

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

INFORMATION TECHNOLOGY



УДК 004

Оригинальное эмпирическое исследование

<https://doi.org/10.23947/2587-8999-2025-9-2-52-64>


Автоматическая обработка первичных данных натурных исследований поведения природно-технических систем при изменении климата и антропогенных воздействиях в условиях Крайнего Севера

С.В. Кушуков¹ , К.Н. Иванов¹ , С.П. Левашкин¹ , М.В. Якобовский² ¹ Поволжский государственный университет телекоммуникаций и информатики, г. Самара, Российская Федерация² Институт прикладной математики имени М.В. Келдыша РАН, г. Москва, Российская Федерация ai_lab@psuti.ru

Аннотация

Введение. Рассматривается научная проблема изучения природно-технических систем (ПТС) Крайнего Севера в условиях изменения климата и антропогенных воздействий. Отмечается актуальность задачи обеспечения их устойчивости, что требует комплексного анализа натурных данных. Выявлены проблемы в методах автоматизированной обработки таких специфических данных. Целью работы является разработка автоматизированных методов обработки натурных данных для выявления закономерностей. В качестве инструментов используются библиотеки Python для анализа, обработки и визуализации данных.

Материалы и методы. Описан объект исследования — Главный корпус Якутской ТЭЦ в условиях вечной мерзлоты. В качестве материалов исследования использованы натурные данные, полученные из инженерно-геологических скважин Якутской ТЭЦ, стационаров Чабыда и Туймаада, а также железнодорожного участка Амуро-Якутской магистрали (АЯМ). Данные включают измерения температуры и влажности грунтов, динамики сезонного слоя, характеристик снежного покрова и др. Представлена детальная последовательность автоматизированной обработки первичных данных из XLS-файлов с использованием библиотеки pandas, включая чтение, очистку, преобразование форматов, заполнение или замену значений, удаление дубликатов, а также сохранение обработанных данных в форматах CSV, JSON и XLSX.

Результаты исследования. Представлены конкретные результаты автоматизированной обработки и систематизации первичных натурных данных. Успешно выполнено приведение разнородных измерений к единому формату, обеспечивающему их корректное использование. Сформирован уникальный массив данных на основе эмпирических наблюдений в специфических условиях Крайнего Севера. Продемонстрировано практическое применение библиотек Python для выполнения основных этапов предобработки и подготовки данных.

Обсуждение и заключение. Доказано, что применение системного подхода и автоматизированной обработки данных существенно повышает качество и надежность анализа натурных данных ПТС. Устранение пропусков и нормализация данных улучшают точность, а итоговые форматы данных удобны для дальнейшего использования в моделировании. Подчеркивается универсальность применения Python. Обозначены перспективы исследования — применение методов машинного обучения, кластеризации и моделирования, предназначенных для выявления закономерностей и прогнозирования поведения природно-технических систем в условиях Крайнего Севера под воздействием климатических и антропогенных факторов.

Ключевые слова: анализ данных, библиотеки Python, подготовка данных, предобработка данных, информационные технологии

Финансирование. Работа выполнена при финансовой поддержке РНФ (грант № 23–61–10032).

Для цитирования. Кушуков С.В., Иванов К.Н., Левашкин С.П., Якобовский М.В. Автоматическая обработка первичных данных натурных исследований поведения природно-технических систем при изменении климата и антропогенных воздействиях в условиях Крайнего Севера. *Computational Mathematics and Information Technologies*. 2025;9(2):52–64. <https://doi.org/10.23947/2587-8999-2025-9-2-52-64>

Automated Processing of Primary Field Data on the Behavior of Natural-Technological Systems under Climate Change and Anthropogenic Impacts in the Far North

Sergey V. Kushukov¹ , Konstantin N. Ivanov¹ , Sergey P. Levashkin¹  , Mikhail V. Yakobovskiy² 

¹ Povolzhskiy State University of Telecommunications and Informatics, Samara, Russian Federation

² Keldysh Institute of Applied Mathematics of the Russian Academy of Sciences (IPM RAS), Moscow, Russian Federation

 ai_lab@psuti.ru

Abstract

Introduction. This work addresses the scientific problem of studying natural-technological systems (NTS) of the Far North under conditions of climate change and anthropogenic impacts. The relevance of ensuring their stability is emphasized, which requires a comprehensive analysis of field data. Problems in automated processing methods of such specific data have been identified. The aim of the study is to develop automated methods for processing field data to reveal patterns. Python libraries for data analysis, processing, and visualization are used as tools.

Materials and Methods. The research object is described — the Main Building of the Yakutsk Thermal Power Plant (TPP) in permafrost conditions. The study materials include field data obtained from engineering-geological boreholes at the Yakutsk TPP, monitoring stations Chabyda and Tuymaada, as well as a section of the Amur-Yakutsk railway (AYR). The data include measurements of soil temperature and moisture, seasonal thaw layer dynamics, snow cover characteristics, and others. A detailed sequence of automated processing of primary data from XLS files using the pandas library is presented, including reading, cleaning, format conversion, filling or replacing missing values, removing duplicates, and saving processed data in CSV, JSON, and XLSX formats.

Results. Specific results of automated processing and systematization of primary field data are presented. Heterogeneous measurements were successfully unified into a single format, ensuring their proper use. A unique data array was formed based on empirical observations under the specific conditions of the Far North. The practical application of Python libraries for executing key stages of preprocessing and data preparation is demonstrated.

Discussion and Conclusion. It is shown that the application of a systematic approach and automated data processing significantly improves the quality and reliability of natural-technological system data analysis. Handling missing data and normalization enhance accuracy, and the final data formats are convenient for further modeling. The universality of Python is highlighted. Prospects for further research include applying machine learning, clustering, and modeling methods aimed at uncovering patterns and forecasting the behavior of natural-technological systems in the Far North under climate and anthropogenic influences.

Keywords: data analysis, Python libraries, data preparation, data preprocessing, information technologies.

Funding. This work was supported financially by the Russian Science Foundation (grant No. 23–61–10032).

For Citation. Kushukov S.V., Ivanov K.N., Levashkin S.P., Yakobovskiy M.V. Automated Processing of Primary Field Data on the Behavior of Natural-Technological Systems under Climate Change and Anthropogenic Impacts in the Far North. *Computational Mathematics and Information Technologies*. 2025;9(2):52–64. <https://doi.org/10.23947/2587-8999-2025-9-2-52-64>

Введение. Изучение природно-технических систем (ПТС) в условиях изменения климата и антропогенных воздействий, особенно в сложных условиях Крайнего Севера, представляет собой важную научную задачу. Эти системы формируются в результате взаимодействия природных процессов и технических объектов, а их устойчивость во многом зависит от динамики внешних факторов. Для выявления закономерностей функционирования ПТС и прогнозирования их возможных изменений требуется комплексный анализ данных.

Обработка информации, полученной в ходе натурных исследований, включает несколько этапов: сбор первичных данных, их предварительную обработку, анализ и интерпретацию результатов. Цель данной работы — разработка методов автоматизированной обработки натурных данных, что позволит выявить ключевые тренды и паттерны во взаимодействии природных и техногенных факторов [1–3].

Актуальность исследования обусловлена необходимостью принятия обоснованных решений для обеспечения устойчивого функционирования и управления природно-техническими системами в условиях нарастающего климатического и антропогенного воздействия [2–8].

Вначале представим алгоритмические библиотеки [4], обеспечивающие разнообразные инструменты обработки, анализа и моделирования [5] данных:

- **scikit-learn:** универсальный инструмент для машинного обучения, охватывающий весь процесс от подготовки данных до оценки моделей. Эта библиотека включает в себя все необходимые функции для решения задач машинного обучения и статистического анализа [9, 10];
- **statsmodels:** библиотека для оценки статистических моделей, которая предоставляет инструменты для статистических тестов, регрессионного анализа и анализа временных рядов.

Далее перечислим библиотеки визуализации, которые обеспечивают возможности для наглядного представления данных:

- **matplotlib**: помогает создавать статические, интерактивные и анимационные графики в Python. Обеспечивает обширный арсенал средств для визуализации данных в различных форматах, раскрывая широкий спектр возможностей для наглядного представления информации [11–14];

- **seaborn**: библиотека визуализации данных, основанная на matplotlib. Предоставляет высокоуровневый интерфейс для создания информативных статистических графиков.

Дополнительно перечислим библиотеки научных вычислений, используемые для математических расчётов и обработки данных:

- **pandas**: фреймворк, позволяющий обрабатывать и анализировать данные. Предоставляет мощные инструменты, включая структуру данных, например, DataFrame, который упрощает работу с табличными данными. Библиотека находит применение в процессах загрузки, очистки, анализа и подготовки данных перед внедрением их в алгоритмы машинного обучения;

- **numpy**: библиотека, предназначенная для манипулирования многомерными массивами и матрицами, а также предоставляющая возможность выполнять математические операции над ними. Предлагает разнообразие функций, оптимизированных под эффективную обработку числовых данных.

- **scipy**: высокоэффективная библиотека для выполнения научных и технических вычислений, которая расширяет функциональность операций оптимизации, интеграции, интерполяции и многих других научных задач.

Материалы и методы

Обзор существующих исследований. В последние годы наблюдается рост интереса к изучению влияния климатических и антропогенных факторов на природно-технические системы. Различные исследования подчеркивают необходимость комплексного подхода к управлению этими системами для обеспечения их устойчивости и функциональности.

Р.С. Рожков и его соавторы в своей работе «Управление природно-техническими системами через концепцию устойчивого развития и приемлемого риска» [15] акцентируют внимание на важности учета экологических, экономических и социальных факторов при управлении ПТС. Авторы предлагают использование системного анализа и оценки рисков для разработки оптимальных стратегий, способствующих снижению антропогенной нагрузки и улучшению экологического состояния окружающей среды.

Н. Дрегуло исследует влияние климатических факторов на эксплуатацию ПТС, связанных с обработкой отходов водоотведения. В работе [16] подчеркивается, что увеличение атмосферных осадков может негативно сказываться на эксплуатационных и природоохранных свойствах таких систем, что требует пересмотра нормативных критериев и адаптации к изменяющимся климатическим условиям.

В статье «Влияние антропогенных факторов на тепловое загрязнение городской среды» [17] рассматривается комплексная оценка воздействия человеческой деятельности на тепловое загрязнение атмосферы. Отмечается, что значительную долю теплового загрязнения вносят теплопотребление и транспорт, что подчеркивает необходимость разработки мер по снижению антропогенного теплового воздействия на урбанизированных территориях.

Исследование, опубликованное на сайте ООО «Биохим-ТЛ», анализирует влияние антропогенных факторов на развитие сельского хозяйства [18]. Особое внимание уделяется изменениям в почвах, вызванным длительным применением удобрений и отходов животноводства, что влияет на экологическое состояние аграрных ландшафтов.

В совокупности эти исследования подчеркивают необходимость комплексного и адаптивного подхода к управлению природно-техническими системами в условиях изменяющегося климата и возрастающего антропогенного воздействия. Учет специфики каждого компонента ПТС и постоянный мониторинг их состояния являются ключевыми элементами для обеспечения их устойчивого функционирования.

Предметная область исследования. Объектом исследования является Главный корпус Якутской ТЭЦ — уникальное инженерное сооружение, построенное в условиях вечной мерзлоты (криолитозоны). Этот объект особенно интересен, потому что он был первым промышленным строением в СССР, возведённым на вечномерзлых грунтах по первому принципу строительства. Это значит, что во время строительства были приняты меры, чтобы сохранить мерзлые грунты в качестве надёжного основания для ТЭЦ. Строительство началось в 1933 году, и в его процессе активно участвовали учёные, специализирующиеся на изучении мерзлоты. В 1937 году ТЭЦ была введена в эксплуатацию и с тех пор обеспечивает Якутск электроэнергией, а с 1961 года — и теплом.

Для того чтобы понять, как ТЭЦ работает в условиях вечной мерзлоты и как на неё влияют различные внешние факторы, была создана расчётная модель. Эта модель охватывает территорию площадью 180×150 метров и глубиной до 30 метров, что позволяет анализировать многолетнемёрзлые породы (ММП), на которых стоит ТЭЦ. Основой для построения модели послужили данные из инженерно-геологических скважин. Эти данные дают возможность определить, как ведут себя грунты при разных температурных и внешних условиях, что крайне важно для обеспечения стабильной работы ТЭЦ.

Особое внимание в исследовании уделено температурному режиму окружающей среды. Были собраны и проанализированы среднесуточные данные о температуре воздуха в районе Якутска за последние 10 лет. Эти данные имеют важное значение для понимания того, как изменяются климатические условия и как эти изменения могут

влиять на состояние мерзлых грунтов и на работу ТЭЦ. Важно отметить, что эти данные отличаются от тех, которые предоставляются управлением гидрометеорологической службы на основе норм с 1966 года. Нормативные данные понимаются как статистические показатели, полученные путём длительного усреднения климатических наблюдений (обычно за 30-летний период). Эти показатели служат эталонными значениями для оценки климатических условий региона. Однако в условиях актуальных климатических изменений такие нормативы, если не обновляются регулярно, могут не отражать современные тенденции и зачастую давать заниженные значения температуры, что ограничивает их применимость при анализе текущего состояния климата.

Для исследования термодинамических процессов в многолетнемёрзлых породах были использованы натурные данные, полученные из инженерно-геологических скважин и стационарных наблюдательных площадок. Эти данные представляют собой детализированные временные ряды, охватывающие многолетние измерения ключевых параметров мерзлотных грунтов в различных районах Якутии.

Данные были собраны с трёх основных источников:

- Якутская ТЭЦ — 8 инженерно-геологических скважин, по которым ведутся систематические измерения температуры и влажности грунтов, а также исследуются литологический состав, физические и теплофизические свойства мерзлых пород;

- Стационары Чабыда и Туймаада — 16 площадок, на которых фиксируются температура и влажность грунтов, сезонная динамика глубины протаивания, межгодовая изменчивость сезонноталого слоя (СТС), высота снежного покрова, плотность снега, а также физико-механические свойства грунтов;

- Железнодорожный участок Амуро-Якутской железнодорожной магистрали — 41 инженерно-геологическая скважина, где собираются данные о температуре и влажности грунтов, изменчивости СТС, высоте снежного покрова и других геокриологических параметрах.

В отличие от обобщённых климатических норм, представленных в официальных источниках, эти данные отражают реальные процессы, происходящие в мерзлотных грунтах, с высокой детализацией по глубине и времени. Это делает их уникальным источником информации для исследования устойчивости природно-технических систем Крайнего Севера в условиях изменения климата и антропогенных воздействий.

Обработка и предварительный анализ данных. Рассмотрим предобработку данных на примере датасета мониторинговых сведений по шестнадцати площадкам Якутии, предоставленных Институтом Мерзлотоведения СО РАН. Каждая площадка содержит свои данные наблюдения. Например, «Площадка 9_Чабыда», содержит сведения о температуре грунтов, влажности, сезонной динамике глубины протаивания, межгодовой изменчивости сезонно-талого слоя, месячной высоты снега, межгодовой динамике плотности снега, сезонной динамике плотности и высоты снежного покрова, литологический разрез, физические и теплофизические свойства напочвенных покровов и грунтов.

Цель исследования — подготовка анализа данных методами машинного обучения и нахождение взаимосвязей между их атрибутами. Следует подчеркнуть, что процессы, которые авторы автоматизируют, обычно выполняются вручную или с использованием интерактивных методов. В результате этого взаимодействия формируются структурированные и подготовленные для анализа наборы данных, преобразованные из исходной информации, предоставленной на начальном этапе.

Структурная схема процесса автоматической обработки данных представлена на рис. 1.

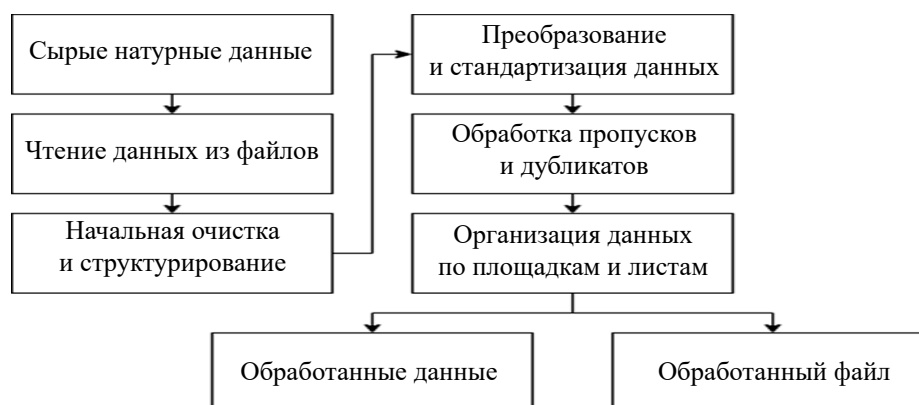


Рис. 1. Структурная схема автоматизированной обработки первичных данных

Для предобработки данных следует выполнить чтение данных из XLS-файла (формат, в котором изначально получены данные мониторинга) и загрузку их в объект DataFrame, используя библиотеку pandas. Этот этап включает в себя обнаружение и обработку пропущенных значений, выбор стратегии заполнения пропусков для столбцов, а также удаление несущественных признаков.

Для анализа датасета следует подготовить все релевантные данные и применить рассмотренные модули Python на практике [7–8]. Первый этап — извлечение данных из XLS-файла. При анализе данных, эффективное хранение и обработка табличных данных являются критическими аспектами.

Получим информацию о количестве файлов по площадкам:

```
def get_excel_files_info(folder_path: str):
    """ """
    excel_files_info = []
    for filename in tqdm(os.listdir(folder_path)):
        if filename.endswith('.xls') or filename.endswith('.xlsx'):
            file_path = os.path.join(folder_path, filename)
            if filename.endswith('.xls'):
                xls_workbook = xlrd.open_workbook(file_path)
                number_sheets = xls_workbook.nsheets
            else:
                workbook = load_workbook(file_path)
                number_sheets = len(workbook.sheetnames)
            name_for_directory, extension = os.path.splitext(filename)
            file_info = {
                'file_path': file_path,
                'name_file': filename,
                'type_file': 'xls' if filename.endswith('.xls') else 'xlsx',
                'number_sheets': number_sheets,
                'name_for_directory': name_for_directory,
            }
            excel_files_info.append(file_info)
    return excel_files_info[:-1]
raw_data = get_excel_files_info(data_cache)
print(f'Number files: {len(raw_data)}\n\nList files:\n')
pprint.pprint(raw_data).
```

После выполнения кода переменная `raw_data` будет содержать сведения о файлах, которые находятся в директории, что позволяет легко увидеть содержимое этих файлов (рис. 2).

```
Number files: 16

List files:

[{'file_path': '/home/user/data_farm/projects/rnf_yakutia/data_cache/monitoring_data/Площадка '
    'Луг_Туймаада.xlsx',
    'name_file': 'Площадка Луг_Туймаада.xlsx',
    'name_for_directory': 'Площадка Луг_Туймаада',
    'number_sheets': 9,
    'type_file': 'xlsx'},
 {'file_path': '/home/user/data_farm/projects/rnf_yakutia/data_cache/monitoring_data/Площадка '
    'Лес_Туймаада.xlsx',
    'name_file': 'Площадка Лес_Туймаада.xlsx',
    'name_for_directory': 'Площадка Лес_Туймаада',
    'number_sheets': 9,
    'type_file': 'xlsx'},
 {'file_path': '/home/user/data_farm/projects/rnf_yakutia/data_cache/monitoring_data/Площадка '
    '(Скважина) 1_Чабыда.xlsx',
    'name_file': 'Площадка (Скважина) 1_Чабыда.xlsx',
    'name_for_directory': 'Площадка (Скважина) 1_Чабыда',
    'number_sheets': 9,
    'type_file': 'xlsx'},
 {'file_path': '/home/user/data_farm/projects/rnf_yakutia/data_cache/monitoring_data/Площадка '
    '10_Чабыда.xlsx',
    'name_file': 'Площадка 10_Чабыда.xlsx',
    'name_for_directory': 'Площадка 10_Чабыда',
    'number_sheets': 9,
    'type_file': 'xlsx'},
 {'file_path': '/home/user/data_farm/projects/rnf_yakutia/data_cache/monitoring_data/Площадка '
    '11_Чабыда.xlsx',
    'name_file': 'Площадка 11_Чабыда.xlsx',
    'name_for_directory': 'Площадка 11_Чабыда',
    'number_sheets': 9,
    'type_file': 'xlsx'},
 {'file_path': '/home/user/data_farm/projects/rnf_yakutia/data_cache/monitoring_data/Площадка '
    '12_Чабыда.xlsx',
    'name_file': 'Площадка 12_Чабыда.xlsx',
    'name_for_directory': 'Площадка 12_Чабыда',
    'number_sheets': 9,
    'type_file': 'xlsx'}]
```

Рис. 2. Информация о файлах

Проанализируем количество листов данных в каждом датасете:

```
def read_non_empty_excel(file_path, sheet_name):
    """ """
    df = pd.read_excel(file_path, sheet_name=sheet_name)
    return df if not df.empty else None
raw_file = raw_data[15]
state_name = raw_file.get('name_for_directory')
df = pd.ExcelFile(raw_file.get('file_path'))
dfs = {name_sheet: read_non_empty_excel(df, name_sheet) for name_sheet in df.sheet_names
if read_non_empty_excel(df, name_sheet) is not None}
key_list = list(dfs.keys())
print(f'Name file: {state_name}\n\nNumber sheets: {len(key_list)}\n\nNames all sheets in
this file:\n\n{key_list}').
```

После выполнения кода у нас будет информация о количестве листов в датасете, наименование каждого листа, что даст нам возможность обращаться непосредственно к определенному листу по его имени (рис. 3).

```
Name file: Площадка 9_Чабыда

Number sheets: 9

Names all sheets in this file:

['Температура грунтов', 'Влажность грунтов', 'Сезон.дина
м.глубины протаивания', 'Межгодовая изменчивость СТС', 'М
есячная высота снега', 'Межгод.динамика плотн.снега', 'Се
зон.динамика плотн.выс.снега', 'Литологический разрез',
'Физические и теплофизические...']
```

Рис. 3. Вывод информации о листах

Однако следует отметить, что в некоторых датасетах есть ошибки в наименовании листов, которые необходимо исправить, чтобы избежать проблем в дальнейшем анализе:

```
# Ошибки наименований 1
if 'Межгод.динамика плотн.снега' in dfs:
    dfs['Межгод.динамика плотн.снега'] = dfs.pop('Межгод.динаика плотн.снега')
if 'Межгод. динамика плотн. снега ' in dfs:
    dfs['Межгод.динамика плотн.снега'] = dfs.pop('Межгод. динамика плотн. снега ')
if 'Межгод.динамика плотн. снега' in dfs:
    dfs['Межгод.динамика плотн.снега'] = dfs.pop('Межгод.динамика плотн. снега')
# Ошибки наименований 2
if 'Сезон. динам. глубины протаивая' in dfs:
    dfs['Сезон.динам.глубины протаивания'] = dfs.pop('Сезон. динам. глубины протаивая')
if 'Сезон.динам глубины протаивания' in dfs:
    dfs['Сезон.динам.глубины протаивания'] = dfs.pop('Сезон.динам глубины протаивания')
if 'Сезон. динам. плот. выс. снега' in dfs:
    dfs['Сезон.динамика плотн.выс.снега'] = dfs.pop('Сезон. динам. плот. выс. снега')
if 'Сезон.динамика плотн.выс. снега' in dfs:
    dfs['Сезон.динамика плотн.выс.снега'] = dfs.pop('Сезон.динамика плотн.выс. снега')
# Ошибки наименований 3
if 'Физические и теплофизические...' in dfs:
    dfs['Физические и теплофизические'] = dfs.pop('Физические и теплофизические...')
if 'Физические и еплофизические...' in dfs:
    dfs['Физические и теплофизические'] = dfs.pop('Физические и еплофизические...')
key_list = list(dfs.keys())
print(f'Update sheets names:\n\n{key_list}').
```

После выполнения кода все наименования листов в документе стали корректными (рис. 4).

Update sheets names:

```
['Температура грунтов', 'Влажность грунтов', 'Сезон.дина  
м.глубины протаивания', 'Межгодовая изменчивость СТС', 'Ме  
сячная высота снега', 'Межгод.динамика плотн.снега', 'Се  
зон.динамика плотн.выс.снега', 'Литологический разрез',  
'Физические и теплофизические']
```

Рис. 4. Обновленные наименования листов

Теперь можно приступить к обработке непосредственно самих листов excel-файла.

Первым листом является «Температура грунтов», который содержит информацию о дате и глубины замера, а также непосредственно сами значения замеров (рис. 5).

Температура грунтов, оС	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10	Unnamed: 11	Ur
0	NaN	Глубина, м	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	Дата	0	п/п	0.15	0.3	0.5	1.0	1.25	1.5	1.75	2.0	3.00
2	1985-06-01 00:00:00	NaN	NaN	NaN	NaN	NaN	-2.1	NaN	NaN	NaN	-2.8	-3.00
3	1985-06-05 00:00:00	NaN	NaN	NaN	NaN	NaN	-2.2	NaN	NaN	NaN	-2.8	-2.90
4	1985-06-10 00:00:00	NaN	NaN	NaN	NaN	NaN	-1.5	NaN	NaN	NaN	-2.3	-2.70
...	...	...	...	...	...	...	...	...	...	...	...	...

Рис. 5. Лист «Температура грунтов»

Для успешной обработки данных необходимо учитывать особенности их представления. В частности, в некоторых записях столбца, содержащего глубину залегания термодатчиков, встречается обозначение «п/п». Оно не является отсутствующим значением NaN (Not a Number), а представляет собой специальную метку, указывающую на размещение датчика непосредственно под снежным покровом. В процессе подготовки данных «п/п» следует заменить на соответствующие числовые значения глубины в соответствии с характеристиками площадок.

Следует также учитывать, что в таблице встречаются пропущенные значения, обозначенные как NaN. В данном контексте NaN означает отсутствие измерения конкретного параметра в заданный момент времени на определенной глубине. Дополнительно требуется привести столбец, содержащий даты, к единому формату, а также проверить данные на наличие дублирующихся записей и удалить их в случае обнаружения. Для этого был написан универсальный код, который выполняет все вышеперечисленное и подходит для всех листов «Температура грунтов» в датасете в целом:

```
df1 = dfs['Температура грунтов']
# Словарь для замены значений «п/п» у определенных площадок
dict_pp_values = {'Площадка 9_Чабыда': 0.08,
                  'Площадка 10_Чабыда': 0.03,
                  'Площадка 11_Чабыда': 0.02}
if state_name in dict_pp_values:
    df1.replace({'п/п': dict_pp_values[state_name]}, inplace=True)
dict_pkr_values = {'Площадка 8_Чабыда': 0.06}
if state_name in dict_pkr_values:
    df1.replace({'под пкр': dict_pkr_values[state_name]}, inplace=True)
date_indices = df1[df1.apply(lambda row: row.astype(str).str.contains('Дата').any(),
axis=1)].index[1:]
df1 = df1.drop(date_indices)
df1 = df1.iloc[1:]
df1 = df1.reset_index(drop=True)
df1.columns = df1.iloc[0]
df1 = df1.iloc[1:]
df1 = df1.reset_index(drop=True)
df1 = df1.rename(columns={'Дата ': 'Дата'})
df1['Дата'] = pd.to_datetime(df1['Дата'])
for column in df1.columns:
```

```

if column != 'Дата':
    df1[column] = df1[column].astype(float)
def drop_duplicate(df, sort_columns):
    """ Remove duplicate rows and sort the dataframe. """
    duplicate_df = df[df.duplicated()]
    if not duplicate_df.empty:
        print(f'Duplicate rows : \n{duplicate_df}\n')
        df = df.drop_duplicates(keep='last')
    else:
        print('Duplicate rows if not found')
    df = df.sort_values(by=sort_columns)
    df = df.reset_index(drop=True)
    return df
df1 = drop_duplicate(df1, 'Дата')
df1.

```

После выполнения кода переменная df1 содержит обработанный лист «Температура грунтов», который легко читается и воспринимается человеком (рис. 6).

	Дата	0	0.08	0.15	0.3	0.5	1.0	1.25	1.5	1.75	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0
0	1985-06-01	NaN	NaN	NaN	NaN	NaN	-2.1	NaN	NaN	NaN	-2.8	-3.00	NaN	-2.8	-1.7	-1.6	-1.4	-1.4	-1.4
1	1985-06-05	NaN	NaN	NaN	NaN	NaN	-2.2	NaN	NaN	NaN	-2.8	-2.90	NaN	-2.8	-1.6	-1.6	-1.4	-1.4	-1.4
2	1985-06-10	NaN	NaN	NaN	NaN	NaN	-1.5	NaN	NaN	NaN	-2.3	-2.70	NaN	-2.8	-1.7	-1.6	-1.5	-1.5	-1.4
3	1985-06-15	NaN	NaN	NaN	NaN	NaN	-1.3	NaN	NaN	NaN	-2.5	-2.70	NaN	-2.7	-1.6	-1.6	-1.5	-1.5	-1.4
4	1985-06-19	NaN	NaN	NaN	NaN	NaN	-1.1	NaN	NaN	NaN	-2.4	-2.70	NaN	-2.7	-1.6	-1.6	-1.5	-1.5	-1.4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
497	2022-08-19	NaN	NaN	NaN	NaN	NaN	4.1	NaN	NaN	NaN	-1.2	-1.60	-2.1	-2.5	-2.7	-2.5	-2.4	-2.4	NaN
498	2022-09-18	NaN	NaN	NaN	NaN	NaN	1.9	NaN	NaN	NaN	-0.9	-1.30	-1.8	-2.2	-2.4	-2.3	-2.3	-2.5	NaN
499	2022-10-18	NaN	NaN	NaN	NaN	NaN	0.7	NaN	NaN	NaN	0.9	0.60	0.0	-0.2	-0.3	-0.3	-0.4	-0.4	NaN
500	2022-11-16	NaN	NaN	NaN	NaN	NaN	-1.5	NaN	NaN	NaN	-0.1	0.00	0.0	-0.2	-0.3	-0.3	-0.4	-0.4	NaN
501	2022-12-14	NaN	NaN	NaN	NaN	NaN	-5.3	NaN	NaN	NaN	-1.1	-0.04	-0.1	-0.2	-0.3	-0.3	-0.4	-0.4	NaN

502 rows × 19 columns

Рис. 6. Обработанные данные листа «Температура грунтов»

Второй лист «Влажность грунтов» содержит аналогичную информацию, что и в листе «Температура грунтов». Однако, необходимо выполнить большее количество преобразований, таких как удаление пустых строк, переименование столбца «Дата определения» в «Дата» и преобразования их в формат даты, а также заменить буквенные значения «Лед» на числовые, выполнить поиск дубликатов и их удаление (рис. 7).

	Влажность грунтов, %	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	...	Unnamed: 17	Unnamed: 18	Unnamed: 19	Unnamed: 20
0	NaN	Глубина, м	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
1	Дата определения	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	...	1.6	1.7	1.8	1.9
2	1985-06-09 00:00:00	31.5	22.9	15.6	19.6	20.0	16.0	16.7	17.0	19.6	...	27.4	33.0	30.5	44.2
3	1985-08-30 00:00:00	46.4	17.4	16.0	NaN	12.8	NaN	8.7	NaN	11.2	...	NaN	NaN	NaN	NaN
4	1985-09-16 00:00:00	35	9.7	7.5	NaN	12.6	NaN	13.9	NaN	10.0	...	NaN	NaN	NaN	NaN
5	1986-06-16 00:00:00	21.9	19.4	14.2	15.2	20.9	NaN	19.4	NaN	NaN	...	NaN	NaN	NaN	NaN
6	1986-07-17 00:00:00	12.3	13.8	15.3	18.9	18.6	NaN	15.8	NaN	22.9	...	NaN	NaN	NaN	NaN
7	1986-08-27 00:00:00	7.8	5.8	6.6	12.2	15.2	15.2	11.7	NaN	14.4	...	NaN	NaN	NaN	NaN
8	1986-09-28 00:00:00	23.2	16.1	17.2	13.7	13.4	16.0	17.9	NaN	18.8	...	NaN	NaN	NaN	NaN
9	1990-10-02 00:00:00	16.1	11.2	10.8	NaN	16.3	NaN	20.4	NaN	23.0	...	28.6	NaN	NaN	NaN
10	1991-04-04 00:00:00	48.5	22.5	22.4	NaN	21.5	NaN	26.0	NaN	28.9	...	46.1	NaN	NaN	NaN
11	2002-09-13 00:00:00	NaN	NaN	15.8	NaN	11.5	NaN	12.0	NaN	8.3	...	NaN	NaN	NaN	NaN

12 rows × 27 columns

Рис. 7. Лист «Влажность грунтов»

Для этого был написан код, который выполняет все эти преобразования для листов «Влажность грунтов»:

```
df2 = dfs['Влажность грунтов']
df2 = df2.iloc[1:]
df2 = df2.reset_index(drop=True)
df2.columns = df2.iloc[0]
df2 = df2.iloc[1:]
df2 = df2.reset_index(drop=True)
df2 = df2.rename(columns={'Дата определения': 'Дата'})
df2 = df2.dropna(how='all')
df2 = df2.dropna(thresh=2)
if state_name == 'Площадка Лес_Туймаада':
    df2['Дата'] = df2['Дата'].astype(str)
    df2['Скважена'] = df2['Дата'].apply(lambda x: 1 if '*' in x else 2)
    df2['Дата'] = df2['Дата'].astype(str).str.replace(' *', '')
    df2['Дата'] = df2['Дата'].astype(str).str.replace('*', '')
    df2['Дата'] = df2['Дата'].apply(lambda x: x + ' 00:00:00' if len(x) == 10 else x)
    for index, row in df2.iterrows():
        try:
            df2.at[index, 'Дата'] = pd.to_datetime(row['Дата'], format='%d.%m.%Y %H:%M:%S').
strftime('%Y-%m-%d %H:%M:%S')
        except ValueError:
            pass
    if state_name == 'Площадка Луг_Туймаада':
        df2['Дата'] = df2['Дата'].astype(str)
        df2 = df2[~df2['Дата'].str.contains('Дата')]
        def fill_skvazhena(row):
            if '*' in row['Дата']:
                return '2'
            elif row['Дата'].count(' ') == 0 and not row['Дата'].isdigit():
                return '3'
            else:
                return '1'
        df2['Скважена'] = df2.apply(fill_skvazhena, axis=1)
        df2['Дата'] = df2['Дата'].astype(str).str.replace(' *', '')
        df2['Дата'] = df2['Дата'].astype(str).str.replace('*', '')
        df2['Дата'] = df2['Дата'].apply(lambda x: x + ' 00:00:00' if len(x) == 10 else x)
        for index, row in df2.iterrows():
            try:
                df2.at[index, 'Дата'] = pd.to_datetime(row['Дата'], format='%d.%m.%Y %H:%M:%S').
strftime('%Y-%m-%d %H:%M:%S')
            except ValueError:
                pass
    months_dict = {'Январь': 'January',
                    'Февраль': 'February',
                    'Март': 'March',
                    'Апрель': 'April',
                    'Май': 'May',
                    'Июнь': 'June',
                    'Июль': 'July',
                    'Август': 'August',
                    'Сентябрь': 'September',
                    'Октябрь': 'October',
                    'Ноябрь': 'November',
                    'Декабрь': 'December'}
    months_dict_number = {'January': '01',
                           'February': '02',
                           'March': '03',
                           'April': '04',
                           'May': '05',
```

```

        'June': '06',
        'July': '07',
        'August': '08',
        'September': '09',
        'October': '10',
        'November': '11',
        'December': '12'}

df2['Год'] = np.where(df2['Скважена'] == '3', '1979', np.nan)
df2['Дата'].replace(months_dict, inplace=True, regex=True)
df2['Дата'].replace(months_dict_number, inplace=True, regex=True)
df2.loc[df2['Скважена'] == '3', 'Дата'] = df2['Год'] + '-' + df2['Дата'] + '-01 00:00:00'
df2 = df2.drop(['Год'], axis=1)
df2 = df2.sort_values(by='Дата')
df2 = df2.reset_index(drop=True)
df2['Дата'] = pd.to_datetime(df2['Дата'])
# Замена значения «Лед» на «-1.0»
df2.replace('Лед', -1.0, inplace=True)
for column in df2.columns:
    if column != 'Дата':
        df2[column] = df2[column].astype(float)
    if 'Скважена' in df2.columns:
        df2['Скважена'] = df2['Скважена'].astype(int)
df2 = drop_duplicate(df2, 'Дата').

```

После выполнения кода переменная df2 хранит обработанный лист «Влажность грунтов», представленный в читаемом для человека виде (рис. 8).

	Дата	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	...	1.6	1.7	1.8	1.9	2.0	2.2	2.4	2.6	2.8	3.0
0	1985-06-09	31.5	22.9	15.6	19.6	20.0	16.0	16.7	17.0	19.6	...	27.4	33.0	30.5	44.2	71.1	31.4	28.3	18.5	19.2	20.9
1	1985-08-30	46.4	17.4	16.0	NaN	12.8	NaN	8.7	NaN	11.2	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	1985-09-16	35.0	9.7	7.5	NaN	12.6	NaN	13.9	NaN	10.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	1986-06-16	21.9	19.4	14.2	15.2	20.9	NaN	19.4	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	1986-07-17	12.3	13.8	15.3	18.9	18.6	NaN	15.8	NaN	22.9	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	1986-08-27	7.8	5.8	6.6	12.2	15.2	15.2	11.7	NaN	14.4	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	1986-09-28	23.2	16.1	17.2	13.7	13.4	16.0	17.9	NaN	18.8	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	1990-10-02	16.1	11.2	10.8	NaN	16.3	NaN	20.4	NaN	23.0	...	28.6	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	1991-04-04	48.5	22.5	22.4	NaN	21.5	NaN	26.0	NaN	28.9	...	46.1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	2002-09-13	NaN	NaN	15.8	NaN	11.5	NaN	12.0	NaN	8.3	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

10 rows × 27 columns

Рис. 8. Обработанные данные листа «Влажность грунтов»

Подобным образом были обработаны оставшиеся семь листов по «Площадка 9_Чабыда» и остальные пятнадцать площадок по девять листов с данными в каждой из них. Для сохранения результатов проделанной работы была написана функция, которая создает директорию с подкаталогами для хранения обработанных данных по площадкам и соответствующим им листам (рис. 9) в таких форматах как CSV (Comma-Separated Values) — это один из наиболее распространенных форматов для представления табличной информации, простой и универсальный формат, который используется для хранения данных в виде текстовых файлов, основанный на разделении значений запятыми; JSON (JavaScript Object Notation) — стандартный текстовый формат для хранения и передачи структурированных данных; XLSX — это формат файла Microsoft Excel, который используется для хранения электронных таблиц.

```

11%|██████████| 1/9 [00:00<00:00, 9.31it/s]
Directory /home/user/data_farm/projects/rnf_yakutia/data/monitoring_data/Площадка 9_Чабыда/csv/ create
Directory /home/user/data_farm/projects/rnf_yakutia/data/monitoring_data/Площадка 9_Чабыда/json/ create
Directory /home/user/data_farm/projects/rnf_yakutia/data/monitoring_data/Площадка 9_Чабыда/xlsx/ create
100%|██████████| 9/9 [00:00<00:00, 42.49it/s]

```

Рис. 9. Сохранение обработанных данных в форматах CSV, JSON, XLSX

Также был написан код, который создает XLSX файл с общей информацией по площадкам:

```
xlsx_files = [f for f in os.listdir(f'{folder_path}/xlsx/') if f.endswith(«.xlsx»)]
xlsx_dfs = {}
# Чтение данных из каждого файла и добавление в словарь
for file in xlsx_files:
    file_path = os.path.join(f'{folder_path}/xlsx/', file)
    xl = pd.ExcelFile(file_path)
    for sheet_name in xl.sheet_names():
        df = xl.parse(sheet_name)
        if sheet_name not in xlsx_dfs:
            xlsx_dfs[sheet_name] = []
        xlsx_dfs[sheet_name].append(df)
    with pd.ExcelWriter(f'{folder_path}/xlsx/{state_name}.xlsx', engine='xlsxwriter') as writer:
        for sheet_name, dataframes in xlsx_dfs.items():
            for i, df in enumerate(dataframes):
                df.to_excel(writer, sheet_name=sheet_name, index=False).
```

После его выполнения, появляется файл с итоговой информацией по каждой из площадок (рис. 10).

Площадка	Широта_дмс	Долгота_дмс	Широта_дд	Долгота_дд	Количество_метрик	Тип_местности	Условия	Первая_дата	Последняя_дата
Площадка Луг_Туйм...	62° 00' 49" с.ш.	129° 39' 24" в.д.	62.0136	129.6567	9		Сливающейся мерзл...	1967-08-14	2022-12-14
Площадка Лес_Туйм...	61° 00' 46" с.ш.	129° 39' 11" в.д.	61.0128	129.6531	8		Сливающейся мерзл...	1967-08-16	2022-12-14
Площадка (Окавкин...	61° 57' 27" с.ш.	129° 24' 50" в.д.	61.9575	129.4139	9	Мелководный	Сливающейся и не с...	1980-09-12	2022-12-14
Площадка 10_Чабыда	61° 57' 10" с.ш.	129° 25' 46" в.д.	61.9528	129.4294	9	Склонный	Сливающейся и не с...	1982-11-01	2022-12-14
Площадка 11_Чабыда	61° 57' 00" с.ш.	129° 25' 48" в.д.	61.9500	129.4300	9	Склонный	Сливающейся и не с...	1982-11-01	2022-12-14
Площадка 3_Чабыда	61° 57' 36" с.ш.	129° 24' 53" в.д.	61.9600	129.4147	9	Склонный	Сливающейся и не с...	1981-01-01	1989-01-01
Площадка 3a_Чабыда	61° 57' 26" с.ш.	129° 25' 17" в.д.	61.9572	129.4214	9	Склонный	Сливающейся и не с...	1982-11-01	2022-12-14
Площадка 4_Чабыда	61° 57' 24" с.ш.	129° 25' 27" в.д.	61.9567	129.4242	9	Склонный	Сливающейся и не с...	1980-07-02	1989-01-01
Площадка 5_Чабыда	61° 57' 28" с.ш.	129° 25' 03" в.д.	61.9578	129.4175	9	Склонный	Сливающейся и не с...	1980-07-02	2022-12-14
Площадка 6_Чабыда	61° 57' 36" с.ш.	129° 24' 53" в.д.	61.9600	129.4147	9	Склонный	Сливающейся и не с...	1982-10-01	1992-04-01
Площадка 6a_Чабыда	61° 57' 24" с.ш.	129° 25' 27" в.д.	61.9567	129.4242	9	Склонный	Сливающейся и не с...	1982-11-01	2022-12-14
Площадка 7_Чабыда	61° 57' 36" с.ш.	129° 24' 53" в.д.	61.9600	129.4147	9	Склонный	Сливающейся и не с...	1982-10-01	1991-04-02
Площадка 7a_Чабыда	61° 57' 26" с.ш.	129° 25' 17" в.д.	61.9572	129.4214	9	Склонный	Сливающейся и не с...	1982-11-01	2022-12-14
Площадка 8_Чабыда	61° 57' 17" с.ш.	129° 25' 13" в.д.	61.9547	129.4203	9	Мелководный	Сливающейся и не с...	1982-10-01	2022-12-14
Площадка 8a_Чабыда	61° 57' 14" с.ш.	129° 25' 09" в.д.	61.9540	129.4192	9	Мелководный	Сливающейся и не с...	1982-11-01	2022-12-14
Площадка 9_Чабыда	61° 57' 03" с.ш.	129° 05' 56" в.д.	61.9508	129.0989	9	Склонный	Сливающейся и не с...	1982-11-01	2022-12-14

Рис. 10. Файл с общей информацией по площадкам

Результаты исследования. Обработка и систематизация данных позволили привести разрозненные измерения к единому формату, обеспечивая их корректное использование в дальнейших расчетах. В результате получен уникальный массив данных, не имеющий аналогов, поскольку он основан на непосредственных эмпирических наблюдениях и охватывает специфические условия изучаемой территории.

Анализ данных с Python является востребованным навыком в современном мире. Этот язык отличается относительной простотой в изучении, что обеспечивает его доступность для начинающих. В работе были рассмотрены основные этапы обработки данных с использованием библиотек Python: предобработка и подготовка данных. На примере датасета «Площадка 9_Чабыда» были продемонстрированы основные приемы предобработки данных, такие как обнаружение и обработка пропущенных значений, трансформация данных, оптимизация, а также удаление несущественных атрибутов.

Обсуждение и заключение. Результаты показывают, что использование системного подхода к обработке данных позволяет значительно повысить качество и надежность анализа. Устранение пропусков и нормализация данных улучшает точность самих данных. Итоговая обработка данных предоставляет удобные форматы для последующего использования в моделировании, включая форматы CSV, JSON и XLSX. Это демонстрирует универсальность применения Python в решении задач обработки данных.

В дальнейшем планируется расширить исследования, включив в них методы анализа данных, такие как кластеризация, построение моделей машинного обучения и визуализация результатов [12, 14, 19]. Эти методы позволяют не только выявить скрытые закономерности в данных, но и предсказать поведение природно-технических систем при изменении внешних условий. Особое внимание будет уделено моделированию воздействия климатических факторов и антропогенной нагрузки на ПТС, что обеспечит более полное понимание процессов, происходящих в условиях Крайнего Севера.

Список литературы / References

1. Поручиков М.А. *Анализ данных*. Самара: издательство Самарского университета; 2016. 88 с. Poruchikov M. A. *Data analysis*. Samara: Samara University Press; 2016. 88 p. (In Russ.)
2. *Введение в анализ данных*. URL: https://mipt-stats.gitlab.io/courses/ad_fivt/titanik.html (дата обращения: 08.08.2024).

- Introduction to Data Analysis*. URL: https://mipt-stats.gitlab.io/courses/ad_fivt/titanik.html (accessed: 08.08.2024)
3. Лесковец Ю. *Анализ больших наборов данных*. Москва: ДМК; 2016. 498 с.
Leskovec Yu. *Analysis of Large Datasets*. Moscow: DMK; 2016. 498 p. (In Russ.)
4. МакКинли У. *Python и анализ данных*. Москва: ДМК; 2015. 482 с.
McKinney W. *Python and Data Analysis*. Moscow: DMK; 2015. 482 p. (In Russ.)
5. Макшанов А.В. *Технологии интеллектуального анализа данных*. Санкт-Петербург: Лань; 2018. 212 с.
Makshanov A.V. *Data Mining Technologies*. Saint Petersburg: Lan; 2018. 212 p. (In Russ.)
6. Миркин Б.Г. *Введение в анализ данных*. Люберцы: Юрайт; 2016. 174 с.
Mirkin B.G. *Introduction to Data Analysis*. Lyubertsy: Yurayt; 2016. 174 p. (In Russ.)
7. Рафалович В. *Data mining, или интеллектуальный анализ данных для занятых. Практический курс*. Москва: SmartBook; 2018. 352 с.
Rafalovich V. *Data Mining, or Data Analysis for the Busy: Practical Course*. Moscow: SmartBook; 2018. 352 p. (In Russ.)
8. Чашкин Ю.Р. *Математическая статистика. Анализ и обработка данных*. Ростов-на-Дону: Феникс; 2017. 236 с.
Chashkin Yu.R. *Mathematical Statistics. Data Analysis and Processing*. Rostov-on-Don: Feniks; 2017. 236 p. (In Russ.)
9. Грон А. *Машинное обучение и анализ данных с помощью Python*. Санкт-Петербург: Питер; 2019. 460 с.
Gron A. *Machine Learning and Data Analysis with Python*. Saint Petersburg: Piter; 2019. 460 p. (In Russ.)
10. Воронцов К.В. *Машинное обучение и интеллектуальный анализ данных*. Москва: издательство МФТИ; 2018. 328 с.
Vorontsov K.V. *Machine Learning and Data Mining*. Moscow: MIPT Publishing; 2018. 328 p. (In Russ.)
11. Алгозина Е.Р. *Обработка и анализ данных с Python и Pandas. Руководство для начинающих*. Москва: ДМК; 2020. 312 с.
Algozina E.R. *Data Processing and Analysis with Python and Pandas: Beginner's Guide*. Moscow: DMK; 2020. 312 p. (In Russ.)
12. Троян И.В. *Анализ данных и машинное обучение на Python*. Санкт-Петербург: БХВ-Петербург; 2019. 320 с.
Troyan I.V. *Data Analysis and Machine Learning in Python*. Saint Petersburg: BHV-Peterburg; 2019. 320 p. (In Russ.)
13. МакКинли У. *Программирование на Python и анализ данных*. Санкт-Петербург: Питер; 2020. 504 с.
McKinney W. *Python Programming and Data Analysis*. Saint Petersburg: Piter; 2020. 504 p. (In Russ.)
14. Мюллер А.К., Гвидо С. *Введение в машинное обучение с Python*. Санкт-Петербург: Питер; 2017. 420 с.
Mueller A.C., Guido S. *Introduction to Machine Learning with Python*. Saint Petersburg: Piter; 2017. 420 p. (In Russ.)
15. Рожков Р. С., Молчанова Е. С., Фадеев М. К., Пименов Н. А. Управление природно-техническими системами через концепцию устойчивого развития и приемлемого риска. *Вестник евразийской науки*. 2024;16(5):1–10.
Rozhkov R.S., Molchanova E.S., Fadeev M.K., Pimenov N.A. Managing Natural-Technical Systems through the Concept of Sustainable Development and Acceptable Risk. *Bulletin of Eurasian Science*. 2024;16(5):1–10. (In Russ.)
16. Дрегуло Н. Влияние климатических факторов на эксплуатацию ИТС, связанных с обработкой отходов водотока. URL: <https://pureportal.spbu.ru/publications/-----%28a9c91d8e-c5e0-43d9-98b5-8aac39cc4a61%29.html> (дата обращения: 13.01.2025)
Dregulo N. Influence of Climatic Factors on the Operation of Natural-Technical Systems for Wastewater Treatment. URL: <https://pureportal.spbu.ru/publications/-----%28a9c91d8e-c5e0-43d9-98b5-8aac39cc4a61%29.html> (accessed: 13.01.2025) (In Russ.)
17. Влияние антропогенных факторов на тепловое загрязнение городской среды. URL: https://www.abok.ru/for_spec/articles.php?nid=7635 (дата обращения: 13.01.2025).
Impact of Anthropogenic Factors on Thermal Pollution of the Urban Environment. URL: https://www.abok.ru/for_spec/articles.php?nid=7635 (accessed: 13.01.2025) (In Russ.)
18. Влияние антропогенных факторов на развитие сельского хозяйства. URL: <https://biohim.com.ru/articles/page2> (дата обращения: 13.01.2025).
Impact of Anthropogenic Factors on Agricultural Development. URL: <https://biohim.com.ru/articles/page2> (accessed: 13.01.2025) (In Russ.)
19. Левашкин С.П., Иванов К.Н., Кушуков С.В. Ферма данных: информационная система сбора, хранения и обработки неструктурированных данных из разнородных источников. *Труды Института системного программирования РАН*. 2023;35(2):57–72. [https://doi.org/10.15514/ISPRAS-2023-35\(2\)-5](https://doi.org/10.15514/ISPRAS-2023-35(2)-5)
Levashkin S.P., Ivanov K.N., Kushukov S.V. Data Farm: An Information System for Collecting, Storing, and Processing Unstructured Data from Heterogeneous Sources. *Proceedings of the Institute for System Programming of the RAS (Proceedings of ISP RAS)*. 2023;35(2):57–72. [https://doi.org/10.15514/ISPRAS-2023-35\(2\)-5](https://doi.org/10.15514/ISPRAS-2023-35(2)-5) (In Russ.)

Об авторах:

Сергей Владимирович Кушуков, инженер Поволжского государственного университета телекоммуникаций и информатики, Научно-исследовательской лаборатории «Искусственный интеллект» (443090, Российская Федерация, г. Самара, Московское ш., 77), [ORCID](https://orcid.org/0000-0001-9151-9151), [SPIN-код](https://spina.ru/author/s.kushukov), s.kushukov@psuti.ru

Константин Николаевич Иванов, инженер Поволжского государственного университета телекоммуникаций и информатики, Научно-исследовательской лаборатории «Искусственный интеллект» (443090, Российская Федерация, г. Самара, Московское ш., 77), [ORCID](#), [SPIN-код](#), k.ivanov@psuti.ru

Сергей Павлович Левашкин, кандидат физико-математических наук, доцент Поволжского государственного университета телекоммуникаций и информатики, заведующий Научно-исследовательской лаборатории «Искусственный интеллект» (443090, Российская Федерация, г. Самара, Московское ш., 77), [ORCID](#), [SPIN-код](#), ai_lab@psuti.ru

Михаил Владимирович Якобовский, член-корреспондент РАН, доктор физико-математических наук, директор Института прикладной математики имени М.В. Келдыша РАН (ИПМ РАН), (125047, Российская Федерация, г. Москва, Миусская пл., 4), [ORCID](#), [SPIN-код](#), lira@imamod.ru

Заявленный вклад авторов:

С.В. Кушук: разработка, тестирование существующих компонентов кода; написание черновика рукописи.

К.Н. Иванов: формальный анализ.

С.П. Левашкин: постановка задачи; формулировка идей исследования, целей и задач.

М.В. Якобовский: рецензирование и редактирование рукописи.

Конфликт интересов: *авторы заявляют об отсутствии конфликта интересов.*

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the Authors:

Sergey V. Kushukov, Engineer, Povolzhskiy State University of Telecommunications and Informatics, Research Laboratory “Artificial Intelligence” (77, Moskovskoe Shosse, Samara, 443090, Russian Federation), [ORCID](#), [SPIN-code](#), s.kushukov@psuti.ru

Konstantin N. Ivanov, Engineer, Povolzhskiy State University of Telecommunications and Informatics, Research Laboratory “Artificial Intelligence” (77, Moskovskoe Shosse, Samara, 443090, Russian Federation), [ORCID](#), [SPIN-code](#), k.ivanov@psuti.ru

Sergey P. Levashkin, Candidate of Physical and Mathematical Sciences, Associated Professor, Povolzhskiy State University of Telecommunications and Informatics, Head of Research Laboratory “Artificial Intelligence” (77, Moskovskoe Shosse, Samara, 443090, Russian Federation), [ORCID](#), [SPIN-code](#), ai_lab@psuti.ru

Mikhail V. Yakobovskiy, Corresponding Member of the Russian Academy of Sciences, Doctor of Physical and Mathematical Sciences, Director, Keldysh Institute of Applied Mathematics of the Russian Academy of Sciences (IPM RAS), (4, Miusskaya pl., Moscow, 125047, Russian Federation), [ORCID](#), [SPIN-code](#), lira@imamod.ru

Contributions of the authors:

S.V. Kushukov: development, testing of existing code components; writing a draft of a manuscript.

K.N. Ivanov: formal analysis.

S.P. Levashkin: problem statement; formulation of research ideas; goals and objectives.

M.V. Yakobovskiy: review and editing of the manuscript.

Conflict of Interest Statement: *the authors declare no conflict of interest.*

All authors have read and approved the final manuscript.

Поступила в редакцию / Received 15.05.2025

Поступила после рецензирования / Revised 02.06.2025

Принята к публикации / Accepted 18.06.2025