследования, представляют собой оригинальный вклад и не повторяют на-

учные разработки, ранее опубликованные другими авторами, обзор которых был представлен в главе 1. *Основные отличия от аналогов* заключаются в следующем.

1. Предложены нейросетевые методы SANNI и SAETI для восстановления потоковых данных, представленных в форме временных рядов, в режимах онлайн и офлайн соответственно. В отличие от передовых аналогов, в предложенных методах для повышения точности восстановления используются поведенческие шаблоны, отражающие типичное поведение исследуемого объекта и применяемые в качестве эталонных примеров ожидаемого поведения. Предложенные методы опережают известные аналоги по точности восстановления на данных, описывающих поведение объектов с активностями, в среднем на 17% и 22% для режимов онлайн и офлайн соответственно.
2. Разработана функция потерь MPDE для обучения нейросетевых моделей восстановления потоковых данных, представленных в форме временных рядов. В отличие от аналогов, MPDE учитывает поведенческое сходство между подпоследовательностями. Результаты вычислительных экспериментов демонстрируют, что использование MPDE обеспечивает повышение точности восстановления в среднем на 44% в 69% случаев по сравнению с передовыми аналогами. Предложен эффективный параллельный алгоритм вычисления MPDE и разработана его реализация для интеграции в современные фреймворки глубокого обучения.
3. Предложен нейросетевой метод tsGAR для прогнозирования ошибки и времени обучения нейросетевых моделей восстановления потоковых данных, представленных в форме временных рядов. В отличие от существующих подходов к прогнозированию качества целевых нейросетевых моделей восстановления, для повышения точности прогнозов в методе tsGAR используются графовые нейронные сети, анализирующие нейросетевую модель восстановления, представленную в виде ориентированного ациклического графа. Вычислительные эксперименты, проведенные на реальных и синтетических данных, продемонстрировали превосходство метода tsGAR над передовыми аналогами в среднем на 21% по прогнозу ошибки целевой модели и на 56.7% по прогнозу времени обучения. Предложенный метод превосходит лучшего конкурента в среднем на 19.6% по прогнозированию ошибки и 4.4% по прогнозированию времени обучения.

Теоретические исследования и практические разработки, выполненные в рамках этой диссертационной работы, предполагается *продолжить по следующим направлениям*: разработка методов и моделей восстановления потоковых данных на основе поведенческих шаблонов, позволяющих производить трансферное обучение; разработка алгоритма предварительной оценки пропусков в потоковых данных на основе использования поведенческих шабло-

нов; повышение эффективности вычисления функции потерь MPDE; разработка средств автоматизации проектирования нейросетевых моделей восстановления потоковых данных на основе метода tsGAR.

## Публикации автора по теме диссертации

1. Yurtin, A. SANNI: Online Imputation of Missing Values in Multivariate Time Series Based on Deep Learning and Behavioral Patterns / A. Yurtin, M. Zymbler // Lobachevskii Journal of Mathematics. – 2024. – Vol. 45, no. 11. – P. 5948–5966. DOI: 10.1134/S1995080224606854. WOS: 001446922600003. (Ядро РИНЦ, Перечень ВАК К1, Scopus Q2).
2. Юртин А.А. Метод прогнозирования ошибки времени обучения нейросетевых моделей восстановления многомерных временных рядов // Проблемы информатики. – 2025. – № 3. – С. 78–95. DOI: 10.24412/2073-0667-2025-3-72-95. (Перечень ВАК К2).
3. Юртин А.А. Об одной функции потерь для обучения нейросетевых моделей восстановления временных рядов / А.А. Юртин // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. – 2024. – Т. 13, № 4. – С. 53–73. DOI: 10.14529/cmse240404. (Ядро РИНЦ, Перечень ВАК К2).
4. Юртин А.А. Восстановление многомерных временных рядов на основе выявления поведенческих шаблонов и применения автоэнкодеров / А.А. Юртин // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. – 2024. – Т. 13, № 2. – С. 39–55. DOI: 10.14529/cmse240203. (Ядро РИНЦ, Перечень ВАК К2).
5. Цымблер М.Л., Восстановление пропущенных значений временного ряда на основе совместного применения аналитических алгоритмов и нейронных сетей / М.Л. Цымблер, А.А. Юртин // Вычислительные методы и программирование. – 2023. – Т. 24, № 3. – С. 243–259. DOI: 10.26089/NumMet.v24r318. (Ядро РИНЦ, Перечень ВАК К1).

Исходные тексты программ, реализующих разработанные в рамках диссертационного исследования алгоритмы и методы, а также наборы данных, с которыми проведены вычислительные эксперименты, подтверждающие эффективность указанных разработок, размещены в сети Интернет в свободно доступных репозиториях:

<https://gitverse.ru/yurtinaa/SANNI>, <https://gitverse.ru/yurtinaa/SAETI>,  
<https://gitverse.ru/yurtinaa/MPDE>, <https://gitverse.ru/yurtinaa/tsGAR2/>

Диссертационное исследование выполнено при финансовой поддержке стипендии Президента Российской Федерации № SPN.2025.04099 (2025 г.) и Российского научного фонда (грант № 23-21-00465).