

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

**ПРИЛОЖЕНИЕ ДЛЯ АНАЛИЗА МЕТЕОПРОГНОЗОВ
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

студентки 4 курса 441 группы

направления 02.03.03 — Математическое обеспечение и администрирование
информационных систем

факультета компьютерных наук и информационных технологий

Черновой Кристины Дмитриевны

Научный руководитель:

проф. кафедры информатики и

программирования, д.т.н

А. С. Фалькович

подпись, дата

Зав. кафедрой:

к. ф.-м. н., доцент

М. В. Огнева

подпись, дата

Саратов 2023

ВВЕДЕНИЕ

Актуальность темы. Интересными для изучения являются прогнозируемые метеорологическими сайтами значения погодных показателей: каждый из них старается дать наиболее точный прогноз, в том числе, на наибольший срок. Поскольку в последние десятилетия точность среднесрочных прогнозов значительно увеличилась, сейчас можно говорить уже не об оправдываемости прогнозов («правильный-неправильный»), а о погрешности отдельных прогнозируемых параметров погоды, как о погрешности любых других измеряемых физических величин. Верификация прогнозов погоды (оценка качества) является обязательным этапом при разработке или усовершенствовании численных моделей. Данная научная оценка предусматривает расчет того или иного ряда показателей успешности для определения сильных и слабых сторон методов и технологий прогнозирования, а также оценивания конкретных путей развития и совершенствования программ.

Цель бакалаврской работы – создание приложения для анализа среднесрочных прогнозов метеоданных.

Поставленная цель определила **следующие задачи:**

1. ознакомление с литературой по прогнозированию погоды;
2. выбор метеорологических сайтов для дальнейшего сбора информации;
3. ознакомление с источниками по автоматизированному сбору данных с веб-страницы;
4. обзор существующих систем для автоматизированного сбора данных;
5. обзор методов автоматизированного сбора данных на языке программирования C#;
6. разработка алгоритма автоматизированного сбора данных о среднесрочных прогнозах погоды с сайтов gismeteo.ru и openweathermap.org на языке программирования C#;

7. реализация методов для вычисления отклонений среднесрочных прогнозов для определенного дня, а также средних отклонений для определенного сайта на языке программирования C#;
8. разработка структуры хранилища данных с использованием MongoDB;
9. создание WPF приложения с возможностью отображения графиков, построенных из полученных погодных показателей, а также сохранения данных в формате .xlsx;
10. приведение примеров предварительного анализа полученных данных.

Теоретическая и/или практическая значимость бакалаврской работы. Значимость данной работы заключается в сохранении прогнозных показателей с веб-страниц gismeteo.ru и openweathermap.org с возможностью их дальнейшего анализа и отображения в графическом формате на примере сайта meteoinfo.ru. Данная информация может быть использована в разных целях: для улучшения алгоритмов прогнозирования погоды, создания и обучения собственных моделей для прогнозирования погоды, выявления сбоев работы сайтов погоды, определения степени достоверности среднесрочных прогнозов на указанных сайтах погоды, определения аномалий среди показателей, а также отклонения прогнозов погоды от фактических погодных показателей, отображаемых на указанных сайтах.

Структура и объём работы. Бакалаврская работа состоит из введения, 5 разделов, заключения, списка использованных источников и 13 приложений. Общий объем работы – 159 страниц, из них 75 страниц – основное содержание, включая 31 рисунок, цифровой носитель в качестве приложения, список использованных источников информации – 27 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Анализ прогнозов погоды и причины ошибок прогнозирования» посвящен основным определениям видов прогнозов погоды: сверхкраткосрочный, краткосрочный, среднесрочный и долгосрочный. Также приведены три основные причины, из-за которых

невозможно избавиться от ошибок прогнозирования погоды: неполнота и неточность знаний о текущем состоянии атмосферы, несовершенство используемых прогностических методов и моделей, проблема предсказуемости. Также в главе описана актуальность и значимость данной работы.

Второй раздел «Определение способа сбора данных. Основные понятия и определения технологии веб-скрейпинга» посвящен основным определениям технологии веб-скрейпинга, юридическим ограничениям использования данной технологии. Также был произведен обзор существующих парсеров веб-сайтов. На основе проведенного анализа удалось сделать следующие выводы: рассмотренные парсеры не справляются с поставленной задачей целиком, из-за решения ими других классов задач или невозможностью систематического сбора данных. Данные выводы приводят к тому, что для разработки приложения для анализа погодных данных, необходимо разработать собственный алгоритм автоматизированного сбора данных с веб-страниц, используя возможности языка программирования C#.

Третий раздел «Описание архитектуры «клиент-сервер»» посвящен основным определениям и видам клиент-серверной архитектуры приложений. Также были описаны основные принципы работы технологии удаленного вызова процедур - gRPC. Данная технология была выбрана в работе, в большей степени, из-за простоты понимания ее принципов работы и реализации. Также огромным плюсом является автоматическая генерация кода, которая позволяет абстрагироваться от удаленного взаимодействия клиента с сервером, что позволяет в их реализациях вызывать нужные методы в привычном формате. Необходимо упомянуть, что при работе с большим количеством данных очень важна скорость передачи объектов, которую данная технология обеспечивает за счет более оптимизированного формата передачи данных.

Четвертый раздел «Программная реализация приложения на языке C#» посвящен реализации приложения для анализа метеопрогнозов. В данной главе приведено описание проектирования приложения.

В качестве архитектуры приложения была выбрана трехуровневая архитектура.

Серверная часть приложения включает в себя основную логику и работу с данными. Сбор данных с веб-страниц производится с помощью отдельного приложения, представленного в виде службы Windows, периодичность работы которого контролируется stop-выражением. Таким образом, сбор прогнозов погоды и их сохранение в базу данных происходит независимо от работы основного сервера. Основной сервер отвечает за работу с клиентской частью приложения, отвечая на запросы клиента о получении данных и их обработку перед передачей клиенту для дальнейшего графического отображения, а также выгрузки данных в формате .xlsx. Серверная часть приложения также содержит реализацию методов для получения данных из базы данных, вычисления отклонений прогнозных значений для выбранного дня от фактических, а также вычисления средних отклонений прогнозных значений от фактических для выбранного пользователем сайта.

Сохранение данных производится в базу данных с использованием системы управления базами данных MongoDB. Данная СУБД не использует реляционную модель баз данных, относится к классу NoSQL СУБД и работает с документами, а не с записями. Вместо таблиц MongoDB использует коллекции, а также использует собственный формат хранения информации – Binary JavaScript Object Notation (BSON). Было принято решение не использовать реляционные базы данных из-за сложностей организации реляционной структуры для сохранения прогнозов погоды, а также, сложностей приведения сохраняемых данных в удобные для использования в работе формат. В качестве сохраняемых данных выбраны следующие – температура воздуха, давление, скорость ветра, влажность.

Одной из задач работы являлось создание алгоритма автоматизированного сбора данных с веб-страниц `gismeteo.ru` и `openweathermap.org`, с использованием возможностей языка программирования C#. Собраны были следующие прогнозные показатели: температура воздуха, скорость ветра, влажность и давление. В работе был произведен анализ различных методов парсинга данных: с помощью библиотек языка программирования C# (`HTMLAgilityPack` и `AngleSharp`), а также с помощью использования API сайта. По итогам сравнения алгоритмов, можно сделать вывод, что использование API является самым простым и оптимальным способом автоматизированного сбора данных с веб-страниц. Тем не менее, для использования API необходимо получить токен, который предоставляют представители веб-сайтов. Представители сайта `gismeteo.ru` не предоставили токен для использования API в данной работе. Поэтому, были проанализированы популярные библиотеки для парсинга данных `HTMLAgilityPack` и `AngleSharp`. Основным преимуществом библиотеки `AngleSharp`, в отличие от `HTMLAgilityPack`, является наличие большого количества методов, позволяющих получить доступ к отдельным элементам страницы, не используя полное xPath-выражение пути до нужного элемента. Также, преимуществом `AngleSharp` можно считать автоматическую загрузку веб-страницы, используя лишь необходимую конфигурацию, в то время, как получение кода веб-страницы, для дальнейшей работы с библиотекой `HTMLAgilityPack`, необходимо реализовать самостоятельно.

Также в работе был произведен анализ работы двух библиотек для автоматического создания и редактирования документа в формате `.xlsx`. В результате сравнения, для работы была выбрана библиотека `DocumentFormat.OpenXml`. Основным преимуществом библиотеки является то, что для создания документа она не привязывается к конкретному приложению. `Microsoft.Office.Interop.Excel` в фоновом режиме открывает приложение `Microsoft Excel` и создает документ. Это создает большие трудности, если на компьютере не установлено или не активировано данное

приложение. За счет этого, увеличивается скорость создания файла с помощью `DocumentFormat.OpenXml`, т.к. она не зависит от замедления работы стороннего приложения или даже возникших в нем ошибок. Недостатки данной библиотеки – сложность написания кода, а также большое количество повторения кода для создания однотипных объектов.

Клиентская часть приложения содержит проект на платформе WPF. В работе был использован паттерн MVVM. Это паттерн разработки, позволяющий разделить приложение на три функциональные части: Model – основная логика программы (работа с данными, вычисления, запросы и так далее), View – вид или представление (пользовательский интерфейс), ViewModel – модель представления, которая служит прослойкой между View и Model. Для рисования графиков в WPF была выбрана библиотека LiveCharts, позволяющая создавать интерактивные графики.

Также, стоит упомянуть, что главным принципом для правильной работы многослойного приложения является – IoC (Inversion of Control). Достигается он использованием технологии внедрения зависимостей – DI (Dependency Injection). В работе в качестве DI-контейнера, библиотеки, позволяющей сохранить ссылки на зарегистрированные объекты, используется Autofac.

Общий функционал работы клиентского интерфейса представлен в следующем виде: сначала пользователь выбирает интересующий его сайт погоды в верхнем меню приложения из выпадающего списка, после этого становится активным календарь, в котором указан диапазон дат, на которые доступны прогнозы для выбранного сайта, далее с правой стороны окна появляется чек-лист с доступными временными промежутками, после выбора которых активируются кнопки, названия которых соответствуют погодным показателям, содержащие внутри себя команды для отрисовки графиков, описанные далее в работе. В верхней части окна представлены графики для выбранного пользователем дня: графики прогнозов и графики отклонений прогнозных значений от фактических. Выбрать интересующий для

просмотра можно с помощью двух кнопок: одна из которых не активна, если для выбранного дня отсутствуют фактические показатели, а значит, невозможно рассчитать график отклонений. В нижней части окна представлены графики средних отклонений для выбранного пользователем сайта. Для его корректного отображения в выпадающем списке слева необходимо выбрать число дней, за которые необходимо увидеть средние отклонения.

Пятый раздел «Примеры предварительного анализа полученных данных» посвящен примерам предварительного анализа полученных данных.

Для собранных с сайта gismeteo.ru показателей было произведено сравнение средних отклонений за зимнее и весеннее время с помощью выгруженных данных в формате `.xlsx`. Были рассчитаны средние отклонения и их абсолютные значения с использованием Microsoft Excel отдельно за весенние и зимние дни с помощью выгруженного из приложения файла. Семидневные прогнозы с данного веб-сайта были собраны с использованием библиотеки `AngleSharp`. В качестве предварительных результатов, можно сделать следующий вывод: в зимнее время лучше всего были показатели на 1, 7, 13, 19 часов, что позволяет выдвинуть гипотезу об использовании показателей с метеостанции, выдающих значения с интервалом в 3 и 6 часов, поэтому, за счет большего числа значений, прогнозы были лучше в эти временные промежутки. Весенние показатели указывают на ухудшение качества прогноза в целом. Самыми плохо прогнозируемыми показателями являются влажность и давление, а также весной плохо прогнозируется скорость ветра.

Анализ средних отклонений прогнозных значений для сайта openweathermap.org проводился с использованием созданного в работе приложения. Стоит заметить, что с помощью API с данного сайта поступали значения прогнозов на 5 дней вперед. В качестве предварительных

результатов, можно сделать следующий вывод: в прогнозах до 3 дней наименьшие отклонения были получены на 0, 6, 12, 18 часов.

Среди всех полученных показателей можно сделать следующие выводы: семидневные прогнозы практически для всех показателей для любых временных промежутков имеют самые большие отклонения, в среднесрочных прогнозах до 4 дней практически для всех погодных показателей наименьшие отклонения от фактических были получены для прогнозов на 1, 7, 13 и 19 часов для сайта gismeteo.ru, а также для прогнозов на 0, 6, 12, 18 часов для сайта openweathermap.org (или по крайней мере для нескольких из этих четырех временных показателей). Это позволяет выдвинуть гипотезу о том, что при формировании метеопрогнозов используются данные с метеостанций, которые выдают показатели не только через каждые 3 часа (8 периодов), но и через каждые 6 часов (4 периода), что ухудшает качество прогноза для промежуточных временных значений. Также, самыми сложно прогнозируемыми показателями являются скорость ветра и влажность.

ЗАКЛЮЧЕНИЕ

Цель работы была успешно выполнена: разработано приложение для анализа среднесрочных прогнозов метеоданных.

В процессе работы была изучена литература по прогнозированию погоды и анализу прогнозов, а также литература по автоматизированному сбору данных с веб-страницы. Были рассмотрены существующие парсинг-системы и проверена их способность решить поставленную в работе задачу по сбору данных о погоде с метеорологических сайтов. проведено сравнение способов автоматизированного сбора данных с веб-страниц, а также был разработан алгоритм автоматизированного сбора прогнозов погоды с сайтов gismeteo.ru и openweathermap.org на языке C#. Была разработана структура хранилища полученных данных с использованием MongoDB. Были разработаны алгоритмы для вычисления отклонений среднесрочных прогнозов для выбранного дня, а также средних отклонений для выбранного

сайта на языке программирования C#. Также было разработано WPF приложение с возможностью отображения графиков, построенных из полученных погодных показателей, а также сохранения данных в формате .xlsx. Были приведены примеры первичного анализа полученных данных.

Основные источники информации:

1. Старченко А.В. Оценка успешности численного прогноза элементов погоды по мезомасштабной модели атмосферы высокого разрешения TSUNM3 / А.В. Старченко, И.В. Кужевская., Л.И. Кижнер, Н.К. Барашкова, М.А. Волкова, А.А. Барт // Оптика атмосферы и океана. 2019. Т. 32, № 1. С. 57-61.

2. Попов А.Ю. Парсинг электронных ресурсов. Библиотека Selenium или fake useragent? / А.Ю. Попов, М.В. Ремез, Е.В. Жилина, М.И. Ожиганова // Информатизация в цифровой экономике. 2022. Т. 3, № 4. С. 197-210.

3. Грязева М.А. Скрапинг интернет-ресурсов / М.А. Грязева // Журнал Суда по интеллектуальным правам. 2020. № 4 (30). С. 42-47.

4. Архитектура «Клиент-Сервер» [Электронный ресурс] URL: <https://itelon.ru/blog/arkhitektura-klient-server/> (дата обращения: 20.02.23)

5. gRPC в качестве протокола межсервисного взаимодействия. Доклад Яндекса [Электронный ресурс] URL: <https://habr.com/ru/company/yandex/blog/484068/> (дата обращения: 20.02.23)

6. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Дж. Рихтер ; 4-е изд. — СПб.: Питер, 2013. — 896 с

7. Creating a Windows Service with .NET 6 [Электронный ресурс] URL: <https://csharp.christiannagel.com/2022/03/22/windows-service-2/> (дата обращения: 25.03.23)

8. MongoDB C# Driver [Электронный ресурс] URL: <https://www.mongodb.com/docs/drivers/csharp/current/> (дата обращения: 25.03.23)

9. MVVM: проектирование приложений для Windows
[Электронный ресурс] (дата обращения: 20.02.23)

10. LiveCharts [Электронный ресурс] URL: <https://v0.lvcharts.com/>
(дата обращения: 25.03.23)

11. Autofac [Электронный ресурс] URL: <https://autofac.org/> (дата
обращения: 25.03.23)