

Разработка портала экологического мониторинга

А.В. Абузов, С.М. Бурков, Д.В. Вегера, А.А. Новикова, И.С. Пархоменко

Тихоокеанский государственный университет, Хабаровск

Аннотация: Статья посвящена разработке веб-портала для мониторинга и прогнозирования качества атмосферного воздуха в Хабаровском крае. В работе представлен анализ существующих решений в области экологического мониторинга, выявлены их ключевые недостатки, такие как отсутствие данных в реальном времени, ограниченный функционал и устаревшие интерфейсы. Авторами предложено современное решение на базе технологического стека Python/Django и PostgreSQL, обеспечивающее сбор, обработку и визуализацию данных с датчиков качества воздуха. Особое внимание уделено реализации прогнозирования концентраций вредных газов с использованием рекуррентной нейронной сети, а также созданию интуитивно понятного пользовательского интерфейса с интерактивной картой на основе OpenStreetMap. В статье подробно описана архитектура системы, включая серверную часть, базу данных и фронтенд-реализацию, а также применяемые методы обеспечения производительности и безопасности. Результатом работы стал функциональный веб-портал, предоставляющий актуальную информацию о состоянии атмосферного воздуха, прогнозные данные и удобные инструменты визуализации. Разработанное решение демонстрирует высокую эффективность и может быть масштабировано для применения в других регионах.

Ключевые слова: экологический мониторинг, качество атмосферного воздуха, веб-портал, прогнозирование, Django, Python, PostgreSQL, нейронные сети, OpenStreetMap

I. Введение

На сегодняшний день на территории Хабаровского края отсутствует сайт, позволяющий любому жителю оценить в реальном времени и спрогнозировать состояние атмосферного воздуха.

Сайт – это цифровой ресурс в сети Интернет, состоящий из совокупности веб-страниц (документов), объединенных общей тематикой и связанных между собой гиперссылками [1]. Каждый сайт принадлежит конкретному юридическому или физическому лицу и идентифицируется уникальным доменным именем, которое служит его адресом. Объем сайта может варьироваться от одностраничного ресурса до крупного портала с тысячами страниц. Технически веб-страницы представляют собой текстовые файлы, написанные с использованием языков разметки (HTML, CSS) и программирования (JavaScript, PHP, Python), которые интерпретируются браузерами для отображения контента пользователям.

Объектом данного исследования является сайт, отображающий данные экологического мониторинга атмосферного воздуха Хабаровского края в режиме реального времени с возможностью прогнозирования концентрации газов.

Результатом данной работы являются описание предметной области, описание реализации сайта.

Анализ существующих решений

В таблице № 1 представлены критерии (функциональность, технологии, UX/UI) и аналоги других сайтов, включая разрабатываемый.

Исходя из сравнительного анализа были выявлены недостатки подобных сайтов, а также их преимущества. Разрабатываемый сайт будет лишен этих недостатков, и при этом обладать всеми достоинствами.

Современные научные исследования в области экологического мониторинга демонстрируют растущую роль технологий искусственного интеллекта и распределённых сетей датчиков для повышения точности прогнозирования качества атмосферного воздуха. Работы [2] и [3] подтверждают эффективность рекуррентных нейронных сетей в прогнозировании концентраций загрязняющих веществ, в то время как исследование [4] обосновывает важность интеграции IoT-устройств для создания комплексных систем мониторинга. Особое значение имеют международные стандарты [5], устанавливающие нормативы качества воздуха и подчёркивающие необходимость открытого доступа к экологическим данным. Эти исследования не только подтверждают актуальность разрабатываемого портала, но и задают научно обоснованные критерии для оценки его эффективности, такие как точность прогнозов, полнота данных и соответствие глобальным рекомендациям по охране здоровья населения.

Таблица № 1

Сравнительный анализ решений

Критерий	ГИС "Метео-ДВ" https://meteo-dv.ru/hydro_dfo/	Мониторинг в Камчатском крае https://eco.kamgov.ru/	Разрабатываемый сайт
Функциональность	Гидрометеорологические данные (осадки, уровень воды, температура) - Статистика и архивы	- Данные о загрязнении воздуха, воды, почвы - Отчеты и аналитика	- Мониторинг атмосферного воздуха - Учет вывозимого леса - Данные в реальном времени
Технологии	- GIS-интеграция (картографические данные) - Возможно, API для данных - Простая верстка (HTML, CSS, JS)	- Возможно, CMS (WordPress или аналоги) - Базовая визуализация данных (графики, таблицы) - Нет сложной аналитики	- Современный стек (Python/Django[6]) - API для интеграции с датчиками - ML для прогнозирования - Базы данных (PostgreSQL)
UX/UI (удобство использования)	- Устаревший дизайн - Минималистичная навигация	- Более современный интерфейс - Удобные фильтры и поиск	- Интерактивная карта с маркерами на базе OpenStreetMap [7]. с использованием библиотеки Leaflet [8]. - Дашборды с аналитикой с использованием Canvas - Адаптивный дизайн
Режим реального времени	нет (в части концентрации газов)	частично (не все показатели обновляются в реальном времени)	полноценный мониторинг в реальном времени
Прогнозирование	отсутствует	отсутствует	на основе рекуррентной нейронной сети

Реализация портала

Структура веб-сайта представляет собой логическую систему организации его страниц и разделов, которая обеспечивает удобство навигации для пользователей. Она выполняет следующие ключевые функции:

- определяет расположение основных информационных блоков и сервисов;
- устанавливает взаимосвязи между различными разделами ресурса;
- формирует понятные маршруты перемещения по контенту и возврата к главным разделам.

Учитывая изложенное, особое внимание при проектировании уделялось: простоте навигационных цепочек, минимизации количества переходов для получения ключевой информации, визуальной понятности структурных элементов.

Главная страница портала представлена на рис. 1. Сайт планируется сделать общедоступным, без прохождения процедуры регистрации с отображением всей необходимой информации: температуры и качества воздуха. На карте, представленной на рис. 2, обозначены местоположения датчиков с возможностью посмотреть актуальную информацию по каждому из них.

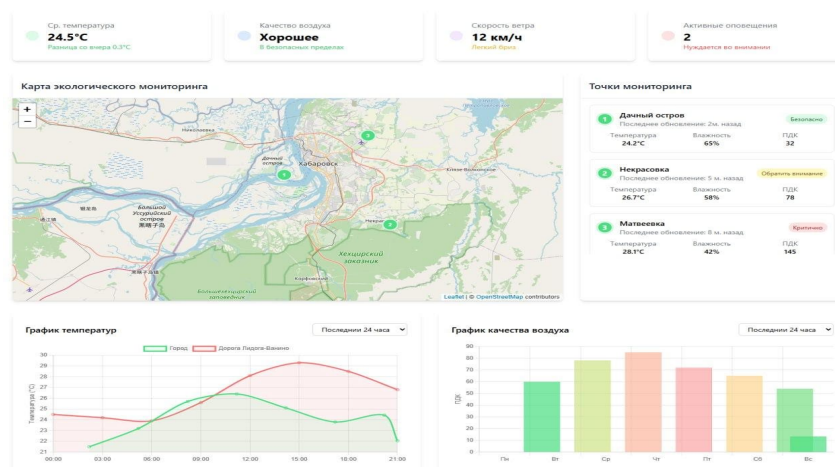


Рис. 1 - Главная страница сайта

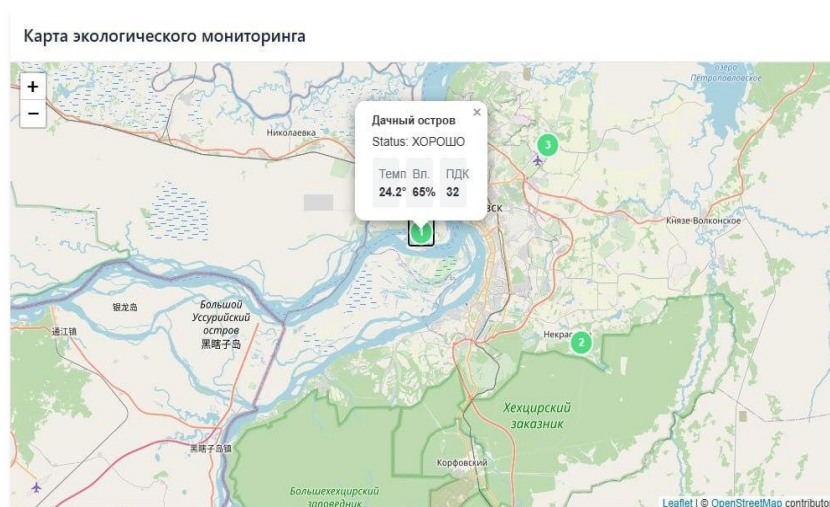


Рис. 2 - Карта с маркером

На странице “Прогнозирование состава воздуха”, представленной на рис. 3, отображаются графики с прогнозированием CO , CO_2 , N_2O и SO_2 , спрогнозированные с помощью рекуррентной нейронной сети, рассмотренной в работе [9].

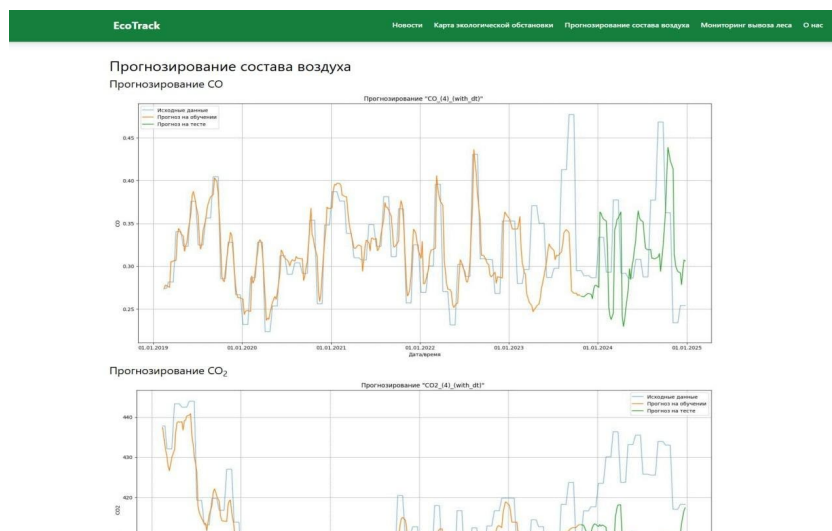


Рис. 3 - Графики с прогнозированием концентраций газов

Методология разработки

Для реализации функциональности веб-портала был выбран язык программирования Python и высокопроизводительный веб-фреймворк Django [10], соответствующий требованиям разработки масштабируемых,

безопасных и поддерживаемых веб-приложений. Архитектура фреймворка, основанная на паттерне MVC (Model–View–Controller), позволила эффективно разделить уровни представления данных, бизнес-логики и пользовательского интерфейса. В рамках проекта были задействованы механизмы аутентификации, авторизации и административного интерфейса, предоставляемые фреймворком, что способствовало ускорению разработки и обеспечению уровня безопасности.

В качестве системы управления базами данных использовалась PostgreSQL [11], поддерживающая сложные запросы, транзакции и обеспечивающая высокую надежность хранения данных. Обмен данными между приложением и системой управления базами данных осуществлялся через ORM-механизм, предоставляемый Django. Это позволило минимизировать использование «сырого» SQL, снизить риски инъекций и повысить читаемость кода. Для оптимизации производительности при выполнении сложных выборок применялись методы `select_related`, `prefetch_related`, а также кастомные запросы с использованием Raw SQL в случаях, требующих максимальной гибкости.

Фронтенд портала был реализован с использованием шаблонизатора Django Template в сочетании с HTML5, CSS3 и JavaScript, с применением адаптивного дизайна для корректного отображения контента на устройствах различных форм-факторов.

Реализация маршрутов и бизнес-логики была выполнена с использованием классовых представлений (Class-Based Views) и декораторов Django, что способствовало снижению дублирования кода и повышению его структурированности и поддерживаемости. Формы обрабатывались с помощью модулей Django Forms и ModelForms, включавших серверную валидацию полей и защиту от CSRF-атак (Cross-Site Request Forgery).

Серверная часть разворачивалась на WSGI-совместимом сервере Gunicorn, с использованием Nginx в роли обратного прокси-сервера, что обеспечило эффективное управление HTTP-запросами и защиту от DDoS-атак. Для унификации окружений разработки, тестирования и эксплуатации была реализована контейнеризация приложения с помощью Docker, включая оркестрацию контейнеров через Docker Compose.

Автоматизация этапов сборки, тестирования и развертывания (CI/CD) была организована с использованием GitLab CI, что позволило создать устойчивый и воспроизводимый процесс доставки кода в продакшн.

Для обработки долгоживущих и фоновых задач (включая работу с внешними API) использовалась асинхронная очередь задач Celery [12] с брокером сообщений Redis, что позволило избежать блокировок основного потока обработки HTTP-запросов.

Мониторинг состояния сервиса и централизованное логирование реализованы с помощью системы Sentry [13]. Для анализа метрик производительности и нагрузки на компоненты системы применялась связка Prometheus [14] и Grafana [15].

Выбранный стек технологий обеспечил высокую степень автоматизации, отказоустойчивость, простоту масштабирования и соответствие современным стандартам разработки веб-приложений.

Обработка данных

Датчики посредством сети интернет (мобильная связь, спутниковая связь и др.) присылают данные на сервер. Информация с датчиков сохраняется в базу данных, имеющую модель, представленную на рис. 4.

Далее, на основе данных, полученных с датчиков, пересчитывается индекс ПДК и актуализируется на сайте. Так же при этом полученные данные передаются рекуррентной нейронной сети и на основе новых данных

корректируется прогноз состава воздуха. Блок схема всех взаимодействий представлена на рис. 5.

```
class Sensor(models.Model):  
    name = models.CharField("Название датчика", max_length=100)  
    description = models.TextField("Описание", blank=True, null=True)  
    location = models.CharField("Местоположение", max_length=255, blank=True, null=True)  
    created_at = models.DateTimeField(auto_now_add=True)  
  
    class Meta:  
        verbose_name = "Датчик"  
        verbose_name_plural = "Датчики"  
  
class SensorType(models.Model):  
    name = models.CharField("Тип данных", max_length=50)  
    unit = models.CharField("Единица измерения", max_length=20)  
  
    class Meta:  
        verbose_name = "Тип датчика"  
        verbose_name_plural = "Типы датчиков"  
  
class SensorData(models.Model):  
    sensor = models.ForeignKey(Sensor, on_delete=models.CASCADE, related_name='data')  
    sensor_type = models.ForeignKey(SensorType, on_delete=models.CASCADE)  
    value = models.FloatField("Значение")  
    timestamp = models.DateTimeField("Время показания", default=timezone.now)  
  
    class Meta:  
        verbose_name = "Показание датчика"  
        verbose_name_plural = "Показания датчиков"  
        ordering = ['-timestamp']
```

Рис. 4 – Модель базы данных для работы с датчиками

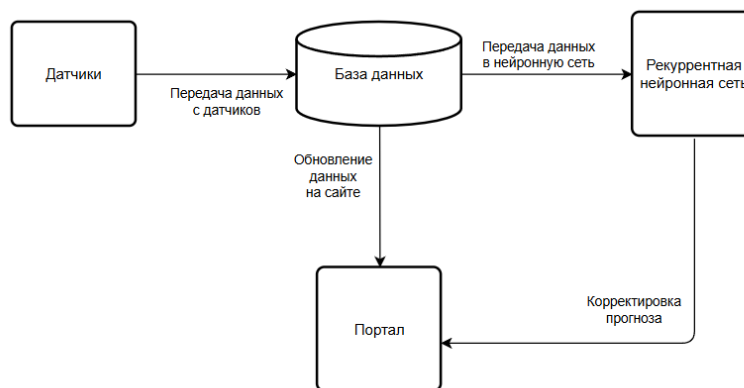


Рис. 5 – Блок схема взаимодействий

Заключение

В ходе проведённой работы был разработан веб-портал для мониторинга и прогнозирования состояния атмосферного воздуха в Хабаровском крае. Реализация проекта позволила устранить ключевые недостатки существующих аналогов, такие как отсутствие данных в реальном времени, ограниченная функциональность и устаревший интерфейс. Использование

современных технологий, включая Python/Django, PostgreSQL, машинное обучение для прогнозирования, а также интерактивные карты на базе OpenStreetMap, обеспечило высокую производительность, удобство использования и масштабируемость системы.

Портал предоставляет пользователям актуальную информацию о качестве воздуха, возможность прогнозирования концентраций вредных газов с помощью рекуррентной нейронной сети, а также интуитивно понятный интерфейс с адаптивным дизайном. Внедрение автоматизированных процессов CI/CD, мониторинга и логирования гарантирует стабильность и безопасность работы системы.

Результаты проекта демонстрируют значительный потенциал для дальнейшего развития, включая интеграцию дополнительных датчиков, расширение аналитических возможностей и применение более сложных моделей прогнозирования. Разработанный портал может служить основой для создания аналогичных решений в других регионах, способствуя улучшению экологического мониторинга и информированности населения.

Литература

1. Таненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е изд. СПб.: Питер, 2020. 992 с.
 2. Zhang Y., Bocquet M., Mallet V., Seigneur C., & Baklanov A. Real-time air quality forecasting, part I: History, techniques, and current status. // Atmospheric Environment. 2020. №60. pp. 632-655.
 3. Sebastian A., Gangoiti G., Albizua J. Deep learning for PM2.5 forecasting: A comparative study of recurrent and convolutional architectures // Environmental Modelling & Software. 2022. №149. pp. 105-318.
 4. Kumar P., Morawska L., Martani C., Biscos G., Neophytou M., Di Sabatino S., Bell M., Norford L., Britter R. The rise of low-cost sensing for managing air pollution in cities // Environment International. 2021. №75. pp. 246-264.
-

5. World Health Organization WHO global air quality guidelines: Particulate matter (PM_{2.5} and PM₁₀), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide. Bonn: WHO European Centre for Environment and Health, 2021. 290 p.
6. Django Web Framework (Python). URL: developer.mozilla.org/ru/docs/Learn/Server-side/Django
7. OpenStreetMap. URL: openstreetmap.org/.
8. Leaflet.js Documentation. URL: leafletjs.com/.
9. Вегера Д.В., Забавин А.С., Новикова А.А., Пархоменко И.С., Похвощев Е.О. Прогнозирование концентраций газов на основе нейросетевого моделирования // Инженерный вестник Дона. 2025. № 4. URL: ivdon.ru/ru/magazine/archive/n4y2025/10018
10. Django documentation URL: docs.djangoproject.com.
11. PostgreSQL: Documentation URL: postgresql.org/docs/
12. Celery Project URL: docs.celeryproject.org.
13. Sentry Documentation. URL: docs.sentry.io.
14. Prometheus URL: prometheus.io.
15. Grafana URL: grafana.com.

References

1. Tanenbaum E., Uezeroll D. Kompiuternye seti. [Computer Networks]. 5-e izd. SPb.: Piter, 2020. p. 992.
 2. Zhang Y., Bocquet M., Mallet V., Seigneur C., & Baklanov A. Atmospheric Environment. 2020. №60. pp. 632-655.
 3. Sebastian A., Gangoiti G., Albizua J. Environmental Modelling & Software. 2022. №149. pp. 105-318.
 4. Kumar P., Morawska L., Martani C., Biscos G., Neophytou M., Di Sabatino S., Bell M., Norford L., Britter R. Environment International. 2021. №75. pp. 246-264.
-

5. World Health Organization WHO global air quality guidelines: Particulate matter (PM_{2.5} and PM₁₀), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide. Bonn: WHO European Centre for Environment and Health, 2021. p. 290.
6. Django Web Framework (Python). URL: developer.mozilla.org/ru/docs/Learn/Server-side/Django
7. OpenStreetMap URL: openstreetmap.org/.
8. Leaflet.js Documentation URL: leafletjs.com/.
9. Vejera D.V., Zabavin A.S., Novikova A.A., Parkhomenko I.S., Pokhvashev E.O. Inzhenernyj vestnik Dona. 2025. № 4. URL: ivdon.ru/ru/magazine/archive/n4y2025/10018
10. Django documentation. URL: docs.djangoproject.com
11. PostgreSQL: Documentation. URL: postgresql.org/docs/
12. Celery Project. URL: docs.celeryproject.org
13. Sentry Documentation. URL: docs.sentry.io
14. Prometheus. URL: prometheus.io
15. Grafana. URL: grafana.com

Дата поступления: 5.06.2025

Дата публикации: 25.08.2025