УДК 519.218.82 DOI 10.55421/3034-4689_2025_28_3_103

З. З. Мингалиев

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ПРОГНОЗИРОВАНИЯ СМЕШАННЫХ ВРЕМЕННЫХ РЯДОВ С ИСПОЛЬЗОВАНИЕМ КЛАСТЕРИЗАЦИИ

Ключевые слова: прогнозирование временных рядов, программный комплекс, кластеризация, DBSCAN, LSTM, предобработка данных, параметризация временных рядов, машинное обучение, ASP.NET Core, Python.

В статье представлен программный комплекс для прогнозирования временных рядов, основанный на методах кластеризации и глубокого обучения. Комплекс включает контроллер управления, модули предобработки данных, параметризации временных рядов, кластеризации с использованием алгоритма DBSCAN, а также прогнозирования с применением рекуррентных нейронных сетей LSTM. Важной особенностью является параметризация временных рядов перед кластеризацией, что позволяет выделять ключевые статистические и спектральные характеристики данных, такие как среднее, дисперсия, коэффициенты ARIMA, тренд, количество пиков и впадин. Применение алгоритма кластеризации DBSCAN обеспечивает автоматическое формирование кластеров без необходимости заранее задавать их количество, что делает алгоритм более гибким и устойчивым к шуму. Использование отдельных моделей LSTM для каждого кластера позволяет учитывать специфические закономерности данных внутри группы, повышая точность предсказаний. Реализация выполнена на Python и ASP.NET Core, что упрощает интеграцию в различные информационные системы и обеспечивает кроссплатформенность. В статье подробно рассматриваются архитектура программного комплекса, его функциональные возможности, алгоритмы обработки данных и методология прогнозирования. Проведённые эксперименты на реальных данных подтвердили эффективность предложенного подхода, демонстрируя снижение ошибки прогноза за счёт предварительной кластеризации временных рядов. Разработанное решение может быть полезно в экономике, медицине, промышленности, энергетике и других областях, где требуется высокая точность прогнозирования. Статья будет полезна разработчикам, работающим с машинным обучением, анализом временных рядов, а также исследователям, изучающим современные методы интеллектуального анализа данных и их практическое применение.

Z. Z. Mingaliev

SOFTWARE COMPLEX FOR FORECASTING MIXED TIME SERIES USING CLUSTERING

Keywords: time series forecasting, software package, clustering, DBSCAN, LSTM, data preprocessing, time series parameterization, machine learning, ASP.NET Core, Python.

The article presents a software package for time series forecasting based on clustering and deep learning methods. The package includes a control controller, modules for data preprocessing, time series parameterization, clustering using the DBSCAN algorithm, and forecasting using LSTM recurrent neural networks. An important feature is the parameterization of time series before clustering, which allows you to highlight key statistical and spectral characteristics of the data, such as mean, variance, ARIMA coefficients, trend, number of peaks and troughs. The use of the DBSCAN clustering algorithm provides automatic formation of clusters without the need to specify their number in advance, which makes the algorithm more flexible and resistant to noise. The use of separate LSTM models for each cluster allows you to take into account specific data patterns within a group, increasing the accuracy of predictions. The implementation is made in Python and ASP.NET Core, which simplifies integration into various information systems and ensures cross-platform. The article discusses in detail the architecture of the software package, its functionality, data processing algorithms and forecasting methodology. Experiments conducted on real data confirmed the effectiveness of the proposed approach, demonstrating a reduction in forecast error due to preliminary clustering of time series. The developed solution can be useful in economics, medicine, industry, energy and other areas where high forecasting accuracy is required. The article will be useful for developers working with machine learning, time series analysis, as well as researchers studying modern methods of data mining and their practical application.

Введение

Временные ряды широко применяются для анализа и прогнозирования в таких областях, как экономика, медицина, промышленность и финансы. Однако традиционные методы прогнозирования часто сталкиваются с проблемами, связанными с высокой изменчивостью данных, наличием шумов и сложными нелинейными зависимостями [1]. В последние годы особый интерес вызывают подходы, основанные на сочетании методов машинного обучения и кластеризации, что позволяет выявлять скрытые закономерности и повышать точность прогнозов [2].

Одним из перспективных направлений является предварительная кластеризация временных рядов перед их прогнозированием [3]. Этот подход позволяет разделять данные на группы со схожими характеристиками, что способствует построению более точных моделей за счёт индивидуальной настройки параметров для каждой группы.

В данной статье представлена разработка программного комплекса, реализующего прогнозирование временных рядов с использованием кластеризации, а также проводится оценка его эффективности на реальных данных.

Постановка задачи

Целью данной работы является разработка программного комплекса, позволяющего проводить прогнозирование временных рядов с предварительной их кластеризацией. Для достижения этой цели необходимо решить следующие задачи:

1. Выбрать алгоритмы кластеризации, наиболее подходящие для группировки временных рядов.
2. Разработать методологию интеграции кластеризации и прогнозирования.
3. Реализовать программный комплекс, включающий модуль кластеризации и модуль прогнозирования.
4. Провести экспериментальное исследование на реальных данных, оценив точность прогнозирования в сравнении с традиционными методами.
5. Разработать рекомендации по использованию предложенного подхода в различных предметных областях.

Алгоритм кластеризации

Кластеризация временных рядов является важным этапом в предлагаемом подходе, так как она позволяет группировать данные по схожим характеристикам, что впоследствии повышает точность прогнозирования. В данной работе кластеризация осуществляется не над самими временными рядами, а над их параметризованными представлениями, что позволяет учитывать ключевые статистические и спектральные особенности данных.

Для разбиения временных рядов на группы предлагается использовать алгоритм DBSCAN, который анализирует плотность точек в многомерном пространстве параметров. Этот подход показал свою эффективность в предыдущих исследованиях [4], поскольку позволяет выявлять структуры данных без необходимости заранее задавать количество кластеров. Применение DBSCAN к параметризованным временным рядам делает процесс кластеризации более устойчивым к шумам и выбросам, а также облегчает обработку данных с разной временной структурой [5].

Этапы кластеризации:

1. Параметризация временных рядов.

Перед применением алгоритма DBSCAN временные ряды преобразуются в векторное представление, включающее ключевые статистические и спектральные характеристики [6]. В качестве параметров используются:

- Среднее, медианное, минимальное и максимальное значения;
- Дисперсия и площадь под графиком;
- Количество пиков и впадин;
- Коэффициенты модели ARIMA;
- Пересечения с нулевым уровнем и перцентилями;
- Показатель тренда.

2. Масштабирование параметров.

Для корректной работы алгоритма DBSCAN все параметры нормализуются, чтобы предотвратить доминирование признаков с большими числовыми диапазонами над другими характеристиками.

3. Применение DBSCAN.

Алгоритм анализирует плотность данных в пространстве признаков и формирует группы временных рядов с учетом заданных параметров ε (эпсилон) – радиуса окрестности точки – и MinPts – минимального количества точек в кластере [7]. Эти гиперпараметры подбираются экспериментально.

4. Формирование кластеров.

В результате работы алгоритма временные ряды распределяются по нескольким группам, каждая из которых содержит данные с похожими характеристиками. Дальнейшее прогнозирование осуществляется индивидуально для каждого кластера с использованием специализированной модели LSTM [8].

Методология интеграции кластеризации и прогнозирования

Интеграция кластеризации и прогнозирования временных рядов является ключевым аспектом разработанного подхода. Основная идея заключается в том, чтобы предварительно группировать временные ряды на основе их параметризованных представлений, а затем обучать специализированные модели прогнозирования для каждой из выделенных групп. Это позволяет учитывать особенности различных типов временных рядов и повышает точность предсказаний.

Разработанная методология включает в себя четыре основных этапа:

1. Параметризация временных рядов

Исходные временные ряды преобразуются в векторы признаков, описывающих их ключевые статистические и спектральные характеристики. Используются такие показатели, как среднее, дисперсия, количество пиков, коэффициенты ARIMA, тренд и другие параметры, которые несут информацию о структуре временного ряда. Нормализация параметров выполняется для приведения их к единому масштабу.

2. Кластеризация временных рядов

К параметризованным рядам применяется алгоритм DBSCAN, который формирует группы временных рядов на основе плотности данных. Этот этап позволяет выделить кластеры с временными рядами, обладающими схожими динамическими характеристиками. В результате формируются подмножества данных, которые далее обрабатываются отдельно.

3. Обучение специализированных моделей прогнозирования

Для каждого сформированного кластера обучается отдельная модель прогнозирования на основе LSTM. Такой подход позволяет учитывать специфические закономерности внутри каждой группы данных, улучшая адаптивность моделей. Обучение ведётся на исторических данных, а модели оптимизируются с использованием метрик точности, таких как средняя абсолютная процентная ошибка (MAPE).

4. Применение модели для нового временного ряда

При поступлении нового временного ряда он сначала параметризуется и классифицируется в один из кластеров, сформированных на этапе кластеризации. Затем для прогнозирования используется LSTM-модель, обученная на данных данного кластера. Такой подход позволяет избежать использования универсальной модели, которая может быть менее точной для данных с разными характеристиками.

Архитектура программного комплекса

Для автоматизированной обработки временных рядов, включая их предобработку, обучение модели

LSTM и прогнозирование, был разработан API-контроллер на основе технологии ASP.NET Core [9]. Этот контроллер обеспечивает удобный интерфейс для взаимодействия с обработчиком данных, реализованным на Python, и интеграции с клиентскими приложениями. Благодаря такому подходу, разработанное решение легко адаптируется к различным сценариям использования, позволяя отправлять запросы как из веб-интерфейса, так и с мобильных и десктопных устройств.

Структура API-контроллера включает три основных блока:

1. Блок предобработки данных – отвечает за очистку и сглаживание временных рядов, заполнение пропущенных значений, выполнение кластеризации и сохранение обученной модели кластеризации.
2. Блок обучения – осуществляет обучение моделей LSTM для каждого кластера временных рядов по отдельности.
3. Блок прогнозирования – выполняет разбиение входного временного ряда на сегменты, определяет их принадлежность к ранее сформированным кластерам и использует соответствующую обученную модель LSTM для предсказания.

Структура архитектуры программного комплекса представлена на Рис. 1.

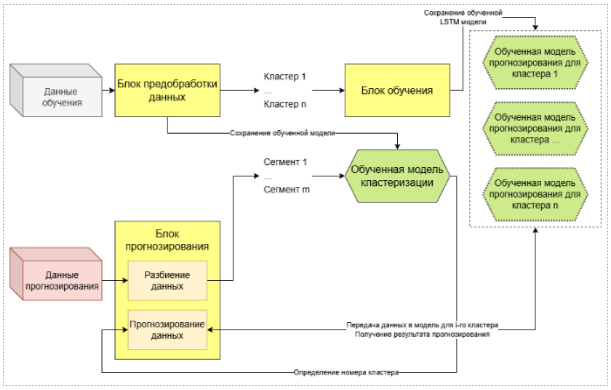

Рис. 1 – Архитектура программного комплекса

Fig. 1 – Architecture of the program complex

Средства разработки контроллера

Серверная часть API разработана на ASP.NET Core, обеспечивающем высокую производительность и кроссплатформенность. В качестве основного языка использован C#, а для документирования и тестирования API — Swagger. Работа с файлами реализована через IFormFile и System.IO, позволяя загружать и обрабатывать Excel-документы. Взаимодействие с Python-скриптами осуществляется через System.Diagnostics.Process для предобработки данных, кластеризации, обучения моделей и прогнозирования.

Средства разработки блока предобработки данных

Блок предобработки данных реализован на Python с использованием pandas для работы с Excel-файлами и numpy для вычислений. Данные разделяются на блоки, анализируются и сохраняются в новый файл с обработанными значениями и статистикой. Для извлечения особенностей временного ряда применяется find_peaks из scipy. Кластеризация выполняется алгоритмом DBSCAN (sklearn) после стандартизации данных через StandardScaler, что улучшает качество группировки [10].

Средства разработки блока обучения

Блок обучения реализован на Python с использованием pandas и numpy для подготовки данных. Временные ряды обрабатываются с интерполяцией пропущенных значений, а точность оценивается через MAPE [11]. Для обучения используется модель на основе LSTM (TensorFlow Keras) с двумя слоями, Dropout и полносвязным выходом. Компиляция выполняется с Adam и MSE. Callbacks (EarlyStopping, ReduceLROnPlateau) оптимизируют обучение. Данные нормализуются MinMaxScaler, модель обучается 500 эпох с batch_size 32. Результаты сохраняются, а прогнозы и MAPE визуализируются в виде графиков [12].

Средства разработки блока прогнозирования

Блок обучения реализован на Python с использованием pandas и numpy для обработки временных рядов, интерполяции и вычисления ошибок прогноза. Данные нормализуются с помощью StandardScaler и MinMaxScaler (scikit-learn) [13]. Для кластеризации применяется KMeans, что позволяет выбирать оптимальную LSTM-модель для прогнозирования.

LSTM-модели, обученные на разных кластерах, загружаются через tensorflow.keras [14]. Данные нормализуются и подготавливаются перед подачей в модель. Прогнозы сравниваются с реальными значениями для оценки точности.

Результаты визуализируются с помощью matplotlib: строятся графики предсказаний и MAPE. Итоговые данные сохраняются в Excel-файл, а графики – в папку с результатами.

Алгоритм работы блока предобработки данных

1. Чтение данных

Сначала выполняется проверка на корректность входных данных. Скрипт ожидает, что в качестве аргумента будет передан путь к Excel-файлу. Входной файл загружается с помощью библиотеки pandas. Если входной файл пуст, генерируется ошибка. Данные предполагаются в виде временного ряда в одном из столбцов. После загрузки данных, каждый блок временного ряда будет обрабатываться отдельно.

2. Разделение временного ряда на сегменты

Для удобства обработки временной ряд делится на несколько сегментов (чанков) с фиксированным размером. В данном случае каждый сегмент содержит 1000 строк данных. Это необходимо для более эффективного анализа и работы с большими наборами данных. Каждое разделение сохраняется как отдельный сегмент (чанк) в списке chunks.

3. Расчет статистических характеристик

Для каждого сегмента вычисляются различные статистические параметры, которые позволяют анализировать особенности временного ряда. Эти параметры включают:

- Минимальное, максимальное, среднее, медианное значения;
- Дисперсия;
- Пики и впадины;
- Площадь под графиком;
- Сумма квадратов производной;
- Тренд;
- Пересечения с уровнем среднего, нуля и перцентилей.

Каждый из этих параметров добавляется в структуру данных для последующего анализа.

4. Сохранение статистики в Excel

После вычисления статистики для каждого сегмента, результаты сохраняются в новый Excel-файл. Статистические данные собираются в отдельный лист «Statistics», который включает все параметры, рассчитанные для каждого сегмента. Для каждого сегмента создается отдельный лист в Excel, содержащий оригинальные данные. Лист «Statistics» содержит сводку по всем сегментам, включая вычисленные параметры и кластерные метки.

5. Кластеризация данных

После того как статистика для всех сегмента вычислена, выполняется кластеризация данных с использованием алгоритма DBSCAN из библиотеки scikit-learn. Для кластеризации используются следующие шаги:

1. Нормализация признаков с использованием методов библиотеки StandardScaler, что улучшает результаты кластеризации, исключая влияние разного масштаба признаков.

2. Данные кластеризуются на основе выбранных статистических признаков (кроме названия листа).

После кластеризации в данные добавляется столбец с метками кластеров, который указывает, к какому кластеру относится каждый сегмент. Эта информация сохраняется в Excel.

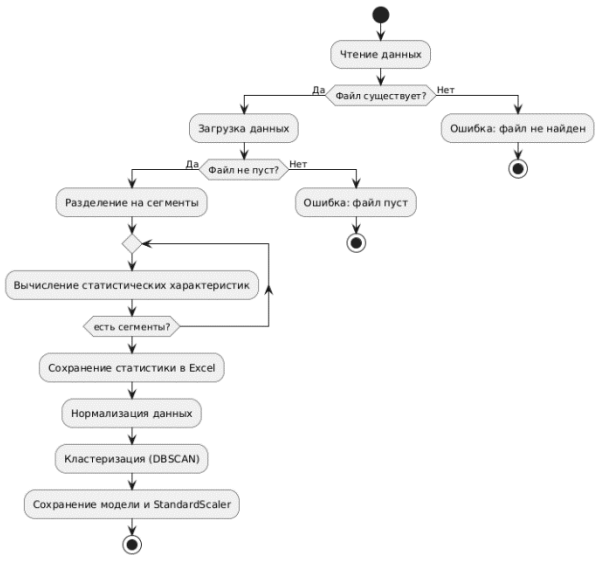

Рис. 2 – Алгоритм работы блока предобработки

Fig. 2 – Algorithm of operation of the preprocessing block

6. Сохранение модели

Модель кластеризации и стандартизатор StandardScaler сохраняются в файлы с помощью библиотеки joblib. Эти файлы могут быть использованы позднее для повторного применения модели кластеризации к новым данным. Это позволяет избежать необходимости повторного обучения модели при последующих запусках скрипта.

Схема алгоритма работы блока предварительной обработки данных представлена на Рис. 2.

Алгоритм работы блока обучения

1. Загрузка данных и модели

На первом этапе происходит загрузка исходных данных:

- Excel-файл с данными статистики, из которого извлекаются характеристики, вычисленные ранее;
- Модель DBSCAN, ранее обученная для кластеризации, загружается для разделения данных на группы.

После загрузки данных проверяется наличие ошибок. Если процесс проходит успешно, выводится сообщение об успешной загрузке.

2. Подготовка данных для каждого кластера

Далее начинается обработка каждого кластера, полученного на предыдущем шаге:

- Для каждого кластера выбираются соответствующие строки из таблицы статистики;
- Извлекается имя листа, соответствующее данным для данного кластера, и загружаются временные ряды для этого кластера из исходного Excel-файла;
- Временные ряды очищаются от пропусков с помощью линейной интерполяции, что позволяет предотвратить потерю информации из-за отсутствующих значений.

3. Сглаживание данных

Для сглаживания временного ряда применяется метод скользящего среднего, который помогает устранить краткосрочные колебания и выделить более четкие долгосрочные тренды. Размер окна для скользящего среднего по умолчанию равен 5, но его можно настроить.

4. Нормализация данных для LSTM

Данные, подготовленные на предыдущем шаге, нормализуются с помощью библиотеки MinMaxScaler. Нормализация необходима для того, чтобы все значения находились в одном и том же масштабе, что важно для корректного обучения модели LSTM.

Преобразованные данные затем делятся на последовательности, каждая из которых будет использоваться в качестве входных данных для модели LSTM. Это делается с помощью функции prepare_data_for_lstm, которая возвращает последовательности X и соответствующие метки y.

5. Построение и обучение модели LSTM

На этом этапе создается и обучается модель LSTM. Модель имеет два слоя LSTM с 100 и 50 нейронами соответственно. Эти слои являются основными для извлечения долгосрочных зависимостей во временных рядах.

Для предотвращения переобучения применяется слой Dropout с вероятностью 0.2, который случайным образом исключает часть нейронов во время обучения [15].

Модель обучается на подготовленных данных с использованием оптимизатора Adam и функции потерь mean_squared_error.

В процессе обучения устанавливаются два callback:

- EarlyStopping – он выполняет остановку обучения, если потери не улучшаются в течение определенного количества эпох (в данном случае — 10).
- ReduceLROnPlateau – он уменьшает скорость обучения, если потери не уменьшаются, чтобы помочь модели найти лучшие решения.

6. Прогнозирование и оценка модели

После обучения модели производится прогнозирование. Модель LSTM используется для предсказания значений на основе обучающей выборки. Прогнозы преобразуются обратно в оригинальный масштаб с помощью функции scaler.inverse_transform. Далее вычисляется MAPE для каждой точки предсказания, что позволяет оценить точность прогноза.

7. Сохранение графиков

Для каждого кластера сохраняются два графика:

- График прогнозов модели и истинных значения временного ряда, позволяя наглядно оценить качество прогноза.
- Столбчатая диаграмма, которая показывает ошибку MAPE для каждой точки во временном ряду, что помогает выявить точки с наибольшей ошибкой.

Эти графики сохраняются в виде изображений в формате PNG для дальнейшего анализа.

8. Сохранение модели

После обучения и прогнозирования модель LSTM сохраняется в файл для каждого кластера с использованием функции model.save. Это позволяет повторно использовать обученную модель для будущих прогнозов без необходимости повторного обучения.

9. Вычисление общей точности

После обработки всех кластеров вычисляется общая точность модели по всем кластерам. Для каждого кластера вычисляется средний MAPE, который затем используется для вычисления общей точности прогноза по всем данным. Результат (средняя точность) выводится на экран и сохраняется в логах.

10. Логирование

Для удобства отслеживания выполнения скрипта используется система логирования, которая сохраняет сообщения об успешной загрузке данных, обучении моделей, возникших ошибках и других ключевых событиях. Логи записываются в файл training_logs.txt.

Схема алгоритма работы блока предварительной обработки данных представлена на Рис. 3.

Рис. 3 – Алгоритм работы блока обучения

Fig. 3 – Algorithm of operation of the training block

Алгоритм работы блока прогнозирования

1. Загрузка данных и моделей

На начальном этапе происходит загрузка всех необходимых данных и моделей:

- Модель кластеризации временных рядов, которая используется для определения, к какому кластеру принадлежит каждый сегмент данных;
- Стандартизатор (scaler), использованный для масштабирования данных во время обучения модели кластеризации. Этот же стандартизатор будет применяться к новым данным для корректного их масштабирования перед кластеризацией;
- Модели LSTM для каждого кластера, которые были предварительно обучены и сохранены в виде файлов.

2. Чтение и подготовка данных

Входной файл с временным рядом загружается с использованием библиотеки pandas. Данные извлекаются из второго столбца, который предполагается быть временным рядом, и затем происходит их разделение на сегменты — последовательности данных размером по 1000 значений.

3. Вычисление параметров временного ряда

Для каждого сегмента временного ряда вычисляются ключевые статистические параметры, которые затем используются для кластеризации:

- Минимум, максимум, среднее, медиана, дисперсия и т.д.;

- Количество пиков и впадин, что помогает анализировать основные экстремумы в ряду;
- Тренд временного ряда, определяемый с использованием полиномиальной регрессии первого порядка;
- Пересечения временного ряда с его средней линией и перцентилями, которые дают информацию о частоте изменения направления временного ряда.

Эти параметры сохраняются в виде объекта DataFrame для последующего использования.

4. Кластеризация данных

После вычисления статистических параметров, они масштабируются с использованием того же scaler, который был применен на этапе обучения модели кластеризации. Затем, с помощью модели DBSCAN, определяется, к какому кластеру принадлежит текущий сегмент данных.

5. Подготовка данных для LSTM

Для каждого сегмента данных, в зависимости от его кластера, подготавливаются данные для предсказания с использованием модели LSTM:

- Данные нормализуются с использованием MinMaxScaler для приведения всех значений в диапазон [0, 1];
- Данные делятся на последовательности длиной time_step, которые затем подаются в модель LSTM для обучения.

6. Прогнозирование с использованием модели LSTM

После того, как данные подготовлены и соответствующая модель LSTM загружена, выполняется прогнозирование:

- Модель LSTM используется для предсказания значений временного ряда для каждого сегмента;
- Прогнозы преобразуются обратно в оригинальный масштаб с помощью инвертирования нормализации, выполненной ранее на данных;
- Истинные значения для сравнения берутся из того же сегмента, но начиная с индекса, равного time_step, так как предсказания начинаются только после первого полного временного шага.

7. Оценка качества прогнозирования

Для оценки точности модели используется метрика MAPE, которая вычисляется как среднее процентное отклонение предсказанных значений от истинных.

8. Визуализация результатов

Для каждого сегмента создаются два графика:

- График временного ряда, который отображает истинные и предсказанные значения на одном графике для визуальной оценки качества прогнозирования;
- График ошибок, который отображает ошибку MAPE по временному ряду для каждого сегмента, что помогает оценить, в каких точках модель дает наибольшие отклонения.

Эти графики сохраняются в каталог prediction_plots, созданный для хранения всех визуализированных результатов.

9. Сохранение результатов

После выполнения прогнозов для всех сегментов, результаты (включая точность и MAPE) сохраняются в файл prediction_results.xlsx в виде таблицы с номером сегмента, кластером, значениями MAPE и точности для каждого сегмента.

10. Финальная визуализация MAPE по всем сегментам

После обработки всех сегментов строится общий график MAPE по сегментам, который позволяет наглядно увидеть, какая точность прогноза для каждого сегмента.

11. Логирование

Весь процесс работы скрипта логируется в файл predict_chunk_lstm.log, что позволяет отслеживать возможные ошибки и получать информацию о ходе выполнения скрипта.

Схема алгоритма работы блока предварительной обработки данных представлена на Рис. 4.

Рис. 4 – Алгоритм работы блока прогнозирования

Fig. 4 – Algorithm of the forecasting block operation

Заключение

В данной работе был представлен программный комплекс для прогнозирования временных рядов с использованием предварительной кластеризации. Разработанный подход позволяет улучшить точность прогноза за счет разделения данных на группы с схожими характеристиками, что позволяет более точно подбирать модели и настраивать их параметры для каждой группы.

Экспериментальное исследование показало, что предложенный метод позволяет значительно повысить точность прогнозирования по сравнению с традиционными подходами, что делает его перспективным инструментом для использования в таких областях, как экономика, медицина, финансы и промышленность.

Предложенный подход открывает новые возможности для применения машинного обучения и кластеризации в анализе временных рядов и может стать полезным инструментом для решения задач прогнозирования в различных сферах деятельности.

Литература

1. Абраменко, Д. А. Анализ точности методов прогнозирования финансовых результатов франчайзинга / Д. А. Абраменко, К. И. Бушмелева // Успехи кибернетики. – 2024. – Т. 5, № 1. – С. 40-46. – DOI 10.51790/2712-9942-2024-5-1-05. – EDN IXDTNM.
2. Рогулин, Р. С. Использование методов анализа данных и машинного обучения для прогнозирования и планирования спроса при управлении цепочками поставок / Р. С. Рогулин // Теоретическая экономика. – 2023. – № 8(104). – С. 35-54. – EDN JIYIYZ.
3. Астахова, Н. Н. Метод прогнозирования групп временных рядов с применением алгоритмов кластерного анализа / Н. Н. Астахова, Л. А. Демидова // Прикаспийский журнал: управление и высокие технологии. – 2015. – № 2(30). – С. 59-79. – EDN UAAUEP.
4. Мингалиев, З. З. Метод адаптивного прогнозирования данных, зависящих от времени, на основе LSTM-моделей с использованием кластеризации временных рядов / З. З. Мингалиев // Вестник Технологического университета. – 2025. – Т. 28, № 2. – С. 70-78. – DOI 10.55421/1998-7072_2025_28_2_70. – EDN QJHKZS.
5. C. -F. Chien and B. Suwattananuruk, "Density-Based Spatial Clustering of Applications With Noise (DBSCAN) for Probe Card Production for Advanced Quality Control of Wafer Probing Test," in IEEE Transactions on Semiconductor Manufacturing, vol. 37, no. 4, pp. 567-575, Nov. 2024, doi: 10.1109/TSM.2024.3468000.
6. Tawakuli, Amal & Havers, Bastian & Gulisano, Vincenzo & Kaiser, Daniel. (2024). Survey: Time-Series Data Preprocessing: A Survey and an Empirical Analysis. Journal of Engineering Research. 10.1016/j.jer.2024.02.018.
7. Трушкова, К. Н. Кластеризация СПУТНИКОВЫХ ИЗОБРАЖЕНИЙ ОБЛАЧНЫХ ПОЛЕЙ НА ОСНОВЕ АЛГОРИТМА DBSCAN / К. Н. Трушкова, В. Т. Калайда // Известия вузов. Физика. – 2013. – Т. 56, № 8-3. – С. 356-358. – EDN RKSBXP.
8. Кондратьева, Т. Н. Прогнозирование тенденции финансовых временных рядов с помощью нейронной сети LSTM / Т. Н. Кондратьева // Интернет-журнал Науковедение. – 2017. – Т. 9, № 4. – С. 61. – EDN ZIGGCR.
9. Ерхов, Р. В. Преимущества разработки веб-приложения на платформе asp.net core / Р. В. Ерхов // Новые информационные технологии в научных исследованиях : материалы XXII Всероссийской научно-технической конференции студентов, молодых ученых и специалистов, Рязань, 15–17 ноября 2017 года / Рязанский государственный радиотехнический университет. – Рязань: Рязанский государственный радиотехнический университет, 2017. – С. 128-130. – EDN ZRXGSF.
10. Ларин, С. Э. Библиотеки Python для анализа данных: предобработка и подготовка данных / С. Э. Ларин, В. Ю. Белаш // Дневник науки. – 2023. – № 12(84). – EDN WWGIYV.
11. Statology. (2020). Mean Absolute Percentage Error (MAPE) in Machine Learning. Retrieved from https://www.statology.org/mape-in-machine-learning/.
12. Картузов, А. В. Программная модель нейронной сети LSTM для прогнозирования временных рядов / А. В. Картузов, Т. В. Картузова, С. В. Храмцов // Научно-технический вестник Поволжья. – 2021. – № 12. – С. 159-162. – EDN HAJJGX.
13. Гребнев, К. Н. Машинное обучение с помощью библиотеки scikit-learn языка Python / К. Н. Гребнев // Математический вестник педвузов и университетов Волго-Вятского региона. – 2017. – № 19. – С. 277-281. – EDN YMABSJ.
14. Эдель, Г. Е. Глубокое обучение с использованием библиотеки TensorFlow / Г. Е. Эдель // Электронные средства и системы управления. Материалы докладов Международной научно-практической конференции. – 2020. – № 1-2. – С. 162-164. – EDN XRLOHY.
15. Харченко, А. С. Методы борьбы с переобучением в нейронных сетях / А. С. Харченко, М. Ю. Новоженов, В. В. Ткаченко // Цифровизация экономики: направления, методы, инструменты : СБОРНИК МАТЕРИАЛОВ V ВСЕРОССИЙСКОЙ НАУЧНО-ПРАКТИЧЕСКОЙ КОНФЕРЕНЦИИ, Краснодар, 16–21 января 2023 года. – Краснодар: Кубанский государственный аграрный университет имени И.Т. Трубилина, 2023. – С. 319-322. – EDN JQTKKJ.

References

1. Abramenko, D. A. Analysis of the accuracy of methods for forecasting the financial results of franchising / D. A. Abramenko, K. I. Bushmeleva // Advances in Cybernetics. - 2024. - Vol. 5, No. 1. - P. 40-46. - DOI 10.51790/2712-9942-2024-5-1-05. - EDN IXDTNM.
2. Rogulin, R. S. Using data analysis and machine learning methods for forecasting and planning demand in supply chain management / R. S. Rogulin // Theoretical Economics. - 2023. - No. 8 (104). - P. 35-54. - EDN JIYIYZ.
3. Astakhova, N. N. Method for forecasting groups of time series using cluster analysis algorithms / N. N. Astakhova, L. A. Demidova // Caspian Journal: Management and High Technologies. - 2015. - No. 2 (30). - P. 59-79. - EDN UAAUEP.
4. Mingaliyev, Z. Z. Method for adaptive forecasting of time-dependent data based on LSTM models using time series clustering / Z. Z. Mingaliyev // Herald Technological University. - 2025. - Vol. 28, No. 2. - P. 70-78. - DOI 10.55421/1998-7072_2025_28_2_70. - EDN QJHKZS.
5. C. -F. Chien and B. Suwattananuruk, "Density-Based Spatial Clustering of Applications With Noise (DBSCAN) for Probe Card Production for Advanced Quality Control of Wafer Probing Test," in IEEE Transactions on Semiconductor Manufacturing, vol. 37, no. 4, pp. 567-575, Nov. 2024, doi: 10.1109/TSM.2024.3468000.
6. Tawakuli, Amal & Havers, Bastian & Gulisano, Vincenzo & Kaiser, Daniel. (2024). Survey: Time-Series Data Preprocessing: A Survey and an Empirical Analysis. Journal of Engineering Research. 10.1016/j.jer.2024.02.018.
7. Trushkova, K. N. Clustering of SATELLITE IMAGES of CLOUD FIELDS BASED ON THE DBSCAN ALGORITHM / K. N. Trushkova, V. T. Kalayda // News of universities. Physics. - 2013. - Vol. 56, No. 8-3. - P. 356-358. - EDN RKSBXP.
8. Kondratieva, T. N. Forecasting the trend of financial time series using the LSTM neural network / T. N. Kondratieva // Internet journal Naukovedenie. - 2017. - Vol. 9, No. 4. - P. 61. - EDN ZIGGCR.
9. Erkhov, R. V. Advantages of developing a web application on the asp.net core platform / R. V. Erkhov // New information technologies in scientific research: Proceedings of the

XXII All-Russian scientific and technical conference of students, young scientists and specialists, Ryazan, November 15-17, 2017 / Ryazan State Radio Engineering University. - Ryazan: Ryazan State Radio Engineering University, 2017. - Pp. 128-130. - EDN ZRXGSF.
10. Larin, S. E. Python libraries for data analysis: preprocessing and data preparation / S. E. Larin, V. Yu. Belash // Science Diary. - 2023. - No. 12 (84). - EDN WWGIYV.
11. Statology. (2020). Mean Absolute Percentage Error (MAPE) in Machine Learning. Retrieved from https://www.statology.org/mape-in-machine-learning/
12. Kartuzov, A. V. Software model of the LSTM neural network for forecasting time series / A. V. Kartuzov, T. V. Kartuzova, S. V. Khramtsov // Scientific and Technical Bulletin of the Volga Region. - 2021. - No. 12. - P. 159-162. - EDN HAJJGX.
13. Grebnev, K. N. Machine learning using the scikit-learn library of the Python language / K. N. Grebnev // Mathematical Bulletin of Pedagogical Universities and Universities of the Volga-Vyatka Region. - 2017. - No. 19. - P. 277-281. - EDN YMABSJ.
14. Edel, G. E. Deep learning using the TensorFlow library / G. E. Edel // Electronic means and control systems. Proceedings of the reports of the International scientific and practical conference. - 2020. - No. 1-2. - P. 162-164. - EDN XRLOHY.
15. Kharchenko, A. S. Methods of combating overfitting in neural networks / A. S. Kharchenko, M. Yu. Novozhenov, V. V. Tkachenko // Digitalization of the economy: directions, methods, tools: COLLECTION OF MATERIALS OF THE V ALL-RUSSIAN SCIENTIFIC AND PRACTICAL CONFERENCE, Krasnodar, January 16-21, 2023. - Krasnodar: Kuban State Agrarian University named after I.T. Trubilin, 2023. - P. 319-322. – EDN JQTKKJ.

© З. З. Мингалиев – аспирант, Казанский национальный исследовательский технический университет им. А.Н. Туполева, Казань, Россия, zaid97abyi@gmail.com

© Z. Z. Mingaliev – PhD-student, Kazan National Research Technical University named after A.N. Tupolev, Kazan, Russia, zaid97abyi@gmail.com

Дата поступления рукописи в редакцию –08.03.25.
Дата принятия рукописи в печать – 13.03.25.