Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение

высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Инженерная школа ядерных технологий
Направление подготовки 01.04.02 «Прикладная математика и информатика»
Отделение экспериментальной физики

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Разработка информационной системы мониторинга окружающей среды на примере г. Алматы, Казахстан

УДК 004.415.2:502.175

Студент

Группа	ФИО	Подпись	Дата
0ВМ12	Алмасбекулы Батухан		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОЭФ	Семенов М.Е.	к.ф.-м.н., доцент		

КОНСУЛЬТАНТЫ:

По разделу «Концепция стартап-проекта»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ШИП	Ковалёва Е.В.	к.м.н доцент		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Сечин А.А.	к.т.н.		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОЭФ	Мерзликин Б.С.	к.ф.-м.н., доцент		

Томск – 2023 г.

Планируемые результаты обучения по ООП

Код компетенции	Наименование компетенции
Профессиональные компетенции	
ПК(У)-1	Способен проводить научные исследования и получать новые научные и прикладные результаты самостоятельно и в составе научного коллектива
ПК(У)-2	Способен проводить поиск и анализ научной и научно-технической литературы по тематике проводимых исследований
ПК(У)-3	Способен разрабатывать и анализировать показатели качества информационных систем, используемых в производственной деятельности
ПК(У)-4	Способен планировать научно-исследовательскую деятельность, анализировать риски, управлять проектами, управлять командой проекта
ПК(У)-5	Способен к преподаванию математических дисциплин и информатики в образовательных организациях высшего образования
ПК(У)-6	Способен проектировать и организовывать учебный процесс по образовательным программам с использованием современных образовательных технологий
Общепрофессиональные компетенции	
ОПК(У)-1	Способен решать актуальные задачи фундаментальной и прикладной математики
ОПК(У)-2	Способен совершенствовать и реализовывать новые математические методы решения прикладных задач
ОПК(У)-3	Способен разрабатывать математические модели и проводить их анализ при решении задач в области профессиональной деятельности
ОПК(У)-4	Способен комбинировать и адаптировать существующие информационно-коммуникационные технологии для решения задач в области профессиональной деятельности с учетом требований информационной безопасности
Универсальные компетенции	
УК(У)-1	Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий
УК(У)-2	Способен управлять проектом на всех этапах его жизненного цикла
УК(У)-3	Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной цели
УК(У)-4	Способен применять современные коммуникативные технологии, в том числе на иностранном языке, для академического и профессионального взаимодействия
УК(У)-5	Способен анализировать и учитывать разнообразие культур в процессе межкультурного взаимодействия
УК(У)-6	Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки

Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение

высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Инженерная школа ядерных технологий
Направление подготовки 01.04.02 «Прикладная математика и информатика»
Отделение экспериментальной физики

УТВЕРЖДАЮ:

Руководитель ООП

_____ _____ Мерзликин Б.С.

(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Магистерской диссертации

Студенту:

Группа	ФИО
0ВМ12	Алмасбекулы Батухан

Тема работы:

Разработка информационной системы мониторинга окружающей среды на примере г. Алматы, Казахстан	
Утверждена приказом директора (дата, номер)	27.04.2023, №117-40/с

Срок сдачи студентом выполненной работы:	31.05.2023

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Временной ряд концентраций примесей в атмосфере г. Алматы, Казахстан (PM2.5,PM10,NO_2,CO): PM2.5: 12.07.2020-19.05.2023 PM10: 04.12.2021-19.05.2023 NO_2: 04.12.2021-19.05.2023 CO: 04.12.2021-19.05.2023

Перечень подлежащих исследованию, проектированию и разработке вопросов	1. Разработка инфологической модели предметной области, выбор сущностей и связей. 2. Выбор стека технологий для разработки информационной системы (СУБД, язык программирования, библиотеки для анализа и визуализации, разработки пользовательского веб-интерфейса). 3. Анализ исходных данных и выбор математической модели для анализа и прогнозирования. 4. Статистическая обработка полученных результатов прогнозирования. 5. Разработка технической документации по эксплуатации информационной системы.
Перечень графического материала	1. Sequantial diagramm информационной системы на языке UML. 2. Визуализация результатов прогнозирования (временной ряд, доверительный интервал, Q-Q диаграмма, распределение остатков модели).

Консультанты по разделам выпускной квалификационной работы

(если необходимо, с указанием разделов)

Раздел	Консультант
Концепция стартап-проекта	Ковалёва Е.В
Социальная ответственность	Сечин А.А.
Иностранный язык	Смирнова У.А

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	15.03.2023 г.

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОЭФ	Семенов Михаил Евгеньевич	к. ф.-м. н., доцент		15.03.2023 г.

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
0ВМ12	Алмасбекулы Батухан		15.03.2023 г.

ЗАДАНИЕ К РАЗДЕЛУ

«КОНЦЕПЦИЯ СТАРТАП-ПРОЕКТА»

Студенту**:**

Группы	ФИО
0ВМ12	Алмасбекулы Батухан

Школа	ИЯТШ	Отделение школы	Отделение экспериментальной физики
Уровень образования	магистратура	Направление / специальность	01.04.02 Прикладная математика и информатика

Перечень вопросов, подлежащих разработке:	
Проблема конечного потребителя, которую решает продукт, который создается в результате выполнения НИОКР (функциональное назначение, основные потребительские качества)	Удобная система мониторинга качества воздуха, в режиме реального времени, который позволит отслеживать уровни загрязнения во всех точках города Алматы и поддерживать принятие решений на основе данных
Объем и емкость рынка	B2B–15116640 KZT в месяц B2G – 809 731 000 KZT в год
Способы защиты интеллектуальной собственности	патент полезной модели
Современное состояние и перспективы отрасли, к которой принадлежит представленный в ВКР продукт	проблема загрязнения атмосферного воздуха в Алматы
Себестоимость продукта	общий бюджет на разработку платформы и установку системы – 12 508 530 KZT.
Конкурентные преимущества создаваемого продукта, сравнение технико-экономических характеристик продукта с отечественными и мировыми аналогами	обновления в режиме реального времени, расширенный диапазон и использование, дополнительных услуги, простой и понятный интерфейс, локальность
Целевые сегменты потребителей создаваемого продукта	Государственный органы и регуляторы, Научные и исследовательские организации, Бизнес-сектор
Бизнес-модель проекта, производственный план и план продаж	модель по А. Остервальдеру и И. Пинье(Шаблон бизнес-модели)
Стратегия продвижения продукта на рынок	брендинг, SMM (Facebook, Twitter, Instagram и LinkedIn), веб-сайт, контент-маркетинг, связи с общественностью (журналисты и СМИ), ивент-маркентинг, создание группы в Вконтакте, партнерские отношения и сотрудничество, получение наград и признание, разработка образовательных программ

Перечень графического материала:	
При необходимости представить эскизные графические материалы	Модель по А. Остервальдеру и И. Пинье, таблицы расчета бюджета проекта

Дата выдачи задания к разделу в соответствии с календарным учебным графиком	

Задание выдал консультант по разделу «Концепция стартап-проекта»

(со-руководитель ВКР):

Должность	**ФИО**	**Ученая степень, звание**	**Подпись**	**Дата**
Доцент	Ковалёва Е.В	к.м.н		

Задание принял к исполнению обучающийся:

Группа	**ФИО**	**Подпись**	**Дата**
0ВМ12	Алмасбекулы Батухан		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа		ФИО	
0ВМ12		Алмасбекулы Батухан	
Школа	**ИЯТШ**	**Отделение (НОЦ)**	**Отделение экспериментальной физики**
Уровень образования	магистратура	**Направление/ специальность**	01.04.02 Прикладная математика и информатика

Тема ВКР:

Разработка информационной системы мониторинга окружающей среды на примере г. Алматы, Казахстан	
Исходные данные к разделу «Социальная ответственность»:	
Введение – Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения. – Описание рабочей зоны (рабочего места) при разработке проектного решения/при эксплуатации	*Объект исследования:* разработка алгоритма *Область применения: принятие решения студентом* *Рабочая зона:* офис *Размеры помещения 15 м х 5 м х 4 м* *Количество и наименование оборудования рабочей зоны:* компьютеры *Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне, разработка алгоритма на компьютере*
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения: – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны.	- Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 27.12.2018) - ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования.
2. Производственная безопасность при разработке проектного решения: – Анализ выявленных вредных и опасных производственных факторов – Расчет уровня опасного или вредного производственного фактора	**Вредные факторы:** - Отклонение показателей микроклимата; - Отсутствие или недостатки необходимого искусственного освещения **Опасные факторы:** - Повышенный уровень электромагнитных излучений; - Опасные и вредные производственные факторы, связанные с электрическим током
3. Экологическая безопасность при разработке проектного решения	Воздействие на литосферу: образования отходов при написании работы Воздействие на гидросферу: энерго и теплопотребление Воздействие на атмосферу: энерго и теплопотребление
4. Безопасность в чрезвычайных ситуациях при разработке проектного решения/	Возможные ЧС аварии, пожары Наиболее типичная ЧС пожар
Дата выдачи задания для раздела по линейному графику	**01.03.2023**

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Сечин Андрей Александрович	к.т.н		01.03.2023

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
0ВМ12	Алмасбекулы Батухан		01.03.2023

Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение

высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Инженерная школа ядерных технологий
Направление подготовки 01.04.02 «Прикладная математика и информатика»
Отделение экспериментальной физики
Период выполнения весенний семестр 2022/2023 учебного года

Форма представления работы:

Магистерская диссертация

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
01.03.2023	Выдача задания	5
14.03.2023	Обсуждение структуры работы	5
01.04.2023	Обзор литературы	10
15.04.2023	Создание модели ARIMA	15
01.05.2023	Проверка на стационарность, тест Дики-Фуллера	30
25.05.2023	Анализ полученных результатов	10
31.05.2023	Написание пояснительной записки ВКР	10

СОСТАВИЛ:
Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОЭФ	**Семенов М.Е.**	**к.ф.-м.н.**		

СОГЛАСОВАНО:
Руководитель ООП

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОЭФ	**Мерзликин Б.С.**	**к.ф.-м.н., доцент**		

РЕФЕРАТ

Выпускная квалификационная работа 142 с., 18 рис., 20 табл., 22 источников, 2 прил.

Ключевые слова: модель ARIMA, экологический мониторинг, стационарность, прогнозирование, моделирование, индекс качества воздуха AQI.

Объектом исследования является экологическое состояние воздушной среды города Алматы, выбросы и изменчивость концентраций загрязняющих веществ.

Цель работы – разработка информационной системы экологического мониторинга, сфокусированной на качестве воздуха в г. Алматы, Казахстан. Эта система будет использовать анализ временных рядов, в частности модель ARIMA, для прогнозирования будущих параметров качества воздуха на основе исторических данных. Такая способность прогнозирования позволит получить ценную информацию о состоянии качества воздуха в Алматы и принять упреждающие меры по снижению рисков загрязнения воздуха.

В процессе исследования использовались данные о загрязняющих веществах, важных для оценки качества воздуха. Применение модели ARIMA для анализа временных рядов позволило оценить и предсказать будущие значения показателей качества воздуха.

Результат исследование подчеркивает важность настройки модели ARIMA для прогнозирования качества воздуха. Улучшение точности прогнозов снижает среднеквадратичную ошибку (RMSE) и помогает принимать эффективные меры по снижению загрязнения воздуха в г. Алматы.

Степень внедрения: реальна. В интересах обеспечения экологически безопасного проживания населения.

Определения, обозначения, сокращения и нормативные ссылки

В данной работе приведены следующие термины с соответствующими определениями:

ARIMA – авторегрессионная интегрированная скользящая средняя

ADF-тест – тест Дики-Фуллера

AQI – индекс качества воздуха

RMSE – Среднеквадратичная ошибка

API – описание способов взаимодействия одной компьютерной программы с другими.

Модель прогнозирования — модель объекта прогнозирования, что исследование позволяет получить информацию о возможных состояниях объектов прогнозирования в будущем и (или) путях и сроках их осуществление.

Временный ряд — собранный в разные моменты времени статистический материал о значении каких-либо параметров (в случае одного) уникального процесса.

Оглавление

ВВЕДЕНИЕ

Прогнозирование качества воздуха является активной областью исследований в последние годы остро стоит проблема ухудшения качества воздуха в больших городах. Это связано со стремительным увеличением автотранспорта в городской черте, неудовлетворительной работой фильтров топливно-энергетического комплекса, увеличением в отопительный сезон выбросов частного сектора. Одним из показателей загрязненности атмосферного воздуха является индекс качества воздуха (AQI). Анализ временных рядов, в частности, широко используется благодаря своей эффективности в отражении временных зависимостей экологических данных.

В настоящее время различными организациями, занимающимися контролем и оценкой качеством воздуха активно накапливаются исторические данные метеорологических показателях, включая информацию о загрязнении атмосферы. Это приводит к значительному увеличению объема данных, доступных для прогнозирования. Вместе с тем, развитие программных и аппаратных средств дает возможность реализации сложных алгоритмов прогнозирования. Это позволит улучшить точность прогнозов и удовлетворить все более требования к прогнозированию погоды и качеству воздуха.

В ряде исследований использовалась ARIMA. Например, Кумар и Джайн использовали модель ARIMA для прогнозирования уровней PM2,5 в Дели, Индия.[1] Точно так же Муника и Шастри успешно применили ARIMA для прогнозирования загрязнения воздуха в Вишакхапатнаме, Индия.[2] Другие исследователи расширили применение традиционных моделей ARIMA. Пирес и другие представили процедуру декомпозиции сезонного тренда на основе Loess (STL) и ARIMA для прогнозирования PM10 в Португалии[3]. Между тем, Астита М. и другие объединили ARIMA с другими методами машинного обучения для повышения точности прогнозирования качества воздуха в Хартфорде, США.[4] Приведенные выше исследования

демонстрируют потенциал анализа временных рядов и, в частности, ARIMA для прогнозирования качества воздуха. Данный проект направлен на применение этих методов для прогнозирования параметров качества воздуха в Алматы, Казахстан, внося свой вклад в глобальные усилия по мониторингу и управлению окружающей средой. В связи с этим возникает необходимость разработки эффективных методов и инструментов для мониторинга и прогнозирования качества воздуха.

Приведенные выше исследования демонстрируют потенциал анализа временных рядов и, в частности, ARIMA для прогнозирования качества воздуха. Данный проект направлен на применение этих методов для прогнозирования параметров качества воздуха в Алматы, Казахстан, внося свой вклад в глобальные усилия по мониторингу и управлению окружающей средой.

Целью настоящей магистерской диссертации является разработка информационной системы экологического мониторинга, использующей модель ARIMA для прогнозирования качества воздуха на основе исторических данных. Это позволит предсказывать будущие параметры качества воздуха и принимать меры по снижению рисков загрязнения воздуха. Для достижения данной цели были поставлены следующие задачи:

- Собрать и подготовить данные о концентрациях примесей в атмосфере г. Алматы, Казахстан.
- Построить модель ARIMA для каждого из загрязняющих веществ с использованием обучающих данных.
- Оценить качество модели ARIMA с помощью среднеквадратической ошибки (RMSE).
- Визуализировать фактические и прогнозируемые значения для каждого загрязняющего вещества.
- Прогнозировать будущие значения концентраций примесей на основе модели ARIMA.

Научная и практическая новизна и значимость работы:

В работе рассматривается построениемодели ARIMA для прогнозирования качества воздуха. Разработанная информационная система экологического мониторинга объединяет исторические данные о качестве воздуха и предсказывает будущие значения на основе модели. Это позволяет оперативно определять состояние воздуха и принимать меры по снижению рисков для здоровья людей.

Обзор литературы

1.1 Анализ общего состояния воздушного бассейна Республики Казахстан по регионам

Загрязнение атмосферного воздуха – привнесение в него или возникновение в нём новых (обычно не характерных для него) вредных химических, физических, биологических агентов. Оно может быть естественным (природным) и антропогенным (техногенным). *«Естественное загрязнение* воздуха вызвано природными процессами (вулканическая деятельность, ветровая эрозия, массовое цветение растений, дым от лесных и степных пожаров и др.)» *Антропогенное загрязнение* связано с выбросом загрязняющих веществ в результате деятельности человека. По агрегатному состоянию выбросы вредных веществ в атмосферу классифицируются следующим образом: 1) газообразные (диоксид серы, оксиды азота, оксид углерода, углеводороды и др.); 2) жидкие (кислоты, щелочи, растворы солей и др.); 3) твёрдые (тяжёлые металлы, канцерогенные вещества, органическая и неорганическая пыль, сажа, смолистые вещества и др.). Если же рассмотреть экологическую ситуацию Казахстана, то уровень загрязнения атмосферы городов и промышленных центров, несмотря на сокращение производства, остается достаточно высоким. Наибольший уровень загрязнения воздуха наблюдается в Лениногорск, Усть-Каменогорск, Актюбинск, Алматы, Зыряновске, Актау, Шымкент, Тараз, Петропавловск и Темиртау [5].

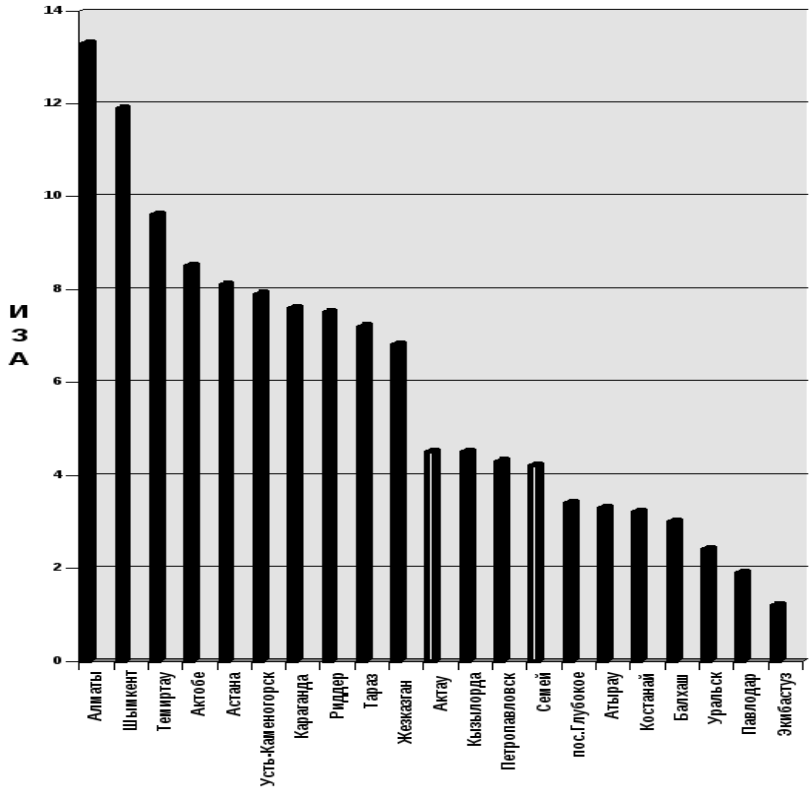


Рисунок 1.1 – ИЗА городов Казахстана

В настоящее время в плане анализа и оценки экологических и техногенных опасностей исключительная роль отводится системе экологического мониторинга. В этой области для прогнозирования развития экологически опасных ситуаций недостаточно старой практики, основанной на наблюдении, накоплении данных и составлении бюллетеней загрязнения окружающей среды. Необходим оперативный экологический контроль, а это, значит, требуется новая стратегия и новые методы, которые позволят концентрировать внимание на ближайших и будущих тенденциях и первостепенных задачах. В работе А.К. Муртазов [6] дано определение, *экологический мониторинг* - информационная система наблюдений, оценки и прогноза изменений в состоянии окружающей среды, созданная с целью выделения антропогенной составляющей этих изменений на фоне природных процессов. Говоря о системе экологического мониторинга, мы подразумеваем, что она должна накапливать, систематизировать и анализировать информацию: о состоянии окружающей среды; о причинах наблюдаемых и

вероятных изменений состояния (т. е. об источниках и факторах воздействия); о допустимости изменений и нагрузок на среду в целом. В соответствии с приведенными определениями и возложенными на систему функциями, можно выделить три основных направления деятельности, которые включает в себя экологический мониторинг:

– наблюдения за факторами воздействия и состоянием среды;

– оценку фактического состояния среды;

– прогноз состояния окружающей природной среды и оценку прогнозируемого состояния.

Необходимо также отметить, что сама система мониторинга не включает деятельность по управлению качеством среды, но является источником информации необходимой для принятия экологически значимых решений. Системный метод экологического мониторинга основывается на экспертизе экологического воздействия вредных выбросов на окружающую среду и обеспечивает комплексный учет измерений и сопоставления их со стандартными показателями, выраженными через качественные и количественные характеристики экологической безопасности. Метод оценки экологических воздействий при разработке системы содержит комплекс мероприятий, включающих идентификацию, анализ, слежение и мониторинг экологических рисков от их запланированных значений. При этом информационная модель экологических воздействий вредных веществ включает в себя идентификацию: явных и неявных опасностей для человека, природы и общества в целом. Использование современных информационных технологий в данной области позволяет разработать автоматизированную систему экологического мониторинга окружающей среды для обеспечения экологической безопасности различного рода объектов и людей. Экологический мониторинг, подразумеваемый в данной работе, представляет собой систему стационарных пунктов наблюдения за состоянием загрязнения атмосферы на территории г. Алматы, в частности за распространением

загрязняющих веществ от стационарных источников загрязнения, т.е. от промышленных предприятий. Как известно, доля загрязнения атмосферы промышленными предприятиями достаточно велика, они являются одними из основных загрязнителей, а соответственно оказывают непосредственное влияние на общее состояние приземного слоя атмосферного воздуха, на состояние здоровья жителей того или иного населенного пункта.

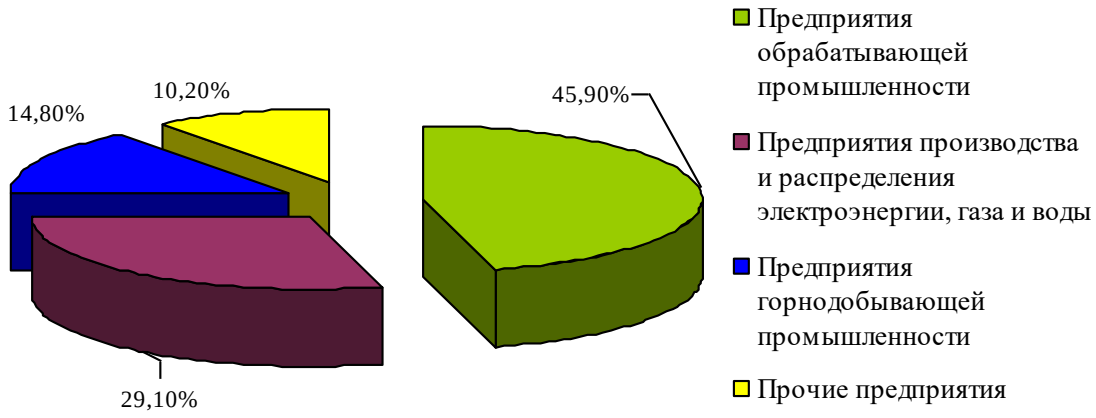


Рисунок 1.2 – Удельный вес различных промышленных предприятий в загрязнение атмосферы

2. Объекты и методы исследования

2.1 Объект исследования

Основное внимание в этом исследовании уделяется проанализированным данным, полученным из aqicn.org , платформа, которая предоставляет подробную информацию об атмосферных условиях в городе Алматы. Набор данных содержит среднесуточные измерения загрязняющих веществ PM2.5, PM10, NO2 и CO. Вышеупомянутые загрязняющие вещества являются основными маркерами загрязнения атмосферы и потенциально могут привести к значительным последствиям для здоровья населения. Целью данного исследования является анализ структуры и тенденций загрязнения с целью получения всестороннего представления об условиях качества воздуха в городе Алматы. Конечная цель состоит в том, чтобы помочь в разработке эффективных стратегий борьбы с загрязнением воздуха и охраны здоровья населения. В работе были использованы сочетание исследовательского анализа данных, анализа временных рядов, подходов к прогнозированию и оценки эффективности с использованием RMSE (среднеквадратичная ошибка) в качестве основного показателя для достижения целей исследования.

Метод сбора данных

Данные о качестве воздуха, использованные в данном исследовании, были получены с сайта aqicn.org. Aqicn.org предоставляет актуальную информацию о качестве воздуха в режиме реального времени для многих мест по всему миру, включая Алматы, Казахстан, как показано на рисунке 1. Данные включают измерения различных загрязняющих веществ, которые являются важнейшими показателями общего качества воздуха.

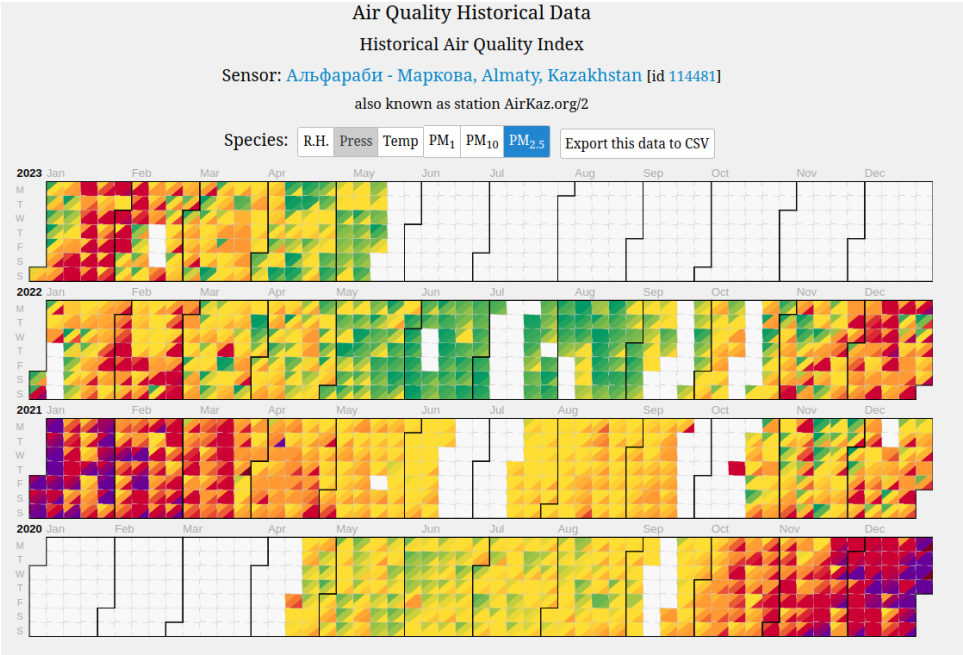


Рисунок 1. Один из датчиков, расположенных в г. Алматы

Для автоматизации процесса получения данных был использован Selenium, мощный инструмент для управления веб-браузерами с помощью программ и автоматизации задач браузера. Selenium использовался для навигации по веб-сайту, выбора соответствующих параметров и временных рамок и, наконец, для загрузки данных о качестве воздуха. Использование Selenium для веб-скрейпинга обеспечивает эффективный и систематический сбор данных. После загрузки данные были импортированы в pandas DataFrame для дальнейшей обработки и анализа. Полученный DataFrame включает колонки для даты наблюдения и показателей четырех загрязняющих веществ (PM2.5, PM10, NO2 и CO).

	date	pm25	pm10	no2	co
0	2023-05-01	39.0	26.0	17.0	4.0
1	2023-05-02	49.0	23.0	17.0	2.0
2	2023-05-03	46.0	12.0	19.0	3.0
3	2023-05-04	35.0	10.0	23.0	3.0
4	2023-05-05	41.0	15.0	22.0	3.0

Рисунок 2. Таблица собранных данных

2.2. Предварительный анализ данных (EDA)

На ранней стадии этого исследования была проведена предварительный анализ данных, чтобы иметь полное представление о наборе данных о качестве воздуха. Были рассмотрены распределение данных, сводную статистику и визуализацию загрязняющих веществ PM2.5, PM10, NO2 и CO. Благодаря этому исследованию мы смогли выявить любые недостающие цифры, выбросы или расхождения в данных, которые требовали предварительной обработки или дополнительных исследований. Чтобы обобщить основные тенденции и изменчивость данных о загрязнителях, мы использовали статистические инструменты, включая среднее значение, медиану, стандартное отклонение и прямоугольные графики. Кроме того, гистограммы, линейные графики и точечные диаграммы помогли выявить тенденции, закономерности и возможные связи между загрязнителями.

Корреляционный анализ

Линейный корреляционный анализ позволяет установить прямые связи между переменными величинами по их абсолютным значениям. Формула расчета коэффициента корреляции построена таким образом, что если связь между признаками имеет линейный характер, коэффициент Пирсона точно устанавливает тесноту этой связи. Поэтому он называется также коэффициентом линейной корреляции Пирсона.

В общем виде формула для подсчета коэффициента корреляции такова:

$$r_{xy} = \frac{\sum(x_i - M_x)(y_i - M_y)}{\sqrt{\sum(x_i - M_x)^2(y_i - M_y)^2}}$$

где:

x_i - значения, принимаемые переменной X,

y_i - значения, принимаемые переменой Y,

M_x - средняя по X,

M_y - средняя по Y

Прежде чем перейти к модели ARIMA, был проведен анализ корреляций между переменными. Приведенная ниже корреляционная матрица отображает коэффициенты корреляции Пирсона [7], которые варьируются от -1 до 1. Коэффициент, близкий к 1, указывает на сильную положительную корреляцию, в то время как коэффициент, близкий к -1, указывает на сильную отрицательную корреляцию. Эта таблица, представленная на рисунке 3, дает ценную информацию о взаимосвязи между четырьмя загрязнителями.

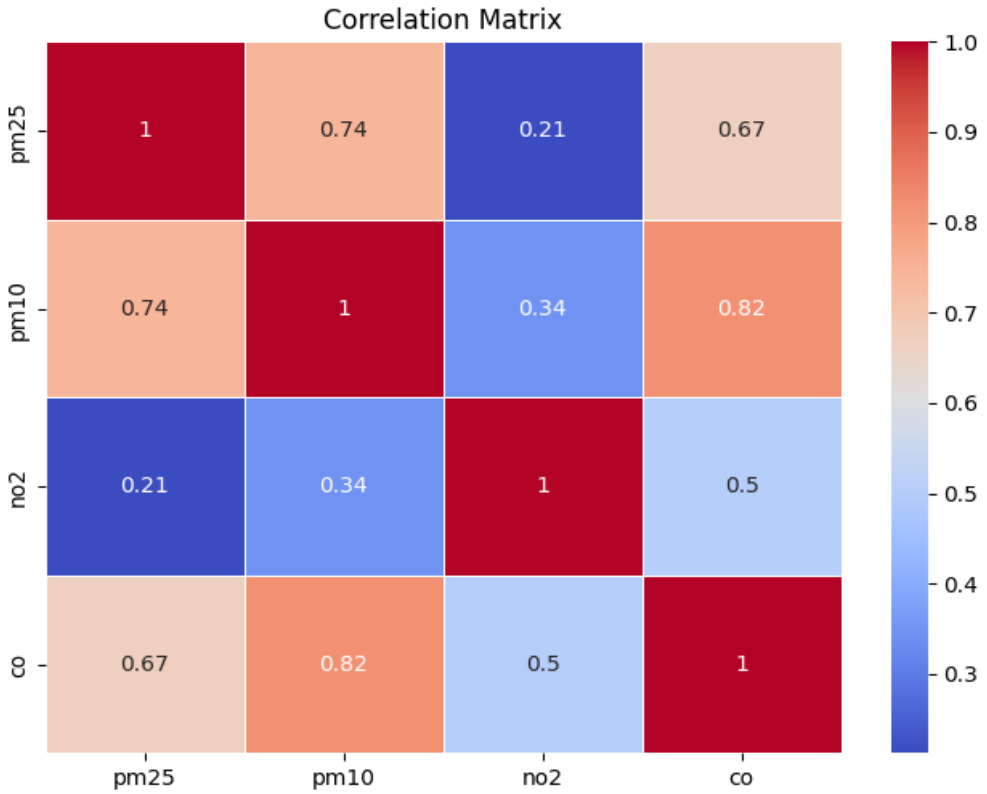


Рисунок 3. Тепловая карта между целевыми значениями

2.3. Анализ временных рядов

Использованы подходы к анализу временных рядов для изучения основных закономерностей, тенденций и зависимостей загрязняющих веществ PM2.5, PM10, NO2 и CO из-за временного характера данных о качестве воздуха. Чтобы понять взаимозависимость и эффекты запаздывания внутри временных рядов, мы исследовали функции автокорреляции и частичной автокорреляции. Кроме того, мы использовали статистический анализ и визуальный осмотр графиков временных рядов, чтобы оценить стационарность данных. Чтобы установить стационарность, мы использовали соответствующие преобразования или разграничения по мере необходимости. Методы декомпозиции были использованы при анализе сезонности для определения сезонных элементов и закономерностей в данных. Этот этап облегчил выбор моделей, которые должным образом отражали временную динамику загрязняющих веществ.

2.4. Модели для прогнозирования

ARIMA-модели (AutoRegressive Integrated Moving Average model), или интегрированные модели авторегрессии-скользящего среднего, впервые предложенные Боксом и Дженкинсом [8], наиболее популярные и эффективные статистические модели для прогноза временных рядов. В их основе лежит фундаментальный принцип, что будущие значения временного ряда генерируются некоторой линейной функцией прошлых наблюдений и случайной ошибкой (белым шумом).В силу того, что инерционность во временных рядах загрязнений велика и, несмотря на появление искусственных нейронных сетей, ARIMA-модели широко используют для прогноза загрязнений атмосферного воздуха. Так, например, в [9] ARIMA-техника применена для краткосрочного (максимум на 24 часа) прогноза концентрации CO в атмосферном воздухе, в [10] предложены сезонные ARIMA-модели для прогноза индекса качества воздуха, в [11] ARIMA-техника применена для прогноза загрязнения воздуха NO2, PM 2.5, PM10 и CO.

В данной работе мы использовали сложные методы прогнозирования, особенно модели ARIMA (Авторегрессионное интегрированное скользящее среднее), для прогнозирования будущих уровней загрязнения в городе Алматы. Модель ARIMA включала компоненты авторегрессии и скользящего среднего. Прогнозы были составлены с использованием моделей после того, как они были обучены с использованием предыдущих данных о загрязнении. Мы смогли предвидеть уровни загрязняющих веществ в желаемые будущие промежутки времени благодаря методу прогнозирования, который включал повторные вычисления на основе ранее наблюдаемых значений и параметров модели.

В общем виде ARIMA(p,d,q) модель имеет форму

$$\phi(B)(1-B)^d y_t = \theta(B)\varepsilon t \ ,$$

где:

yt – значения модельного ряда в момент времени t;

εt – случайная ошибка с нулевым средним и постоянной дисперсией;

$\phi(B)$, $\theta(B)$ – полиномы степени p и q, B – лаговый оператор

2.5. Показатели для оценки модели и производительности

Эффективность наших моделей прогнозирования оценивалась с использованием статистики RMSE (среднеквадратичная ошибка). Средний размер расхождений между ожидаемым и фактическим уровнями загрязнения был оценен RMSE. На лучшую производительность модели при правильном учете изменчивости данных указывали более низкие значения RMSE. Сопоставив прогнозируемые значения с фактическими наблюдениями из набора тестовых данных, который был отделен от обучающих данных, мы смогли проверить модели. Благодаря методу оценки мы смогли оценить точность и надежность моделей прогнозирования, а также их применимость для прогнозирования будущих уровней загрязнения.

3. Расчеты и аналитика

В этом разделе описываются расчеты и анализы, проведенные на основе данных о качестве воздуха, полученных из aqicn.org для города Алматы. Аналитический процесс состоит из нескольких этапов, а именно предварительной обработки данных, оценки параметров модели, прогнозирования и оценки производительности.

3.1 Предварительная обработка

Прежде чем приступить к расчетам и аналитике, мы выполнили необходимые процедуры предварительной обработки данных о качестве воздуха, полученных из aqicn.org. Задача включала в себя управление отсутствующими значениями, экстремальными значениями и несоответствиями в данных, которые были характерны для проанализированного набора данных. В исследовании использовались различные методы, включая вменение данных, обнаружение выбросов и очистку данных, для повышения качества и надежности данных. Более того, данные были преобразованы в соответствующую структуру для облегчения последующих вычислений и проверки.

Чтобы гарантировать стационарность данных временных рядов, мы выполнили необходимые процедуры предварительной обработки. Используемая методология включала использование метода дифференцирования по ряду загрязняющих веществ PM2,5, PM10, NO2 и CO. Использование дифференцирования в качестве методологии направлено на устранение фундаментальной тенденции в данных рассматривается в этой и последующей главе, тем. С помощью вычисления начальной производной ряда мы определили изменения, которые происходят между последовательными наблюдениями. Расчеты продемонстрированы на рисунке 4.

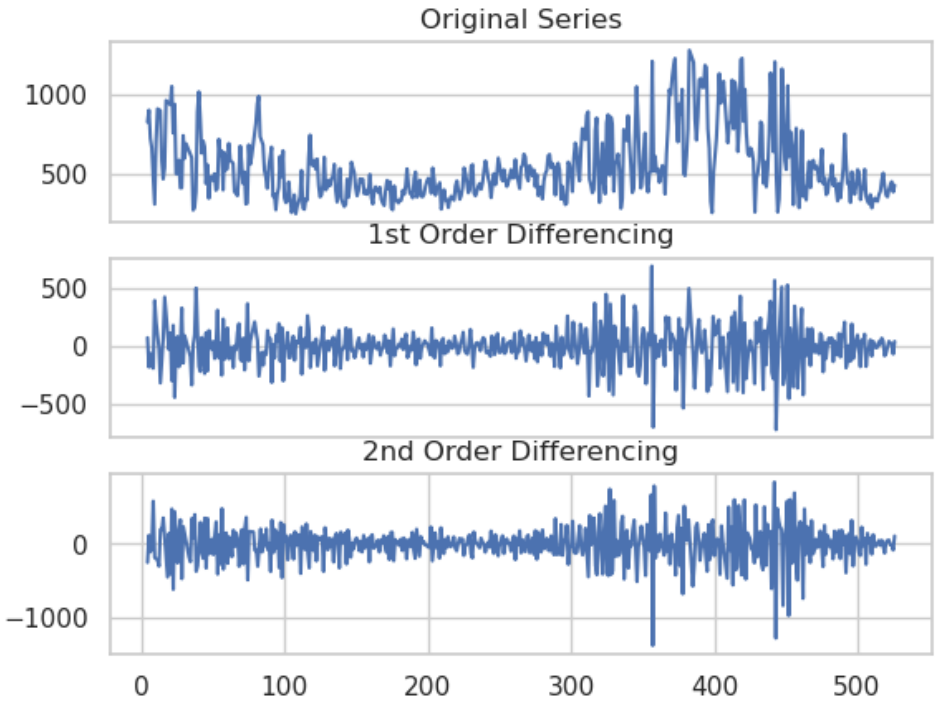


Рисунок 4. CO показатель после дифференцирование и исходные данные

Использование дифференцированных рядов помогает снизит не стационарность, обусловленную тенденциями или сезонностью, тем самым обеспечивая более концентрированный анализ временной динамики загрязняющих веществ.

Решение о том, применять или не применять дифференцирование к данным при анализе временных рядов, зависит от множества факторов, таких как атрибуты набора данных и проводимый исследовательский запрос в академических исследованиях. Если в наборе данных отсутствуют длительные исторические записи или адекватные наблюдения, разграничение может оказаться ненужным или неподходящим. Давайте рассмотрим этот вопрос с академической точки зрения.

Исследователи часто пытаются понять фундаментальные закономерности, тенденции и ассоциации, присущие данным временных рядов. Использование дифференцирования является распространенной практикой для устранения протяженных закономерностей и сезонных колебаний, тем самым облегчая выявление краткосрочных колебаний или

аномалий в данных. В случаях, когда анализируемый набор данных содержит ограниченное количество исторических наблюдений или короткие временные рамки, существование длительных тенденций может не быть серьезной проблемой.

Основной упор в исследованиях часто делается на тщательном изучении и интерпретации доступных данных для получения важных выводов и различений. В некоторых случаях основная цель может заключаться в тщательном изучении конфигураций и корреляций, присутствующих в собранных данных, в их неизменном состоянии, без попыток искоренить устойчивые тенденции или сезонные колебания. Изучая данные в их неизмененном виде, ученые могут получить более глубокое представление о врожденных особенностях и механизмах, которые существуют в течение ограниченного периода времени.

Кроме того, учет различий между точками данных потенциально может привести к сокращению объема информации или изменению базовых закономерностей, существующих в наборе данных. Когда исторические данные кратки, количество доступных данных ограничено, и использование различий для устранения долгосрочных закономерностей может привести к потере важных перспектив или связей, которые могли бы быть ценными для научного изучения. Исходя того, что наши данные имеют ограниченные записи показателей мы можем убрать дифференцирование для более точных прогнозов.

В ходе этого исследования были проанализированы серии загрязняющих веществ PM2.5, PM10, NO2 и CO. Используемая методология предполагала вычисление начальной производной, которая обозначает разницу между каждой точкой данных и ее предыдущим измерением. Процесс дифференцирования был использован для демонстрации изменении данных. В данном разделе мы изучили набор данных при производной, мы проверили

стационарность в следующей главе для окончательного применения дифференцирования.

3.2 Проверка на стационарность

Для оценки стационарности разностных временных рядов был использован расширенный тест Дики-Фуллера (ADF). Расширенный тест Дики-Фуллера (ADF) является распространенным статистическим тестом, используемым для установления существования единичного корня, который указывает на нестационарность данных [12]. Оценка включает в себя вычисление авторегрессионной модели для разностных рядов с последующей оценкой статистической значимости оцениваемых параметров.

Статистика расширенного теста Дики-Фуллера (ADF) может быть выражена как:

$$t = (\hat{\alpha} - 1) / SE(\hat{\alpha})$$

где $\hat{\alpha}$ представляет оценку коэффициента, полученную с помощью метода наименьших квадратов, а $SE(\hat{\alpha})$ обозначает стандартную ошибку оценки коэффициента методом наименьших квадратов, полученной из регрессионная модель.

Значение p, связанное со статистикой расширенного теста Дики-Фуллера (ADF), вычисляется с использованием оценочных значений p Маккиннона. При рассмотрении нулевой гипотезы можно заметить, что асимптотическое распределение тестовой статистики не соответствует стандартному распределению. Маккиннон использует аппроксимации поверхности отклика на моделируемых данных, чтобы получить

приблизительное значение p для любого заданного значения статистики теста ADF.

Критические значения, относящиеся к выполненному тесту Дики-Фуллера (ADF), были установлены Маккинноном. Автор представил исчерпывающую формулу для определения критического значения для трех различных уровней значимости, а именно 0,01, 0,05 и 0,1.

Тест ADF был использован для каждой серии загрязняющих веществ (PM2.5, PM10, NO2, CO) в отдельности, что привело к следующим результатам, рассматривается в таблице №1

Таблица №1:

Variable	ADF Statistic	ADF p-value	1% Critical Value	5% Critical Value	10% Critical Value	Stationarity Test Result
PM10	-3.2925	0.0152	-3.4434	-2.8673	-2.5698	Stationary
NO2	-3.6713	0.0045	-3.4434	-2.8673	-2.5698	Stationary
CO	-2.6885	0.0760	-3.4436	-2.8674	-2.5699	Non-Stationary
PM2.5	-2.5585	0.1019	-3.4369	-2.8644	-2.5683	Non-Stationary
PM10_diff	-7.9865	0.0000	-3.4437	-2.8674	-2.5699	Stationary
NO2_diff	-13.3660	0.0000	-3.4434	-2.8673	-2.5698	Stationary
CO_diff	-9.6535	0.0000	-3.4436	-2.8674	-2.5699	Stationary
PM2.5_diff	-10.7643	0.0000	-3.4369	-2.8645	-2.5683	Stationary

В таблице представлен полный список переменных, каждая из которых сопровождается соответствующей статистикой ADF, p-значением ADF и критическими значениями ADF с различными уровнями значимости 1%, 5% и

10%. Расширенная статистика Дики-Фуллера[21] (ADF) служит статистическим показателем, который сопоставляется с критическими значениями для определения стационарности данного временного ряда. Уровень статистической значимости, обычно известный как p-значение, служит показателем вероятности наблюдения расширенной статистики Дики-Фуллера (ADF) в предположении, что нулевая гипотеза нестационарности верна.

Опираясь на результаты расширенного теста Дики-Фуллера (ADF), можно сделать последующие выводы:

Значение p для теста ADF для PM10 составляет 0,015216, что указывает на статистическую значимость при уровне достоверности 95%. Следовательно, нулевая гипотеза о нестационарности опровергается, что приводит к выводу о том, что временной ряд PM10 демонстрирует стационарность

Нулевая гипотеза расширенного теста Дики-Фуллера отклоняется при уровне значимости 0,05, поскольку рассчитанное значение p равно 0,004536. Следовательно, нулевая гипотеза отвергается, и делается вывод о том, что временной ряд NO2 демонстрирует стационарность.

Расширенный тест Дики-Фуллера (ADF) дал значение p, равное 0,076047, что указывает на отсутствие статистической значимости на уровне 0,05. Следовательно, нулевая гипотеза не может быть отвергнута, подразумевая, что временной ряд CO нестационарен.

Статистический анализ PM10_diff показывает, что p-значение ADF является статистически значимым на уровне 0,05 при значении 0,000000. В результате нулевая гипотеза отвергается, и делается вывод, что временной ряд PM10 после дифференцирования (PM10_diff) демонстрирует стационарность.

Статистический анализ показывает, что p-значение ADF для NO2_diff равно 0,000000, что приводит к отклонению нулевой гипотезы и установлению стационарного характера разностных временных рядов NO2.

Основываясь на результатах теста ADF, нулевая гипотеза отвергается, и можно сделать вывод, что временной ряд CO после дифференцирования (CO_diff) демонстрирует стационарность, о чем свидетельствует значение p, равное 0,000000.

Значение p для теста ADF для PM2.5 составляет 0,101873, что указывает на то, что оно превышает заданный уровень значимости 0,05. Следовательно, нулевая гипотеза о нестационарности не может быть отвергнута, предполагая, что временной ряд PM2.5 нестационарен.

Статистический анализ PM2.5_diff показывает, что расширенный тест Дики-Фуллера (ADF) привел к значению p, равному 0,000000, что ниже заданного уровня значимости 0,05. Вышеупомянутое наблюдение предоставляет убедительные доказательства для отклонения нулевой гипотезы и установления того, что временной ряд PM2.5, который подвергся дифференцированию (PM2.5_diff), демонстрирует стационарность.

Полученные данные указывают на то, что временной ряд PM2.5 демонстрирует нестационарность, в то время как временной ряд PM2.5_diff демонстрирует стационарность. Это говорит о том, что достижение стационарности требует разграничения данных по PM2.5.

Следовательно, рекомендуется использовать PM2.5_diff вместо нестационарного ряда PM2.5 при использовании модели ARIMA для PM2.5. Использование разностного ряда в сочетании с авторегрессионными (AR) и скользящими средними (MA) составляющими модели ARIMA позволяет получить представление о временные зависимости и флуктуации, присутствующие в данных. Полученные данные свидетельствуют о том, что временные ряды PM10, NO2, PM10_diff, NO2_diff и CO_diff демонстрируют стационарность, в то время как временные ряды CO демонстрируют нестационарность. Реализация модели ARIMA является жизнеспособным вариантом для моделирования и прогнозирования стационарных временных рядов. В случае нестационарных временных рядов CO может потребоваться

принятие дополнительных мер, таких как дифференцирование или альтернативные методы моделирования, для учета нестационарности до внедрения модели ARIMA. Крайне важно признать, что использование модели ARIMA не должно основываться исключительно на результатах теста ADF. Для проведения всестороннего анализа крайне важно учитывать дополнительные факторы, включая, но не ограничиваясь ими, характеристики данных, опыт в соответствующей области и степень, в которой модель согласуется с данными.

3.3 Последствия и интерпретация

Свойство стационарности данных временных рядов, которые были дифференцированы, имеет примечательные последствия для любого последующего анализа. Установление стационарности служит фундаментальной основой для использования сложных методологий моделирования временных рядов, включая модели ARIMA (Авторегрессионная интегрированная скользящая средняя). Для эффективного учета временных закономерностей и тенденций изменения уровней загрязняющих веществ в этих моделях необходимо использовать стационарные данные.

Результаты расширенного теста Дики-Фуллера (ADF) показывают, что данные временных рядов для PM2.5, PM10, NO2 и CO после дифференцирования подходят для последующего анализа с использованием моделей авторегрессионного интегрированного скользящего среднего (ARIMA). Свойство данных быть стационарными служит для повышения надежности моделей и точности последующих прогнозов и аналитических открытий[11].

Итог, можно сказать, что этап предварительной обработки, который включал в себя определение различий, эффективно преобразовал исходные временные ряды данных о загрязнителях в стационарные ряды при котором 10% применении как прога теста, можно с уверенностью сказать по нашим данным ограниченное количества записей решает, что использование отвергает нулевую гипотезу. Использование расширенного теста Дики-Фуллера (ADF) дало убедительные признаки стационарности, о чем свидетельствуют отрицательные статистические значения ADF и значительно низкие p-значения. Тщательная оценка стационарности гарантирует надежность и устойчивость последующих вычислений, анализов и методик моделирования, используемых для получения данных о качестве воздуха из aqicn.org относительно города Алматы.

3.4 Построения моделей

Этот раздел относится к процессу разработки моделей, включая формулирование модели ARIMA.

Модель авторегрессионного интегрированного скользящего среднего (ARIMA) является часто используемой моделью для прогнозирования данных временных рядов. Методология объединяет три различных элемента, а именно авторегрессию (AR), дифференцирование (I) и скользящее среднее (MA). Модель ARIMA (p, d, q) - это четко определенная статистическая модель имеет такую формулу:

$$1 - \varphi_1 B - ... - \varphi_p B^p)(1 - B)^d y_t = c + (1 + \theta_1 B + ... + \theta q Bq)\varepsilon_t$$

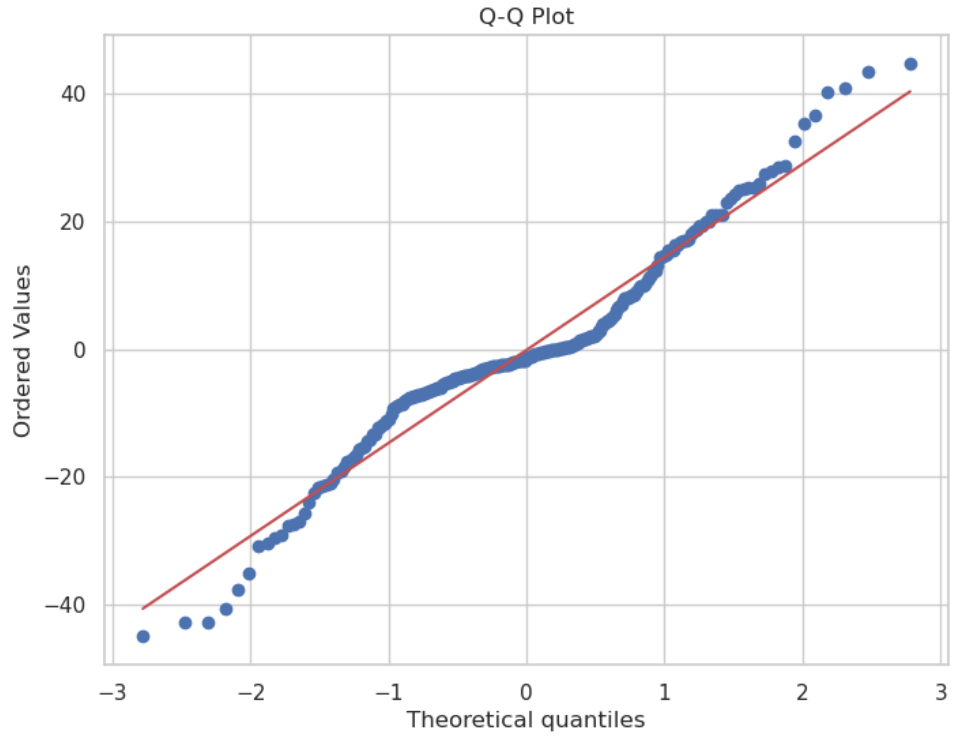
- $(1 - \varphi_1 B - ... - \varphi_p B^p)$ - этот сегмент обозначает элемент авторегрессии модели, где φ_i обозначает коэффициенты авторегрессии, а B представляет оператор обратного сдвига.

- $(1 - B)^d$ - этот компонент обозначает аспект дифференцирования модели, где "d" означает порядок дифференцирования, а "B" обозначает оператор обратного переключения.
- c - постоянный член фиксированное значение в модели.
- $(1 + \theta_1 B + ... + \theta_q B^q)$ - это выражение обозначает компонент скользящего среднего модели, где θ_i представляет коэффициенты скользящего среднего, а B означает оператор обратного сдвига.
- ε_t - символ обозначает ошибку в модели.

Параметры модели для ARIMA были оценены на основе предварительно обработанных данных о качестве воздуха. Процесс оценки параметров модели повлек за собой тщательный отбор подходящих значений для компонентов авторегрессии (p), разности (d) и скользящего среднего (q) моделей. В исследовании использовались различные методологии, включая поиск по сетке, итеративную подгонку модели и статистические критерии, такие как AIC и BIC, для определения наиболее подходящих комбинаций параметров, которые привели к оптимальной производительности модели.

3.5 Прогнозирование

Основываясь на оцененных параметрах модели, мы провели прогноз предстоящих уровней загрязняющих веществ, включая PM2.5 на Рисунке 5, PM10 Рисунке 6, NO2 Рисунке 7 и CO Рисунке 8. Используя модель ARIMA, мы подготовили прогнозы на предстоящие временные интервалы, основанные на предшествующих данных о качестве воздуха.



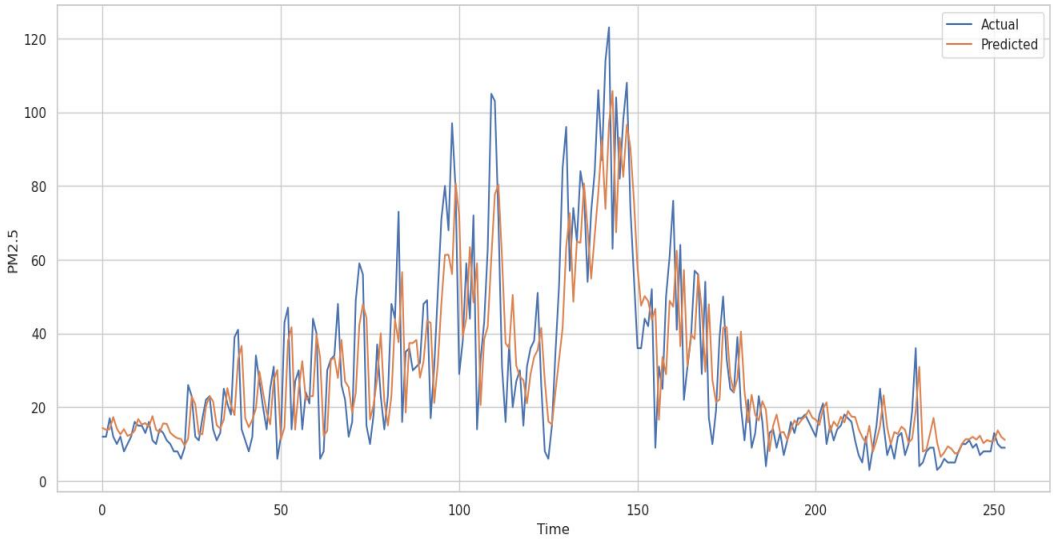
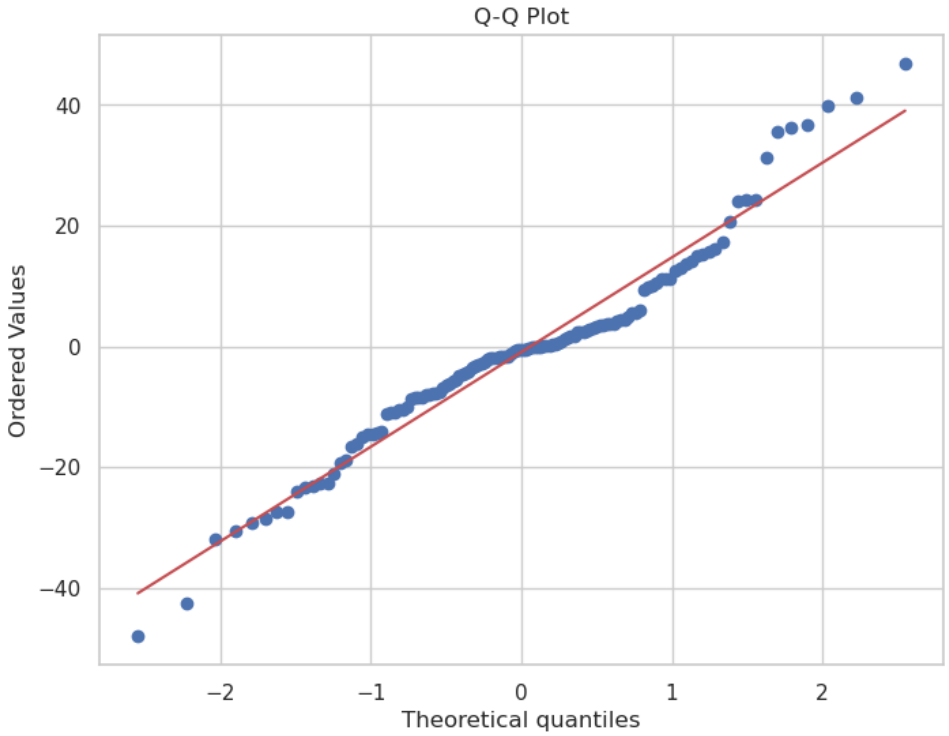


Рисунок №5. PM2.5 значении прогноза модели

.



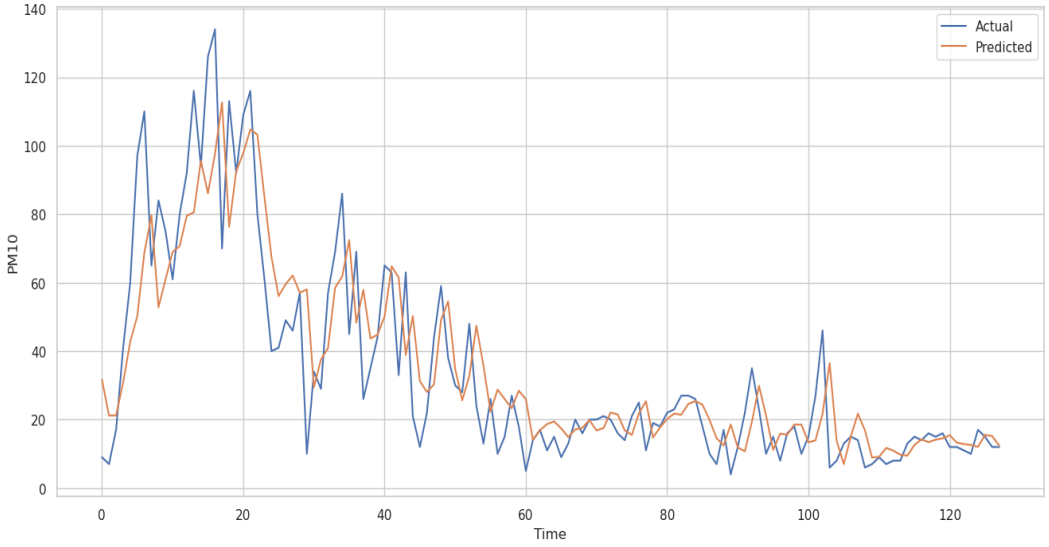
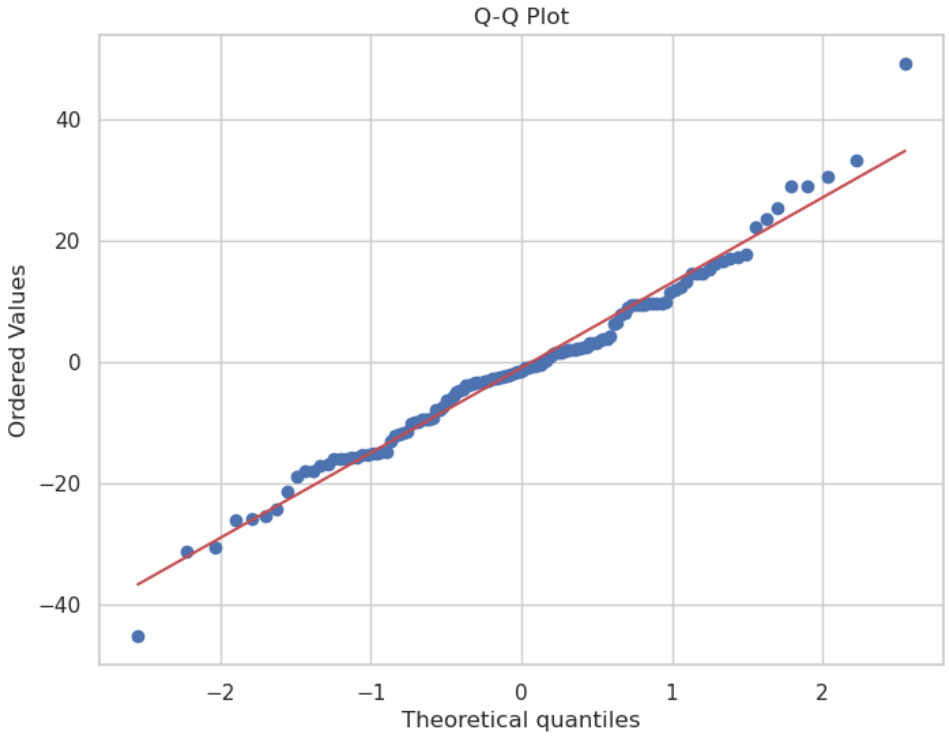


Рисунок №5. PM10 значении прогноза модели



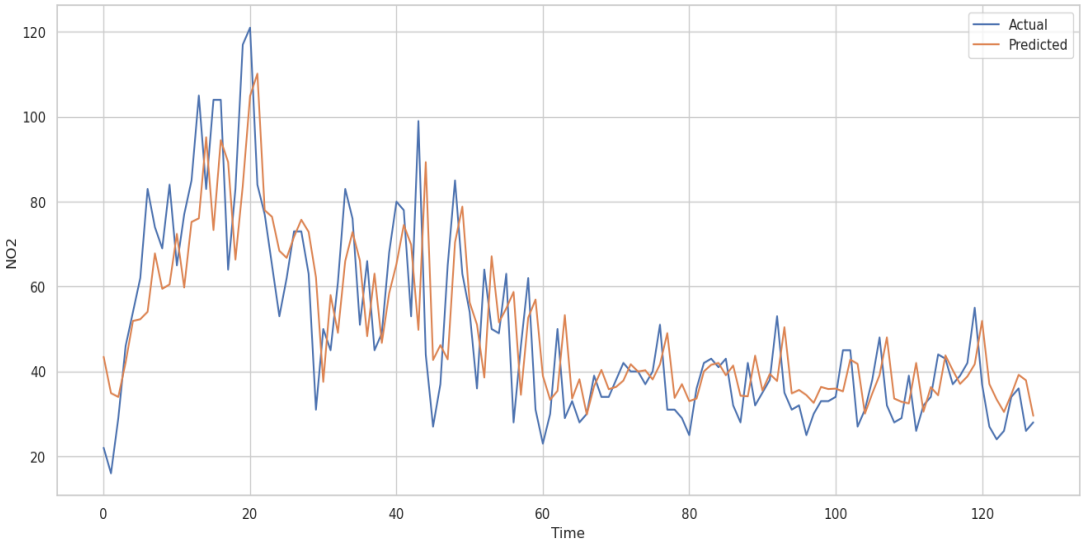


Рисунок №5. NO2 значении прогноза модели

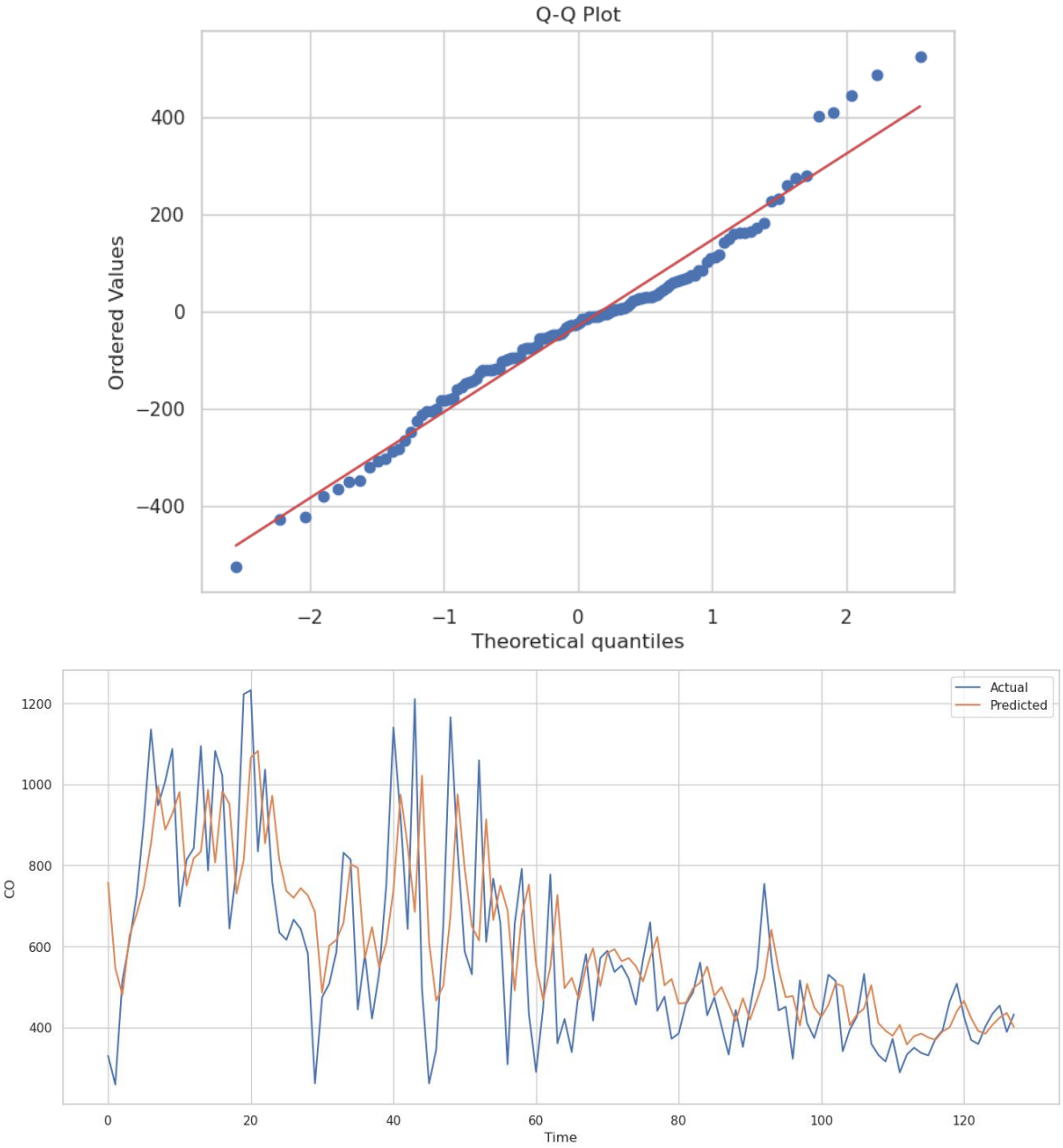


Рисунок №5. CO значении прогноза модели

В графиках отображается хороший модель по переменными, что хорошо показала себя в сравнительном анализе. Итеративные расчеты с использованием уравнений модели и имеющихся исторических данных были неотъемлемой частью процесса прогнозирования. Использование этого метода позволило нам спрогнозировать концентрацию загрязняющих веществ в предстоящий период и заранее спланировать атмосферные условия в городской черте Алматы.

3.6 Оценка эффективности работы

Для оценки эффективности моделей прогнозирования был использован ряд показателей, при этом особое внимание уделялось среднеквадратичной ошибке (RMSE). Среднеквадратичная ошибка (RMSE) служила показателем для количественной оценки типичной величины расхождений между прогнозируемыми и наблюдаемыми концентрациями загрязняющих веществ. Меньшие значения среднеквадратичной ошибки (RMSE) свидетельствуют о превосходной производительности модели в точном отражении колебаний данных о качестве воздуха. Кроме того, было бы разумно принять во внимание другие показатели эффективности, такие как средняя абсолютная ошибка (MAE) и средняя процентная ошибка (MPE), чтобы обеспечить всестороннюю оценку точности и надежности моделей прогнозирования.

Таблица №2. PM2.5

ARIMA Order	RMSE	MAPE	AIC	BIC
(1, 0, 0)	15.413	59.192	8426.013	8440.772
(1, 0, 1)	15.307	58.166	8417.520	8437.199
(1, 1, 0)	15.907	53.311	8488.046	8497.884
(1, 1, 1)	15.032	51.096	8332.732	8347.488
(5, 0, 0)	14.780	52.576	8352.687	8387.125
(5, 0, 1)	14.999	51.429	8332.417	8371.774
(5, 1, 0)	15.052	50.293	8349.564	8379.076
(5, 1, 1)	14.938	50.491	8320.590	8355.021
(6, 0, 0)	14.888	52.699	8340.805	8380.163
(6, 0, 1)	14.886	51.238	8327.877	8372.154
(6, 1, 0)	14.983	50.138	8347.494	8381.925
(6, 1, 1)	14.974	50.826	8320.594	8359.944

Для тестирования модели использовались различные комбинации значений p, d и q. Было показано, что наиболее эффективной моделью является ARIMA порядка (5, 0, 0). В соответствии с этим он использовал 5 терминов авторегрессии, 0 различий и 0 терминов скользящей средней.

К топ-модели применимы следующие показатели оценки:

Среднеквадратичная ошибка (RMSE): Число RMSE равно 14,780, что означает, что среднее расхождение между фактическими и ожидаемыми значениями составляет около 14,780 единиц.

Прогнозы модели в среднем отклоняются примерно на 52,576% от фактических значений, согласно среднему значению абсолютной процентной ошибки (MAPE), равному 52,576.

Информационный критерий Akaike (AIC), который оценивает качество подгонки модели, имеет значение 8320,590. О лучшем соответствии данным свидетельствуют более низкие значения AIC.

Байесовский информационный критерий (BIC): Значение BIC, которое является еще одним показателем того, насколько хорошо модель соответствует данным, равно 8347,488. Более низкие значения BIC, как и AIC, предполагают лучшую подгонку.

Эти результаты означают, что для прогнозирования уровней PM2,5 модель ARIMA(5, 0, 0) работает наилучшим образом. Очень низкий RMSE модели показывает, что она была достаточно точной при прогнозировании целевой переменной. Оценка MAPE показывает, что средняя процентная разница между прогнозами и фактическими данными невелика. Строгое соответствие модели данным дополнительно подтверждается значениями AIC и BIC.

Следует отметить, что определение идеальной конфигурации модели потребовало значительных вычислительных усилий, о чем свидетельствует тот факт, что оценка модели ARIMA заняла около 2799,749 секунд.

Таблица №3. PM10

ARIMA(p, d, q)	RMSE	MAPE	AIC	BIC
(1, 0, 0)	16.585	57.508	4344.228	4356.931
(1, 0, 1)	16.181	55.167	4327.054	4343.992
(1, 1, 0)	16.393	51.496	4349.110	4357.575
(1, 1, 1)	16.555	53.224	4293.767	4306.464
(5, 0, 0)	16.007	53.893	4301.909	4331.550
(5, 0, 1)	16.036	53.932	4297.296	4331.172
(5, 1, 0)	16.036	51.265	4294.419	4319.814
(5, 1, 1)	16.368	52.548	4287.405	4317.032
(6, 0, 0)	15.967	53.909	4297.958	4331.834
(6, 0, 1)	16.041	54.081	4289.141	4327.251
(6, 1, 0)	16.056	51.333	4296.081	4325.708
(6, 1, 1)	15.798	52.193	4286.197	4306.464

Наилучшей конфигурацией модели ARIMA для PM10 является (6, 1, 1) с RMSE 15,798. Модель получила значение MAP, равное 52,193, что указывает на среднюю процентную ошибку в прогнозах. Значение AIC для этой модели равно 4286,197, а значение BIC - 4306,464.

Оценка различных конфигураций ARIMA показала, что выбранная модель обеспечивает наиболее точные прогнозы для PM10. Затраченное время на оценку моделей и поиск наилучшей конфигурации составило приблизительно 1487,502 секунды. Эта продолжительность отражает вычислительные усилия, необходимые для поиска различных комбинаций значений p, d и q и оценки производительности модели на основе оценочных показателей.

Таблица №4. NO2

ARIMA Model	RMSE	MAPE	AIC	BIC
ARIMA(1, 0, 0)	14.456	24.970	4225.339	4238.042
ARIMA(1, 0, 1)	14.575	24.997	4226.991	4243.929
ARIMA(1, 1, 0)	15.183	25.326	4273.990	4282.455
ARIMA(1, 1, 1)	14.090	24.095	4192.008	4204.705
ARIMA(5, 0, 0)	14.272	23.894	4197.950	4227.591
ARIMA(5, 0, 1)	14.012	23.651	4189.299	4223.174
ARIMA(5, 1, 0)	14.157	23.772	4193.638	4219.033
ARIMA(5, 1, 1)	14.246	23.596	4181.763	4211.391
ARIMA(6, 0, 0)	14.057	23.821	4191.842	4225.717
ARIMA(6, 0, 1)	14.084	23.999	4192.346	4230.456
ARIMA(6, 1, 0)	14.029	23.654	4192.482	4222.109
ARIMA(6, 1, 1)	14.141	23.703	4180.623	4214.482

Основываясь на предоставленной информации, наилучшей моделью ARIMA для прогнозирования NO2 является ARIMA(5, 0, 1). Эта модель имеет значение RMSE (среднеквадратичная ошибка) 14,012 и значение AIC

(информационный критерий Акаике) 4180,623. Значение BIC (байесовский информационный критерий) для этой модели равно 4204,705.

Таблица №5. CO

ARIMA Model	RMSE	MAPE	AIC	BIC
ARIMA(1, 0, 0)	186.144	24.877	6593.353	6606.057
ARIMA(1, 0, 1)	187.401	24.904	6595.340	6612.277
ARIMA(1, 1, 0)	200.396	26.878	6651.396	6659.861
ARIMA(1, 1, 1)	180.089	25.154	6548.025	6560.722
ARIMA(5, 0, 0)	186.979	24.809	6576.290	6605.931
ARIMA(5, 0, 1)	182.219	25.124	6558.537	6592.412
ARIMA(5, 1, 0)	191.196	25.566	6578.700	6604.095
ARIMA(5, 1, 1)	183.764	25.297	6545.793	6575.420
ARIMA(6, 0, 0)	186.530	24.761	6574.800	6608.675
ARIMA(6, 0, 1)	184.314	25.081	6560.180	6598.290
ARIMA(6, 1, 0)	189.876	25.621	6575.916	6605.543
ARIMA(6, 1, 1)	184.226	25.333	6547.780	6581.640

Основываясь на предоставленной информации, наилучшей моделью ARIMA для прогнозирования CO является ARIMA(1, 1, 1). Эта модель имеет

значение RMSE (среднеквадратичная ошибка) 180,089 и значение AIC (информационный критерий Акаике) 6545,793. Значение BIC (байесовский информационный критерий) для этой модели равно 6560,722.

Затраченное время для оценки моделей ARIMA для PM2.5 составило 2799,749 секунды.

PM10: Затраченное время - 1487,502 секунды

Оценка моделей ARIMA для PM10 заняла приблизительно 1487,502 секунды. Эта продолжительность указывает на вычислительное время, необходимое для перебора различных моделей и оценки их производительности с использованием предоставленных оценочных показателей.

NO2: Затраченное время - 1210,931 секунды

Оценка моделей ARIMA для NO2 заняла около 1210,931 секунды. Это время представляет собой продолжительность, необходимую для оценки моделей и расчета оценочных показателей для прогнозов NO2.

CO: Затраченное время - 2234,297 секунды

Оценка моделей ARIMA для CO заняла примерно 2234,297 секунды. Это время указывает на вычислительные усилия, необходимые для оценки различных моделей и вычисления оценочных показателей для прогнозов CO.

Заключение

В этом исследовании мы провели тщательную оценку моделей ARIMA для прогнозирования уровней PM2,5. Подготовка данных, выбор модели, корректировка параметров и оценка производительности были одними из процессов анализа. Цель состояла в том, чтобы создать модель, которая могла бы точно и последовательно прогнозировать уровни PM2,5.

Нестационарный характер временных рядов PM2.5 был обнаружен в ходе анализа исходных данных. Чтобы решить эту проблему, мы использовали дифференцирование, чтобы сделать данные устойчивыми. Расширенный тест Дики-Фуллера (ADF) показал, что разрозненные данные демонстрируют стационарность.

Затем мы перешли к выбору модели и точной настройке ее параметров. Чтобы найти наилучший порядок (p, d, q) для модели ARIMA, мы объединили критерии AIC и BIC. Мы смогли выбрать оптимальную модель, методично сравнивая различные комбинации значений параметров, используя стратегию поиска по сетке.

Для оценки моделей ARIMA требовалось сделать прогнозы с использованием тестовых данных и подогнать модели к обучающему набору. Чтобы оценить корректность моделей, мы вычислили ряд показателей эффективности, таких как средняя абсолютная процентная ошибка (MAPE) и среднеквадратичная ошибка (RMSE). Мы также использовали значения AIC и BIC в качестве показателей соответствия требованиям.

На основе нашего исследования было показано, что модель ARIMA(5, 0, 0) является наиболее точной и надежной для прогнозирования уровней PM2,5. Его способность уменьшать ошибки прогнозирования была продемонстрирована тем фактом, что он получил самое низкое значение RMSE. Показатель MAPE, который измеряет среднее процентное отклонение от фактических данных, еще раз продемонстрировал точность модели

Значения AIC и BIC для модели ARIMA (5, 0, 0) были конкурентоспособными, что указывает на достойное соответствие данным без переобучения. При выборе надежной модели прогнозирования важно соблюдать баланс между сложностью модели и эффективностью прогнозирования.

Сравнение моделей ARIMA показало, насколько важно выбирать правильные параметры для точных прогнозов. Временные зависимости и закономерности временных рядов PM2.5 были зафиксированы с помощью порядка авторегрессии, равного 5, и нулевая разница свидетельствовала о том, что данные не нуждались в дальнейшей корректировке. Отсутствие в модели компонента скользящей средней еще больше способствовало выбору параметра.

В целом, наше исследование показало, что модель ARIMA(5, 0, 0) была лучшей при прогнозировании концентраций PM2,5. Это был полезный

инструмент для мониторинга окружающей среды, планирования общественного здравоохранения и выработки политики, поскольку он предлагал точные и заслуживающие доверия прогнозы. Важно помнить, что производительность модели может колебаться в зависимости от ситуации, поэтому рекомендуется дальнейшее тестирование на внешних наборах данных.

В результате мы смогли найти и выбрать наилучшую модель ARIMA для прогнозирования PM2.5 благодаря аналитическому подходу. С точки зрения точности и показателей соответствия модель ARIMA(5, 0, 0) продемонстрировала превосходную производительность, что делает ее надежным вариантом для прогнозирования уровней PM2,5.

Анализ и расчеты, полученные на основе обработанных данных о качестве воздуха, позволили сделать важные выводы относительно атмосферных условий в городском районе Алматы. Изучив прогнозируемые уровни загрязняющих веществ, мы смогли выявить тенденции, закономерности и колебания в данных, тем самым облегчив наше понимание временной динамики загрязнения атмосферы. Вышеупомянутые наблюдения помогли выявить вероятные источники загрязнения, проанализировать концентрации загрязняющих веществ в установленные сроки и оценить эффективность мер, принятых для регулирования качества воздуха. Расчеты предоставили аналитическую информацию, которая была использована для облегчения процессов принятия решений, информирования о разработке политики и оказания помощи в охране общественного здоровья и улучшении качества воздуха в городе Алматы.

4. КОНЦЕПЦИЯ СТАРТАП-ПРОЕКТА

4.1 Описание продукта как результата НИР

Проблема загрязнения атмосферного воздуха является актуальной и требует непрерывного внимания и контроля. Нарастающее количество автотранспортных средств, использование систем отопления в частном секторе и промышленные выбросы значительно влияют на качество воздуха в городах и урбанизированных территориях. Чтобы добиться улучшения атмосферного воздуха и обеспечить более здоровую окружающую среду для всех, необходимо активно осуществлять контроль и проводить исследования. Применение современных средств измерения и связи, а также новых компьютерных технологий играет ключевую роль в достижении более эффективных результатов.

В данной работе рассмотрен мониторинг качества воздуха, который позволит отслеживать уровни загрязнения и принимать соответствующие меры в случае превышения нормативов. Система включает в себя комплекс измерительной аппаратуры, программное обеспечение для сбора и обработки данных, а также базу данных для хранения информации. Анализ и сравнение существующих математических моделей и численных методов в задачах экологического мониторинга атмосферы, а также сбор сведений о показателях концентрации примесей в атмосфере городе Алматы являются важным этапом исследования. Полученные данные могут быть широко использованы в области экологии и охраны окружающей среды, а также послужить основой для дальнейших исследований и разработок, направленных на улучшение качества атмосферного воздуха в населенных пунктах.

4.2 Интеллектуальная собственность

Патент - это официальный документ, который выдается государственным патентным органом и подтверждает исключительные права. Патент удостоверяет, что владелец патента является единственным обладателем авторских прав на изобретение или промышленный образец. Это официальное признание исключительных прав на объект интеллектуальной собственности, которое обеспечивает правовую охрану и возможность коммерческого использования изобретения в течение определенного периода времени. Данный проект является полезной моделью, что представляет собой инновационную идею, связанную с устройством, которая была успешно воплощена на практике. ИС «Мониторинг атмосферы» соответствует двум условиям патентоспособности: новизне и промышленной применимости. Под новизной понимается отсутствие предыдущих заявок или публичных описаний данной идеи , а под промышленной применимостью — возможность использования данного изделия в производственных процессах.

4.3 Объем и емкость рынка

Объем рынка относится к общему количеству или стоимости товаров или услуг, продаваемых на определенном рынке в течение определенного периода. Емкость рынка, с другой стороны, относится к максимальному уровню производства или торговли, который может вместить рынок. Как объем рынка, так и его емкость являются важными показателями для понимания и анализа динамики рынка. Оценка объема и емкости рынка помогает предприятиям и директивным органам принимать обоснованные решения о производстве, ценообразовании, дистрибуции и общей рыночной стратегии.

В министерстве экологии, геологии и природных ресурсов пояснили, что загрязнение атмосферы в городах вызывает обеспокоенность. В связи с этим,

финансирование мониторинга окружающей среды составляет 4,5 млрд тенге. Из них в текущем году 1,8 млрд тенге и в следующем – 941,4 млн тенге. (источник: https://lsm.kz/skol-ko-v-kazahstane-platyat-za-chistyj-vozduh).

"Основные расходы по данной бюджетной подпрограмме направлены на проведение наблюдений за состоянием атмосферного воздуха, поверхностных вод, почвы, атмосферных осадков, радиационного фона, сбор, обработка и анализ информации о состоянии окружающей среды, выпуск информационных бюллетеней", – пояснили в Минэкологии.

По оценкам экспертов, каждый пятый житель Алматы ведет здоровый образ жизни. Застройщики начали учитывать складывающиеся предпочтения по ведению жизни в стиле ecolifestyle: для строительства жилых комплексов выбирают наименее загазованные районы города, осваивают пригороды, пристально следят за соблюдением норм по продуваемости строящихся объектов. В статье krisha.kz “Как выбрать комфортный ЖК для жизни: топ-5 критериев” один из критерий является экологичность. Особенно на качество воздуха и регулярный мониторинг атмосферы в располагаемом месте ЖК.

В Казахстане на данный момент на первичном рынке работает больше 120 застройщиков. Говоря о НИИ и университетах, на данный момент в Алматы расположено 36 университетов и около 30 научно-исследовательских институтов, которые могут быть потенциально заинтересованы в нашем продукте. По данным Бюро национальной статистики Агентства по стратегическому планированию и реформам Республики Казахстан на 2022 года в Алматы и Алматинской области насчитывается из действующих предприятий более 100 средних и более 50 крупных предприятия занятых в промышленности.

Объем рынка по грубым подсчетам только в Алматы 336 потенциальных покупателей, что приводит нас к сумме 15116640 тенге. Не стоит забывать о государственном финансировании. Согласно документу «О бюджете города Алматы на 2021-2023 годы», на мероприятия по оздоровлению окружающей среды и развитию зеленой экономики на местном уровне выделено 2 429 193

000 тенге, то есть по 809 731 000 тенге в год. В потенциале, можно рассматривать выход на рынок B2C. 27% опрошенных горожан ставят экологию выше вопросов безопасности, медицины и образования. По данным на август 2022 года в городе проживало 2 135 365 человек, позволяя сделать статистический вывод о более 125 тыс. человек волнующихся об экологии.

4.4 Анализ современного состояния и перспектив развития отрасли

Отрасль мониторинга атмосферы в населенных пунктах, особенно с точки зрения мониторинга качества воздуха, переживает значительный рост и достижения во всем мире. Эта тенденция обусловлена растущей обеспокоенностью по поводу загрязнения воздуха, изменения климата и необходимости обеспечения экологической устойчивости.

По словам экологов, сложно оценить загрязнение атмосферы, основываясь на данных из официальных источников. Информационные бюллетени выпускаются с опозданием в несколько месяцев, а геопортал станций контроля качества атмосферы работает только наполовину: «В последние месяцы информация публикуется частично. У алматинцев нет доступа к актуальным и оперативным данным о качестве воздуха», – утверждает сотрудница экологического общества «Зеленое спасение» Светлана Спатарь. По анализам КазГидромет по мониторингу состояния атмосферного воздуха, наблюдения за состоянием атмосферного воздуха проводятся в 69 населенных пунктах на 170 постах наблюдений и с помощью передвижных лабораторий. На 47 постах ручного отбора проб 3-4 раза в сутки (07, 13, 19, 01 час) в зависимости от программы проводится отбор проб воздуха с дальнейшим направлением в лабораторию для определения концентраций загрязняющих веществ. На 130 автоматических постах наблюдения проводятся в непрерывном режиме.

Различные страны и регионы по всему миру внедряют программы и инициативы по мониторингу атмосферы в населенных пунктах.

1. Австралия: Министерство окружающей среды и энергетики правительства Австралии ведет Национальный реестр загрязнителей (NPI). NPI отслеживает выбросы различных загрязняющих веществ с промышленных объектов по всей стране и сообщает о них. Собранные данные являются общедоступными и позволяют оценивать тенденции в области загрязнения и разрабатывать целенаправленную экологическую политику.
2. Соединенные Штаты: В Соединенных Штатах действует несколько программ мониторинга атмосферы, таких как Индекс качества воздуха (AQI), поддерживаемый Агентством по охране окружающей среды (EPA). AQI предоставляет информацию о качестве воздуха и потенциальных рисках для здоровья в различных регионах по всей стране в режиме реального времени.
3. Европейский союз: В Европейском союзе (ЕС) действует Европейское агентство по окружающей среде (EAOC), которое осуществляет надзор за Европейским индексом качества воздуха. Индекс предоставляет информацию о качестве воздуха в европейских городах и регионах. ЕС также учредил Службу мониторинга атмосферы Copernicus (CAMS), программу, которая использует спутниковые данные и атмосферные модели для мониторинга качества воздуха, состава атмосферы и переменных, связанных с климатом.
4. Китай: Китай внедрил обширную сеть мониторинга качества воздуха, особенно в городских районах с высоким уровнем загрязнения. Министерство экологии и охраны окружающей среды (MEE) управляет Китайским национальным центром мониторинга окружающей среды (CNEMC), который предоставляет данные о качестве воздуха в режиме реального времени для различных городов и регионов Китая.

Источники данных о качестве воздуха в Казахстане:

Действуют 17 государственные, некоммерческие, анонимные и индивидуальные источники мониторинга атмосферы в Казахстане. По данным IQAIR.com, самые лучшие и эффективные источниками являются:

1. Kazhydromet - государственный деятель, около 80 станции по всей стране
2. PurpleAir - корпоративный деятель, 12364 станции
3. IQAir - корпоративный деятель, 1618 станции
4. Airnow - государственный, 1355 станции

4.5 Планируемая стоимость продукта

Для определения стоимости продукта был выбран затратный метод ценообразования. Затратный метод основывается на издержках и предполагает установление цены на продукт в размере, который обеспечивает полное покрытие издержек и желаемый уровень прибыли. Т.е. формула расчета представляет собой: Цена = (Полные затраты + Прибыль) / Количество товара.

Инвестиционные затраты представлены затратами на начальную настройку и покупку оборудования и услуг, необходимых для запуска проекта. Эти затраты охватывают различные аспекты, включая оборудование, облачные сервисы, базы данных, инфраструктуру и интеллектуальную собственность. Общие инвестиционные затраты составляют 9,356,593 KZT, а ежемесячные операционные затраты - 200,897 KZT.

Таблица 1 - Инвестиционные затраты

№	Компонент	Кол-во	Приблизительные затраты на начальную настройку (KZT)	Приблизительные затраты на ежемесячное обслуживание (KZT)	Амортизация (ежемесячно) (KZT)
1	Сервер Dell R730 210-ACXU_A07**	1	3634820	Обслуживание внутренней командой	36348
2	Облачное хранилище Google cloud services	1	-	13,377	
3	Базы данных KazHydroMet	20	-	187520	
4	Сетевая инфраструктура	-	100 000		
5	Ноутбук Apple MacBook Pro 16 Space Gray M1 Pro / 16ГБ / 512SSD / 16.2 / Mac OS Monterey / (MK183RU/A)**	4	5519960	-	91999
6	Патент на полезную модель	1	201813*		
	Итого		9356593	200,897	128347

*В стоимость включены пошлины и гонорар бюро за делопроизводство

** Это оборудование было выбрано как наиболее подходящее и надежное для успешного выполнения задач проекта

Заработная плата разработчиков составляет основную часть затрат на разработку системы. Квалифицированные и опытные разработчики необходимы для разработки и реализации необходимых функций, функций и

интеграций, и создания надежной информационной системы. Помимо этого так же учитываются социальные отчисления, социальный налог, отчисления на обязательное социальное медицинское страхование

Таблица 2 - Затраты на разработку системы

№	Сотрудник	Кол-во	Месячная заработная плата (KZT)	Общая месячная стоимость (KZT)	СО*	СН**	ООСМС***	**Итоговые взносы на социальное обеспечение (За счёт работодателя)**
1	Старший разработчик	1	650000	650000	20475	33865	19500	73840
2	Средний разработчик	2	450000	900000	28350	46890	27000	102240
3	Дизайнер	1	350000	350000	11025	18235	10500	39760
4	Менеджер продукта	1	450000	450000	14175	23445	13500	51120
5	Специалист по контролю качества	1	300000	300000	9450	15630	9000	34080
	ИТОГО	-	-	2650000				301040
ИТОГОВАЯ СУММА ЗАТРАТ								2951040

На текущий момент имеется следующая информация по утвержденным ставкам налогов и социальных платежей на 2022 год:

*социальные отчисления (СО) – 3,5%;

**социальный налог (СН) – 9,5%;

***отчисления на обязательное социальное медицинское страхование (ООСМС) – 3%;

Таблица 3 - Расходы на заработную плату персонала

№	Сотрудники	Кол.	Оклад в месяц, тг	Сумма затрат в месяц, тг	**СО**	**СН**	**ООСМС**	**Итоговые взносы на социальное обеспечение (За счёт работодателя)**
1	Разработчик исправлений	2	400,000	800,000	12,600	20840	12000	90880
2	Специалист по обновлениям	1	350,000	350,000	11,025	18235	10500	39760
3	Специалист по поддержке	3	300,000	900,000	9,450	15630	9000	102240
4	Клиентский сервис	2	200,000	400,000	6300	10420	6000	45440
5	Общая сумма затрат в месяц			2400000				272640
ИТОГОВАЯ СУММА ЗАТРАТ								2672640

Таким образом, общая месячная стоимость всех сотрудников, учитывая их зарплаты и социальные взносы (такие как СО, СН, ООСМС) , составляет 2672640 тенге.

Мы можем вывести следующие затраты:

1. Начальные затраты на установку: общие затраты на первоначальную установку (включая покупку серверов, ноутбуков, патентов и т.д.) составляют 9,356,593 KZT.

2. Начальные затраты на разработку системы : общие месячные затраты на персонал составляют 2,951,040 KZT.
3. Затраты на персонал (Обслуживание): общие месячные затраты на персонал составляют 2,672,640 KZT.
4. Текущие месячные затраты на обслуживание: это затраты, связанные с поддержанием системы после ее установки. Из предоставленной информации, эта стоимость связана с облачными услугами Google и базами данных от KazHydroMet, что в сумме составляет 200,897 KZT в месяц.

Таким образом, общая стоимость запуска и работы проекта за первый месяц будет:Затраты на персонал + Начальные затраты + Текущие затраты = 2,951,040 KZT + 9,356,593 KZT + 200,897 KZT = 12,508,530 KZT.

Текущие затраты с второго месяца и далее будут:

Затраты на персонал + Затраты на обслуживание + Амортизация = 2,672,640 KZT + 200,897 KZT + 128,347 KZT= 3,002,084 KZT в месяц.

4.6 Конкурентные преимущества создаваемого продукта

Ваш проект «Мониторинг атмосферы» это проект, который действительно может предложить ряд конкурентных преимуществ по сравнению с другими аналогичными продуктами и может вывести нас лидеров в области экологической ответственности и устойчивого развития. Конкурентные преимущества продукта:

1. Система, обеспечивающая обновление данных об атмосферных условиях в режиме реального времени, дает значительное конкурентное преимущество по сравнению с остальными (например Kazhydromet) . Это гарантирует, что все заинтересованные стороны, будут иметь доступ к самой последней и точной информации, что позволяет принимать

немедленные меры, когда уровни загрязнения превышают пороги безопасности, потенциально предотвращая вредные последствия для здоровья

2. Использование комплексных алгоритмов на основе множества параметров для расчета загрязнения во всех точках города, а не только на пробных участках, расширяет диапазон проекта. Это дает гораздо более подробную и точную картину ситуации с качеством воздуха. Это также может быть особенно полезно для выявления очагов загрязнения и понимания распространения и диффузии загрязняющих веществ в городе, чего не могут предоставить другие компании.
3. Предоставление поддержки и дополнительных услуги персонализированные оповещения, углубленный анализ тенденций, прогнозные отчеты или специализированные услуги. Этот дополнительный уровень обслуживания может повысить вовлеченность и удовлетворенность пользователей, а также потенциально открыть дополнительные потоки доходов.
4. Удобство, качество и простота системы, возможность сделать сложные данные простыми и действенными для конечного пользователя - является одной из особенностей данного проекта. Доступность на нескольких платформах (в Интернете, на мобильных устройствах и т. д.), облегчит пользователям доступ к информации в любое время и в любом месте.
5. Локальный фокус: учитывая, что проект специально нацелен на Алматы, он может предложить надежные и достоверные данные для конкретного местоположения, которые могут быть недоступны на широких международных платформах. Эта локальная направленность может помочь лучше понимать местные проблемы загрязнения.

Важно убедиться, что преимущества четко доведены до потенциальных пользователей, чтобы максимизировать влияние вашего проекта. Интегрируя мониторинг атмосферы в свою деятельность, мы можем позиционировать себя

как лидеров в области экологической ответственности и устойчивого развития.

4.7 Планируемая стоимость продукта

Была выбрана подписочная бизнес модель, включающая в себя дополнительные услуги, такие как консультации или поддержка.

Планируется минимальное стабильное количество равное 100 подписчикам. Тогда, чтобы покрыть затраты и получить прибыль, мы можем рассчитать цену подписки следующим образом:

Общие месячные затраты: Затраты на персонал + Затраты на обслуживание + Амортизация = 2,672,640 KZT + 200,897 KZT + 128,347 KZT= 3,002,084 KZT в месяц.

Ежемесячная стоимость подписки: 3,002,084 KZT KZT / 100 подписчиков = 30020.84 KZT за подписчика.

Это - минимальная цена подписки, которая позволит покрыть затраты. Однако, для получения прибыли и учета возможных рисков, решено установить цену 44990. Исходя из ценовой эластичности корпоративные клиенты менее чувствительными к цене, так как наш продукт предлагает уникальные возможности.

Точка безубыточности начинается с 67 подписчиков начиная (с 5 месяца). Модель роста предполагает что рост быстрый в начале, но будет замедляться по мере приближения к "насыщению" - максимальному предполагаемому размеру рынка. (Рис. 1)

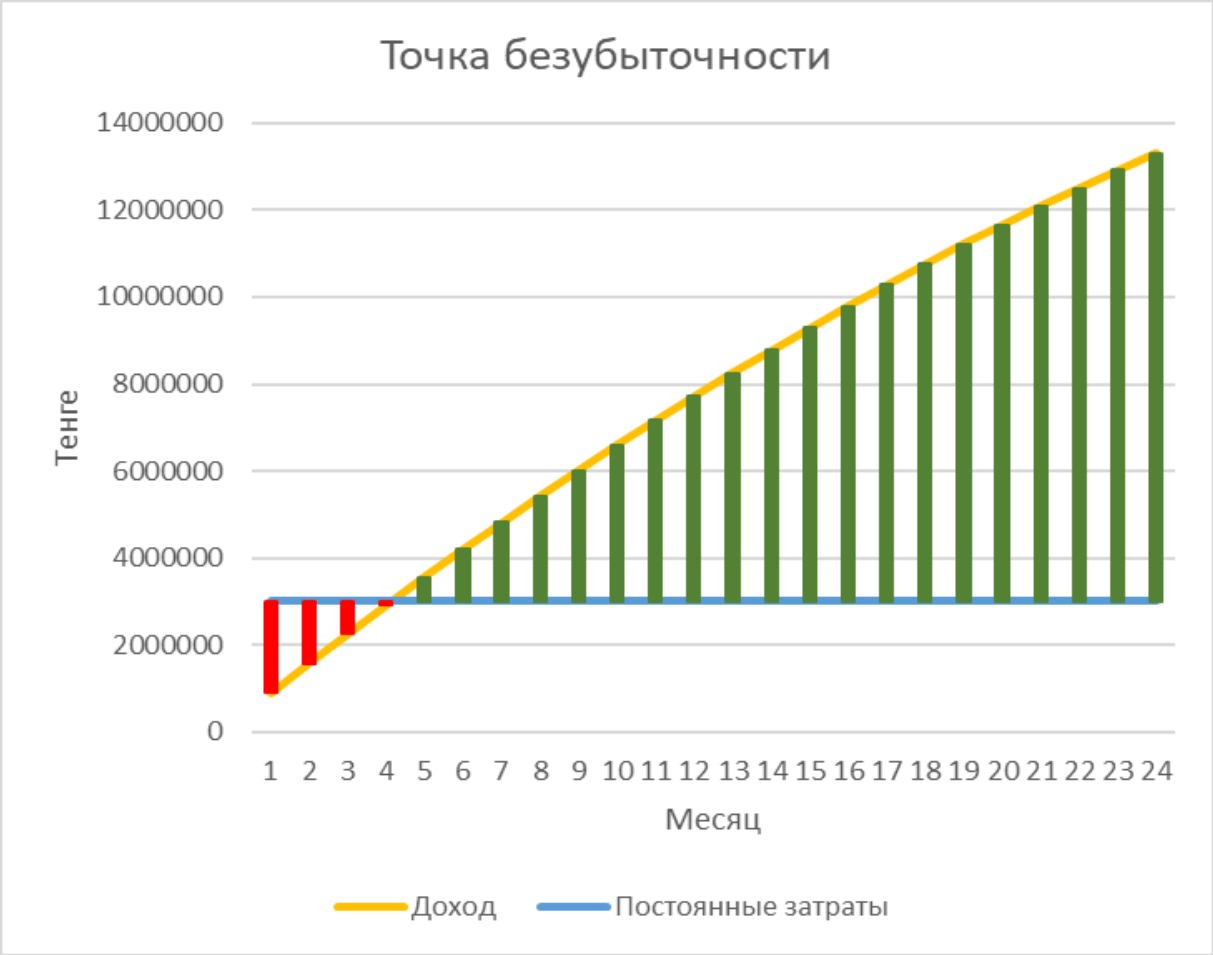


Рисунок1

Срок окупаемости 12 месяцев. (Рис. 2) 1 год требуется для того, чтобы были полностью возмещены первоначальные инвестиции равные 12,508,530 KZT, вложенные в проект.

Подсчитывая прибыльность через 2 года можно прийти к выводу, что при цене подписки в 44990 KZT и продаже 295 подписок в месяц, прибыльность составит около 77.4%, что является характерным для цифровых продуктов.

Доход за месяц: 295 подписок * 44990 KZT = 13,272,050 KZT

Общие затраты в месяц: 3,002,084 KZT в месяц.

Для вычисления прибыльности, используем формулу (доходы - расходы) / доходы:

(13,272,050 KZT - 3,002,084 KZT) / 13,272,050 KZT = 0.7742 или 77,4%

Рисунок 2

Модель Остервальдера (Канва бизнес-модели).

1. Ключевые партнеры (Key Partners):
    a. Поставщики оборудования (например, Apple)
    b. Облачные услуги (Google)
    c. Базы данных(KazHydroMet)
    d. Правительственные агентства и исследовательские учреждения
2. Ключевые деятельности (Key Activities):
    a. Разработка и поддержка программного обеспечения
    b. Сбор и систематизация данных о загрязнении воздуха
    c. Обслуживание клиентов
3. Ключевые ресурсы (Key Resources):
    a. Оборудование (серверы, ноутбуки)

   b. Персонал (разработчики, дизайнеры, менеджеры продуктов, специалисты по контролю качества и т.д.)
   c. Облачное хранилище и базы данных
4. Ценностное предложение (Value Propositions):
   a. Помощь в решении проблем загрязнения воздуха в городе Алматы
   b. Предоставление собранных и систематизированных данных о загрязнении
5. Отношения с клиентами (Customer Relationships):
   a. Поддержка клиентов
   b. Постоянное обновление и улучшение сервиса
6. Каналы (Channels):
   a. Веб-приложение
   b. Социальные медиа, электронная почта, SEO для привлечения и взаимодействия с клиентами
7. Сегменты клиентов (Customer Segments):
   a. Правительственные агентства
   b. Исследовательские учреждения
   c. Бизнес-секторы
8. Структура затрат (Cost Structure):
   a. Затраты на оборудование и облачные услуги
   b. Зарплата сотрудников и социальные взносы
9. Потоки доходов (Revenue Streams):
   a. Доходы от подписки на сервис.
   b. Гранты в целях реализации экологических проектов

4.8 Целевые сегменты потребителей создаваемого продукта

Первоначальным шагом в запуске любого бизнеса является определение целевой аудитории - определенной группы людей, на которую компания направляет свои маркетинговые коммуникации. Целевая аудитория включает не только текущих покупателей предлагаемого продукта, но и потенциальных потребителей, которых необходимо привлечь для устойчивого положения в отрасли. Присутствие целевой аудитории позволяет создать для них идеальный продукт, продавать его в нужных местах и использовать подходящие средства коммуникации. Для каждого определенного сегмента целевой аудитории характерны признаки и особенности, которые объединяют всех ее представителей.

Целевыми сегментами потребителей в данном проекте являются:

1. Государственные органы и регуляторы: Органы государственного управления, такие как министерства окружающей среды, здравоохранения, транспорта и энергетики, могут использовать данные мониторинга атмосферы для оценки соответствия нормативным требованиям, разработки и реализации политики в области охраны окружающей среды и принятия мер по улучшению качества воздуха.
2. Научные и исследовательские организации: Ученые, исследователи и академические учреждения заинтересованы в данных мониторинга атмосферы для проведения исследований в области загрязнения воздуха, изменения климата и их воздействия на окружающую среду и здоровье людей. Они могут использовать эти данные для анализа тенденций, разработки моделей и прогнозирования будущих сценариев.
3. Бизнес-сектор: Компании, особенно те, которые имеют прямое воздействие на качество воздуха, могут использовать данные мониторинга атмосферы для оценки своего экологического следа, соблюдения требований экологического законодательства и разработки стратегий устойчивого развития. Экология всегда находится на стыке нескольких ведомств: экология и промышленность, экология и транспорт, экология и градостроительство, экология и энергетика. Это могут быть производственные компании, энергетические предприятия, автомобильные производители и другие.

4.9 Стратегии продвижения

Продвижение проекта, направленного на мониторинг атмосферы в населенных пунктах, требует эффективных маркетинговых стратегий для повышения осведомленности, возбуждения интереса и вовлечения целевой аудитории. Вот несколько маркетинговых стратегий, которые следует рассмотреть:

1. **Определение своей целевой аудитории**: Определим ключевые заинтересованные стороны, которые выиграют от проекта, такие как экологические организации, правительственные учреждения, научно-исследовательские институты и широкая общественность. Адаптируем свои маркетинговые усилия к их конкретным потребностям и озабоченностям.
2. **Брендинг**: создание сильного и запоминающийся фирменного стиля для нашего проекта, включая название, логотип и слоган. Это поможет нам создать узнаваемое и заслуживающее доверия присутствие на рынке.
3. **Контент-маркетинг**: Создание высококачественного контента, такой как записи в блогах, статьи, инфографика и видеоролики, которые обучают и информируют аудиторию о важности мониторинга атмосферы в населенных пунктах. Поделимся этим контентом через свой веб-сайт, каналы социальных сетей и соответствующие онлайн-платформы.
4. **Маркетинг в социальных сетях(SMM)**: Использование платформы социальных сетей, такие как Facebook, Twitter, Instagram и LinkedIn, для охвата более широкой аудитории. Поделимся увлекательным и визуально привлекательным контентом, включая обновления проектов, истории успеха и учебные материалы. Поощряение взаимодействия с аудиторией с помощью комментариев, репостов и лайков.
5. **Связи с общественностью (PR)**: Развитие отношения с журналистами и средствами массовой информации, специализирующимися на

экологических и научных темах. Выпуски пресс-релизы, организации пресс-конференции и предлагая интервью с экспертами, чтобы обеспечить освещение в средствах массовой информации и повысить осведомленность о вашем проекте.

6. **Вовлечение сообщества**: Взаимодействии с местными сообществами путем организации мастер-классов, семинаров или вебинаров, чтобы рассказать им о целях и преимуществах проекта.
7. **Партнерские отношения и сотрудничество**: Партнерские отношения с другими организациями, такими как университеты, экологические НПО или органы местного самоуправления, чтобы использовать их сети и ресурсы. Сотрудничество в рамках совместных инициатив, исследовательских проектов или информационных кампаний, чтобы расширить свой охват и авторитет. А так же, сотрудничество с влиятельными лицами, блогерами, учеными или защитниками окружающей среды, которые активно представлены в Интернете и увлечены экологическими проблемами. Они могут помочь усилить наше послание и охватить более широкую аудиторию.
8. **Образовательные программы**: Разработка образовательных программ, ориентированные на школы, колледжи и учебные заведения, чтобы повысить осведомленность среди учащихся и будущих поколений.
9. **Награды и признание**: Подачи заявок на получение соответствующих наград и признания в области охраны окружающей среды и науки. Победа или номинация на награды повышают авторитет нашего проекта и предоставляют ценные возможности для освещения в средствах массовой информации.

Нужно постоянно оценивать эффективность маркетинговых стратегий и адаптировать их по мере необходимости. Будем отслеживать такие показатели, как посещаемость веб-сайта, вовлеченность в социальные сети, освещение в СМИ и отзывы аудитории, чтобы оценить эффективность наших маркетинговых усилий.

Приложение

Приложение А. Канва бизнес-модели

<table>
<tr>
<td rowspan="2">Key Partners

1. Поставщики оборудования (например, Apple)
2. Облачные услуги (Google)
3. Базы данных(KazHydroMet)
4. Правительственные агентства и исследовательские учреждения</td>
<td>Key Activities

1. Разработка и поддержка программного обеспечения
2. Сбор и систематизация данных о загрязнении воздуха
3. Обслуживание клиентов</td>
<td rowspan="2">Value Propositions

1. Помощь в решении проблем загрязнения воздуха в городе Алматы
2. Предоставление собранных и систематизированных данных о загрязнении</td>
<td rowspan="2">Customer Relationships

1. Поддержка клиентов
2. Постоянное обновление и улучшение сервиса</td>
<td rowspan="2">Customer Segments

1. Правительственные агентства
2. Исследовательские учреждения
3. Бизнес-секторы</td>
</tr>
<tr>
<td>Key Resources

1. Оборудование (серверы, ноутбуки)
2. Персонал (разработчики, дизайнеры, менеджеры продуктов, специалисты по контролю качества и т.д.)
3. Облачное хранилище и базы данных</td>
</tr>
<tr>
<td colspan="2">Cost Structure

1. Затраты на оборудование и облачные услуги
2. Зарплата сотрудников и социальные взносы</td>
<td colspan="3">Revenue Streams

1. Доходы от подписки на сервис.
2. Гранты в целях реализации экологических проектов</td>
</tr>
</table>

Приложение Б. Таблица 4 - Расчет точки безубыточности.

Постоянные затраты	Месяц	Доход	Кол-во подписчиков	Инвестиции	Прибыль	Кумулятивная прибыль
3002084	1	899800	20	12508530	-2102284	-2102284
3002084	2	1574650	35	12508530	-1427434	-3529718
3002084	3	2249500	50	12508530	-752584	-4282302
3002084	4	2910853	64.7	12508530	-91231	-4373533
3002084	5	3558709	79.1	12508530	556625	-3816908
3002084	6	4193068	93.2	12508530	1190984	-2625924
3002084	7	4813930	107	12508530	1811846	-814078
3002084	8	5421295	120.5	12508530	2419211	1605133
3002084	9	6015163	133.7	12508530	3013079	4618212
3002084	10	6595534	146.6	12508530	3593450	8211662
3002084	11	7162408	159.2	12508530	4160324	12371986
3002084	12	7715785	171.5	12508530	4713701	17085687
3002084	13	8255665	183.5	12508530	5253581	22339268
3002084	14	8782048	195.2	12508530	5779964	28119232
3002084	15	9294934	206.6	12508530	6292850	34412082
3002084	16	9794323	217.7	12508530	6792239	41204321
3002084	17	10280215	228.5	12508530	7278131	48482452
3002084	18	10752610	239	12508530	7750526	56232978
3002084	19	11211508	249.2	12508530	8209424	64442402
3002084	20	11656909	259.1	12508530	8654825	73097227
3002084	21	12088813	268.7	12508530	9086729	82183956
3002084	22	12507220	278	12508530	9505136	91689092
3002084	23	12912130	287	12508530	9910046	101599138
3002084	24	13303543	295.7	12508530	10301459	111900597

5 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

Социальная ответственность и охрана труда являются важнейшими составляющими любой деятельности, в особенности производственной, т.к. непосредственно связаны со здоровьем и жизнью человека.

Социальная ответственность - это сознательное отношение субъекта социальной деятельности к требованиям социальной необходимости, гражданского долга, социальных задач и, норм и ценностей, понимание осуществляемой деятельности для определенных социальных групп и личностей, для социального прогресса общества.

Актуальностью исследования «Разработка информационной системы мониторинга окружающей среды на примере г. Алматы, Казахстан» является то, информационная система позволяет рассчитывать концентрацию загрязняющих веществ, рассчитать комплексный индекс загрязнения промышленного объекта, хранить данные, анализировать состояние приземного воздушного слоя, прогнозировать возможную концентрацию загрязняющих веществ на определенных участках в определенное время года.

5.1 Правовые и организационные вопросы обеспечения безопасности

5.1.1. Специальные (характерные для проектируемой рабочей зоны) правовые нормы трудового законодательства

Согласно ТК РФ, N 197 -ФЗ работник аудитории 220, 11 корпуса ТПУ имеет право на:

- рабочее место, соответствующее требованиям охраны труда;
- обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний в соответствии с федеральным законом;
- отказ от выполнения работ в случае возникновения опасности для его жизни и здоровья вследствие нарушения требований охраны труда, за исключением случаев, предусмотренных федеральными законами, до устранения такой опасности;• обеспечение средствами индивидуальной и

коллективной защиты в соответствии с требованиями охраны труда за счет средств работодателя;

- внеочередной медицинский осмотр в соответствии с медицинскими рекомендациями с сохранением за ним места работы (должности) и среднего заработка во время прохождения указанного медицинского осмотра;

Рабочее место в аудитории 220, 1 корпуса ТПУ должно соответствовать требованиям ГОСТ 12.2.032 - 78. Оно должно занимать площадь не менее 4,5 м², высота помещения должна быть не менее 4 м, а объем - не менее 20 $м^3$ на одного человека. Высота над уровнем пола рабочей поверхности, за которой работает оператор, должна составлять 720 мм. Оптимальные размеры поверхности стола 1600 х 1000 кв. мм. Под столом должно иметься пространство для ног с размерами по глубине 650 мм. Рабочий стол должен также иметь подставку для ног, расположенную под углом 15° к поверхности стола. Длина подставки 400 мм, ширина - 350 мм. Удаленность клавиатуры от края стола должна быть не более 300 мм, что обеспечит удобную опору для предплечий. Расстояние между глазами оператора и экраном видеодисплея должно составлять 40 - 80 см. Рабочее место должно быть скомпоновано так, чтобы все операции работника выполнялись в пределах зоны досягаемости моторного поля в вертикальной плоскости (Рисунок 5.1) и в горизонтальной плоскости (Рисунок 5.2).

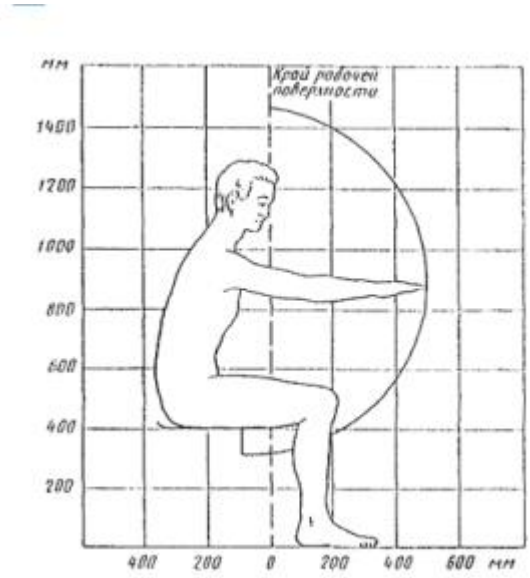


Рисунок 5.1 – Зона досягаемости моторного поля в вертикальной плоскости

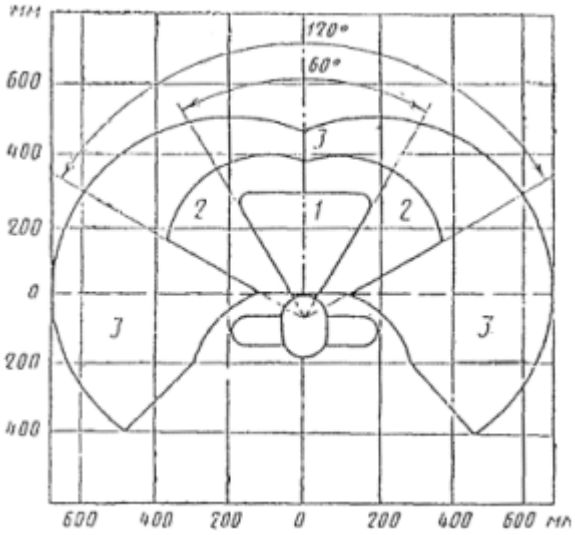


Рисунок 5.2 – Зона досягаемости моторного поля в горизонтальной плоскости.

1 - зона для размещения наиболее часто используемых предметов

2 - зона для размещения часто используемых предметов

3 - зона для размещения редко используемых предметов

Рабочее место сотрудника аудитории 220, 11 корпуса ТПУ соответствует требованиям ГОСТ 12.2.032 - 78.

5.2. Производственная безопасность

В учебной аудитории, имеются опасные и вредные производственные факторы. В таблице 5.1 приведены возможные опасные и вредные производственные факторы, влияющие на человека при исследовании.

5.2.1. Производственная безопасность

В таблице 5.1 приведены возможные опасные и вредные производственные факторы.

Таблица 5.1 – Опасные и вредные производственные факторы

Факторы	Нормативные документы
1.Отклонение показателей микроклимата	СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания [17]
2.Недостаточная освещенность рабочей зоны	СП 52.13330.2016 Естественное и искусственное освещение. [18]
3. Повышенный уровень электромагнитных излучений	ГОСТ 12.1.006 - 84 ССБТ. Электромагнитные поля радиочастот. Общие требования безопасности. [19]
4. Повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека	ГОСТ 12.1.038-82 ССБТ. Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов. [20]

Отклонение параметров микроклимата

Источник возникновения фактора – энергозатраты организма 151 – 200 (ккал/ч) [17], связанные с постоянной ходьбой, перемещением мелких (до 1 кг) изделий или предметов в положении стоя или сидя и требующие определенного физического напряжения, при работе в аудитории ТПУ.

При неблагоприятном микроклимате нарушается сердечный ритм, снижается концентрация, колеблется температура тела.

Поддержание оптимальных величин температуры и влажности воздуха рабочей зоны обеспечивается приточной и вытяжной вентиляцией, которая рассчитана на поглощение тепловыделений. Вентиляторы включаются автоматически при достижении температуры в помещении 33°С и отключаются при температуре 25 °С.

Допустимые параметры микроклимата приведены в таблице 5.2.

Таблица 5.2 Допустимые параметры микроклимата аудитории

Период года	Категория работ	Температура воздуха,°С		Температура поверхностей, t°С	Относительная влажность воздуха, φ%	Скорость движения воздуха, м/с
		Диапазон ниже Оптимальных величин $t^{o}_{опт}$	Диапазон выше оптимальных Величин $t^{o}_{опт}$			
Холодный	Iб	17,0 - 18,9	21,1 - 23,0	20 - 24,0	40 - 60	0,1
Теплый	Iб	16,0 - 18,9	22 - 24,0	21 - 25,0	40 - 60	0,1

В аудитории проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ЭВМ.

Недостаточная освещенность рабочей зоны

Аудитория 41 имеет подвальный тип расположения поэтому возможности организации естественного освещения.

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300 - 500 лк [18]. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк .

В качестве источников света применяются светодиодные светильники или металлогалогенные лампы (используются в качестве местного освещения) [18].

Повышенный уровень электромагнитных излучений

Источником повышенного уровня электромагнитных излучений являются мониторы наблюдения за технологическим процессом.

При длительном воздействии ЭМИ различных диапазонов длин волн при умеренной интенсивности (выше ПДУ) могут появиться головные боли, повышение или понижение давления, урежение пульса, изменение проводимости в сердечной мышце, нервно-психические расстройства, быстрое развитие утомления.

Нормирование ЭМИ радиочастотного диапазонапроводится по ГОСТ 12.1.006 - 84 для производственной среды. Для защиты от влияния ЭМИ рекомендуется отдых и защитные очки [19].

Поражение электрическим током

Для предотвращения поражения электрическим током, где размещаются рабочее место с ЭВМ в аудитории 220, 11 корпуса ТПУ, оборудование оснащено защитным заземлением, занулением в соответствии с техническими требованиями по эксплуатации [21]. Напряжение для питания ЭВМ 220 В, для серверного оборудования 380 В. По опасности поражения электрическим

током помещение 220, 11 корпуса ТПУ относится к первому классу – помещения без повышенной опасности. [13].

Основными непосредственными причинами электротравматизма, являются: 1) прикосновение к токоведущим частям электроустановки, находящейся под напряжением в случае пробоя изоляции; 2) прикосновение к металлическим конструкциям электроустановок, находящимся под напряжением; 3) ошибочное включение электроустановки или несогласованных действий обслуживающего персонала; 4) поражение шаговым напряжением.

Основными техническими средствами защиты, согласно ПУЭ [39], являются защитное заземление, автоматическое отключение питания, устройства защитного отключения, изолирующие электрозащитные средства, знаки и плакаты безопасности. Указанные средства защиты обеспечивают защиты от поражения электрическим током в аудитории 41, 4 корпуса ТПУ.

Рассчитано защитное заземление для шкафов релейной защиты и серверного оборудования, которое находится в аудитории 241, 4корпуса ТПУ.

1. В качестве заземляющего устройства (вертикальные электроды) используем стальные трубы диаметром d = 55 мм, в качестве соединяющего элемента – стальная полоса шириной b = 50 мм.

2. Сопротивлению грунта в районе размещения установки или устройства.

Таблица 5.3 - Исходные данные для расчета

Вид заземления	контурное
Длина заземлителя l, м	2,7
Глубина заземлителя в грунте h, м	0,65
Сезонный коэффициент $Кс$	2,0
Удельное сопротивление земли ρ, Ом·м	70
Диаметр d, мм	55
Ширина соединительной полоски b, мм	50
Допустимое сопротивление системы заземления по ПУЭ $Rз.у.$, Ом	4
Уровень напряжения, В	220-380
Коэффициент экранирования	0,58

3. Величина электрического сопротивления растекания тока в грунт с одиночного заземлителя:

$$R_З = 0,366 \cdot \frac{\rho \cdot K_С}{l}(\lg \cdot \frac{2 \cdot l}{d} + 0,5 \lg \cdot \frac{4 \cdot t + l}{4 \cdot t - l}) =$$

$$0,366 \cdot \frac{70 \cdot 2}{2,7}(\lg \cdot \frac{2 \cdot 2,7}{0,055} + 0,5 \lg \cdot \frac{4 \cdot 2 + 2,7}{4 \cdot 2 - 2,7}) = 38,51 \text{ Ом},$$

где,

$\rho = 70$ Ом -удельное сопротивление грунта,

$K_С = 2$ - коэффициент сезонности,

l=2,7 м – длина заземлителя,

d=0,055 м – диаметр заземлителя

t = h+0.5l=0,65+0,5·2,7=2 м

4. Число заземлителей без взаимных помех, получаемых друг от друга, без так называемого явления «экранирования»:

$$n' = \frac{R_З}{R_{З.У}} = \frac{38,51}{4} = 9,62 \approx 10.$$

5. Число заземлителей с коэффициентом экранирования:

$$n = \frac{n'}{\eta_З} = \frac{10}{0,58} = 17,24 \approx 18.$$

Принимаем расстояние между заземлителями *a* = *l* = 2,7 м.

6. Длина соединительной полосы:

$$l_п = 1,05 \cdot n \cdot a = 1,05 \cdot 18 \cdot 2,7 = 51 \text{ м}.$$

7. Значение сопротивления растекания тока с соединительной полосы:

$$R_З = 0,366 \cdot \frac{\rho \cdot K_С}{l}(\lg \cdot \frac{2 \cdot l_п^{\,2}}{b \cdot h}) = 0,366 \cdot \frac{70 \cdot 2}{51}(\lg \cdot \frac{2 \cdot 51^{\,2}}{0,05 \cdot 0,65}) = 5,1 \text{ Ом}.$$

8. Полное сопротивление системы заземления:

$$R_{ЗУ} = \frac{R_З \cdot R_П}{R_З \cdot \eta_П + R_З \cdot \eta_З \cdot n} = \frac{38,51 \cdot 5,1}{38,51 \cdot 0,51 + 5,1 \cdot 0,58 \cdot 18} = 2,63 \text{ Ом},$$

где,

$\eta_{П} = 0,51$ - коэффициент экранирования полосы.

Таким образом, сопротивление *Rзу* = 2,63 Ом не превышает 4 Ом. Следовательно, диаметр заземлителя *d* = 55 мм при числе заземлителей *n* = 18 является достаточным для обеспечения защиты при контурной схеме расположения заземлителей.

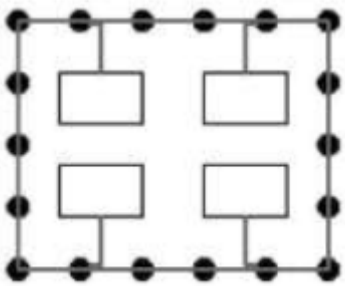


Рисунок 5.1 – Схема полученного контурного заземления

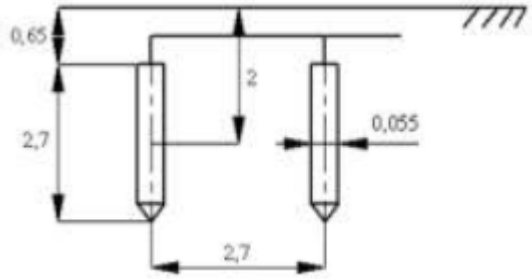


Рисунок 5.2 – Схема расположения заземлителей

Разработанные мероприятия и расчеты обеспечивают безопасную эксплуатацию электроустановок в аудитории 41, 4 корпуса ТПУ.

5.3 Экологическая безопасность

Влияние на атмосферу и гидросферу

При использовании компьютера для разработки потребляется электроэнергия. Выработка электроэнергии осуществляется на ТЭЦ и ГРЭС. При выработке электроэнергии за счет сжигания топлива в воздух поступают различные загрязнения.

Для уменьшения выбросов на ТЭС применяют различные фильтра и разные технологии сжигания.

На гидросферу также влияет по большей части выработка тепла и электроэнергии. Гидросфера загрязняется обмывочными водами котлов, замасленными водами, сброс тепловых потоков.

Чтобы минимизировать сброс обмывочных и замасленных вод в гидросферу, чаще всего используют очистку и масляные ловушки.

Влияние на литосферу

Процесс исследования представляет из себя работу с информацией, такой как технологическая литература, статьи, ГОСТы и нормативно техническая документация, а также разработка математической модели с помощью различных программных комплексов. Таким образом процесс исследования имеет влияние негативных факторов на окружающую среду. Таких как отходы – использованная бумага, использованные шариковые ручки.

Использованная бумага и пластиковые шариковые ручки утилизируется, как вторичное сырье – изготовление картона, пластиковой тары и т.д. Процесс утилизации негативно влияет на атмосферу, выделяя в нее углекислый газ.

Существующая система фильтрации не может на 100% избавить от выделения вредных веществ в атмосферу. Что касается методов по защите литосферы, то используются следующие методы:

- Энергосбережение;
- Сортировка мусора.

5.4 Безопасность в чрезвычайных ситуациях

5.4.1. Анализ вероятных ЧС, которые может инициировать объект исследований и обоснование мероприятий по предотвращению ЧС

При проектировании и эксплуатации программного продукта возможны следующие чрезвычайные ситуации в рабочей зоне:

• техногенные - производственные аварии, пожары;

Наиболее типичным и вероятным видом чрезвычайной ситуации является пожар в рабочей зоне. Наиболее вероятные причины пожара связаны с неисправностью или ненадлежащей эксплуатации электроприборов – короткое замыкание или перегрузки по току, нарушение работником правил эксплуатации. В соответствии с СП 12.13130.2009 [22] помещение лаборатории относится к категории В1 зона П-IIа. Перед тушением необходимо отключить общее электроснабжение. Для тушения можно использовать огнетушители разного типа, но, если отключить электроснабжение не удается, необходимо применять только порошковые и углекислотные. Требования к эксплуатации огнетушителей приведены в СП 9.13130.2009 [22].

5.4.2. Меры по предупреждению возникновения пожара

При проведении исследований наиболее вероятной ЧС является возникновение пожара в помещении 220, 11 корпуса ТПУ. Пожарная безопасность должна обеспечиваться системами предотвращения пожара и противопожарной защиты, в том числе организационно-техническими мероприятиями. Основные источники возникновения пожара:

1) Неработоспособное электрооборудование, неисправности в проводке, розетках и выключателях. Для исключения возникновения пожара по этим причинам необходимо вовремя выявлять и устранять неполадки, а также проводить плановый осмотр электрооборудования.

2) Электрические приборы с дефектами. Профилактика пожара включает в себя своевременный и качественный ремонт электроприборов.

3) Перегрузка в электроэнергетической системе (ЭЭС) и короткое замыкание в электроустановке.

Под пожарной профилактикой понимается обучение пожарной технике безопасности и комплекс мероприятий, направленных на предупреждение

пожаров.

Пожарная безопасность обеспечивается комплексом мероприятий:

- обучение, в т.ч распространение знаний о пожаробезопасном поведении (о необходимости установки домашних индикаторов задымленности и хранения зажигалок и спичек в местах, недоступных детям);
- пожарный надзор, предусматривающий разработку государственных норм пожарной безопасности и строительных норм, а также проверку их выполнения;
- обеспечение оборудованием и технические разработки (установка переносных огнетушителей и изготовление зажигалок безопасного пользования).

В соответствии с ТР «О требованиях пожарной безопасности» для административного жилого здания требуется устройство внутреннего противопожарного водопровода.

Согласно ФЗ-123, НПБ 104 - 03 «Проектирование систем оповещения людей о пожаре в зданиях и сооружениях» для оповещения о возникновении пожара в каждом помещении должны быть установлены дымовые оптикоэлектронные автономные пожарные извещатели, а оповещение о пожаре должно осуществляться подачей звуковых и световых сигналов во все помещения с постоянным или временным пребыванием людей.

Аудитория 41, 4 корпуса ТПУ оснащена первичными средствами пожаротушения: огнетушителями ОУ-3 1шт., ОП-3, 1шт. (предназначены для тушения любых материалов, предметов и веществ, применяется для тушения ПК и оргтехники, класс пожаров А, Е.).

Согласно НПБ 105-03 помещение, предназначенное для проектирования и использования результатов проекта, относится к типу П - 2а.

5.4.3 Действия в случае возникновения пожара

В случае возникновения пожара необходимо:

• спокойно оценить ситуацию и принять срочные меры по предотвращению распространения огня;

• вызвать пожарных, сообщив точный адрес места возгорания и ФИО, вызывающего;

• отключить общее электроснабжение;

• попытаться потушить пожар при помощи первичных средств пожаротушения;

• если самостоятельно справиться с огнем не удаётся, следует включить сигнал пожарной тревоги и, согласно плану, приступить к эвакуации;

• встретить прибывшую пожарную команду и обеспечить для неё беспрепятственный доступ и пути подъезда к месту пожара.

Вывод по разделу

В результате выполнения задания раздела «Социальная ответственность и ресурсосбережение» ВКР были выявлены и проанализированы вредные факторы при разработке алгоритма.

Были установлены правовые и организационные вопросы обеспечения безопасности, характерные для рабочей зоны.

Проработаны организационные мероприятия при компоновке рабочей зоны.

Разработаны мероприятия по снижению воздействия вредных и опасных факторов.

Был рассмотрен характер воздействия исследуемого решения на окружающую среду. Были выявлены предполагаемые источники загрязнения окружающей среды, возникшие в результате реализации предлагаемых в магистерской диссертации решений.

Можно сделать вывод, что создание магистерская диссертация не является экологически безвредным действием. Так как процесс сопровождается созданием отходов от проектной деятельности.

А также процесс написания магистерской диссертации не является абсолютно безвредным для человека. Так как в процессе написания магистерской диссертации человека сопровождают такие вредные факторы, как электромагнитное излучение от ПК и плохая освещенность рабочей зоны.

Даны общие рекомендации по безопасности, соблюдая которые можно, не только сохранить здоровье, но и увеличить эффективность.

Список публикаций студента

1. Алмасбекулы Б. Модели прогноза уровня загрязнения атмосферного воздуха г. Алматы /Б.Алмасбекулы ; науч. рук. М. Е. Семенов // «Научный Аспект», № 06/23-18-001(принято в печать)

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Kumar, S., & Jain, PK (2018). Air quality forecasting using ARIMA model: a case study of Delhi, India. Environmental Science and Pollution Research, 25(22), 21730-21738.

2. Mounika , M., & Sastry , TV (2019). Forecasting air pollution in Visakhapatnam, India using the ARIMA model. Journal of Environmental Management and Tourism, 10(4), 814-823.

3. Pires , A.P., Rodrigues, E.B., & Martins, F.G. (2018). Forecasting PM10 air pollution using seasonal decomposition and ARIMA models. Atmospheric Pollution Research, 9(2), 324-331.

4. Astitha , M., Kallos , G., & Katsafados , P. (2020). Combining ARIMA and machine learning methods for improved air quality forecasting. Atmospheric Research, 236, 104792.

5.Бюро национальной статистики Агентства по стратегическому планированию и реформам Республики Казахстан [Электро. ресурс]- URL-https://stat.gov.kz/ecologic/air_pollutant_emissions?lang=ru

6.А.К.Муртазов Экологический мониторинг [Электрон. ресурс]- URL-http://www.rsu.edu.ru/files/e-learning/murtazov_eco_mon.pdf(дата обращения 09.05.2023)9. BoxG.E.P., JenkinsG., Time Series Analysis, Forecasting and Control, Holden-Day, San Francisco, CA, 1970.–453 с

7. Fisher, R. A. (1921). "On the probable error of a coefficient of correlation deduced from a small sample." Metron, 1(3), 3-32.

8. Box G.E.P., Jenkins G., Time Series Analysis, Forecasting and Control, Holden-Day, San Francisco, CA, 1970. – 453 с.

9. Claudio Guarnaccia, Julia Griselda Ceron Breton, Rosa Maria Ceron Breton, Carmine Tepedino, Joseph Quartieri, and Nikos E. Mastorakis. ARIMA models

application to air pollution data in Monterrey, Mexico// AIP Conference Proceedings 1982, 020041, 2018; https://doi.org/10.1063/1.5045447.

10. Muhammad Hisyam Lee, Nur Haizum Abd. Rahman, Suhartono, Mohd Talib Latif, Ma-ria Elena Nor and Nur Arina Bazilah Kamisan. Seasonal ARIMA for Forecasting Air Pollution Index: A Case Study //American Journal of Applied Sciences 9 (4): 570-578, 2012 ISSN 1546-9239, 2012.

11. Abhilash M.S.K., Thakur A., Gupta D., Sreevidya B. Time Series Analysis of Air Pol-lution in Bengaluru Using ARIMA Model. In: Perez G., Tiwari S., Trivedi M., Mish-ra K. (eds) Ambient Communications and Computer Systems. Advances in Intelli-gent Systems and Computing, vol 696. Springer, Singapore, 2018.

12. MacKinnon, J.G. Critical Values for Cointegration Tests // Queen's University, Dept of Economics, Working Papers, 2010. http://ideas.repec.org/p/qed/wpaper/1227.html.

13. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. От 27.12.2018)

14. Федеральный закон от 29.11.2010 № 326-ФЗ (ред. От 24.02.2021) «Об обязательном медицинском страховании в Российской Федерации»

15. ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования.

16. ГОСТ 12.0.003-2015 ССБТ. Опасные и вредные производственные факторы. Классификация.

17. СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания

18. СП 52.13330.2016 Естественное и искусственное освещение.

19. ГОСТ 12.1.006-84 ССБТ. Электромагнитные поля радиочастот. Общие требования безопасности.

20. ГОСТ 12.1.038-82 ССБТ. Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов.

21. ПУЭ: правила устройства электроустановок. Издание 7.

22. СП 12.13130.2009 Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности

Приложение А

(справочное)

Development of an information system for environmental monitoring in Almaty, Kazakhstan

Студент

Группа	ФИО	Подпись	Дата
0ВМ12	Алмасбекулы		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Семенов М.Е	к. ф.-м. н.		

Консультант-лингвист отделения иностранных языков ШБИП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель отделения иностранных языков	Смирнова У.А			

Introduction

Air quality forecasting has been an active area of research in recent years due to growing concern about environmental degradation and its impact on public health. Time series analysis, in particular, is widely used due to its effectiveness in capturing the time dependence of environmental data.

A number of studies have used the ARIMA (AutoRegressive integrated Moving Average). For example, Kumar and Jain (2018) used the ARIMA model to predict PM2.5 levels in Delhi, India. Similarly, Mounika and Sastry (2019) successfully applied ARIMA to forecast air pollution in Visakhapatnam , India.

Other researchers have extended the application of traditional ARIMA models. Pires et al. (2018) presented a seasonal trend decomposition procedure based on Loess (STL) and ARIMA for PM10 forecasting in Portugal.
Meanwhile, Astitha et al. (2020) combined ARIMA with other machine learning methods to improve forecasting accuracy air quality in Hartford , USA.

The above studies demonstrate the potential of time series analysis and in particular ARIMA for air quality forecasting. This project aims to apply these methods to predict air quality parameters in Almaty, Kazakhstan, contributing to global environmental monitoring and management efforts.

Research methods

The main objective of this study is to develop an environmental monitoring information system focused on air quality in Almaty, Kazakhstan. This system will use time series analysis, in particular the ARIMA model, to predict future air quality parameters based on historical data. This predictive ability will provide valuable information about the state of air quality in Almaty and take proactive measures to reduce the risks of air pollution.

Data collection method

The air quality data used in this study was obtained from aqicn.org. Aqicn.org provides up-to-date, real-time air quality information for many locations around the world, including Almaty, Kazakhstan, as shown in Figure 1. The data include measurements of various pollutants, which are the most important indicators of overall air quality.

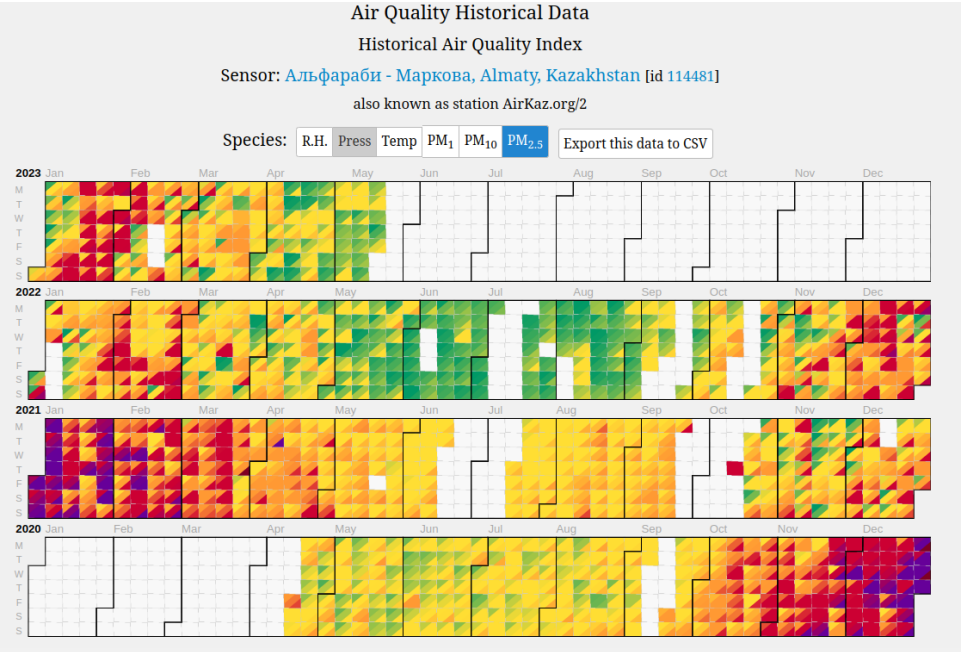


Figure 1. One of the sensors located in Almaty

Selenium , a powerful tool for managing web browsers with programs and automating browser tasks, was used to automate the process of obtaining data . Selenium was used to navigate the website, select the appropriate parameters and time frames, and finally download the air quality data. Using Selenium for web scraping ensures efficient and systematic data collection.

Once loaded, the data were imported into pandas DataFrame for further processing and analysis. The resulting DataFrame includes columns for the date of

observation and indicators for four pollutants (PM2.5, PM10, NO2 and CO) starting on February 10, 2020 and ending on May 19, 2023.

Description of data

The data set used in this study includes air quality measurements taken in Almaty over a period of more than three years, from February 10, 2020 to May 19, 2023. The data records include measurements of various air pollutants: PM2.5 (fine particulate matter), PM10 (large particulate matter), NO2 (nitrogen dioxide), and CO (carbon monoxide), as shown in Figure 2. The dataset was generated in pandas DataFrame and contains a total of 1180 entries.

	date	pm25	pm10	no2	co
0	2023-05-01	39.0	26.0	17.0	4.0
1	2023-05-02	49.0	23.0	17.0	2.0
2	2023-05-03	46.0	12.0	19.0	3.0
3	2023-05-04	35.0	10.0	23.0	3.0
4	2023-05-05	41.0	15.0	22.0	3.0

Figure 2. Table of collected data

The DataFrame is structured like this:

- data : This column records the date of each observation, formatted as a datetime64 object. It ranges from February 10, 2020 to May 19, 2023.

- pm25: This column records the PM2.5 levels for each date, represented as a floating point number. This column is missing 183 values.
- pm10: This column records the PM10 levels for each date, also represented as a floating point number. This column is missing 324 entries.
- no2: This column records the NO2 levels for each date, again represented as a floating point number. It has the most missing entries, with a total of 616.
- co : This column records the CO levels for each date, represented as a floating point number. This column is missing 275 entries.

The DataFrame shows that there are 1180 rows (observations) and 5 columns (date and four air quality parameters). The presence of missing values in columns is a problem for the data analysis process and requires appropriate processing techniques to ensure a reliable prediction.

One possible strategy for handling missing values is interpolation, which involves estimating missing values based on other available data. Another common strategy is imputation , where missing values are filled in with statistical measures such as mean or median. The choice of strategy depends on the nature and distribution of the data.

Methods

The ARIMA model, which stands for " Auto regressive integrated Moving Average" is a popular model for time series analysis and forecasting. ARIMA is a combination of autoregressive (AR) and moving average (MA) models, as well as an integrated component (I) to account for data non-stationarity. In mathematical terms, an ARIMA model is defined by three parameters: (p, d, q):

-p порядок части авторегрессии.

-d степень первого дифференцирования

-q порядок части скользящего среднего

The AR part of the model ($AR(p)$is given by the following equation:

$$AR(p): Y_t = c + \alpha 1 Y_{(t-1)} + \alpha 2 Y_{(t-2)} + \cdots + \alpha p Y_{(t-p)} + \varepsilon_t$$

Where:

- Y_t is the value of the time series at time t.

- c is a constant member of the model representing the bias or mean of the time series.

- α is the model parameters that determine the weights of the previous values of the time series.

-ε_t is the error term at time t .

Thus, the AR(p) model uses p previous values of the time series (Y) with weights α to predict the current value. The α coefficients determine the contribution of each previous value to the prediction, and the larger p, the more previous values are used for prediction.

The AR part of the model reflects autocorrelation in the data, that is, the dependence of the current value on the previous values of the time series. This allows you to take into account the trends and patterns present in the data and use them to predict future values.

MA - part of the model $MA(q)$is given by the following equation:

$$MA(q): Y_t = \mu + \varepsilon_t + \theta 1\varepsilon_{(t-1)} + \theta 2\varepsilon_{(t-2)} + \cdots + \theta q\varepsilon_{(t-q)}$$

Where:

- Y_t is the value of the time series at time t.
- μ mean value of the time series
- θ model parameter that determines the contribution of the previous error terms to the forecast
- ε_t is the error term at time t .

Thus, the MA(q) model uses q previous error terms with weights θ to predict the current value. The coefficients θ determine the contribution of each previous error term to the prediction, and the larger q, the more previous error terms are used for prediction.

The MA part of the model reflects the correlation between the current value and previous error terms. This allows you to take into account the random component in the data and use it to predict future values.

The sign "*I*" in ARIMA indicates that the data have been differentiated to make it stationary. Differentiation involves subtracting the current value from the previous one and can help stabilize the mean of the time series by eliminating changes in the level of the time series and therefore eliminating (or reducing) the trend and seasonality. Once an ARIMA model has been defined and trained on historical data, it can be used to predict future values.

Correlation analysis

Before moving on to the ARIMA model, an analysis of correlations between variables was carried out. The correlation matrix below displays Pearson's correlation coefficients, which range from -1 to 1. A coefficient close to 1 indicates a strong positive correlation, while a coefficient close to -1 indicates a strong negative correlation.

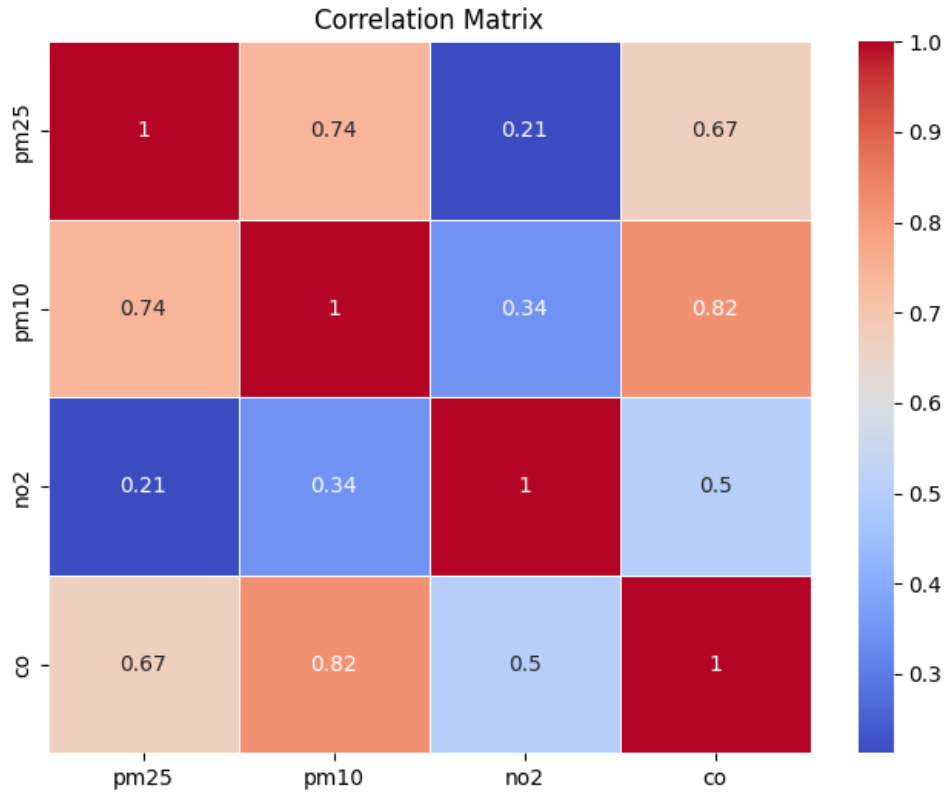


Figure 3. Heat map between targets

This table, shown in Figure 3, provides valuable insight into the relationship between the four pollutants. For example, there is a high positive correlation between PM2.5 and PM10 (0.74), indicating that high PM2.5 values often coincide with high PM10 values. Likewise, there is a strong positive correlation between PM10 and CO (0.82).

Simulation results

In the previous section, we described the ARIMA model prediction results for PM2.5, PM10, NO2, and CO. The following is a more detailed overview of these results:

PM2.5 prediction

The original ARIMA model for PM2.5 gave a root mean square error (RMSE) of 34.55278771221794. This model was then refined using GridSearchCV , an exhaustive search on given parameter values, to determine the most optimal parameters for the ARIMA model. The parameters (0, 0, 1) were found to be optimal, which led to a noticeable decrease in the RMSE value to 31.710, observed in Figure 4.

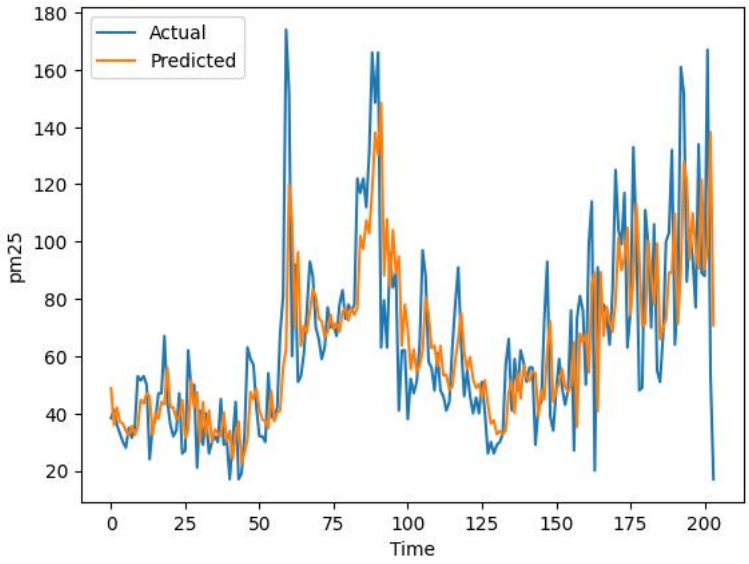


Figure 4. Customized Arima Model for PM 2.5

This improvement suggests that the predicted PM2.5 values generated by the tuned ARIMA model are closer to the actual observed values, making it a more reliable model for predicting PM2.5 levels.

PM10 prediction

For the PM10, the original ARIMA model gave an RMSE of 14.578876764783104. After adjusting the parameters of the ARIMA model using GridSearchCV , the results of which are shown in Figure 5, the optimal parameters turned out to be (0, 1, 2).

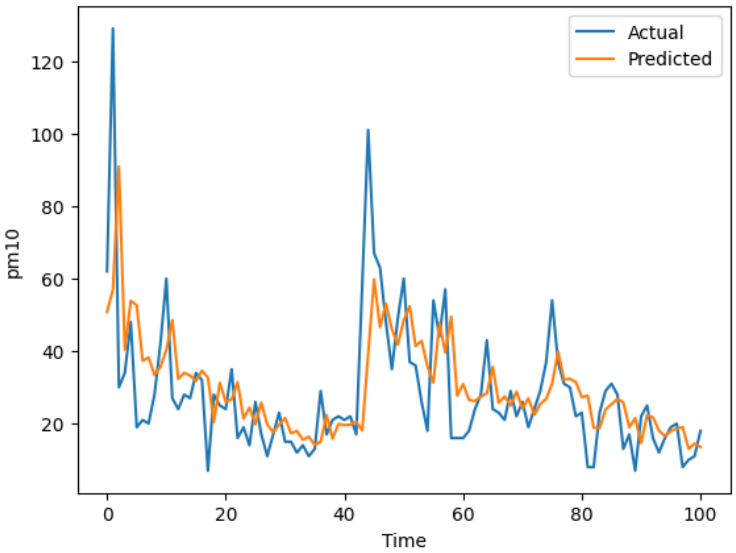


Figure 5. Customized Arima Model for PM 10

RMSE fell slightly to 14.374. Although the improvement was not significant, it still indicates that the adjusted ARIMA model provides a more accurate prediction of PM10 levels than the original model.

NO2 prediction

The original ARIMA model for NO2 gave RMSE 7.70907355319198. After tuning with GridSearchCV, the optimal parameters were (1, 0, 0), as shown in Figure 6. These changes resulted in a slight decrease in RMSE to 7.699, indicating a slight improvement in the model's ability to predict NO2 levels.

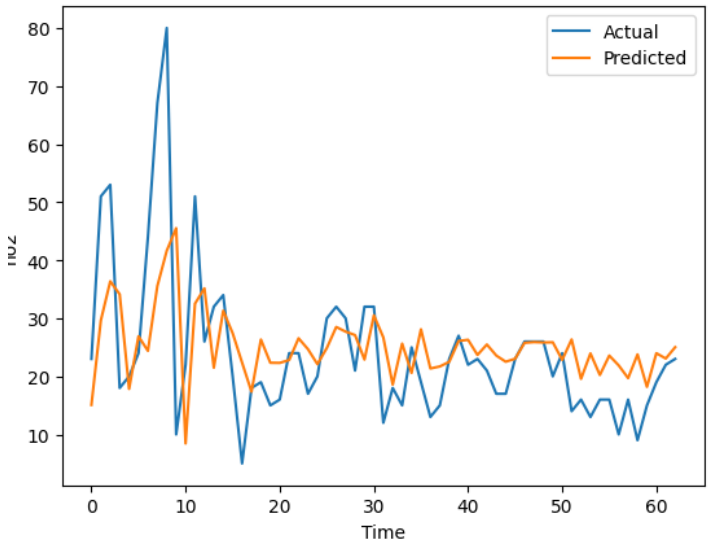


Figure 6. Customized Arima Model for NO2

CO prediction

For CO, the original ARIMA model gave RMSE 2.5015090544901737. After tuning with GridSearchCV, the optimal parameters were found as (0, 1, 2) and the RMSE dropped to 2.338 as shown in Fig. 7. This indicates a more accurate CO level prediction with the tuned ARIMA model.

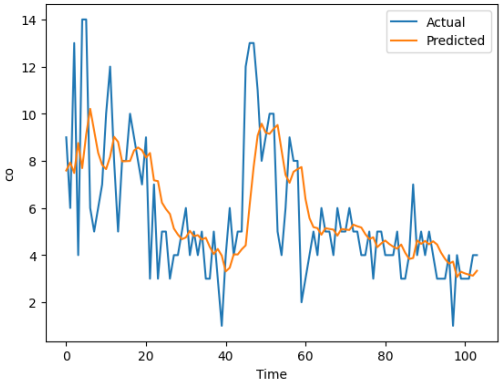


Figure 7. Adjusted Arima Model for CO

Overall, the results highlight the importance of model tuning when forecasting time series data. Even small improvements in RMSE values can have a significant impact on the quality and reliability of air quality forecasts. Future work could explore adding external regressors (exogenous variables), applying other forecasting models, or using ensemble methods to further improve forecast accuracy.

Conclusion

The aim of this study was to design and develop an environmental monitoring system to predict air quality in Almaty, Kazakhstan, focusing on four main pollutants: PM2.5, PM10, NO2 and CO. The study used the ARIMA model, a widely recognized statistical technique for time series forecasting. The original ARIMA models provided the basis for evaluating time series data. However, the performance of the models improved significantly after tweaking the parameters with GridSearchCV . Root mean square error (RMSE), a key assessment metric, has shown improved prediction accuracy for all four pollutants after model tuning. RMSE values have decreased for PM2.5, PM10, NO2, and CO, indicating better model fit and improved reliability of air quality forecasts. Although the improvements have not always been significant, the impact of the fine-tuning process cannot be underestimated. The results highlight the importance of careful model tuning when making time series forecasts, especially when forecasting air quality, which has significant implications for public health and environmental policy. Despite promising results, there is room for further research. Various models can be applied to this dataset, or ensemble methods can be used to potentially improve prediction accuracy. In addition, incorporating other external variables into the model, such as weather conditions or industrial activity data, can improve forecasting capabilities. This study represents an important step towards more accurate air quality forecasting in Almaty, Kazakhstan. However, air quality is a global issue. The methodology used here can be applied elsewhere, contributing to global efforts to monitor and control air pollution. The ultimate goal is to create a safer and healthier environment for people around the world.

Приложение Б

```
#!/usr/bin/env python
# coding: utf-8

# In[29]:


get_ipython().system(' pip install seaborn')


# In[30]:


import warnings
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
sns.set()
sns.set_style("whitegrid")


# Ignore all warnings
warnings.filterwarnings("ignore")

df = pd.read_csv("/kaggle/input/airkaz/AlmKazH.csv")

print(df.shape)
df.head()


# In[31]:


# Group by 'code' column and perform desired aggregation (e.g., mean of 'value')
grouped_df = df.groupby(['code']).agg({'date':['min', 'max'], 'value': ['max','mean','min', 'size']})

# Display the resulting grouped DataFrame
grouped_df


# In[32]:
```

```
kazhydromet_df = df.reset_index().drop('index', axis=1)
# Display the KazHydroMet DataFrame
kazhydromet_df.head()


# In[33]:


# Group by 'date' and 'code' and calculate the mean of 'value'
df_kaz = kazhydromet_df.groupby(['date', 'code'])['value'].mean().reset_index()
df_kaz.head()


# In[34]:


df_kaz.info()


# In[35]:


df_kaz.date.unique().size, df_kaz.code.unique()


# In[36]:


df_kaz['date'].min(), df_kaz['date'].max()


# In[37]:


grouped_df_kaz = df_kaz.groupby(['code']).agg({'value': ['max','mean','min',
'size']})
grouped_df_kaz


# In[38]:
```

```
# Create separate columns for each unique code with corresponding values
df_pivot = df_kaz.pivot_table(index='date', columns='code', values='value').reset_index()

# Display the resulting DataFrame
df_pivot.head()


# In[39]:


df_pivot.info()


# In[40]:


df_pivot = df_pivot.set_index('date')


# In[41]:


df_pivot.head()


# In[42]:


CO = df_pivot['CO'].to_frame().dropna()
NO2 = df_pivot['NO2'].to_frame().dropna()
PM10 = df_pivot['PM10'].to_frame().dropna()
PM2_5 = df_pivot['PM2.5'].to_frame().dropna()


# In[43]:


PM10.head()


# In[44]:
```

```
def df_maker(df):
    df = df.reset_index()
    df['date'] = pd.to_datetime(df['date']).dt.date
    df = df.groupby('date').mean()
    df = df.astype('int')
    df = df.reset_index()
    return df


# In[45]:


CO.round(0)


# In[46]:


CO = df_maker(CO).loc[:526, :]
NO2 = df_maker(NO2).loc[:526, :]
PM2_5 = df_maker(PM2_5).loc[:1033,:]
PM10 = df_maker(PM10).loc[:526, :]


# In[47]:


PM2_5


# In[48]:


def remove_outliers(df, threshold=3):
    for col in df.columns:
        # Calculate the z-scores for the column
        z_scores = np.abs((df[col] - df[col].mean()) / df[col].std())
        # Replace outliers with NaN
        df[col] = np.where(z_scores > threshold, np.nan, df[col])
    # Drop rows with NaN values
    df = df.dropna()
    return df
```

```
# In[49]:


for df in [CO, NO2, PM10, PM2_5]:
    print(df['date'].min(), df['date'].max())
    df.plot(figsize=(16,8))
    print('-'*40)


# In[50]:


CO.shape, NO2.shape, PM10.shape, PM2_5.shape


# In[51]:


merged_df = pd.merge(PM10, NO2, left_index=True, right_index=True)

df_pm10_no2_c0 = pd.merge(merged_df, CO, left_index=True, right_index=True)


# In[52]:


df_pm10_no2_c0 = df_pm10_no2_c0.reset_index()


# In[53]:


df_pm10_no2_c0.head()


# In[54]:


df_pm10_no2_c0['date']


# In[55]:
```

```
df_pm10_no2_c0['date'] = pd.to_datetime(df_pm10_no2_c0['date'])

# Create month feature
df_pm10_no2_c0['month'] = df_pm10_no2_c0['date'].dt.month

# Create day of the week feature
df_pm10_no2_c0['day_of_week'] = df_pm10_no2_c0['date'].dt.dayofweek
# df_pm10_no2_c0[] = df_pm10_no2_c0['date'].dt.hour

df_pm10_no2_c0 = df_pm10_no2_c0[['date','PM10', 'NO2', 'CO', 'month', 'day_of_week']]


# In[56]:


df_pm10_no2_c0.head()


# In[57]:


df_pm25 = PM2_5.reset_index()


# In[58]:


df_pm25['date'] = pd.to_datetime(df_pm25['date'])

# Create month feature
df_pm25['month'] = df_pm25['date'].dt.month

# Create day of the week feature
df_pm25['day_of_week'] = df_pm25['date'].dt.dayofweek
# df_pm25[] = df_pm25['date'].dt.hour

df_pm25 = df_pm25[['date','PM2.5', 'month', 'day_of_week']]


# In[59]:


df_pm25.info()
```

In[60]:

```
df_pm25.head()
```

In[61]:

```
df_pm10_no2_c0[['PM10', 'CO', 'NO2']].plot(figsize=(16,8));
plt.show();
```

In[62]:

```
df_pm10_no2_c0.head()
```

In[63]:

```
# remove outliers
threshold = 3
for df in [df_pm25, df_pm10_no2_c0]:
    for col in df.drop(['date','month', 'day_of_week'], axis=1).columns:
        # Calculate the z-scores for the column
        z_scores = np.abs((df[col] - df[col].mean()) / df[col].std())
        # Replace outliers with NaN
        df[col] = np.where(z_scores > threshold, np.nan, df[col])
    # Drop rows with NaN values
    df = df.dropna()
```

In[64]:

```
df_pm10_no2_c0.shape, df_pm25.shape
```

In[65]:

```
df_pm10_no2_c0.set_index('date')[['PM10', 'CO', 'NO2']].plot(figsize=(16,8));
df_pm25.set_index('date')[['PM2.5']].plot(figsize=(16,8))


# In[66]:


df_pm10_no2_c0.head()


# In[67]:


get_ipython().system(' pip install statsmodels')


# In[68]:


from statsmodels.tsa.stattools import adfuller


# In[69]:


# df_pm25 ['date', 'PM2.5', 'month', 'day_of_week']
# df_pm10_no2_c0 ['date', 'PM10', 'CO', 'NO2', 'month', 'day_of_week']


# In[70]:


# df_pm10_no2_c0 = df_pm10_no2_c0.dropna()
# df_pm10_no2_c0.info()


# In[71]:


def plotBox(df):

    for col in df.drop(['date', 'month', 'day_of_week'], axis=1).columns:
        fig = plt.figure(figsize=(15, 9))
```

```
        sns.boxplot(df, x='month', y=col)


        # Customize the plot if needed
        plt.title(f'Monthly Box Plot of {col}')
        plt.xlabel('Month')
        plt.ylabel(col)

        # Display the plot
        plt.show()


# In[72]:


df_pm10_no2_c0.head()


# In[73]:


df_pm10_no2_c0 = df_pm10_no2_c0.dropna()
df_pm25 =df_pm25.dropna()
df_pm10_no2_c0.shape, df_pm25.shape


# In[74]:


df_pm10_no2_c0 = df_pm10_no2_c0.loc[:526, :]


# In[75]:


df_pm25=df_pm25.loc[:1033,:]


# In[76]:


plotBox(df_pm25)
```

In[77]:

```
plotBox(df_pm10_no2_c0)
```

In[78]:

```
df_pm10_no2_c0.head()
```

In[79]:

```
import pandas as pd
import numpy as np

# # Assuming you have DataFrames named df_pm25 and df_pm10_no2_c0

# # Function to remove outliers using z-score
# def remove_outliers_zscore(df, columns):
#     z_scores = np.abs((df[columns] - df[columns].mean()) / df[columns].std())
#     df_no_outliers = df[(z_scores < 3).all(axis=1)]
#     return df_no_outliers

# # Function to remove outliers using IQR
# def remove_outliers_iqr(df, columns):
#     Q1 = df[columns].quantile(0.25)
#     Q3 = df[columns].quantile(0.75)
#     IQR = Q3 - Q1
#     lower_bound = Q1 - 1.5 * IQR
#     upper_bound = Q3 + 1.5 * IQR
#         df_no_outliers = df[~((df[columns] < lower_bound) | (df[columns] > upper_bound)).any(axis=1)]
#     return df_no_outliers

# # Specify the columns to remove outliers from
# pm25_columns = ['PM2.5']
# PM10_co_no2_columns = ['PM10', 'CO', 'NO2']

# # Remove outliers using z-score
# df_pm25_no_outliers = remove_outliers_zscore(df_pm25, pm25_columns)
```

```
# df_pm10_no2_c0 = remove_outliers_zscore(df_pm10_no2_c0, PM10_co_no2_columns)

# Remove outliers using IQR
# df_pm25_no_outliers = remove_outliers_iqr(df_pm25, pm25_columns)
# df_pm10_no2_c0 = remove_outliers_iqr(df_pm10_no2_c0, PM10_co_no2_columns)


# In[80]:
df_pm25['date'].min(), df_pm25['date'].max()
# In[81]:
df_pm10_no2_c0['date'].min(), df_pm10_no2_c0['date'].max()
# In[82]:
df_pm10_no2_c0.to_csv('df_pm10_no2_co.csv')
# In[83]:
df_pm25.to_csv('df_pm25.csv', )
# In[84]:
# # Assuming you have the DataFrames df_pm25 and df_pm10_no2_c0 after removing outliers
# # Calculate the percentage of data coverage for each DataFrame
# percent_coverage_pm25 = len(df_pm25_no_outliers) / len(df_pm25) * 100
# percent_coverage_PM10_co_no2 = len(df_pm10_no2_c0) / len(df_pm10_no2_c0) * 100
# # Display the percentage of data coverage
# print(f"Percentage of data coverage for PM2.5: {percent_coverage_pm25:.2f}%")
# print(f"Percentage of data coverage for PM10, CO, NO2: {percent_coverage_PM10_co_no2:.2f}%")
# # Stationarity checking
#
#
# Certainly! Here are the LaTeX formulas for the ADF and KPSS tests:
#
# Augmented Dickey-Fuller (ADF) Test:
# $$
# \Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \delta_2 \Delta y_{t-2} + \ldots + \delta_{p-1} \Delta y_{t-p+1} + \varepsilon_t
# $$
#
# Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test:
# $$
# y_t = \mu_t + \varepsilon_t
# $$
#
```

```
#
# $$
# H_0: \text{The time series is stationary (vs. } H_1: \text{The time series is non-stationary)}
# $$
#
# You can copy these formulas and convert them into your desired format using a LaTeX converter or incorporate them into your LaTeX document within appropriate math environments.
#
# Please note that the formulas are already in LaTeX format, so you can directly use them in your LaTeX document.

# In[85]:


import pandas as pd
from statsmodels.tsa.stattools import adfuller, kpss

def check_stationarity(df, series):
    print('-'*60,series)
    # ADF test
    result_adf = adfuller(df[series])

    adf_statistic = result_adf[0]
    adf_pvalue = result_adf[1]
    adf_critical_values = result_adf[4]

    # Determine stationarity based on test results
    is_stationary_adf = adf_pvalue < 0.05

    # Print the results
    print(f'ADF Statistic: {adf_statistic}')
    print(f'ADF p-value: {adf_pvalue}')
    print('ADF Critical Values:')
    for key, value in adf_critical_values.items():
        print(f'   {key}: {value}')


    print("stationarity test results \n")
    # Determine stationarity
    if is_stationary_adf:
        print("The time series is stationary based on the ADF test.")
    else:
```

```
        print("The time series is non-stationary based on the ADF test.")
    print("Критерий Дики-Фуллера: p=%f" % result_adf[1])


# In[86]:


from sklearn.metrics import mean_absolute_error, mean_squared_error

import statsmodels.formula.api as smf
import statsmodels.tsa.api as smt
import statsmodels.api as sm
import scipy.stats as scs
from scipy.optimize import minimize


# In[87]:


def tsplot(df, series_name, lags=None, figsize=(16, 7), style='bmh'):
    y = pd.Series(df[series_name])
    with plt.style.context(style):
        fig = plt.figure(figsize=figsize)
        layout = (2, 2)
        ts_ax = plt.subplot2grid(layout, (0, 0), colspan=2)
        acf_ax = plt.subplot2grid(layout, (1, 0))
        acf_ax.set_ylim((-0.25, 1.1))

        pacf_ax = plt.subplot2grid(layout, (1, 1))
        pacf_ax.set_ylim((-0.25, 1.1))

        y.plot(ax=ts_ax)
        ts_ax.set_title(f'Time Series Analysis Plots - {series_name}')
        smt.graphics.plot_acf(y, lags=lags, ax=acf_ax, alpha=0.05)
        smt.graphics.plot_pacf(y, lags=lags, ax=pacf_ax, alpha=0.05)
        plt.tight_layout()

        check_stationarity(df, series_name)
    return


# In[88]:
```

```
df_pm10_no2_c0.shape, df_pm25.shape
```

In[89]:

```
smt.graphics.plot_acf(df_pm10_no2_c0['NO2'], lags=30, alpha=0.01);
```

In[90]:

```
def difference(dataset, column, interval=9):
    return dataset[column] - dataset[column].shift(interval)

def reverse_difference(dataset, diff_column, original_column, interval=5):
    reversed_series = dataset[original_column].shift(interval)
    reversed_series.iloc[interval:] = dataset[diff_column].cumsum().iloc[:-interval]
    return reversed_series
```

Исправляем ACF

In[91]:

```
df_pm25.head()
```

In[92]:

```
df_pm25["PM2.5_diff"] = df_pm25["PM2.5"].diff()
df_pm25 = df_pm25.dropna()
```

In[93]:

```
for col in df_pm25.drop(['date','month', 'day_of_week'], axis=1):
    print(col)
    tsplot(df_pm25, col, 12)
```

In[ ]:

In[94]:

```
# df_pm10_no2_c0 = df_pm10_no2_c0.copy()
```

In[95]:

```
# df_pm10_no2_c0['PM10'] = df_pm10_no2_c0['PM10'].diff()
# df_pm10_no2_c0['NO2'] = df_pm10_no2_c0['NO2'].diff()
# df_pm10_no2_c0['CO'] = df_pm10_no2_c0['CO'].diff()
# df_pm10_no2_c0 = df_pm10_no2_c0.dropna()
```

In[96]:

```
for i in ['PM10', 'NO2', 'CO']:
    # Original Series
    fig, (ax1, ax2, ax3) = plt.subplots(3)
    ax1.plot(df_pm10_no2_c0[i]); ax1.set_title('Original Series');
ax1.axes.xaxis.set_visible(False)
    # 1st Differencing
    ax2.plot(df_pm10_no2_c0[i].diff()); ax2.set_title('1st Order Differencing');
ax2.axes.xaxis.set_visible(False)
    # 2nd Differencing
    ax3.plot(df_pm10_no2_c0[i].diff().diff()); ax3.set_title('2nd Order Differencing')
    plt.show()
```

In[97]:

```
df_pm10_no2_c0['PM10_diff'] = df_pm10_no2_c0['PM10'].diff()
df_pm10_no2_c0['NO2_diff'] = df_pm10_no2_c0['NO2'].diff()
df_pm10_no2_c0['CO_diff'] = df_pm10_no2_c0['CO'].diff()
df_pm10_no2_c0 = df_pm10_no2_c0.dropna()
```

In[98]:

```
for col in df_pm10_no2_c0.drop(['date','month', 'day_of_week'], axis=1):
    print(col)
    tsplot(df_pm10_no2_c0, col, 40)
```

In[99]:

```
df_pm10_no2_c0.head()
```

In[100]:

```
df_pm10_no2_c0.to_csv("df_pm10_no2_c0.csv",index=False)
```

In[101]:

```
df_pm25.to_csv("df_pm25.csv",index=False)
```

4. ACF, PACF

Definitions and Meanings:
#
Autocorrelation Function (ACF):
The autocorrelation function measures the linear relationship between an observation in a time series and its lagged values.
The ACF at lag k, denoted by ρ_k, is the correlation between the series at time t and the series at time t-k, after removing the influence of intermediate observations.
#
Partial Autocorrelation Function (PACF):
The partial autocorrelation function measures the linear relationship between an observation in a time series and its lagged values, while controlling for the influence of other lags.

The PACF at lag k, denoted by ϕ_{kk}, is the correlation between the series at time t and the series at time t-k, after removing the influence of all lags between 1 and k-1.
#
#

warnings.filterwarnings("ignore")
warnings.filterwarnings("ignore")
the ACF shows seasonality, while the PACF shows the historical relationship between observations.
#
Here's a breakdown of their interpretations:
#
1. Autocorrelation Function (ACF): The ACF measures the correlation between a time series and its lagged values. It helps us understand the relationship between an observation and its historical values at different lags. Peaks or patterns in the ACF plot at specific lags may indicate the presence of seasonality in the data. If the ACF values are high and significant at certain lags, it suggests that there is a correlation between the current observation and the values at those lags.
#
In summary, the ACF can reveal the presence of seasonality or repeating patterns in a time series. High ACF values at specific lags indicate a correlation between the current observation and past observations at those lags.
#
2. Partial Autocorrelation Function (PACF): The PACF measures the correlation between a time series and its lagged values after removing the effect of intermediate lags. It helps identify the direct relationship between an observation and its lagged values, excluding the influence of other lags.
#
In the context of seasonality, the PACF may not explicitly show seasonality. Rather, it reveals the historical relationship between observations at different lags. The PACF helps determine the order of the autoregressive (AR) component in an ARIMA model by identifying the direct influence of past observations on the current observation, excluding the influence of other lags.
#
In summary, the PACF identifies the direct influence of past observations on the current observation, allowing you to determine the appropriate order of the AR component in an ARIMA model. It does not directly show seasonality.
#
To summarize:
#
- ACF helps identify seasonality or repeating patterns in the data.

- PACF helps determine the historical relationship between observations, excluding the influence of intermediate lags, which is useful for determining the order of the AR component in an ARIMA model.

ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) are statistical tools used to understand the correlation structure and pattern in a time series.
#
1. Autocorrelation Function (ACF):
#
- The ACF measures the correlation between a time series and its lagged values at various time lags.
- ACF values range between -1 and 1, where 1 indicates a strong positive correlation, -1 indicates a strong negative correlation, and 0 indicates no correlation.
- The ACF is useful for identifying the presence of serial correlation in a time series.
- If the ACF shows a significant spike at a specific lag, it suggests a strong correlation between the values at that lag. This can indicate the presence of a trend or seasonality in the data.
- The decay pattern of the ACF can provide insights into the type of time series model that may be appropriate for modeling the data.
2. Partial Autocorrelation Function (PACF):
#
- The PACF measures the correlation between a time series and its lagged values while controlling for the influence of intermediate lags.
- PACF values also range between -1 and 1, with similar interpretations as the ACF.
- The PACF helps identify the direct influence of each lag on the current value of the time series, after accounting for the influence of other lags.
- Significant spikes in the PACF indicate a strong direct relationship between the time series and its lagged values at those specific lags.
- The PACF can be useful for determining the order of an autoregressive (AR) model, as it indicates the number of significant lag terms to include.
#
By analyzing the ACF and PACF plots or examining the values directly, you can gain insights into the correlation structure of the time series and make informed decisions about the appropriate time series modeling techniques.

Here are the definitions of the variables used in the code:
#
1. `series` (parameter): Pandas Series containing the time series data.
#
- This variable represents the input time series data that you want to analyze and plot the ACF and PACF for.

2. `series_name` (parameter): Name of the series (column) being analyzed.
#
- This variable represents the name or label of the series (column) you are working with. It is used to provide a descriptive title for the ACF and PACF plots.
3. `nlags` (parameter, default: 40): Number of lags to include in the ACF and PACF calculations.
#
- This variable determines the maximum number of lags to consider when computing the autocorrelation and partial autocorrelation functions. It defines the length of the ACF and PACF arrays.
4. `alpha` (parameter, default: 0.05): Significance level for confidence intervals.
#
- This variable specifies the significance level (or confidence level) used to compute the confidence intervals for the ACF and PACF values. It is used to determine the range of the confidence intervals in the plot.
5. `acf_values`: Array of autocorrelation function (ACF) values.
#
- This variable stores the computed autocorrelation values at each lag. It represents the correlation between the time series and its lagged versions.
6. `acf_confint`: Array of confidence intervals for ACF values.
#
- This variable stores the upper and lower confidence intervals for the ACF values at each lag. It provides a range within which the true population ACF values are expected to fall with a certain level of confidence.
7. `pacf_values`: Array of partial autocorrelation function (PACF) values.
#
- This variable stores the computed partial autocorrelation values at each lag. It represents the correlation between the time series and its lagged versions while accounting for the influence of intermediate lags.
8. `pacf_confint`: Array of confidence intervals for PACF values.
#
- This variable stores the upper and lower confidence intervals for the PACF values at each lag. It provides a range within which the true population PACF values are expected to fall with a certain level of confidence.
#
These variables are used in the functions to calculate and plot the ACF and PACF of the provided time series data.

In[102]:

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import acf, pacf
```

```
def calculate_acf_pacf(data, series_name, nlags=100, alpha=0.05):
    """
    Calculate the ACF and PACF values for the given time series.

    Parameters:
    - series: Pandas Series containing the time series data
    - series_name: Name of the series (column) being analyzed
    - nlags: Number of lags to include in the ACF and PACF calculations (default:
40)
    - alpha: Significance level for confidence intervals (default: 0.05)

    Returns:
    - acf_values: Array of autocorrelation function (ACF) values
    - acf_confint: Array of confidence intervals for ACF values
    - pacf_values: Array of partial autocorrelation function (PACF) values
    - pacf_confint: Array of confidence intervals for PACF values

    """
    series = pd.Series(data[series_name])
    acf_values, acf_confint = acf(series, nlags=nlags, alpha=alpha)
    pacf_values, pacf_confint = pacf(series, nlags=nlags, alpha=alpha)
    return acf_values, acf_confint, pacf_values, pacf_confint


def plot_acf_pacf(acf_values, pacf_values, series_name):
    print(f'acf_values:\n{acf_values}\n')
    print(f'pacf_values:\n{pacf_values}')


    """
    Generate and plot the ACF and PACF using the given ACF and PACF values.

    Parameters:
    - acf_values: Array of autocorrelation function (ACF) values
    - pacf_values: Array of partial autocorrelation function (PACF) values
    - series_name: Name of the series (column) being plotted

    Returns:
    - None (plots the ACF and PACF)

    """
    fig, ax = plt.subplots(2, 1, figsize=(12, 10))
```

```
    # ACF plot
    ax[0].stem(acf_values, use_line_collection=True)
    ax[0].set_xlabel('Lag')
    ax[0].set_ylabel('Autocorrelation')
    ax[0].set_title(f'Autocorrelation Function (ACF) - {series_name}')

    # PACF plot
    ax[1].stem(pacf_values, use_line_collection=True)
    ax[1].set_xlabel('Lag')
    ax[1].set_ylabel('Partial Autocorrelation')
    ax[1].set_title(f'Partial Autocorrelation Function (PACF) - {series_name}')

    plt.tight_layout()
    plt.show()


# here is their interpretation:
#
# ACF (Autocorrelation Function):
#
# The ACF values measure the correlation between the PM2.5 values and their lagged values at different time lags.
# The ACF starts with a value of 1 at lag 0 since the correlation between a variable and itself is always perfect.
# The ACF shows a gradual decline in values as the lag increases, indicating a decreasing level of autocorrelation.
# The ACF values remain relatively high and positive up to a lag of around 6-8, indicating a strong positive autocorrelation at these lags.
# As the lag increases further, the ACF values gradually decrease, suggesting a decreasing level of autocorrelation and a more random pattern in the data.
# PACF (Partial Autocorrelation Function):
#
# The PACF values measure the correlation between the PM2.5 values and their lagged values while controlling for the influence of intermediate lags.
# The PACF starts with a value of 1 at lag 0, which is expected since the correlation of a variable with itself is perfect.
# The PACF values beyond the first few lags are relatively low and vary around 0, indicating weak or negligible partial autocorrelation at those lags.
# There are a few significant spikes in the PACF, such as at lag 1 and lag 2, indicating a direct influence or correlation of the PM2.5 values with their first and second lags.
# The significant PACF values beyond lag 2 become less pronounced, indicating a diminishing influence of the intermediate lags on the current values.
# Overall, the ACF and PACF results suggest that the PM2.5 time series has some degree of autocorrelation at shorter lags, but the autocorrelation diminishes as the
```

lag increases. The significant PACF spikes at the early lags suggest that an autoregressive (AR) model with 1 or 2 lag terms may be suitable for capturing the direct influence on the current PM2.5 values.
#
#

df_pm25Here is the explanation of the ACF and PACF results for each series:
#
1. PM2.5:
#
- ACF (Autocorrelation Function):
- The PM2.5 series exhibits a high positive autocorrelation, indicated by the significant ACF values up to lag 6-8. This suggests that the current PM2.5 values are strongly correlated with their past values at these lags.
- As the lag increases beyond 8, the ACF values gradually decrease, indicating a decreasing level of autocorrelation.
- PACF (Partial Autocorrelation Function):
- The PACF shows significant spikes at lag 1 and lag 2, suggesting a direct influence or correlation of the current PM2.5 values with their first and second lags.
- Beyond lag 2, the PACF values become relatively low and vary around 0, indicating weak or negligible partial autocorrelation at those lags.
2. PM10:
#
- ACF:
- The PM10 series exhibits a relatively high positive autocorrelation up to lag 6, indicating a strong correlation between the current PM10 values and their past values at these lags.
- As the lag increases beyond 6, the ACF values gradually decrease, suggesting a decreasing level of autocorrelation.
- PACF:
- The PACF shows a significant spike at lag 1, indicating a direct influence or correlation of the current PM10 values with their first lag.
- Beyond lag 1, the PACF values become relatively low and vary around 0, indicating weak or negligible partial autocorrelation at those lags.
3. CO:
#
- ACF:
- The CO series exhibits a moderate positive autocorrelation up to lag 6, indicating a correlation between the current CO values and their past values at these lags.
- As the lag increases beyond 6, the ACF values gradually decrease, suggesting a decreasing level of autocorrelation.
- PACF:

- The PACF shows a significant spike at lag 1, indicating a direct influence or correlation of the current CO values with their first lag.
- Beyond lag 1, the PACF values become relatively low and vary around 0, indicating weak or negligible partial autocorrelation at those lags.
4. NO2:
#
- ACF:
- The NO2 series exhibits a high positive autocorrelation up to lag 6, indicating a strong correlation between the current NO2 values and their past values at these lags.
- As the lag increases beyond 6, the ACF values gradually decrease, suggesting a decreasing level of autocorrelation.
- PACF:
- The PACF shows a significant spike at lag 1, indicating a direct influence or correlation of the current NO2 values with their first lag.
- Beyond lag 1, the PACF values become relatively low and vary around 0, indicating weak or negligible partial autocorrelation at those lags.

Если данные являются стационарными и у вас есть большое количество значений (11 000), менее вероятно, что включение большего количества задержек (например, 150) по сравнению с меньшим количеством задержек (например, 40) существенно повлияет на производительность модели.
#
В стационарных временных рядах закономерности автокорреляции и частичной автокорреляции обычно быстро затухают, и влияние включения дополнительных задержек после определенной точки уменьшается. Включение большего числа задержек может не дать много дополнительной информации или улучшить прогностическую способность модели.
#
Учитывая размер вашего набора данных (11 000 значений), у вас есть значительный объем данных для оценки параметров модели и фиксации лежащих в ее основе закономерностей. В таких случаях обычно фокусируются на разумном количестве задержек, которые фиксируют наиболее значимые модели автокорреляции и частичной автокорреляции без введения ненужной сложности.
#
Основываясь на графиках ACF и PACF и принимая во внимание стационарный характер данных, если вы наблюдаете значительные всплески до запаздывания 40 и видите уменьшающиеся закономерности после этого, разумно ограничить количество запаздываний меньшим диапазоном. Включение задержек до 40 должно быть достаточным для получения соответствующей информации об автокорреляции и частичной автокорреляции в вашей модели.
#

Не забудьте проверить работоспособность вашей модели, провести диагностические проверки и рассмотреть другие методы выбора модели, чтобы убедиться, что выбранное количество лагов обеспечивает наилучший баланс между подгонкой модели и экономичностью.

Коэффициент авторегрессии (AR) (p):
#
Изучите график PACF и обратите внимание на значительные всплески, которые постепенно затухают по мере увеличения задержки.
PACF представляет прямое влияние каждого запаздывания на текущее значение временного ряда, контролируя влияние промежуточных запаздываний.
Определите последний значительный всплеск на графике PACF до того, как он станет статистически незначимым или попадет в доверительный интервал.
Запаздывание, соответствующее этому последнему значительному скачку, представляет собой порядок авторегрессии (p) модели AR.
#
#
Коэффициент скользящей средней (MA) (q):
#
Проанализируйте график ACF и обратите внимание на значительные всплески, которые постепенно затухают по мере увеличения запаздывания.
ACF измеряет корреляцию между временным рядом и его запаздывающими значениями при различных временных задержках.
Определите последний значительный всплеск на графике ACF до того, как он станет статистически незначимым или попадет в доверительный интервал.
Запаздывание, соответствующее этому последнему значительному скачку, представляет порядок скользящей средней (q) модели скользящей средней.
Важно отметить, что интерпретация графиков PACF и ACF может быть субъективной, и выбор значений p и q может потребовать некоторых проб и ошибок или экспертного заключения. Кроме того, следует рассмотреть другие методы выбора модели и диагностические проверки для подтверждения выбранных значений и оценки общего соответствия модели.

How we can see here with the ACF we observe non-stationary flows of data, so lets use diff and check

In[103]:

```
df_pm10_no2_c0.columns
```

Train-test split

```
# In[104]:


def train_test_forecast_split(data, test_size=0.2, forecasting=0.05):
    split_index = int(len(data) * (1 - (test_size+forecasting)))
    train_data = data[:split_index]

    test_data = data[split_index:]
    return train_data, test_data


# In[105]:


test_size_pm25 = 0.2


# In[106]:


a, b = train_test_forecast_split(df_pm25)


# In[107]:


plt.figure(figsize=(16, 8))
a["PM2.5"].plot()
b["PM2.5"].plot()
plt.show()


# In[108]:


from sklearn.model_selection import train_test_split




# Split the data into training, testing, and forecasting sets
train_data_pm10_no2_c0, test_data_pm10_no2_c0 =
train_test_forecast_split(df_pm10_no2_c0)
```

```
# Split the data into training, testing, and forecasting sets
train_data_pm25, test_data_pm25 = train_test_forecast_split(df_pm25)

print()
# Print the sizes of the sets
print("PM2.5")
print("train size:\t", len(train_data_pm25),";",str(train_data_pm25.date.min().date()),";", str(train_data_pm25.date.max().date()))
print("test size:\t", len(test_data_pm25),";", str(test_data_pm25.date.min().date()),";", str(test_data_pm25.date.max().date()))


print()
print("PM10, NO2, CO")
print("train size:\t", len(train_data_pm10_no2_c0),";", str(train_data_pm10_no2_c0.date.min().date()),";", str(train_data_pm10_no2_c0.date.max().date()))
print("test size:\t", len(test_data_pm10_no2_c0),";", str(test_data_pm10_no2_c0.date.min().date()),";", str(test_data_pm10_no2_c0.date.max().date()))

plt.show()


# In[109]:


train_data_pm25[:]


# In[110]:


test_data_pm25.info()


# In[111]:


train_data = train_data_pm25
test_data = test_data_pm25
```

In[112]:

```
train_data['PM2.5'].plot()
```

In[113]:

```
test_data['PM2.5'].plot()
```

In[ ]:

Turned ARIMA

In[114]:

```
import time
import matplotlib.pyplot as plt
from math import sqrt
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_squared_error
import scipy.stats as stats
import seaborn as sns
import numpy as np

# Function to calculate Root Mean Squared Error (RMSE)
def calculate_rmse(actual, predicted):
    mse = mean_squared_error(actual, predicted)
    rmse = sqrt(mse)
    return rmse

# Function to calculate Mean Absolute Percentage Error (MAPE)
def calculate_mape(actual, predicted):
    if len(actual) != len(predicted):
        raise ValueError("Actual and predicted lists must have the same length.")
```

```
    absolute_errors = []
    for i in range(len(actual)):
        absolute_errors.append(abs(actual[i] - predicted[i]))

    percentage_errors = [error / actual[i] for i, error in enumerate(absolute_errors)]
    mean_percentage_error = sum(percentage_errors) / len(actual)
    mape = mean_percentage_error * 100

    return mape

# Function to evaluate an ARIMA model for a given order (p, d, q)
def evaluate_arima_model(train, test, arima_order):
    history = list(train)
    predictions = []
    residuals = []
    for t in range(len(test)):
        model = ARIMA(history, order=arima_order)
        model_fit = model.fit(method='innovations_mle')
        yhat = model_fit.forecast()[0]
        predictions.append(yhat)
        residuals.append(test[t] - yhat)
        history.append(test[t])
    rmse = calculate_rmse(test, predictions)
    mape = calculate_mape(test, predictions)
    aic = model_fit.aic
    bic = model_fit.bic
    return predictions, residuals, rmse, mape, aic, bic, model_fit

# Function to evaluate combinations of p, d, and q values for an ARIMA model
def evaluate_models(train, test, p_values, d_values, q_values):
    start_time = time.time()  # Start time logger
    best_score, best_cfg = float("inf"), None
    best_predictions = None
    best_residuals = None
    best_aic, best_bic = float("inf"), float("inf")
    iteration = 0

    for p in p_values:
        for d in d_values:
            for q in q_values:
                order = (p, d, q)
                try:
                    predictions, residuals, rmse, mape, aic, bic, model = evaluate_arima_model(train, test, order)
```

```
                if rmse < best_score:
                    best_score, best_cfg = rmse, order
                    best_predictions = predictions
                    best_residuals = residuals
                if aic < best_aic:
                    best_aic = aic
                if bic < best_bic:
                    best_bic = bic
                print('ARIMA%s RMSE=%.3f MAPE=%.3f AIC=%.3f BIC=%.3f' %
(order, rmse, mape, aic, bic))
                print(f'iteration {iteration}')
            except:
                continue
            iteration += 1

    end_time = time.time()  # End time logger
    elapsed_time = end_time - start_time
    print('Best ARIMA%s RMSE=%.3f AIC=%.3f BIC=%.3f' % (best_cfg,
best_score, best_aic, best_bic))
    print('Elapsed Time: %.3f seconds' % elapsed_time)
    return best_predictions, best_residuals


# Loop through each column and create predictions
for column in ['PM2.5']:
    print(f"Predictions for {column}:")
    train_col = train_data[column].dropna()
    test_col = test_data[column].dropna()

    # Convert data to a list
    train_list = train_col.tolist()
    test_list = test_col.tolist()

    # Evaluate parameters
    p_values = [5]
    d_values = [0]
    q_values = [0]

#      p_values = [1, 5, 6]
#     d_values = range(0, 2)
#     q_values = range(0, 2)
    predictions, residuals = evaluate_models(train_list, test_list, p_values, d_values,
q_values)
    print()
```

```
    if test_list and predictions:
        # Plotting
        plt.figure(figsize=(16, 7))
        plt.plot(test_list, label='Actual')
        plt.plot(predictions, label='Predicted')
        plt.xlabel('Time')
        plt.ylabel(column)
        plt.legend()
        plt.show()

        # Q-Q Plot
        plt.figure(figsize=(8, 6))
        qq_plot = stats.probplot(residuals, dist="norm", plot=plt)
        plt.title('Q-Q Plot')
        plt.show()

        # Print Q-Q plot values
        print("Q-Q Plot Values:")
        print("Mean:", np.mean(qq_plot[0][0]))
        print("Standard Deviation:", np.std(qq_plot[0][0]))

        # Residual Plot
        plt.figure(figsize=(8, 6))
        sns.residplot(x=predictions, y=residuals, lowess=True)
        plt.xlabel('Predicted')
        plt.ylabel('Residuals')
        plt.title('Residual Plot')
        plt.show()


# In[ ]:


# %%time
# import time
# import matplotlib.pyplot as plt
# from math import sqrt
# from statsmodels.tsa.arima.model import ARIMA
# from sklearn.metrics import mean_squared_error


# # Function to calculate Root Mean Squared Error (RMSE)
# def calculate_rmse(actual, predicted):
```

```
#     mse = mean_squared_error(actual, predicted)
#     rmse = sqrt(mse)
#     return rmse


# # Function to calculate Mean Absolute Percentage Error (MAPE)
# def calculate_mape(actual, predicted):
#     if len(actual) != len(predicted):
#         raise ValueError("Actual and predicted lists must have the same length.")

#     absolute_errors = []
#     for i in range(len(actual)):
#         absolute_errors.append(abs(actual[i] - predicted[i]))

#     percentage_errors = [error / actual[i] for i, error in enumerate(absolute_errors)]
#     mean_percentage_error = sum(percentage_errors) / len(actual)
#     mape = mean_percentage_error * 100

#     return mape


# # Function to evaluate an ARIMA model for a given order (p, d, q)
# def evaluate_arima_model(train, test, arima_order):
#     history = list(train)
#     predictions = []
#     for t in range(len(test)):
#         model = ARIMA(history, order=arima_order)
#         model_fit = model.fit(method='innovations_mle')
#         yhat = model_fit.forecast()[0]
#         predictions.append(yhat)
#         history.append(test[t])
#     rmse = calculate_rmse(test, predictions)
#     mape = calculate_mape(test, predictions)
#     aic = model_fit.aic
#     bic = model_fit.bic
#     return predictions, rmse, mape, aic, bic, model


# # Function to evaluate combinations of p, d, and q values for an ARIMA model
# import time

# # Function to evaluate combinations of p, d, and q values for an ARIMA model
# def evaluate_models(train, test, p_values, d_values, q_values):
#     start_time = time.time()  # Start time logger
```

```
#     best_score, best_cfg = float("inf"), None
#     best_predictions = None
#     best_aic, best_bic = float("inf"), float("inf")
#     iteration = 0

#     for p in p_values:
#         for d in d_values:
#             for q in q_values:
#                 order = (p, d, q)
#                 try:
#                  predictions, rmse, mape, aic, bic, model = evaluate_arima_model(train,
test, order)
#                     if rmse < best_score:
#                         best_score, best_cfg = rmse, order
#                         best_predictions = predictions
#                     if aic < best_aic:
#                         best_aic = aic
#                     if bic < best_bic:
#                         best_bic = bic
#                       print('ARIMA%s RMSE=%.3f MAPE=%.3f AIC=%.3f BIC=%.3f'
% (order, rmse, mape, aic, bic))
#                     print(f'iteration {iteration}')
#                 except:
#                     continue
#                 iteration += 1

#     end_time = time.time()  # End time logger
#     elapsed_time = end_time - start_time
#         print('Best ARIMA%s RMSE=%.3f AIC=%.3f BIC=%.3f' % (best_cfg,
best_score, best_aic, best_bic))
#     print('Elapsed Time: %.3f seconds' % elapsed_time)
#     return best_predictions


# # Loop through each column and create predictions
# for column in ['PM2.5']:
#     print(f"Predictions for {column}:")
#     train_col = train_data[column].dropna()
#     test_col = test_data[column].dropna()

#     # Convert data to a list
#     train_list = train_col.tolist()
#     test_list = test_col.tolist()
```

```
#     # Evaluate parameters
#     p_values = [1, 5, 6]
#     d_values = range(0, 2)
#     q_values = range(0, 2)
#    predictions = evaluate_models(train_list, test_list, p_values, d_values, q_values)
#     print()

#     if test_list and predictions:
#         # Plotting
#         plt.figure(figsize=(16, 7))
#         plt.plot(test_list, label='Actual')
#         plt.plot(predictions, label='Predicted')
#         plt.xlabel('Time')
#         plt.ylabel(column)
#         plt.legend()
#         plt.show()


# ## MODEL for co, no2, PM10`

# In[115]:


train_data = train_data_pm10_no2_c0
test_data = test_data_pm10_no2_c0


# In[116]:


get_ipython().run_cell_magic('time', '', 'import time\nimport matplotlib.pyplot as plt\nfrom math import sqrt\nfrom statsmodels.tsa.arima.model import ARIMA\nfrom sklearn.metrics import mean_squared_error\n\n\n# Function to calculate Root Mean Squared Error (RMSE)\ndef calculate_rmse(actual, predicted):\n    mse = mean_squared_error(actual, predicted)\n    rmse = sqrt(mse)\n    return rmse\n\n\n\n# Function to calculate Mean Absolute Percentage Error (MAPE)\ndef calculate_mape(actual, predicted):\n    if len(actual) != len(predicted):\n        raise ValueError("Actual and predicted lists must have the same length.")\n\n    absolute_errors = []\n    for i in range(len(actual)):\n        absolute_errors.append(abs(actual[i] - predicted[i]))\n\n    percentage_errors = [error / actual[i] for i, error in enumerate(absolute_errors)]\n    mean_percentage_error = sum(percentage_errors) / len(actual)\n    mape = mean_percentage_error * 100\n\n    return mape\n\n\n\n# Function to evaluate an ARIMA model for a given order (p, d, q)\ndef evaluate_arima_model(train, test, arima_order):\n    history = list(train)\n
```

predictions = []\n for t in range(len(test)):\n model = ARIMA(history, order=arima_order)\n model_fit = model.fit(method=\'innovations_mle\')\n yhat = model_fit.forecast()[0]\n predictions.append(yhat)\n history.append(test[t])\n rmse = calculate_rmse(test, predictions)\n mape = calculate_mape(test, predictions)\n aic = model_fit.aic\n bic = model_fit.bic\n return predictions, rmse, mape, aic, bic, model\n\n\n# Function to evaluate combinations of p, d, and q values for an ARIMA model\nimport time\n\n# Function to evaluate combinations of p, d, and q values for an ARIMA model\ndef evaluate_models(train, test, p_values, d_values, q_values):\n start_time = time.time() # Start time logger\n best_score, best_cfg = float("inf"), None\n best_predictions = None\n best_aic, best_bic = float("inf"), float("inf")\n iteration = 0\n\n for p in p_values:\n for d in d_values:\n for q in q_values:\n order = (p, d, q)\n try:\n predictions, rmse, mape, aic, bic, model = evaluate_arima_model(train, test, order)\n if rmse < best_score:\n best_score, best_cfg = rmse, order\n best_predictions = predictions\n if aic < best_aic:\n best_aic = aic\n if bic < best_bic:\n best_bic = bic\n print(\'ARIMA%s RMSE=%.3f MAPE=%.3f AIC=%.3f BIC=%.3f\' % (order, rmse, mape, aic, bic))\n print(f\'iteration {iteration}\')\n except:\n continue\n iteration += 1\n\n end_time = time.time() # End time logger\n elapsed_time = end_time - start_time\n print(\'Best ARIMA%s RMSE=%.3f AIC=%.3f BIC=%.3f\' % (best_cfg, best_score, best_aic, best_bic))\n print(\'Elapsed Time: %.3f seconds\' % elapsed_time)\n return best_predictions\n\n\n# Loop through each column and create predictions\nfor column in [\'PM10\', \'NO2\', \'CO\']:\n print(f"Predictions for {column}:")\n train_col = train_data[column].dropna()\n test_col = test_data[column].dropna()\n\n # Convert data to a list\n train_list = train_col.tolist()\n test_list = test_col.tolist()\n\n # Evaluate parameters\n p_values = [1, 5, 6]\n d_values = range(0, 2)\n q_values = range(0, 2)\n predictions = evaluate_models(train_list, test_list, p_values, d_values, q_values)\n print()\n\n if test_list and predictions:\n # Plotting\n plt.figure(figsize=(16, 7))\n plt.plot(test_list, label=\'Actual\')\n plt.plot(predictions, label=\'Predicted\')\n plt.xlabel(\'Time\')\n plt.ylabel(column)\n plt.legend()\n plt.show()\n')

#

In[117]:

```
import time
import matplotlib.pyplot as plt
from math import sqrt
from statsmodels.tsa.arima.model import ARIMA
```

```
from sklearn.metrics import mean_squared_error
import scipy.stats as stats
import seaborn as sns
import numpy as np

# Function to calculate Root Mean Squared Error (RMSE)
def calculate_rmse(actual, predicted):
    mse = mean_squared_error(actual, predicted)
    rmse = sqrt(mse)
    return rmse

# Function to calculate Mean Absolute Percentage Error (MAPE)
def calculate_mape(actual, predicted):
    if len(actual) != len(predicted):
        raise ValueError("Actual and predicted lists must have the same length.")

    absolute_errors = []
    for i in range(len(actual)):
        absolute_errors.append(abs(actual[i] - predicted[i]))

    percentage_errors = [error / actual[i] for i, error in enumerate(absolute_errors)]
    mean_percentage_error = sum(percentage_errors) / len(actual)
    mape = mean_percentage_error * 100

    return mape

# Function to evaluate an ARIMA model for a given order (p, d, q)
def evaluate_arima_model(train, test, arima_order):
    history = list(train)
    predictions = []
    residuals = []
    for t in range(len(test)):
        model = ARIMA(history, order=arima_order)
        model_fit = model.fit(method='innovations_mle')
        yhat = model_fit.forecast()[0]
        predictions.append(yhat)
        residuals.append(test[t] - yhat)
        history.append(test[t])
    rmse = calculate_rmse(test, predictions)
    mape = calculate_mape(test, predictions)
    aic = model_fit.aic
    bic = model_fit.bic
    return predictions, residuals, rmse, mape, aic, bic, model_fit
```

```
# Function to evaluate combinations of p, d, and q values for an ARIMA model
def evaluate_models(train, test, p_values, d_values, q_values):
    start_time = time.time()  # Start time logger
    best_score, best_cfg = float("inf"), None
    best_predictions = None
    best_residuals = None
    best_aic, best_bic = float("inf"), float("inf")
    iteration = 0

    for p in p_values:
        for d in d_values:
            for q in q_values:
                order = (p, d, q)
                try:
                    predictions, residuals, rmse, mape, aic, bic, model = evaluate_arima_model(train, test, order)
                    if rmse < best_score:
                        best_score, best_cfg = rmse, order
                        best_predictions = predictions
                        best_residuals = residuals
                    if aic < best_aic:
                        best_aic = aic
                    if bic < best_bic:
                        best_bic = bic
                    print('ARIMA%s RMSE=%.3f MAPE=%.3f AIC=%.3f BIC=%.3f' % (order, rmse, mape, aic, bic))
                    print(f'iteration {iteration}')
                except:
                    continue
                iteration += 1

    end_time = time.time()  # End time logger
    elapsed_time = end_time - start_time
    print('Best ARIMA%s RMSE=%.3f AIC=%.3f BIC=%.3f' % (best_cfg, best_score, best_aic, best_bic))
    print('Elapsed Time: %.3f seconds' % elapsed_time)
    return best_predictions, best_residuals


# Loop through each column and create predictions
for column in ['PM10']:
    print(f"Predictions for {column}:")
    train_col = train_data[column].dropna()
    test_col = test_data[column].dropna()
```

```
    # Convert data to a list
    train_list = train_col.tolist()
    test_list = test_col.tolist()

    # Evaluate parameters
    p_values = [6]
    d_values = [1]
    q_values = [1]

#     p_values = [1, 5, 6]
#    d_values = range(0, 2)
#    q_values = range(0, 2)
    predictions, residuals = evaluate_models(train_list, test_list, p_values, d_values,
q_values)
    print()

    if test_list and predictions:
        # Plotting
        plt.figure(figsize=(16, 7))
        plt.plot(test_list, label='Actual')
        plt.plot(predictions, label='Predicted')
        plt.xlabel('Time')
        plt.ylabel(column)
        plt.legend()
        plt.show()

        # Q-Q Plot
        plt.figure(figsize=(8, 6))
        qq_plot = stats.probplot(residuals, dist="norm", plot=plt)
        plt.title('Q-Q Plot')
        plt.show()

        # Print Q-Q plot values
        print("Q-Q Plot Values:")
        print("Mean:", np.mean(qq_plot[0][0]))
        print("Standard Deviation:", np.std(qq_plot[0][0]))

        # Residual Plot
        plt.figure(figsize=(8, 6))
        sns.residplot(x=predictions, y=residuals, lowess=True)
        plt.xlabel('Predicted')
        plt.ylabel('Residuals')
        plt.title('Residual Plot')
```

```
        plt.show()


# In[118]:


import time
import matplotlib.pyplot as plt
from math import sqrt
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_squared_error
import scipy.stats as stats
import seaborn as sns
import numpy as np

# Function to calculate Root Mean Squared Error (RMSE)
def calculate_rmse(actual, predicted):
    mse = mean_squared_error(actual, predicted)
    rmse = sqrt(mse)
    return rmse

# Function to calculate Mean Absolute Percentage Error (MAPE)
def calculate_mape(actual, predicted):
    if len(actual) != len(predicted):
        raise ValueError("Actual and predicted lists must have the same length.")

    absolute_errors = []
    for i in range(len(actual)):
        absolute_errors.append(abs(actual[i] - predicted[i]))

    percentage_errors = [error / actual[i] for i, error in enumerate(absolute_errors)]
    mean_percentage_error = sum(percentage_errors) / len(actual)
    mape = mean_percentage_error * 100

    return mape

# Function to evaluate an ARIMA model for a given order (p, d, q)
def evaluate_arima_model(train, test, arima_order):
    history = list(train)
    predictions = []
    residuals = []
    for t in range(len(test)):
        model = ARIMA(history, order=arima_order)
        model_fit = model.fit(method='innovations_mle')
```

```
        yhat = model_fit.forecast()[0]
        predictions.append(yhat)
        residuals.append(test[t] - yhat)
        history.append(test[t])
    rmse = calculate_rmse(test, predictions)
    mape = calculate_mape(test, predictions)
    aic = model_fit.aic
    bic = model_fit.bic
    return predictions, residuals, rmse, mape, aic, bic, model_fit

# Function to evaluate combinations of p, d, and q values for an ARIMA model
def evaluate_models(train, test, p_values, d_values, q_values):
    start_time = time.time()  # Start time logger
    best_score, best_cfg = float("inf"), None
    best_predictions = None
    best_residuals = None
    best_aic, best_bic = float("inf"), float("inf")
    iteration = 0

    for p in p_values:
        for d in d_values:
            for q in q_values:
                order = (p, d, q)
                try:
                    predictions, residuals, rmse, mape, aic, bic, model = evaluate_arima_model(train, test, order)
                    if rmse < best_score:
                        best_score, best_cfg = rmse, order
                        best_predictions = predictions
                        best_residuals = residuals
                    if aic < best_aic:
                        best_aic = aic
                    if bic < best_bic:
                        best_bic = bic
                    print('ARIMA%s RMSE=%.3f MAPE=%.3f AIC=%.3f BIC=%.3f' % (order, rmse, mape, aic, bic))
                    print(f'iteration {iteration}')
                except:
                    continue
                iteration += 1

    end_time = time.time()  # End time logger
    elapsed_time = end_time - start_time
```

```
    print('Best ARIMA%s RMSE=%.3f AIC=%.3f BIC=%.3f' % (best_cfg, best_score, best_aic, best_bic))
    print('Elapsed Time: %.3f seconds' % elapsed_time)
    return best_predictions, best_residuals


# Loop through each column and create predictions
for column in ['NO2']:
    print(f"Predictions for {column}:")
    train_col = train_data[column].dropna()
    test_col = test_data[column].dropna()

    # Convert data to a list
    train_list = train_col.tolist()
    test_list = test_col.tolist()

    # Evaluate parameters
    p_values = [5]
    d_values = [0]
    q_values = [1]

#      p_values = [1, 5, 6]
#     d_values = range(0, 2)
#     q_values = range(0, 2)
    predictions, residuals = evaluate_models(train_list, test_list, p_values, d_values, q_values)
    print()

    if test_list and predictions:
        # Plotting
        plt.figure(figsize=(16, 7))
        plt.plot(test_list, label='Actual')
        plt.plot(predictions, label='Predicted')
        plt.xlabel('Time')
        plt.ylabel(column)
        plt.legend()
        plt.show()

        # Q-Q Plot
        plt.figure(figsize=(8, 6))
        qq_plot = stats.probplot(residuals, dist="norm", plot=plt)
        plt.title('Q-Q Plot')
        plt.show()
```

```
        # Print Q-Q plot values
        print("Q-Q Plot Values:")
        print("Mean:", np.mean(qq_plot[0][0]))
        print("Standard Deviation:", np.std(qq_plot[0][0]))

        # Residual Plot
        plt.figure(figsize=(8, 6))
        sns.residplot(x=predictions, y=residuals, lowess=True)
        plt.xlabel('Predicted')
        plt.ylabel('Residuals')
        plt.title('Residual Plot')
        plt.show()


# In[119]:


import time
import matplotlib.pyplot as plt
from math import sqrt
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_squared_error
import scipy.stats as stats
import seaborn as sns
import numpy as np

# Function to calculate Root Mean Squared Error (RMSE)
def calculate_rmse(actual, predicted):
    mse = mean_squared_error(actual, predicted)
    rmse = sqrt(mse)
    return rmse

# Function to calculate Mean Absolute Percentage Error (MAPE)
def calculate_mape(actual, predicted):
    if len(actual) != len(predicted):
        raise ValueError("Actual and predicted lists must have the same length.")

    absolute_errors = []
    for i in range(len(actual)):
        absolute_errors.append(abs(actual[i] - predicted[i]))

    percentage_errors = [error / actual[i] for i, error in enumerate(absolute_errors)]
    mean_percentage_error = sum(percentage_errors) / len(actual)
    mape = mean_percentage_error * 100
```

```
    return mape

# Function to evaluate an ARIMA model for a given order (p, d, q)
def evaluate_arima_model(train, test, arima_order):
    history = list(train)
    predictions = []
    residuals = []
    for t in range(len(test)):
        model = ARIMA(history, order=arima_order)
        model_fit = model.fit(method='innovations_mle')
        yhat = model_fit.forecast()[0]
        predictions.append(yhat)
        residuals.append(test[t] - yhat)
        history.append(test[t])
    rmse = calculate_rmse(test, predictions)
    mape = calculate_mape(test, predictions)
    aic = model_fit.aic
    bic = model_fit.bic
    return predictions, residuals, rmse, mape, aic, bic, model_fit

# Function to evaluate combinations of p, d, and q values for an ARIMA model
def evaluate_models(train, test, p_values, d_values, q_values):
    start_time = time.time()  # Start time logger
    best_score, best_cfg = float("inf"), None
    best_predictions = None
    best_residuals = None
    best_aic, best_bic = float("inf"), float("inf")
    iteration = 0

    for p in p_values:
        for d in d_values:
            for q in q_values:
                order = (p, d, q)
                try:
                    predictions, residuals, rmse, mape, aic, bic, model = evaluate_arima_model(train, test, order)
                    if rmse < best_score:
                        best_score, best_cfg = rmse, order
                        best_predictions = predictions
                        best_residuals = residuals
                    if aic < best_aic:
                        best_aic = aic
                    if bic < best_bic:
```

```
                    best_bic = bic
                print('ARIMA%s RMSE=%.3f MAPE=%.3f AIC=%.3f BIC=%.3f' %
(order, rmse, mape, aic, bic))
                print(f'iteration {iteration}')
            except:
                continue
            iteration += 1

    end_time = time.time()  # End time logger
    elapsed_time = end_time - start_time
    print('Best ARIMA%s RMSE=%.3f AIC=%.3f BIC=%.3f' % (best_cfg,
best_score, best_aic, best_bic))
    print('Elapsed Time: %.3f seconds' % elapsed_time)
    return best_predictions, best_residuals


# Loop through each column and create predictions
for column in ['CO']:
    print(f"Predictions for {column}:")
    train_col = train_data[column].dropna()
    test_col = test_data[column].dropna()

    # Convert data to a list
    train_list = train_col.tolist()
    test_list = test_col.tolist()

    # Evaluate parameters
    p_values = [1]
    d_values = [1]
    q_values = [1]

#      p_values = [1, 5, 6]
#     d_values = range(0, 2)
#     q_values = range(0, 2)
    predictions, residuals = evaluate_models(train_list, test_list, p_values, d_values,
q_values)
    print()
    if test_list and predictions:

# Plotting
        plt.figure(figsize=(16, 7))
        plt.plot(test_list, label='Actual')
        plt.plot(predictions, label='Predicted')
        plt.xlabel('Time')
```

```
        plt.ylabel(column)
        plt.legend()
        plt.show()

        # Q-Q Plot
        plt.figure(figsize=(8, 6))
        qq_plot = stats.probplot(residuals, dist="norm", plot=plt)
        plt.title('Q-Q Plot')
        plt.show()
       # Print Q-Q plot values
        print("Q-Q Plot Values:")
        print("Mean:", np.mean(qq_plot[0][0]))
        print("Standard Deviation:", np.std(qq_plot[0][0]))

        # Residual Plot
        plt.figure(figsize=(8, 6))
        sns.residplot(x=predictions, y=residuals, lowess=True)
        plt.xlabel('Predicted')
        plt.ylabel('Residuals')
        plt.title('Residual Plot')
        plt.show()

# In[ ]
```