

Сравнительный анализ эффективности
Channel Independence и Channel Mixing
для прогнозирования многомерных
временных рядов

Рахимова ГР

Введение

Прогнозирование многомерных временных рядов представляет собой фундаментальную задачу, имеющую широкое применение в различных областях от энергетики и транспорта до финансовых рынков и здравоохранения. Точные прогнозы позволяют оптимизировать использование ресурсов, снижать издержки и принимать обоснованные управленческие решения. Однако построение эффективных моделей прогнозирования осложняется высокой размерностью данных, сложными нелинейными зависимостями и изменчивостью распределений во времени [1].

Развитие методов глубокого обучения существенно расширило возможности анализа временных рядов. Архитектуры на основе рекуррентных нейронных сетей (LSTM, GRU), свёрточных сетей (TCN) и трансформеров (Informer, FEDformer, PatchTST) продемонстрировали высокую эффективность в моделировании долгосрочных зависимостей и сложной структуры сигналов [1], [2]. При этом особое внимание исследователей привлекают многомерные временные ряды, для которых критически важным становится вопрос организации обработки отдельных переменных (каналов) и учёта их взаимосвязей.

Одной из ключевых открытых проблем в области прогнозирования многомерных временных рядов является выбор стратегии моделирования межканальных связей. В современной литературе выделяются два принципиально различных подхода: Channel Independence (CI) и Channel Mixing (CM) [1]. Стратегия Channel Independence предполагает независимое моделирование каждого канала, что позволяет устранять шум и влияние нерелевантных переменных, упрощает обучение и повышает устойчивость модели. Однако такой подход может упрощать сложные взаимодействия между каналами, что потенциально приводит к потере важной информации о межканальных зависимостях. Стратегия Channel Mixing, напротив, направлена на совместное моделирование всех каналов, что позволяет эксплуатировать межканальные зависимости и извлекать общие закономерности. Однако этот подход может приводить к проблемам чрезмерного сглаживания (over-smoothing), когда модель смешивает информацию всех каналов, включая нерелевантные зависимости, что в итоге может снижать точность прогнозирования и увеличивать вычислительную сложность, особенно при высоких размерностях данных.

Современные исследования показывают, что проблема балансирования между независимостью и смешиванием каналов остаётся одной из центральных и во многом

не решённых в области глубокого обучения для временных рядов. Различные архитектуры по-разному балансируют между этими стратегиями, а эффективность подходов существенно зависит от домена, масштаба и статистических свойств данных [1]. В этой связи особую актуальность приобретают систематические исследования, направленные на сравнительный анализ эффективности стратегий Channel Independence и Channel Mixing применительно к многомерным временным рядам, которые позволят выявить условия, при которых каждый из подходов демонстрирует наилучшие результаты.

Целью данной выпускной квалификационной работы является сравнительный анализ эффективности подходов Channel Independence и Channel Mixing для прогнозирования многомерных временных рядов на основе современных методов глубокого обучения.

Для достижения поставленной цели в работе решаются следующие основные задачи:

- проанализировать теоретические основы подходов Channel Independence и Channel Mixing, а также их реализацию в современных моделях прогнозирования временных рядов;
- выполнить обзор существующих методов прогнозирования многомерных временных рядов, включая классические статистические подходы и модели глубокого обучения (RNN, CNN, Transformer и др.) с точки зрения обработки каналов;
- разработать и реализовать экспериментальный стенд для сравнения моделей, основанных на Channel Independence и Channel Mixing, на реальных многомерных данных;
- провести экспериментальную оценку эффективности рассматриваемых подходов по ряду метрик качества и вычислительной сложности;
- сформулировать практические рекомендации по выбору стратегии моделирования каналов в зависимости от свойств данных и прикладного сценария.

Экспериментальная часть работы направлена на сравнение двух подходов к прогнозированию многомерных временных рядов: независимого прогнозирования по каналам Channel Independence и совместного прогнозирования с перемешиванием каналов Channel Mixing и выявление условий, при которых каждый из подходов даёт наилучший результат. Для достижения этой цели решаются следующие экспериментальные задачи:

Анализ набора данных. Собственный датасет будет изучен на предмет пригодности, идентифицированы отдельные временные ряды и их статистические свойства (наличие

сезонности, трендов, всплесков). Если собственный датасет недостаточно разнороден, будут привлечены открытые многомерные датасеты с разнообразными по природе каналами чтобы обеспечить широкий спектр случаев. На этом этапе важно понять, присутствуют ли группы каналов с похожим поведением.

Кластеризация рядов по поведению. Будут выделены кластеры каналов со схожими характеристиками например, кластеры с выраженной сезонностью, кластеры с резкими редкими всплесками, кластеры с плавным трендом и малой сезонностью и т.п. Для этого могут быть использованы метрики подобия временных рядов или методы кластеризации (k-means, DBSCAN на признаках сезонности, автокорреляций и пр.). В результате кластеризации ожидается выделить несколько групп каналов, каждая из которых будет характеризоваться схожими временными паттернами.

Обучение моделей в рамках каждого подхода. Для каждого кластера будет проведено сравнение двух схем. Channel Independent обучаются отдельные модели для каждого канала в кластере, при этом каналы обрабатываются независимо друг от друга, Channel Mixing модель обрабатывает каналы совместно, перемешивая информацию между ними в определённые моменты обработки (например, через механизмы внимания или специальные слои смешивания каналов). В качестве базовых моделей планируется использовать простые алгоритмы типа ARIMA, Prophet в обеих схемах. А далее подключать более современные (например, для Mixing многомерный LSTM или трансформер с механизмами перемешивания каналов, для Independent та же модель, но запущенная раздельно по каждому ряду). Также возможно применение библиотек вроде ETNA или PyTorch Forecasting для удобства реализации этих экспериментов.

Экспериментальное сравнение качества. Для каждого кластера выполняется многократное прогнозирование на тестовых отрезках данных. Сравниваются метрики ошибки (MAE, MAPE, SMAPE, RMSE) между подходами CI и CM при разном количестве каналов. Планируется специально варьировать размерность модели (сколькими каналами она обучается одновременно), объём обучающих данных. Это позволит выяснить, как масштаб данных влияет на выигрыш от совместного обучения. Ожидаемый результат зависимость качества прогноза от числа каналов. Возможно, при малом количестве серий и их неоднородности локальные модели работают лучше, а при увеличении числа похожих серий глобальная модель начинает выигрывать.

Проведённое исследование позволит сравнить эффективность подходов Channel Independence и Channel Mixing на различных типах кластеров каналов и выявить условия, при которых каждый из подходов показывает наилучшие результаты.

Ожидается, что результаты исследования внесут вклад в развитие теории и практики прогнозирования многомерных временных рядов, расширят понимание роли межканальных зависимостей в моделях глубокого обучения и предоставят исследователям и практикам обоснованные рекомендации по выбору стратегии моделирования каналов в зависимости от характеристик данных и требований прикладной задачи. Это, в свою очередь, должно способствовать повышению точности прогнозов и более эффективному использованию временных рядов при принятии решений в различных предметных областях.

Глава 1. Классические и многомерные модели временных рядов

1.1 Временные ряды.

Временным рядом называют последовательность наблюдений некоторой величины, упорядоченных во времени. Формально временной ряд можно представить как набор значений $\{x_t\}$, где каждое значение соответствует определённому моменту времени t . Примерами временных рядов являются динамика цен финансовых активов, объёмы продаж товаров по дням, показатели потребления электроэнергии, метеорологические измерения (температура, влажность, осадки), данные датчиков в промышленности и медицине. Прогнозирование временных рядов (Time Series Forecasting, TSF) играет критически важную роль в экономическом планировании, управлении цепочками поставок, прогнозировании спроса, а также при поддержке принятия решений в здравоохранении и других отраслях [1].

Особенность временных рядов состоит в том, что значения наблюдаемой величины зависят не только от текущих условий, но и от её прошлой динамики. При этом реальные данные, как правило, обладают целым рядом характерных свойств. Часто в временных рядах присутствует тренд - долгосрочная направленная компонента, отражающая общую тенденцию роста или спада показателя во времени. Во многих рядах также наблюдается сезонность - повторяющиеся паттерны, связанные с суточными, недельными или годовыми циклами (например, пики потребления электроэнергии вечером, рост продаж в выходные дни). Существенную роль играет шум - случайные колебания, не объясняемые трендом и сезонностью, а также редкие аномальные события (выбросы), которые могут вызывать резкие отклонения значений

временного ряда [1]. Важно отметить, что при декомпозиции временного ряда остатки после выделения тренда и сезонности должны представлять собой именно шум - случайную компоненту без явных закономерностей. Если остатки после декомпозиции имеют структуру, это указывает на то, что модель не полностью уловила все паттерны в данных. Сочетание тренда, сезонности, шума и возможной нестационарности делает задачу прогнозирования существенно более сложной. Модель должна, с одной стороны, выделять устойчивые закономерности в данных, а с другой уметь подстраиваться под изменения поведения ряда во времени.

По числу одновременно рассматриваемых переменных временные ряды принято делить на одномерные и многомерные. В одномерном случае анализируется динамика одной величины во времени. Такие модели, как правило, проще в построении, требуют меньше данных и вычислительных ресурсов, что делает их привлекательными для ряда практических задач. Однако они используют ограниченный объём информации, так как игнорируют возможное влияние других, связанных показателей.

Многомерные временные ряды могут образовываться различными способами, например когда несколько переменных измеряются одновременно (например, показания различных датчиков), когда один процесс разбивается на несколько компонент, или когда данные собираются из разных источников и объединяются по временной оси. Например, при прогнозировании погоды для конкретного региона могут одновременно учитываться температура, влажность, скорость и направление ветра, атмосферное давление и другие метеорологические параметры. В таких данных каждый показатель образует отдельный «канал», между которыми часто существуют нетривиальные зависимости. Дополнительно на поведение каналов влияют внешние факторы, такие как календарные эффекты (дни недели, праздники), изменения макроэкономической ситуации или особенности режима работы системы [1], [2].

С одной стороны, использование многомерной информации позволяет выявлять более сложные и содержательные закономерности и, как следствие, потенциально повышать точность прогноза. С другой стороны, это приводит к росту размерности данных, усложняет модель и повышает риск переобучения, особенно при ограниченном объёме наблюдений. Кроме того, временные ряды из разных предметных областей (метеорология, финансы, промышленность) могут существенно отличаться по своим характеристикам структуре, частоте изменений, наличию трендов и сезонности, что ограничивает возможность создания универсальных моделей и требует адаптации под конкретную предметную область [1].

Таким образом, временные ряды представляют собой сложный тип данных, для которого характерны тренды, сезонность, шум, а также влияние различных внешних факторов. Различие между одномерными и многомерными временными рядами является принципиальным при выборе методов моделирования и прогнозирования, в первом случае основное внимание уделяется динамике одной переменной, тогда как во втором важно совместно учитывать несколько показателей и характер их взаимосвязей. Именно в условиях работы с многомерными временными рядами на первый план выходят вопросы организации каналов и выбора стратегии их моделирования, что в дальнейшем служит основой для анализа подходов Channel Independence и Channel Mixing.

1.2 Классические модели AR, ARMA, ARIMA, SARIMA, SARIMAX

Развитие методов прогнозирования временных рядов начиналось с классических статистических моделей, которые до сих пор широко используются как базовые подходы. Ключевую роль здесь сыграли модели авторегрессии (AR), скользящего среднего (MA) и их комбинации ARMA/ARIMA, заложившие фундамент для формального подхода к анализу и моделированию временных зависимостей [3].

Авторегрессионная модель (AR) описывает текущее значение ряда как линейную комбинацию нескольких его прошлых значений и случайной ошибки. Модель скользящего среднего (MA), напротив, выражает текущее значение через линейную комбинацию прошлых ошибок прогноза. Модели ARMA объединяют оба подхода и хорошо подходят для стационарных временных рядов, где статистические свойства (среднее, дисперсия, автокорреляции) не меняются во времени. Однако важным ограничением этих моделей является требование стационарности ряда, если временной ряд не является стационарным, его необходимо привести к стационарности определёнными методами (например, дифференцированием или логарифмированием). Важным вкладом в систематизацию построения таких моделей стал метод Бокса–Дженкинса. Он предусматривает последовательное выполнение трёх этапов, выбор структуры модели (порядков AR и MA) на основе анализа автокорреляционных функций, оценку параметров и проверку адекватности модели [3].

Для работы с нестационарными данными была предложена модель ARIMA (Autoregressive Integrated Moving Average), в которой применяется дифференцирование временного ряда для устранения тренда и приведения его к стационарному виду. Для моделирования сезонности была разработана модель SARIMA, которая дополнительно

учитывает сезонные эффекты. Модель SARIMAX расширяет SARIMA возможностью включения экзогенных переменных. ARIMA-модели позволяют эффективно учитывать краткосрочные зависимости и компоненты тренда и сезонности. Классические модели этого класса показали высокую эффективность в задачах краткосрочного прогноза и долгое время оставались стандартом в прикладных областях [3]. Однако, как отмечают современные исследования, такие модели с трудом справляются с долгосрочными зависимостями, нелинейными закономерностями и многомерными данными.

Для многомерных временных рядов, где требуется явно учитывать взаимосвязи между переменными, были разработаны модели векторной авторегрессии, VAR (Vector Autoregression). В таких моделях каждая переменная(канал) описывается как линейная комбинация прошлых значений всех переменных системы, что позволяет учитывать межканальные зависимости и моделировать взаимное влияние показателей друг на друга. Однако возможности VAR-подходов ограничены, при росте размерности и наличии нелинейных взаимодействий между переменными приходится прибегать к упрощениям или методам регуляризации, что снижает их эффективность в сложных практических задачах.

Кроме того, в классическом наборе методов прогнозирования присутствуют модели экспоненциального сглаживания (например, метод Холта–Уинтерса), которые эффективны для одномерных рядов с трендом и сезонностью, но имеют ограничения при работе с многомерными и нелинейными данными [3].

В совокупности классические статистические модели (AR, MA, ARMA, ARIMA, SARIMA, VAR) сформировали основу теории и практики прогнозирования временных рядов и до сих пор широко используются как интерпретируемые, устойчивые и относительно простые в настройке базовые решения, в том числе в качестве baseline-моделей. При этом их ограничения - линейность, ориентированность преимущественно на краткосрочные зависимости и трудности с моделированием сложных многомерных структур стали одной из ключевых причин активного развития методов машинного обучения и глубоких нейронных сетей. На практике классические и современные методы часто используются совместно, это находит отражение в современных фреймворках, таких как ETNA, который объединяет традиционные модели (например, SARIMAX, экспоненциальное сглаживание) и современные алгоритмы (Facebook Prophet, CatBoost, глубокие RNN вроде DeepAR) для прогнозирования множества рядов. В таких системах классические модели

используются вместе с моделями глубокого обучения, они могут служить baseline, входить в ансамбли или использоваться как интерпретируемые опорные решения [4].

1.3 Подходы к моделированию многомерных временных рядов

При работе с многомерными временными рядами каждая переменная (например, отдельный показатель, сенсор или признак, измеряемый одновременно с другими) образует отдельный «канал». Например, в метеорологических данных каналами могут быть температура, влажность, скорость ветра и атмосферное давление, в финансовых данных цены, объёмы торгов и волатильность по группе инструментов, в промышленных системах показания различных датчиков, регистрируемых синхронно. Вопрос о том, как моделировать эти каналы независимо или совместно, является одним из ключевых при построении моделей прогнозирования многомерных временных рядов [5].

В современной литературе выделяются два основных подхода к обработке каналов. Первый подход Channel Independent (CI), или стратегия независимого моделирования каналов. В этом случае каждый канал обрабатывается отдельно, для каждого показателя строится своя модель, которая использует только собственную историю значений. Это упрощает архитектуру, снижает вычислительную нагрузку и уменьшает влияние шума, возникающего из-за сложных межканальных связей. Показано, что стратегия CI способна улучшать качество прогноза за счёт более точного учета временной структуры внутри каждого канала [5]. Вместе с тем такой подход полностью игнорирует возможные взаимодействия между переменными, что может снижать обобщающую способность моделей и не позволяет использовать информацию о совместной динамике показателей.

Второй подход Channel Dependent (CD), или стратегия зависимого моделирования каналов. Здесь модель одновременно учитывает все каналы, пытаясь уловить взаимосвязи между ними. CD-стратегия позволяет моделировать взаимное влияние показателей друг на друга и формализовать межканальные зависимости, что теоретически должно повысить точность прогноза за счёт использования более полной информации. Большинство моделей глубокого обучения традиционно принимают именно CD-стратегию, стремясь использовать весь спектр информации по каналам. К ним относятся, например, ранние Transformer-подходы для временных рядов, а также

современные модели вроде iTransformer и C-Mamba, в которых межканальные зависимости закладываются прямо в архитектуру [1],[5]. Однако, как отмечают исследования, CD-подход может приводить к проблемам чрезмерного сглаживания (over-smoothing), когда модель смешивает информацию всех каналов, включая шум, что в итоге может снижать точность прогнозирования. Кроме того, по сравнению с CI-подходом CD, как правило, лучше улавливает общие закономерности между каналами, но хуже отражает индивидуальные особенности отдельных каналов [5].

В качестве развития подхода CD в некоторых современных моделях используется метод Channel Mixing, который предполагает избирательное перемешивание информации между каналами в определённые моменты обработки, а не совместную обработку всех каналов с самого начала. В отличие от CD, где все каналы обрабатываются совместно на всех этапах, Channel Mixing позволяет контролировать, когда и как происходит обмен информацией между каналами, что может способствовать более эффективному использованию межканальных зависимостей. Примером реализации этого метода является модель CSformer, которая использует двухэтапный механизм самовнимания, на первом этапе извлекается специфичная для каждого канала информация (Channel Independence), а на втором этапе происходит интеграция информации из различных каналов с учётом межканальных зависимостей (Channel Mixing) [14]. Такой подход позволяет сочетать преимущества независимого моделирования каналов (устранение шума, фокус на индивидуальных особенностях) с возможностями совместного использования информации между каналами, что повышает устойчивость и точность модели.

Всё многообразие методов говорит о том, что ни одна из стратегий не является универсальной, и возникает необходимость в подходах, которые учитывали бы сильные стороны CI и CD. В связи с этим разрабатываются различные методы, которые пытаются комбинировать преимущества обоих подходов.

Один из таких методов основан на кластеризации каналов, как показано в работе [5], каналы сначала группируются по сходству их временных характеристик, после чего для каждого кластера обучается отдельная модель прогнозирования. Это позволяет учитывать общие паттерны внутри однородных групп и одновременно избегать излишнего смешивания информации всех каналов сразу.

Другой такой пример представлен в модели CT-PatchTST, где многомерные временные ряды разбиваются на патчи в пространстве канал-время, что обеспечивает

более детальный учёт как временной структуры, так и организации каналов при долгосрочном прогнозировании [3].

Ещё одной современной моделью, которая работает и с отдельными каналами, и с их связями, является TSMixer. В ней используется блок MLP-Mixer, чтобы отдельно обрабатывать данные по времени и по каналам. Это позволяет учесть особенности каждого канала, но при этом уловить и общие закономерности между ними [6].

Такие подходы показывают, что между стратегиями CI и CD возможны решения, которые одновременно учитывают и особенности отдельных каналов, и их совместное поведение.

В дальнейшем идея кластеризации каналов и разбиение на патчи будут подробнее рассмотрены и использованы в экспериментальной части настоящей работы.

Глава 2. Обзор существующих исследований

2.1 Методы прогнозирования многомерных временных рядов: статистические и алгоритмические

Прогнозирование многомерных временных рядов представляет собой более сложную задачу по сравнению с одномерными рядами, поскольку требует учёта как временных зависимостей, так и взаимосвязей между различными переменными (каналами). Аналогичная проблема возникает при работе с множеством временных рядов, для которых традиционно использовались два основных подхода. Первый подход локальное моделирование, при котором для каждого ряда строится своя модель. В классических статистических методах это означает, что для каждого ряда подбирается своя ARIMA-модель с индивидуальными параметрами. Такой подход позволяет учитывать особенности каждого ряда, но не использует общие закономерности, которые могут быть в группе связанных рядов [7].

Второй подход глобальное моделирование, одна модель обучается сразу на всех рядах. Переход к глобальным моделям произошёл из-за того, что они могут использовать информацию из всех рядов одновременно. Глобальная модель смешивает данные из разных рядов, извлекая общие паттерны и используя их для улучшения прогноза отдельных рядов.

Научные обоснования эффективности глобальных моделей были получены в ряде исследований. В работе [8] показано, что глобальные модели не менее универсальны, чем локальные, а при достаточном объёме данных могут аппроксимировать поведение отдельных рядов не хуже индивидуальных моделей. Исследования показывали, что объединение данных многих временных рядов может улучшить точность прогноза за счёт переноса знаний между рядами, особенно когда отдельные временные ряды содержат мало данных. Глобальная модель требует меньше вычислительных ресурсов и упрощает процесс обучения.

Для работы с множеством рядов появились специализированные инструменты. В частности, библиотека ETNA от Tinkoff.AI. Она предоставляет модульный подход, позволяет использовать как классические модели (например, SARIMA для каждого ряда), так и алгоритмы машинного обучения и нейросети, которые могут обучаться сразу на всех рядах. ETNA поддерживает как локальные модели (на уровне отдельного сегмента или ряда), так и глобальные (одна модель на все ряды), а также их сочетания [4].

Аналогичные возможности есть в других фреймворках, GluonTS от Amazon для глубокого обучения временных рядов, Facebook Prophet для прогнозирования с учётом трендов и сезонности [9]. Эти инструменты также поддерживают как локальные, так и глобальные модели, что позволяет выбирать подход в зависимости от характеристик данных.

Для автоматизированного машинного обучения (AutoML) временных рядов есть библиотеки, которые автоматически выбирают модели и оптимизируют гиперпараметры. Например, AutoTS позволяет строить статистические модели и модели машинного обучения для прогнозирования большого количества рядов, а PyCaret поддерживает моделирование временных рядов и обнаружение аномалий [9]. Подходы к прогнозированию множества временных рядов перешли от локального моделирования к глобальным моделям, которые смешивают информацию из разных рядов. Современные фреймворки поддерживают оба подхода. Аналогичная проблема выбора стратегии возникает при работе с многомерными временными рядами моделировать ли каналы независимо (CI) или смешивать информацию из разных каналов (CM). Сравнительная эффективность этих подходов требует дальнейшего исследования.

2.2 Рекуррентные нейросети для прогнозирования временных рядов.

Развитие методов глубокого обучения привело к появлению рекуррентных нейросетей, которые стали активно применяться для прогнозирования временных рядов. Рекуррентные нейросети, особенно архитектуры LSTM и GRU, смогли преодолеть ограничения классических методов, научившись моделировать долгосрочные временные зависимости и нелинейности.

Классические статистические методы, такие как ARIMA, хорошо подходят для краткосрочных зависимостей и линейных паттернов, но сталкиваются с трудностями при моделировании долгосрочных зависимостей и сложных нелинейных закономерностей. Рекуррентные нейросети решают эту проблему, так как могут хранить информацию о предыдущих состояниях и обрабатывать последовательности переменной длины.

Однако обычные RNN сталкиваются с проблемой затухания градиента, при обратном распространении ошибки градиенты экспоненциально уменьшаются при прохождении через множество временных шагов, что делает обучение долгосрочных зависимостей практически невозможным. Архитектура LSTM (Long Short-Term Memory) решает эту проблему, используя специальные вентили (gates), которые контролируют поток информации через сеть. Вентили решают, какую информацию сохранить, какую забыть и какую передать дальше. Это позволяет LSTM сохранять информацию о дальних временных шагах и предотвращает затухание градиента при обучении [10]. Благодаря такой структуре LSTM может запоминать долгосрочные зависимости и эффективно работать с последовательностями переменной длины. LSTM особенно эффективна для задач прогнозирования, где важно учитывать долгосрочные паттерны и тренды.

Одной из первых успешных работ, применивших RNN для прогнозирования временных рядов, стала модель DeepAR, предложенная в 2017 году [2]. DeepAR обучает один LSTM на большой коллекции похожих временных рядов, что позволяет модели извлекать общие закономерности и использовать их для прогнозирования каждого отдельного ряда. Модель может генерировать вероятностные прогнозы (probabilistic forecasts), что важно для оценки неопределённости. DeepAR показала примерно 15% улучшение точности по сравнению с лучшими классическими подходами, что подтвердило практический потенциал RNN для прогнозирования временных рядов [2].

Другим направлением стало применение архитектур типа sequence-to-sequence, заимствованных из области обработки естественного языка. Seq2Seq модели состоят из энкодера и декодера на основе RNN, что позволяет предсказывать сразу последовательность будущих значений, а не одно значение за раз. Однако ранние Seq2Seq модели теряли информацию при работе с длинными входными последовательностями. Введение механизма внимания позволило модели автоматически выделять наиболее важные моменты во входной последовательности, что значительно улучшило качество долгосрочного прогноза [11].

Для многомерных временных рядов были предложены гибридные архитектуры, сочетающие свёрточные и рекуррентные слои. Примером такой архитектуры является модель, которая объединяет свёрточные слои и рекуррентный блок (GRU). CNN выделяет короткие локальные циклы (например, недельные паттерны), а RNN хранит долгосрочные зависимости. Также добавлен линейный AR-компонент для улавливания простых корреляций. Такая комбинация позволяет эффективно обрабатывать как локальные, так и глобальные временные паттерны в многомерных данных [12].

Ещё одним направлением стало объединение статистических методов с глубоким обучением. В гибридной модели [13] каждый временной ряд предварительно сглаживается для выделения базового уровня и сезонности, а остатки прогнозируются LSTM, обученным глобально на всех рядах. Такой ансамбль достиг лучшего качества, что подтвердило эффективность объединения статистических моделей с RNN. Этот результат показал, что классические и современные методы могут успешно дополнять друг друга [13].

Таким образом, рекуррентные нейросети, особенно LSTM и GRU, стали важным шагом в развитии методов прогнозирования временных рядов. Они позволили преодолеть ограничения классических методов в работе с долгосрочными зависимостями и нелинейностями. Модели типа DeepAR показали практическую эффективность RNN на реальных данных, а гибридные подходы, сочетающие CNN, RNN и статистические методы, продемонстрировали возможность улучшения качества прогноза за счёт комбинации различных техник [1].

2.3 Трансформеры для прогнозирования временных рядов.

Трансформеры, изначально развитые для языковых задач, привлекли внимание в TSF благодаря мощному механизму самовнимания, позволяющему моделям фокусироваться на долгосрочных связях в данных.

При адаптации стандартного Transformer для временных рядов исследователи столкнулись с рядом проблем [15]. Во-первых, квадратичная сложность механизма самовнимания при длинных последовательностях создавала вычислительные ограничения. Во-вторых, возникла необходимость учитывать специфические особенности временных рядов, такие как сезонность и тренды. В-третьих, требовалось включение внешних переменных и работа с разнородными каналами в многомерных рядах.

Для решения проблемы квадратичной сложности механизма самовнимания были предложены модели с разреженным вниманием. В отличие от стандартного механизма внимания, который вычисляет внимание между всеми парами элементов последовательности, разреженное внимание выбирает только наиболее релевантные пары для вычисления, что значительно снижает вычислительную сложность. Выбор релевантных пар основан на оценке важности элементов например, через статистический анализ распределения внимания или оценку сходства между элементами последовательности. Например, Informer использует механизм ProbSparse attention, который выбирает запросы с наибольшей дисперсией в распределении внимания, что позволяет снизить вычислительную нагрузку, сохраняя при этом способность моделировать долгосрочные зависимости [16]. Preformer применяет многоуровневые корреляции между сегментами (multi-scale segmentwise correlations) для эффективного вычисления внимания на длинных последовательностях [17]. Эти подходы значительно снизили вычислительную сложность трансформеров при работе с длинными последовательностями.

Для учёта сезонности и трендов были разработаны модели, вводящие декомпозицию временного ряда прямо внутри архитектуры. Autoformer использует декомпозицию серии на тренд и сезонную компоненту, что позволяет модели более эффективно обрабатывать эти компоненты отдельно [18]. Такой подход улучшает способность модели улавливать долгосрочные тренды и сезонные паттерны.

Для работы с многомерными временными рядами были предложены специализированные Transformer-архитектуры, которые явно учитывают

межканальные зависимости. Примером такой архитектуры является Crossformer [15]. Модель преобразует входной многомерный ряд в двумерный массив «время × переменные» и применяет Two-Stage Attention, чередуя внимание по временной оси и по оси переменных. Такой подход позволяет уловить как временные зависимости, так и межканальные связи, что улучшает качество прогноза на рядах, где каналы существенно связаны между собой.

Параллельно, другая линия исследований шла в противоположном направлении: упрощение трансформеров для временных рядов. В работе [19] авторы показывают, что на ряде стандартных датасетов простая линейная однослойная модель DLinear превосходит сложные Transformer-модели по точности. Авторы аргументируют, что механизм самовнимания может терять информацию о временном порядке и излишне усложнять задачу, тогда как линейные модели более эффективно извлекают основные сезонные компоненты. Этот результат вызвал всплеск исследований по пересмотру архитектур. Выяснилось, что многие трансформеры плохо работали с нестационарными рядами, и после должной нормализации данных или применения простых базовых подходов их преимущества перед линейными моделями значительно сокращаются [19].

В 2023–2024 годах появилось несколько работ, пытающихся объединить сильные стороны Channel-Independent и Channel-Dependent подходов внутри трансформеров. Модель PatchTST предложила разбивать временные ряды на подпоследовательности-патчи (subseries-level patches), которые служат входными токенами для трансформера, и применять стратегию Channel Independence: каждый канал обрабатывается как отдельный одномерный ряд, но использует общие embedding и веса трансформера [20]. Такой подход показал отличные результаты на длинных горизонтах прогнозирования, подтвердив эффективность стратегии CI для трансформеров.

Последующие работы обращают внимание на ограничение PatchTST, например отсутствие взаимодействия между каналами может снижать качество прогноза в случаях, когда каналы сильно коррелируют. Модель CT-PatchTST расширяет PatchTST, добавляя механизм channel attention, который работает на уровне патчей в пространстве «канал-время» [3]. Архитектура использует двухступенчатый процесс: сначала применяется channel attention для выделения и интеграции информации между каналами на основе патчей, затем для каждого канала отдельно применяется временное self-attention. Такой подход позволяет модели одновременно учитывать как

межканальные зависимости, так и временную структуру внутри каждого канала. CT-PatchTST показал более высокую точность по сравнению с PatchTST на данных возобновляемой энергетики, где показатели (например, скорость ветра и солнечная радиация) демонстрируют явные корреляции между собой [3].

Другой подход к комбинированию Channel-Independent и Channel-Mixing стратегий представлен в модели CSformer [14]. Архитектура использует двухэтапный механизм многоголовой самовнимательности: на первом этапе извлекается специфичная для каждого канала информация (CI), что позволяет устранить шум и сфокусироваться на индивидуальных особенностях каждого канала, а на втором этапе происходит интеграция информации из различных каналов с учётом межканальных зависимостей (CM). Такой подход позволяет сочетать преимущества независимого моделирования каналов с возможностями совместного использования информации между каналами, что повышает устойчивость и точность модели на различных типах данных.

Альтернативный подход к балансированию между Channel Independent и Channel Dependent стратегиями предложен в работе по кластеризации каналов [5]. Вместо прямого смешивания всех каналов или их полной независимости, этот подход предлагает группировку схожих каналов. Модуль CCM (Channel Clustering Module) динамически кластеризует каналы по внутреннему сходству и заставляет модель смешивать информацию только в пределах каждого кластера, а не по всему набору каналов сразу. Тем самым достигается баланс, внутри группы модель учитывает взаимодействия между схожими каналами (избегая проблемы полной независимости каналов), а между далёкими по характеристикам каналами не происходит перемешивания, что позволяет избежать включения лишнего шума (избегая проблемы излишнего смешивания в CD-подходе). Эксперименты показали улучшение точности как у чисто CI, так и у CD моделей при подключении модуля кластеризации (в среднем на 2–7%), а также более устойчивое качество на новых данных [5].

Отдельно следует отметить появление в 2023–2025 годах концепции больших предварительно обученных моделей для временных рядов (Time-series Foundation Models). Например, TimeGPT и Time-LLM представляют собой попытки адаптировать большие языковые модели для работы с временными рядами, обучая их на множествах рядов или представляя временные последовательности в виде текстовых токенов [21]. В этих моделях часто используется контекстное прогнозирование через промпты, исторические данные и сопутствующий контекст (например, календарные данные, описание условий) подаются модели в виде текстового описания, а модель генерирует

прогноз. Эти подходы также сталкиваются с выбором стратегии обработки многомерных данных: некоторые модели остаются приверженцами Channel independence (например, Time-LLM разбивает многомерные ряды на отдельные каналы при вводе в языковую модель) [21]. Это демонстрирует, что проблема выбора между Channel Independent и Channel Dependent подходами актуальна и для современных фундаментальных моделей временных рядов.

Глава 3. Актуальность исследования

Прогнозирование множественных временных рядов представляет собой задачу большого практического значения, находящую применение в различных областях от энергетики и финансов до здравоохранения и транспорта. В последние годы в области прогнозирования многомерных временных рядов наблюдается развитие двух противоположных подходов. С одной стороны, происходит активное усложнение моделей глубокого обучения, направленное на учёт всех возможных взаимосвязей между переменными через механизмы смешивания каналов (CM). С другой стороны, появляются исследования, показывающие эффективность упрощённых подходов и раздельного моделирования для каждого ряда или канала (CI). Возникает естественный вопрос: насколько действительно необходимы сложные механизмы смешивания каналов для достижения высокой точности прогнозирования?

С одной стороны, появляется всё больше свидетельств того, что сложные модели типа трансформеров не всегда превосходят простые методы. Наиболее показательным в этом отношении является исследование, представленное в работе [19], где авторы демонстрируют, что на ряде стандартных наборов данных простая линейная однослойная модель DLinear превосходит по точности несколько видов сложных трансформеров, включая Informer, Autoformer и FEDformer. Авторы отмечают, что механизм самовнимания может не сохранять информацию о временном порядке и излишне усложнять задачу, тогда как линейные модели более эффективно извлекают основные сезонные компоненты и тренды. Этот результат ставит под сомнение необходимость сложных механизмов смешивания каналов в случаях, когда каналы слабо связаны между собой или когда основная информация содержится во временной структуре каждого канала, а не в межканальных зависимостях [20].

С другой стороны, ряд исследований подчёркивает важность межканальных зависимостей и демонстрирует, что их учёт может существенно повысить точность

прогнозирования. В практических задачах, которые содержат сильно коррелированные показатели, игнорирование этих связей означает потерю важной информации. Например, в данных возобновляемой энергетики скорость ветра и солнечная радиация демонстрируют явные корреляции между собой, и модели, учитывающие межканальные зависимости, показывают более высокую точность по сравнению с подходами, использующими стратегию Channel Independence [3]. Аналогично, модели с механизмами смешивания каналов улучшают качество прогноза на рядах, где каналы существенно связаны между собой [15].

Новейшие результаты показывают, что эффективность стратегий Channel Independence и Channel Mixing существенно зависит от характера данных. Исследования демонстрируют, что улучшение качества при добавлении даже небольшого межканального взаимодействия явно проявляется на наборах данных с сильной корреляцией между каналами, тогда как на данных, где каналы практически независимы, включение смешивания почти не даёт выгоды или даже может слегка ухудшить результат из-за переобучения на шуме [5]. Другими словами, ни одна стратегия не является универсально лучшей всё зависит от структуры ряда. Однако в существующей литературе пока нет однозначных рекомендаций: как понять, в каких случаях лучше применять глобальную модель на все каналы, а где отдельные модели? И как влияет количество каналов (размерность ряда) на качество прогноза?

Эти вопросы важны в контексте растущей популярности трансформеров и моделей с механизмами смешивания каналов. Трансформеры стали своего рода «стандартом» в ряде соревнований и исследований, но их применение требует больших вычислительных ресурсов и сложной настройки. Если выяснится, что для ряда практических случаев достаточно более простого подхода (например, кластеризовать каналы и применять локальные модели внутри групп), это даст серьёзный вклад в упрощение и интерпретацию прогнозных моделей. С другой стороны, если определены сценарии, где без учёта межканальных влияний не обойтись, это позволит фокусировать усилия на разработке гибридных моделей как Channel Clustering именно для таких случаев [5].

Исследование эффективности Channel Independence и Channel Mixing является своевременным и востребованным. Оно лежит на стыке современных теоретических дискуссий и практических потребностей индустрии в массовом прогнозировании. Результаты данного исследования могут прояснить, когда действительно нужны сложные архитектуры вроде трансформеров, а когда простые методы демонстрируют

сопоставимую эффективность. Это соответствует актуальной тенденции в научном сообществе по переоценке избыточной сложности моделей в анализе данных [22]. Проведённое исследование направлено на восполнение этого пробела путём систематического сравнительного анализа эффективности подходов Channel Independence и Channel Mixing на различных типах многомерных временных рядов. Результаты исследования позволят выявить условия, при которых каждый из подходов демонстрирует наилучшие результаты, и сформулировать практические рекомендации по выбору стратегии моделирования каналов в зависимости от характеристик данных и требований прикладной задачи.

Глава 4. Постановка задачи

Целью работы является экспериментальное сравнение двух подходов к прогнозированию многомерных временных рядов независимого прогнозирования по каналам Channel Independence и совместного прогнозирования с перемешиванием каналов Channel Mixing и выявление условий, при которых каждый из подходов демонстрирует наилучшие результаты.

Для достижения поставленной цели в работе решаются следующие задачи:

4.1. Анализ набора данных.

Собственный датасет будет изучен на предмет пригодности для исследования: идентифицированы отдельные временные ряды и их статистические свойства (наличие сезонности, трендов, всплесков). Если собственный датасет недостаточно разнороден, будут привлечены открытые многомерные датасеты с разнообразными по природе каналами чтобы обеспечить широкий спектр случаев. На этом этапе важно понять, присутствуют ли группы каналов с похожим поведением, что позволит в дальнейшем провести осмысленную кластеризацию.

4.2. Кластеризация рядов по поведению.

Будут выделены кластеры каналов со схожими характеристиками: например, кластеры с выраженной сезонностью, кластеры с резкими редкими всплесками, кластеры с плавным трендом и малой сезонностью и т.п. Для этого могут быть использованы метрики подобия временных рядов или методы кластеризации (k-means на признаках сезонности, автокорреляций и пр.). В результате кластеризации ожидается получить несколько осмысленных групп каналов, каждая из которых будет характеризоваться схожими временными паттернами.

4.3. Обучение моделей в рамках каждого подхода

Для каждого кластера будет проведено сравнение двух схем моделирования:

4.3.1 Channel Independent: обучаются отдельные модели для каждого канала в кластере, при этом каналы обрабатываются независимо друг от друга.

4.3.2 Channel Mixing: обучается единая многомерная модель на совокупности каналов кластера (например, многовыходный регрессор или нейросеть, учитывающая все каналы одновременно).

В качестве базовых моделей планируется использовать простые алгоритмы (ARIMA, Prophet) в обеих схемах. Далее будут применены современные модели глубокого обучения. Для Channel Mixing будут использованы многомерный LSTM или трансформер с механизмами смешивания каналов, для Channel Independent — та же модель, но запущенная раздельно по каждому ряду. Также возможно применение библиотек вроде ETNA или PyTorch Forecasting для удобства реализации этих экспериментов.

На каждом кластере выполняется backtesting — многократное прогнозирование на тестовых отрезках данных. Сравниваются метрики ошибки (MAE, MAPE, SMAPE, RMSE) между подходами Channel Independence и Channel Mixing при разном количестве каналов. Планируется специально варьировать: (а) размерность модели (сколькими каналами она обучается одновременно), (б) объём обучающих данных. Это позволит выяснить, как масштаб данных влияет на выигрыш от совместного обучения. Ожидаемый результат — зависимость качества прогноза от числа каналов: возможно, при малом количестве серий и их неоднородности локальные модели работают лучше, а при увеличении числа похожих серий глобальная модель начинает выигрывать.

Список используемой литературы:

1. Jongseon Kim, Hyungjoon Kim, HyunGi Kim, Dongjun Lee, Sungroh Yoon. A Comprehensive Survey of Deep Learning for Time Series Forecasting: Architectural Diversity and Open Challenges. 2024. - 79p.
2. David Salinas, Valentin Flunkert, Jan Gasthaus. DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks. 2017. - 12p.
3. Kuan Lu , Menghao Huo ,Yuxiao Li , Qiang Zhu , Zhenrui Chen CT-PatchTST: Channel-Time Patch Time-Series Transformer for Long-Term Renewable Energy Forecasting 2023. - 10p.
4. <https://medium.com/its-tinkoff/forecasting-with-etna-fast-and-furious-1b58e1453809>
5. Jialin Chen, Jan Eric Lenssen, Aosong Feng, Weihua Hu, Matthias Fey, Leandros Tassioulas, Jure Leskovec, Rex Ying. From Similarity to Superiority: Channel Clustering for Time Series Forecasting. 2024. - 23p.
6. Vijay Ekambaram, Arindam Jati, Nam Nguyen, Phanwadee Sinthong and Jayant Kalagnanam. TSMixer: Lightweight MLP-Mixer Model forMultivariate Time Series Forecasting. 2023.
7. Rob J Hyndman , George Athanasopoulos. Forecasting: Principles and Practice. Monash University, Australia
8. Pablo Montero-Manso, Rob J Hyndman. Principles and Algorithms for Forecasting Groups of Time Series: Locality and Globality. 2021. - 37p.
9. Christian O’Leary , Farshad Ghassemi Toosi , Conor LynchA. Review of AutoML Software Tools for Time Series Forecasting and Anomaly Detection. 2023
10. Sepp Hochreiter, Jürgen Schmidhuber. Long. Short-Term Memory. Neural Computation, 1997.
11. Shiyu Wang, Haixu Wu, Xiaoming Wu, Jinjie Gu, Jianmin Wang, Mingsheng Long. Sequence-to-Sequence Learning with Attention Mechanism for Time Series Forecasting. 2024.
12. Guokun Lai, Wei-Cheng Chang, Yiming Yang, Hanxiao Liu. Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. 2018.
13. Ensheng Xue, Yuxuan Liang, Ziyang Wu, Yuxuan Liang, Ziyang Wu, Yanfei Kang. A Hybrid Model for Time Series Forecasting. 2021.

14. Haoxin Wang , Yipeng Mo , Kunlan Xiang , Nan Yin , Honghe Dai Bixiong Li , Songhai Fan , Site Mo. CSformer: Combining Channel Independence and Mixing for Robust Multivariate Time Series Forecasting. 2024
15. Yunhao Zhang , Junchi YanCrossformer: Transformer utilizing crossdimention dependency for multivariate time series forecasting
16. Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wan Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In AAAI Conference on Artificial Intelligence (AAAI), 2021.
17. Dazhao Du, Bing Su, and Zhewei Wei. Preformer: Predictive transformer with multi-scale segmentwise correlations for long-term time series forecasting. arXiv preprint, 2022.
18. Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In Advances in Neural Information Processing Systems (NeurIPS), 2021.
19. Ailing Zeng, Muxi Chen, Lei Zhang , Qiang Xu. Are Transformers Effective for Time Series Forecasting? 2022.
20. Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, Jayant Kalagnanam. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. 2022
21. Donghwa Shin, Edwin Zhang. Enhancing Channel-Independent Time-Series Forecasting via Cross-Variate Patch Embedding. 2025
22. Bai S, Kolter JZ, Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. 2018