

Содержание

1	Введение	2
1.1	Назначения проекта	2
1.2	Требования к модулю прогнозирования и постановка задачи	2
1.3	Технические особенности проекта	2
1.4	Структура проекта	3
1.5	Анализ имеющихся данных	3
2	Теоретические сведения	7
2.1	Молоко и процесс пастеризации	7
2.2	Временные ряды и их прогнозирование	8
3	Модуль DataManipulator	11
3.1	Назначение модуля DataManipulator	11
3.2	Главное меню модуля DataManipulator.	11
3.3	Меню обработки данных OCDF-формата модуля DataManipulator.	11
3.3.1	Считывание OCDF-данных из файла.	11
3.3.2	Возможности по обработке OCDF-данных.	13
3.4	Обезка OCDF-данных.	14
3.4.1	Обрезка OCDF-данных по заданному проценту.	14
3.4.2	Обрезка OCDF-данных по заданному количеству.	15
3.4.3	Обрезка OCDF-данных по заданному моменту времени.	16
3.4.4	Возможные ошибки при обрезке OCDF-данных.	16
3.5	Парсинг OCDF-данных.	18
3.5.1	Процесс парсинга OCDF-данных.	18
3.5.2	Возможные ошибки при парсинге OCDF-данных.	18
3.6	Выравнивание диапазонов OCDF-данных.	19
3.6.1	Математические основы выравнивания диапазонов OCDF-данных.	20
3.6.2	Процесс выравнивания диапазонов OCDF-данных.	20
3.6.3	Возможные ошибки при выравнивании диапазонов OCDF-данных.	21
3.7	Добавление OCDF-данных.	21
3.8	Визуализация OCDF-данных.	22
3.8.1	Управление визуализацией OCDF-данных.	23
3.9	Сохранение OCDF-данных в файл.	24
3.9.1	Сохранение OCDF-данных в файл формата csv.	24
3.9.2	Сохранение OCDF-данных в бинарный файл.	24
3.9.3	Возможные ошибки при сохранении OCDF-данных в файл.	25
3.10	Выход из меню работы с OCDF-данными.	25
3.11	Сохранение TDF-данных в файл.	25
3.11.1	Сохранение TDF-данных в файл формата csv.	26
3.11.2	Сохранение TDF-данных в бинарный файл.	26
3.12	Выход из меню работы с TDF-данными.	26

1 Введение

1.1 Назначения проекта

Проект предназначен для прогнозирования данных временных рядов технологического процесса пастеризационной установки в режиме реального времени и развёртывания его на программируемом микроконтроллере архитектуры PLCnext. Данный проект является модулем прогнозирования данных временных рядов технологического процесса пастеризационной установки.

1.2 Требования к модулю прогнозирования и постановка задачи

Модуль прогнозирования должен выполнять прогноз технологического процесса пастеризационной установки. Для этого ему необходимо получать данные технологического процесса пастеризационной установки с помощью программируемого микроконтроллера, использующего платформу PLCnext Technology. По этой причине, программа должна быть написана на низкоуровневом языке программирования, а именно на C++, который поддерживается данным микроконтроллером.

Поскольку процесс пастеризации происходит непрерывно, то программа должна работать в режиме реального времени и своевременно снабжать необходимой информацией о прогнозах поведения пастеризационной установки.

Программа должна быть устойчива к различным сбоям и авариям, а также не создавать их сама.

Для постановки задачи, необходимо понимать, что сама программа, которая будет в дальнейшем использоваться для прогнозирования данных временных рядов технологического процесса пастеризационной установки, будет использовать нейросетевое решение задачи прогнозирования. Для обучения модели нейронной сети применяются алгоритмы машинного обучения, от которых зависит качество модели прогнозирования. Оттого данная работа в основном и будет развиваться по этапам инженерии машинного обучения, и потому данная работа будет являться проектом машинного обучения.

Задачей для проекта машинного обучения будет являться разработка модуля прогнозирования данных временных рядов пастеризационной установки с целью прогноза поведения пастеризационной установки, планирования производства, а также выявления аномального поведения, возможных сбоев и выбросов пастеризационной установки.

Для выполнения поставленной задачи необходимо изучать данные технологического процесса, а также провести их анализ и обработку. Далее необходимо выбрать подходящую архитектуру нейронной сети, построить и обучить модель сети, после чего организовать эффективное обучение сети, используя алгоритмы машинного обучения. Написать код нейронной сети необходимо на низкоуровневом языке программирования для дальнейшего развёртывания на программируемом контроллере, взаимодействующим с пастеризационной установкой.

Также необходимо произвести тестирование и оценку качества полученной модели. После чего непосредственно выполнить процесс развёртывания нейронной сети.

1.3 Технические особенности проекта

Проект написан в основном на низкоуровневом языке программирования C++20 с использованием кроссплатформенной утилиты Smake.

Проект использует также и высокоуровневый язык программирования Python для визуализации данных. В случае, если поддержки Python не будет, то не будет возможности визуализировать данные.

Документация к проекту создана с помощью XeLaTeX.

1.4 Структура проекта

1.5 Анализ имеющихся данных

Модуль DataManipulator предназначен для различного рода обработки данных, с целью получения наборов данных (датасетов) необходимых для обучения моделей нейронных сетей. DataManipulator помогает преодолеть этап анализа и обработки данных, а также этап конструирования признаков.

Данные сидов: 1 2 3 4 5 6

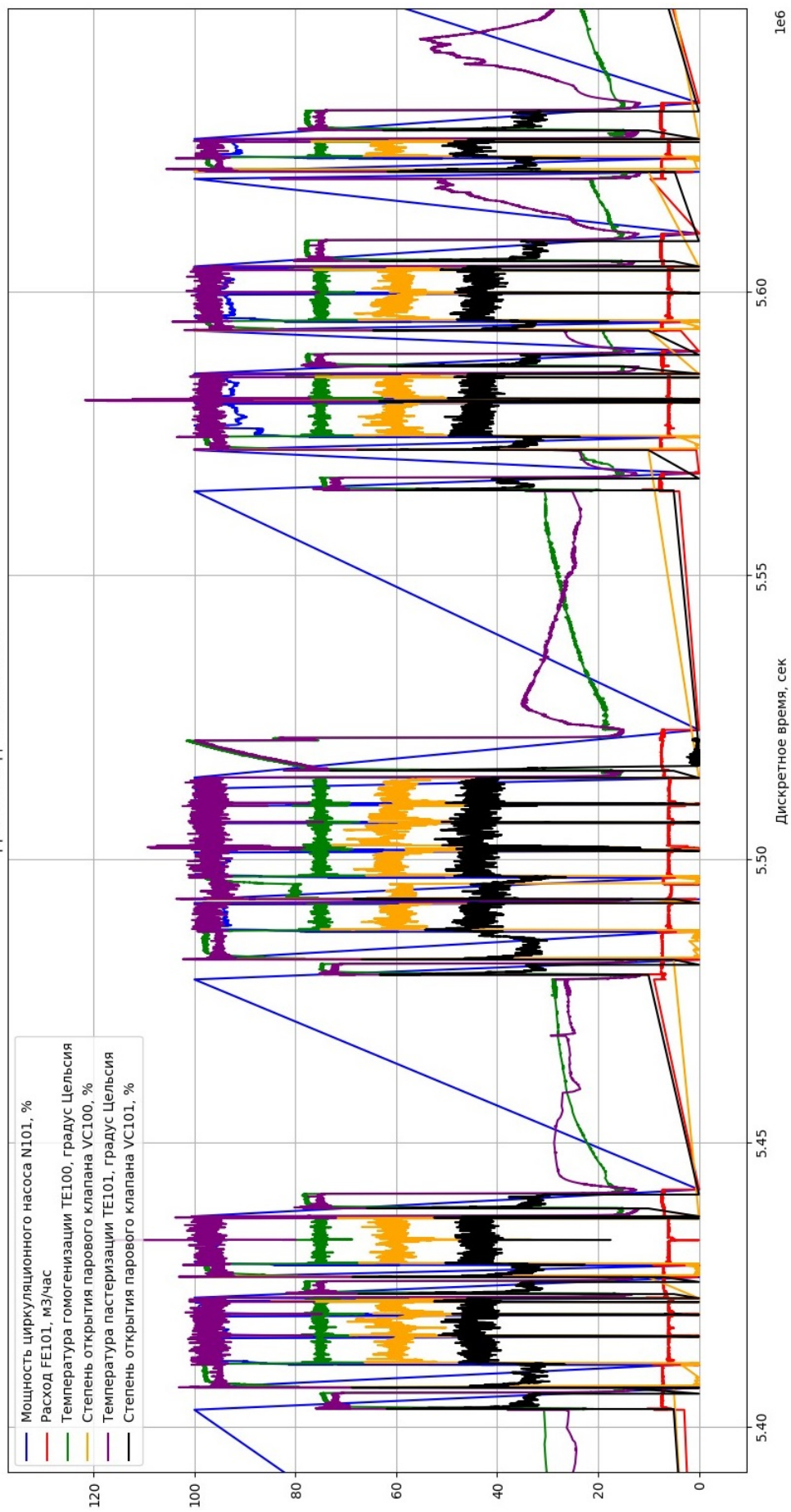


Рис. 1: Визуализация данных 1-ого файла

Данные сидов: 1 2 3 4 5 6

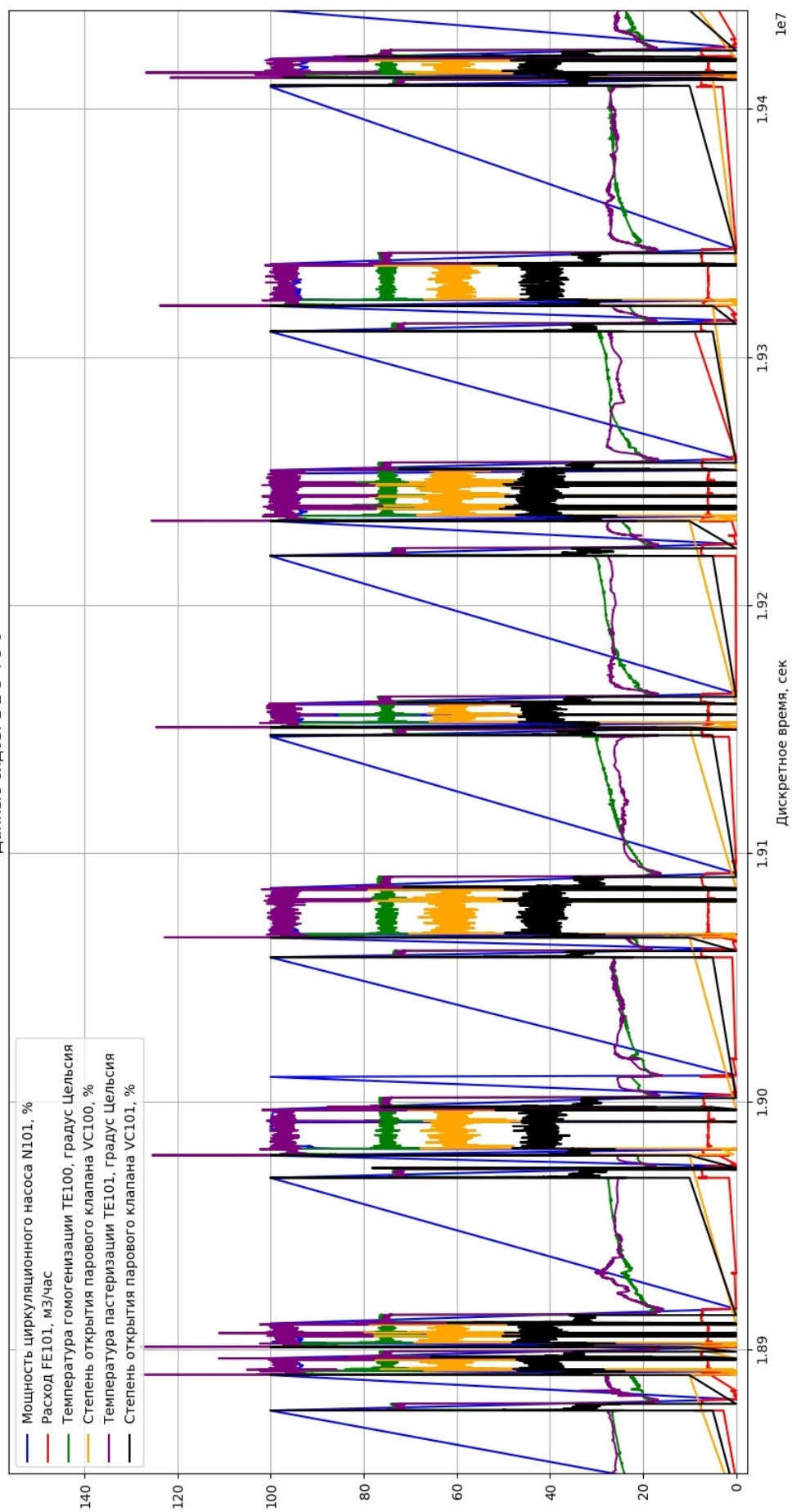


Рис. 2: Визуализация данных 2-ого файла с растянутым масштабом по оси абсцисс

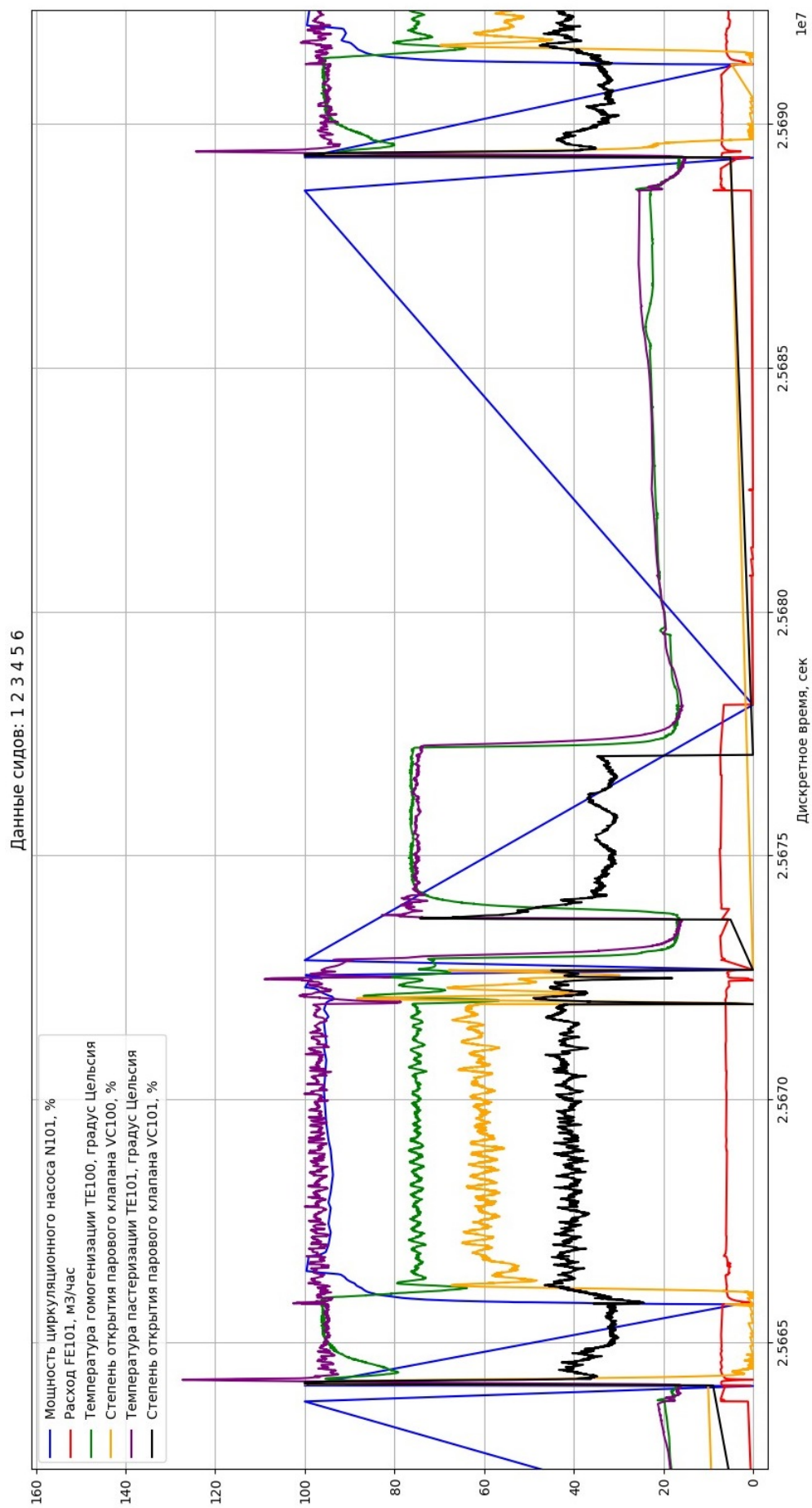


Рис. 3: Визуализация данных 3-го файла с суженным масштабом по оси абсцисс

2 Теоретические сведения

2.1 Молоко и процесс пастеризации

Объектом исследования и внедрения нейронной сети является пастеризационная установка. Для понимания её технологического процесса, необходимо для начала понять с чем она работает и что она производит, а также какими функциями обладает пастеризационная установка. Поэтому в данной главе мы будем рассматривать пастеризационную установку как некий «серый ящик»: опишем объекты, которые попадают на вход и выходят из пастеризационной установки, а также опишем самые основные процессы, которые происходят с подаваемым на вход объектом.

И так, основной объект обработки пастеризационной установки является молоко. Поэтому в дальнейшем молоко будет рассматриваться как объект технической обработки. Рассматривая его таким образом, мы понимаем, что оно должно обладать некоторыми показателями, например, состав молока, степень чистоты, кислотность, наличие токсичных и нейтрализующих веществ. При этом молоко обладает ещё и различными свойствами: органолептическими, физико-механическими и биохимическими. И так, разберём лишь те параметры и свойства, которые будут нам в дальнейшем интересны.

Молоко можно разделить на две составляющих: вода и распределённые в этой воде пищевые вещества. К таким веществам относят жиры, белки, углеводы, ферменты, различные минеральные вещества и газы. Помимо этого, в молоке могут находиться различные микроорганизмы. И как известно, некоторые из этих микроорганизмов, содержащиеся в молоке, являются опасными или вредными, например, бруцеллеза, ящура, возбудитель кишечной палочки и другие. Но как избавиться от вредных и опасных микроорганизмов? Для этого используется процесс пастеризации – уничтожение различных форм вредных и опасных микроорганизмов в молоке. Но при этом молоко должно сохранить свою биологическую и питательную ценность, а также и своё качество.

Однако перед тем, как перейти к пастеризации, необходимо определить, какое молоко можно пастеризовать, каким требованиям оно должно соответствовать. Для определения подходящего перед пастеризацией молока существует множество различных показателей и требований. Так, например, пригодное для пастеризации молоко должно быть кислотностью не более 22 °Т, а бактериальная обсеменённость молока должна быть один миллион клеток на сантиметр кубический. При этом, молоко не должно быть вспененным. Перед пастеризацией, молоко должно быть также предварительно очищено на фильтрах или на сепараторах-молокоочистителях. Что ж, основные требования перечислены.

А теперь к самой пастеризации. Основные параметры пастеризации есть температура пастеризации, а также время выдержки, то есть время нахождения молока в данном процессе. Относительно данных параметров существует выражение, выведенное Г. А. Куком и называемая критерием Пастера. Этот критерий можно рассчитать по формуле 1.

$$P = \frac{t}{p} \quad (1)$$

где t – время действия температуры пастеризации, с;

p – время бактерицидного действия температуры пастеризации, с.

Что такое бактерицидное действие температуры пастеризации? Это, как раз и есть эффект, в результате которого происходит уничтожение вредных и опасных микроорганизмов.

Также известна и ещё одна немаловажная зависимость: продолжительность выдержки зависит от температуры пастеризации. Зависимость показана в формуле 2.

$$\ln t = 36.84 - 0.48T \quad (2)$$

где T – температура пастеризации, °C.

Завершение процесса пастеризации характеризуется полным уничтожением содержащихся в молоке микроорганизмов. Это можно будет определить благодаря уже известному критерию Пастера, значение которого должно быть не меньше единицы, для того чтобы считать, что процесс пастеризации завершён.

Перед тем, как перейти к описанию пастеризационной установки, следует коротко разобрать ещё одно понятие, с которым мы будем сталкиваться в дальнейшем, а именно гомогенизация. Она представляет собой процесс дробления или уничтожения жировых шариков, образовавшихся в ходе хранения молока. Под воздействием внешних сил можно достичь значительного уменьшения объёма жировых шариков. Процесс гомогенизации позволяет предотвратить самопроизвольное отстаивание жира в молоке на производстве или при его хранении. При этом, гомогенизация даёт возможность сохранить однородную консистенцию молока. Далее рассматривать гомогенизацию так подробно, как процесс пастеризации, не имеет смысла, поскольку процесс гомогенизации не является центральным понятием предметной области.

2.2 Временные ряды и их прогнозирование

Одним из ключевых понятий данной работы является понятие временных рядов. Что же это? Временные ряды – это, по сути, некоторая последовательность, каждый элемент из которой состоит из двух или более параметров, а один из них обязательно должен обозначать время. Причём, все эти элементы в последовательности расположены в хронологическом порядке, то есть в порядке возрастания параметра времени.

Параметр времени может быть представлен в разных форматах. Выбор формата времени зависит от задачи, удобства использования, длительности, в пределах которой будут собираться данные, а также от требуемой точности. Например, в случае если данные фиксируются раз в день, то хорошо подойдёт отсчёт времени по дням с указанием месяца и года. Если же данные фиксируются в определённые моменты дня, то к вышеописанному стоит прибавить указание часа и минуты фиксации. При необходимости можно указывать и секунды, и миллисекунды. Но что, если не особо-то и важно знать, в какой год, месяц или день это происходило, когда важно знать, сколько прошло времени от начала того или иного процесса? Тогда, скорее подойдёт формат дискретного времени. С помощью этого формата можно узнать длительность процесса в единственной выбранной нами единице измерения времени. Например, если сохранять время в секундах, то 1000-ча секунд сохранит свой формат 1000-чи секунд, время не будет переведено в 16-ать минут и 40-ок секунд. Всё это позволяет не привязываться к датам, которые не особо-то и влияют на технологический процесс промышленного оборудования.

Остальные параметры могут уже характеризовать или описывать какой-либо процесс или какие-либо процессы, причём даже не обязательно одного элемента, а даже целой системы элементов. Так, например, когда элемент последовательности состоит из двух признаков, а один из которых, как уже известно, время, то второй, конечно же, уже будет обозначать характеристику или состояние изучаемого нами элемента. Но как только появляется три или более признаков, тогда уже можно говорить о фиксации характеристик или

состояний разных элементов изучаемой системы в один и тот же момент времени или же о фиксации характеристики или состояния одного из множества элементов системы, но с указанием этого элемента, например, с помощью идентификационного номера элемента. Как можно убедиться, временные ряды дают весьма гибкую возможность описания процессов или систем относительно времени.

Технологический процесс пастеризационной установки как раз и представляет собой временной ряд, в котором имеется информация о показаниях различных датчиков в определённые моменты времени. Поэтому, говоря о данных технологического процесса пастеризационной установки, мы будем понимать, что они имеют форму временных рядов.

А что из себя представляет работа с временными рядами? В основном работа делится на две части. Первая часть – это понимание структуры временного ряда, его закономерностей, таких как цикличность, тренд, сезонность и так далее, обработка данных временного ряда, визуализация данных, в общем, это всесторонний анализ временного ряда. Если опускать различные математические, статистические и тому подобные подробности, то анализ также может дать нам возможность понять, как начинался процесс, как шёл, развивался и на чём он закончился или остановился на данном моменте времени. Строго говоря, нейронным сетям, как и исследователям временных рядов, тоже необходимо это понять, чтобы выполнить вторую часть работы, а именно, составление прогноза, что, зачастую, и является основной задачей работы с временным рядом. Да, анализ данных временного ряда технологического процесса делается с целью понять, что будет происходить с этим процессом дальше. Но чем нам так полезна информация о будущем, зачем она нужна? Для этого необходимо понять, что есть прогноз.

Сам прогноз – это некоторая случайная величина, характеризующая вероятность того, что график в будущем пройдёт через определённую точку или некоторую область. Тогда прогнозирование – это получение максимально точных прогнозов, или, если говорить в отрыве от понятия прогноза, то это точное предсказание будущего, учитывающее исторические данные об объекте прогнозирования, а также знания о любых будущих событиях, которые могут повлиять на прогнозы.

Как понять, что прогнозы действительные? Прогнозы являются таковыми, если они отражают подлинные закономерности и взаимосвязи, которые есть в исторических данных, при этом не повторяя прошлые события, которые более не актуальны или уже не повторяются.

Что необходимо, чтобы составить хороший прогноз на основе временных рядов? Для этого обычно требуется выполнить следующие шаги:

1. Определить задачу.
2. Собрать информацию.
3. Произвести предварительный анализ.
4. Выбрать и создать модель прогнозирования.
5. Использовать и оценить модель прогнозирования.

Вкратце разберём каждый пункт. При определении задачи необходимо понять, что вообще будет прогнозироваться, как будут использоваться прогнозы, благодаря чему будут получены прогнозы, кому эти прогнозы нужны или для чего. Второй пункт подразумевает непосредственный сбор или получение данных, а также оценка накопленного опыта людей, которые собирают данные и используют прогнозы. Для выполнения третьего пункта необходимо визуализировать данные, составить инфографику, если она необходима, определить взаимосвязь с признаками, закономерности временных рядов, качество данных и так далее, другими словами, провести анализ данных. На четвёртом пункте необходимо

либо приобрести и адаптировать готовую модель, либо создать её самостоятельно. Но модель должна учитывать исторические данные, силы взаимосвязи между прогнозируемым признаком и любым другим признаком, а также способы использования прогнозов. В заключении проводится тестирование, оценивание, развёртывание и сопровождение модели прогнозирования.

3 Модуль DataManipulator

3.1 Назначение модуля DataManipulator

Модуль DataManipulator предназначен для различного рода обработки данных, с целью получения наборов данных (датасетов) необходимых для обучения моделей нейронных сетей. DataManipulator помогает преодолеть этап анализа и обработки данных, а также этап конструирования признаков.

3.2 Главное меню модуля DataManipulator.

В главном меню модуля DataManipulator представлены основные направления для обработки данных. Предоставляется выбор, с каким форматом данных будет происходить работа: с OCDF- или TDF-форматом. В главном меню также можно выбрать пункт, позволяющий создать TDF-данные на основе OCDF-данных. Также имеется пункт, который выводит на экран краткую информацию о форматах данных и о самом модуле DataManipulator. Внешний вид меню можно увидеть на рисунке 4

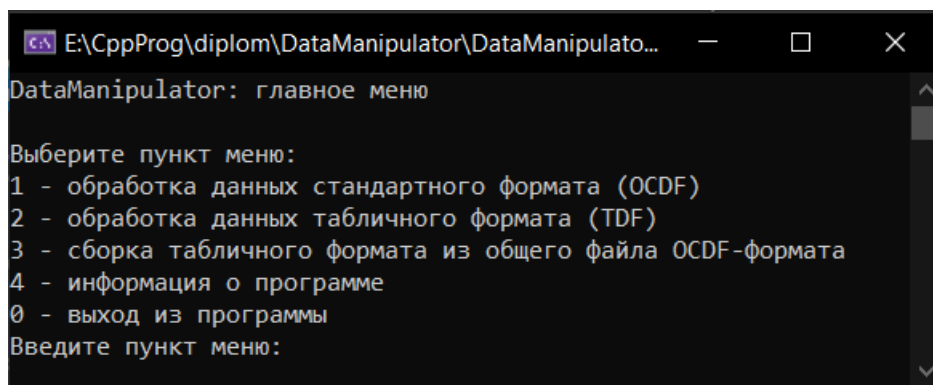


Рис. 4: главное меню модуля DataManipulator

3.3 Меню обработки данных OCDF-формата модуля DataManipulator.

3.3.1 Считывание OCDF-данных из файла.

Для того, чтобы попасть в меню по обработке данных OCDF-формата будет предложено для начала их загрузить из файла. Это показано на рисунке на рисунке 5.

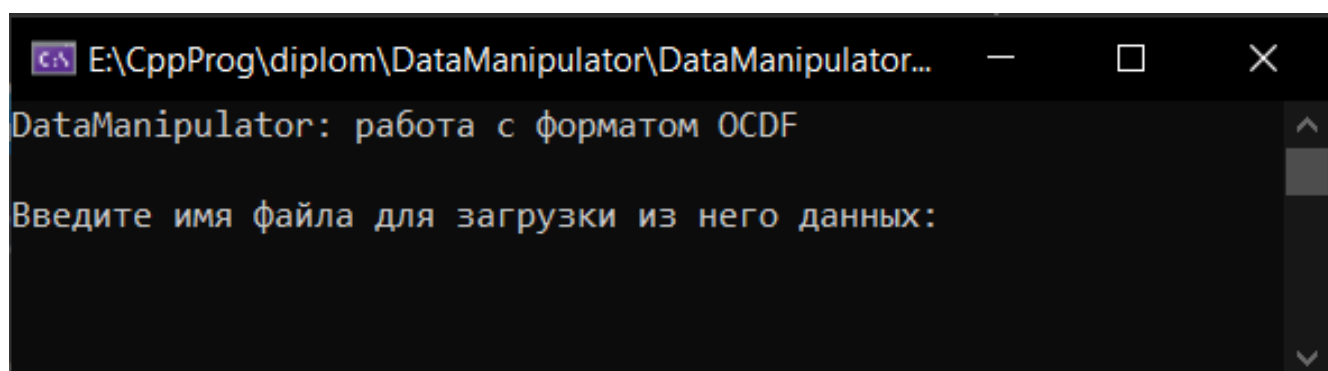
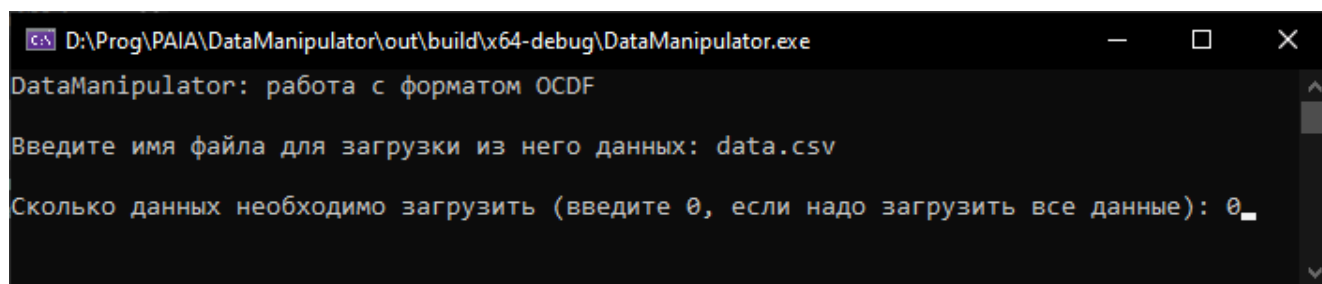


Рис. 5: указание файла для загрузки из него OCDF-данных.

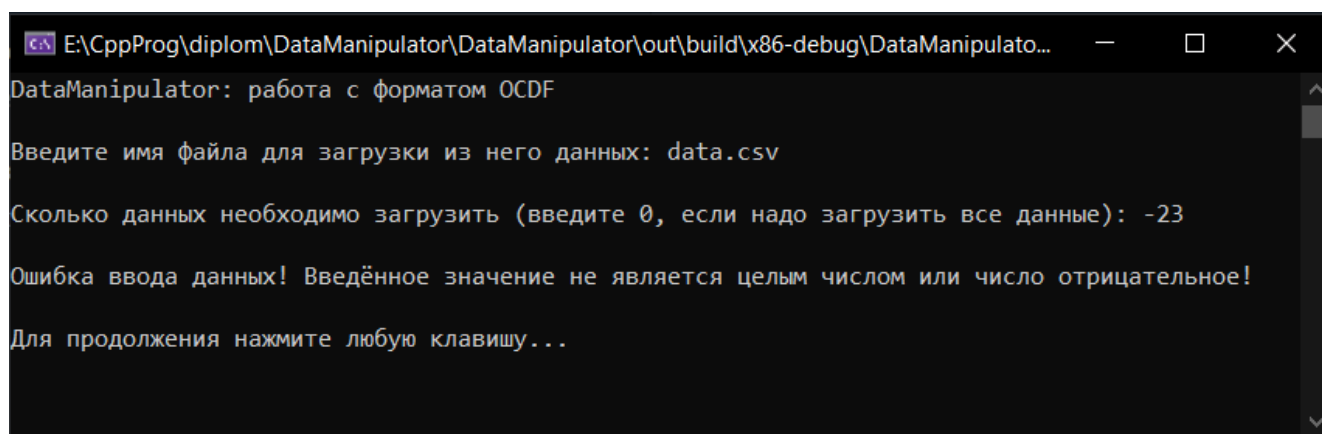
Вводим путь к файлу или перетаскиваем файл в консоль для получения его абсолютного пути и нажимаем Enter. После этого будет предложено ввести количество данных, которые необходимо считать из файла. Это показано на рисунке 6.



```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: работа с форматом OCDF
Введите имя файла для загрузки из него данных: data.csv
Сколько данных необходимо загрузить (введите 0, если надо загрузить все данные): 0
```

Рис. 6: указание необходимого количества данных для считывания.

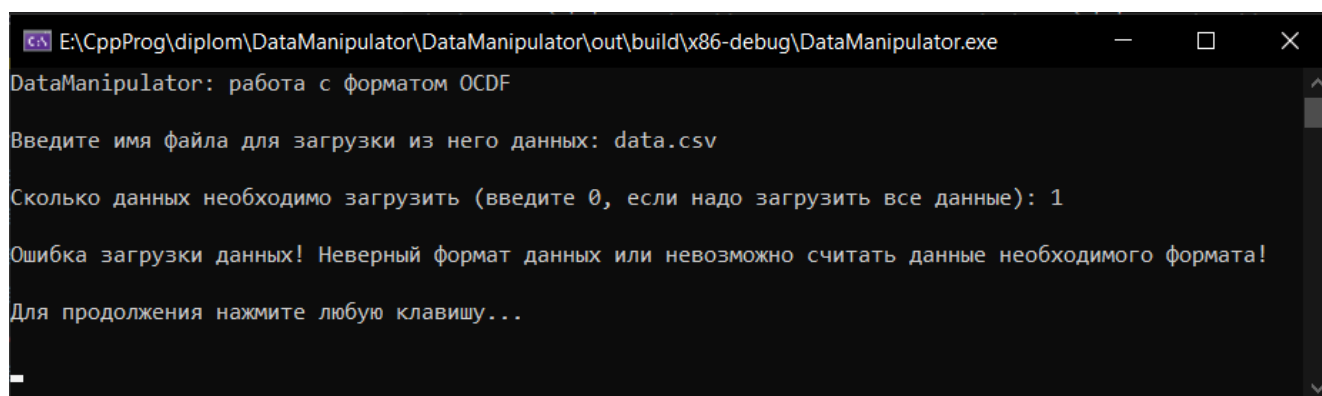
Есть несколько особенностей, при вводе количества данных. В случае, если будет введён 0 или число, превышающее количество данных в файле, то будут считаны все данные, которые есть в файле. Если будет введено отрицательное число или хотя бы один символ, отличный от цифр, то программа выдаст предупреждение о некорректном вводе данных, показанное на рисунке 7, и предложит ввести путь к файлу и необходимое количество данных для считывания снова.



```
E:\CppProg\diplom\DataManipulator\DataManipulator\out\build\x86-debug\DataManipulator.exe
DataManipulator: работа с форматом OCDF
Введите имя файла для загрузки из него данных: data.csv
Сколько данных необходимо загрузить (введите 0, если надо загрузить все данные): -23
Ошибка ввода данных! Введённое значение не является целым числом или число отрицательное!
Для продолжения нажмите любую клавишу...
```

Рис. 7: предупреждение об ошибке при вводе необходимого количества данных для считывания.

Могут возникнуть и другие предупреждения, связанные непосредственно с самим файлом. Если файл с данными не существует, или этот файл пуст, или данные в нём не представлены в необходимом формате, то возникнет предупреждение о невозможности чтения данных, представленное на рисунке 8.



```
E:\CppProg\diplom\DataManipulator\DataManipulator\out\build\x86-debug\DataManipulator.exe
DataManipulator: работа с форматом OCDF
Введите имя файла для загрузки из него данных: data.csv
Сколько данных необходимо загрузить (введите 0, если надо загрузить все данные): 1
Ошибка загрузки данных! Неверный формат данных или невозможно считать данные необходимого формата!
Для продолжения нажмите любую клавишу...
```

Рис. 8: предупреждение об ошибке чтения данных.

Как уже понятно, чтение данных происходит в текстовом режиме. Однако существует возможность считывать данные, которые представлены в бинарном виде. Для этого необходимо, чтобы файл имел расширение .bin, иначе чтение данных будет происходить в тек-

стовом режиме. Пример файла с данными в бинарном виде можно увидеть на рисунке 9.

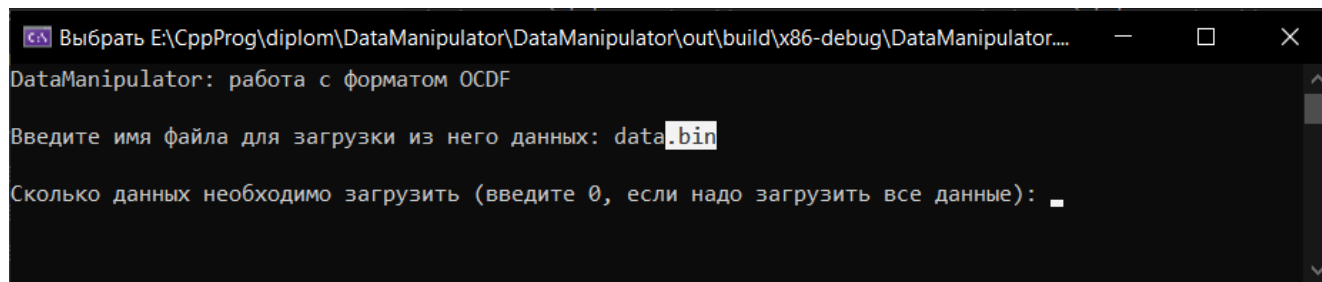


Рис. 9: ввод файла с данными в бинарном виде.

После указания файла с данными в бинарном виде будет предложено ввести количество данных, которое необходимо считать из файла. Все предупреждения, которые связаны с чтением данных и ввода значений уже разобраны выше.

3.3.2 Возможности по обработке OCDF-данных.

После считывания OCDF-данных из файла открывается меню для работы с данными, представленное на рисунке 10.

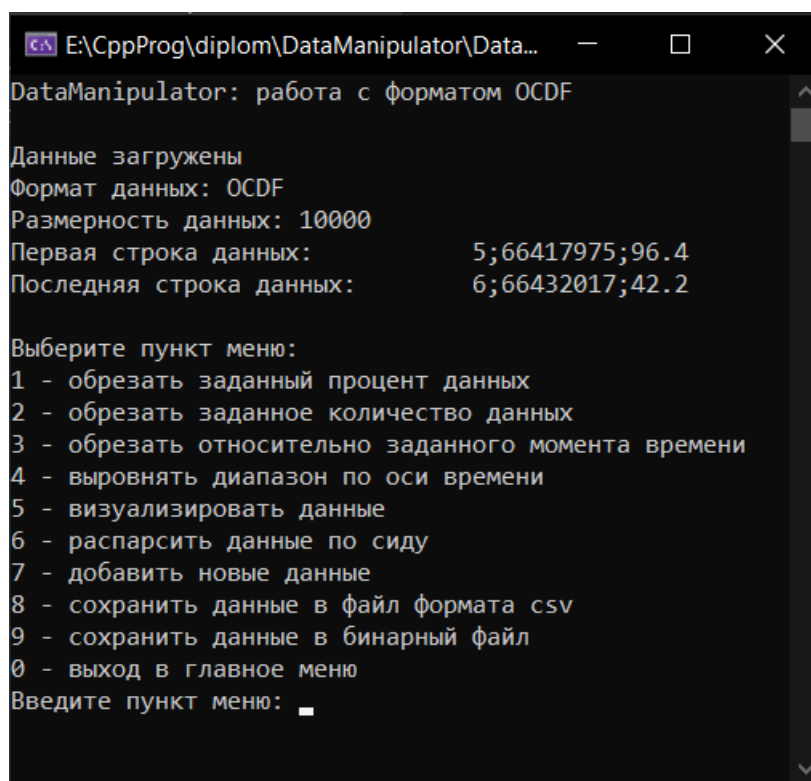


Рис. 10: ввод файла с данными в бинарном виде.

В появившемся меню будет указана информация о считанных данных: формат данных, их количество, а также первая и последняя строки последовательности данных. После идёт выбор действий, которые можно совершить над данными: обрезка данных, визуализация данных, приведение данных к равноинтервальному виду, парсинг данных, добавление данных и сохранение результатов обработки. Каждое из них подробно будет разобрано ниже.

3.4 Обрезка OCDF-данных.

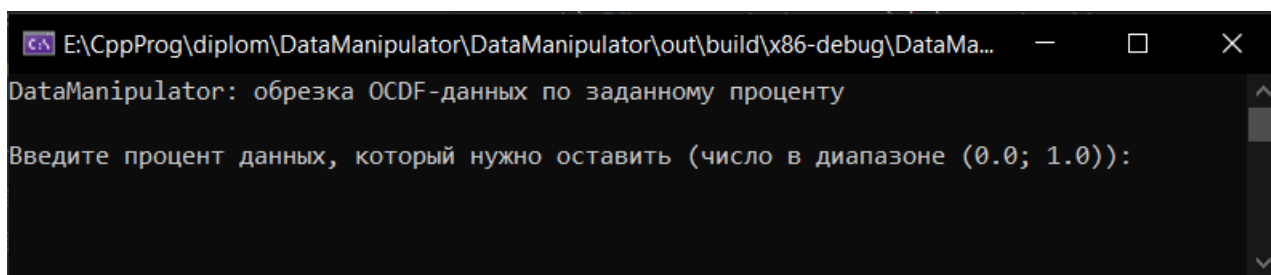
Существует 3 вида обрезки данных: обрезка заданного процента данных, обрезка заданного количества данных и обрезка по заданному моменту времени. Также у каждого вида обрезки данных существует 2 режима обрезки данных: обрезка данных слева направо или обрезка данных справа налево. Рассмотрим каждый вид обрезки данных на примерах данных, показанных на рисунке 11.

```
Данные загружены
Формат данных: OCDF
Размерность данных: 69870
Первая строка данных:      1;1256;0
Последняя строка данных:  1;10612923;100
```

Рис. 11: пример данных для дальнейшей их обрезки.

3.4.1 Обрезка OCDF-данных по заданному проценту.

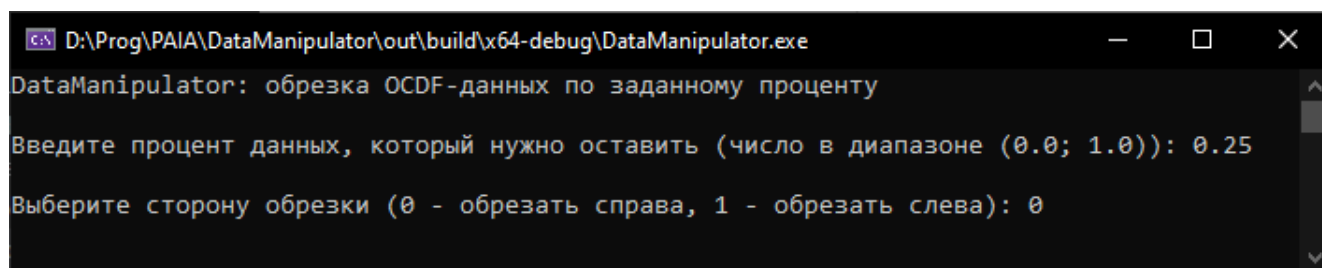
Обрезка по заданному проценту данных оставляет введённое пользователем процентное количество данных от общего их количества. Для ввода процента обрезки данных необходимо задать число в промежутке (0,00; 1,00), где 0,00 - 0%, а 1,00 - 100%, как это показано на рисунке 12.



```
E:\CppProg\diplom\DataManipulator\DataManipulator\out\build\x86-debug\DataMa...
DataManipulator: обрезка OCDF-данных по заданному проценту
Введите процент данных, который нужно оставить (число в диапазоне (0.0; 1.0)):
```

Рис. 12: ввод процента для обрезки данных.

После ввода процента для обрезки данных, программа предлагает выбрать режим обрезки, введя значения 0 или 1, как показано на рисунке 13.



```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: обрезка OCDF-данных по заданному проценту
Введите процент данных, который нужно оставить (число в диапазоне (0.0; 1.0)): 0.25
Выберите сторону обрезки (0 - обрезать справа, 1 - обрезать слева): 0
```

Рис. 13: ввод режима для обрезки по заданному проценту.

Если ввести 0, то заданный процент данных будет оставлен слева, т.е. в начале. Если ввести 1, то заданный процент данных будет оставлен справа, т.е. в конце. Вводим режим обрезки данных 0 и смотрим результаты на рисунке 14.

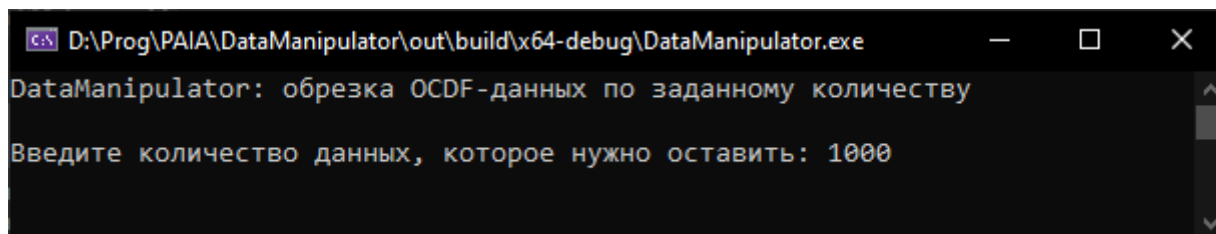
```
Данные загружены
Формат данных: OCDF
Размерность данных: 17467
Первая строка данных:      1;1256;0
Последняя строка данных:   1;1981280;79.7
```

Рис. 14: результаты обрезки данных по заданному проценту.

Была произведена обрезка 75% данных от общего количества, поскольку было введён 25% данных, которые необходимо оставить, в режиме справо налево, т.е. с конца.

3.4.2 Обрезка OCDF-данных по заданному количеству.

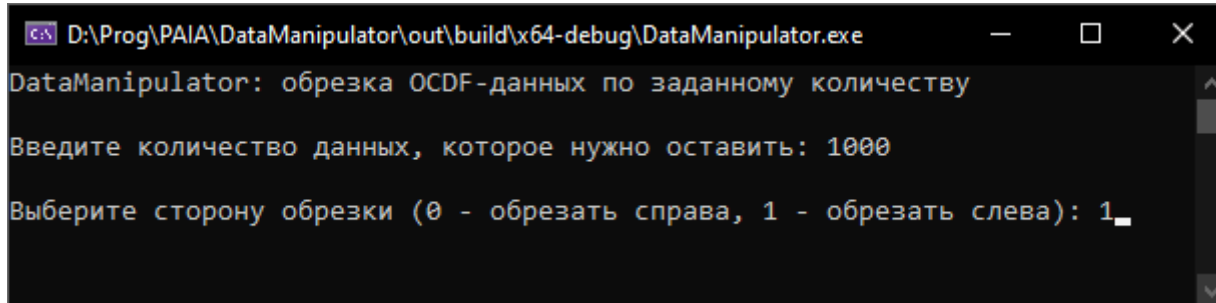
Обрезка по заданному количеству данных оставляет введённое пользователем количество данных. Для ввода количества обрезки данных необходимо задать целое положительное число отличное от нуля, как это показано на рисунке 15.



```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: обрезка OCDF-данных по заданному количеству
Введите количество данных, которое нужно оставить: 1000
```

Рис. 15: ввод значения количества для обрезки данных.

После ввода количества для обрезки данных, программа предлагает выбрать режим обрезки, введя значения 0 или 1, как показано на рисунке 16.



```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: обрезка OCDF-данных по заданному количеству
Введите количество данных, которое нужно оставить: 1000
Выберите сторону обрезки (0 - обрезать справа, 1 - обрезать слева): 1
```

Рис. 16: ввод режима для обрезки по заданному количеству.

Если ввести 0, то заданное количество данных будет оставлено слева, т.е. в начале. Если ввести 1, то заданное количество данных будет оставлено справа, т.е. в конце. Вводим режим обрезки данных 1 и смотрим результаты на рисунке 17.

```
Данные загружены
Формат данных: OCDF
Размерность данных: 1000
Первая строка данных:      1;1709916;98.8
Последняя строка данных:   1;1981280;79.7
```

Рис. 17: результаты обрезки данных по заданному количеству.

Была произведена обрезка 16467 данных, поскольку было введено 1000 данных, которые необходимо оставить, в режиме справо налево, т.е. в конце.

3.4.3 Обрезка OCDF-данных по заданному моменту времени.

Обрезка по заданному моменту времени оставляет промежуток данных до или после указанного момента. Для ввода момента времени для обрезки данных необходимо задать целое положительное число отличное от нуля, как это показано на рисунке 18.

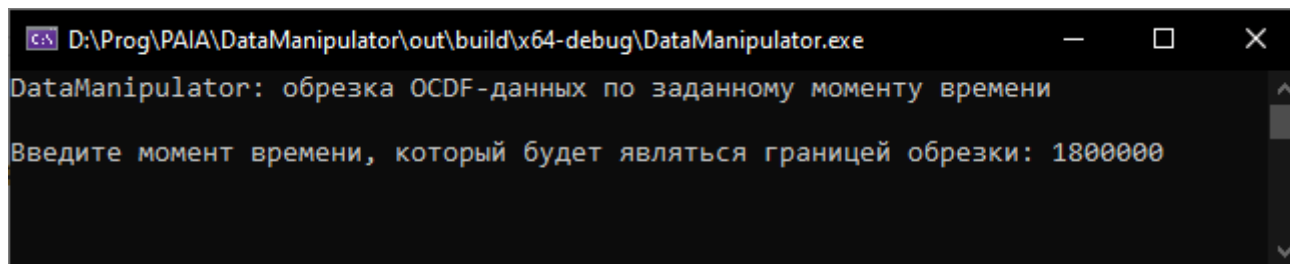


Рис. 18: ввод значения момента времени для обрезки данных.

После ввода момента времени для обрезки данных, программа предлагает выбрать режим обрезки, введя значения 0 или 1, как показано на рисунке 19.

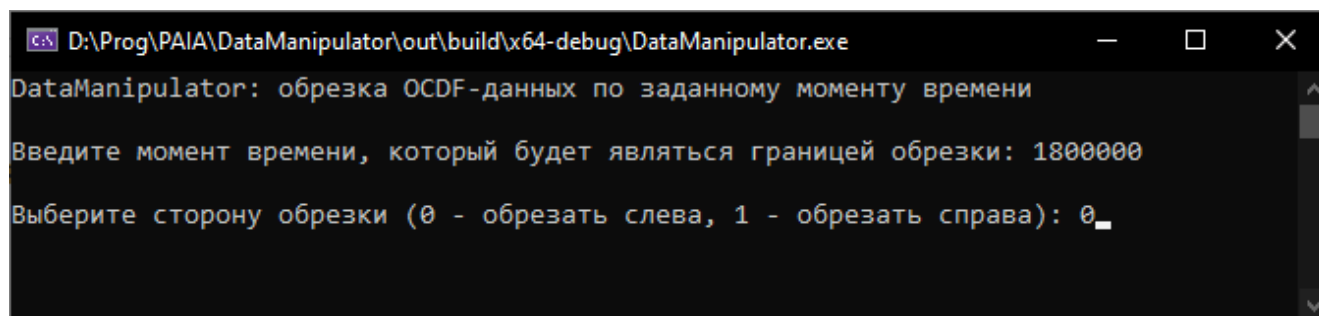


Рис. 19: ввод режима для обрезки по заданному моменту времени.

Если ввести 0, то данные будут оставлены слева относительно заданного момента времени, т.е. в начале. Если ввести 1, то данные будут оставлены справа относительно заданного момента времени, т.е. в конце. Вводим режим обрезки данных 0 и смотрим результаты на рисунке 20.

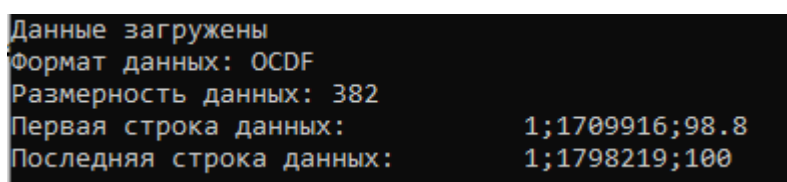


Рис. 20: результаты обрезки данных по заданному моменту времени.

3.4.4 Возможные ошибки при обрезке OCDF-данных.

При обрезке данных могут возникать различные ситуации, которые способны вызывать всяческие ошибки или исключительные случаи. Самым основным, что может вызвать ошибку, это некорректный ввод данных. Так, например, при обрезке данных по заданному проценту не допустимо вводить числа не принадлежащие диапазону (0,00; 1,00), хоть и ввод 1,00 доступен, однако при этом данные никак не изменятся. Ввод отрицательных или не целых чисел при обрезке по заданному моменту времени и по заданному количеству данных также вызывает ошибки некорректного ввода. Пример такой ошибки продемонстрирован на рисунке 21.

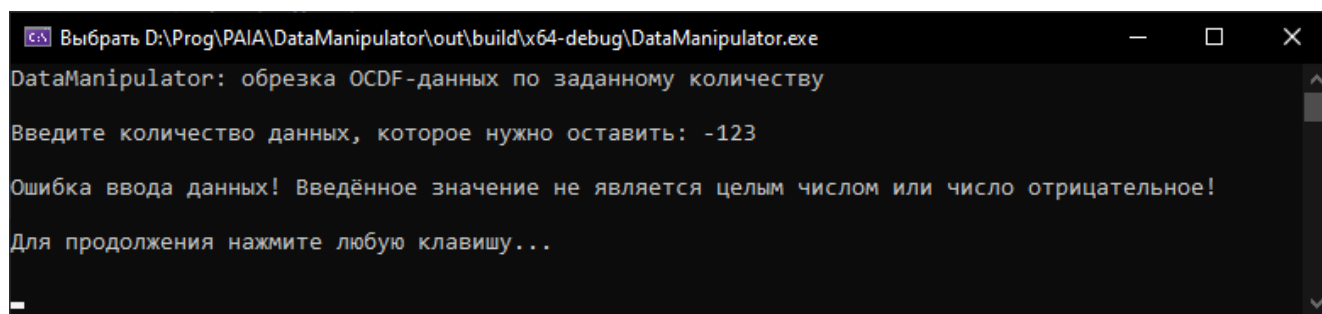


Рис. 21: пример ввода недопустимых чисел при обрезке OCDF-данных.

Также может быть вызвана ошибка некорректного ввода, при использовании символов, отличных от цифр, характерное для всех видов обрезки. Пример такой ошибки продемонстрирован на рисунке 22.

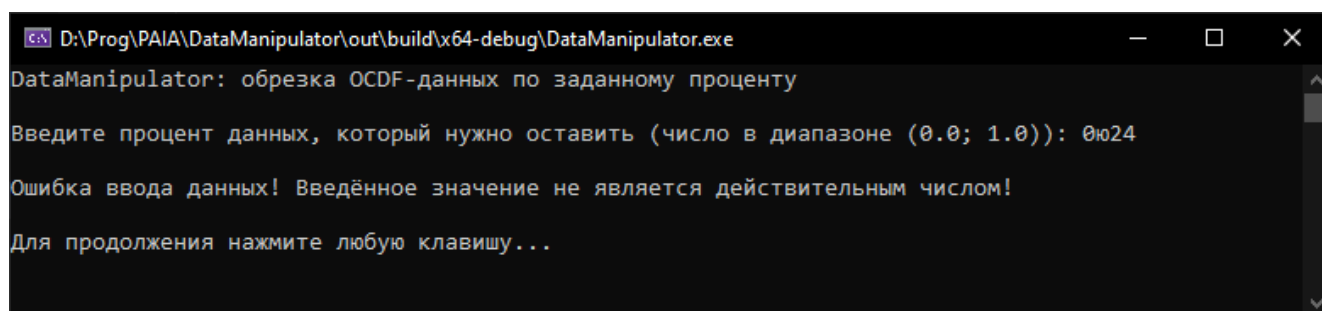


Рис. 22: пример ввода недопустимых символов при обрезке OCDF-данных.

При обрезке данных по заданному количеству есть вероятность того, что пользователь введёт число больше общего количества данных. В таком случае данные не изменятся и ошибка не появится.

При обрезке данных по заданному моменту времени есть вероятность того, что пользователь введёт момент времени, который находится раньше начального момента времени или позже конечного момента времени в самих данных. В таком случае, если будет выбран соответствующий режим обрезки, данные либо никак не изменятся, либо появится ошибка о возврате пустых данных, что продемонстрирована на рисунке 23.

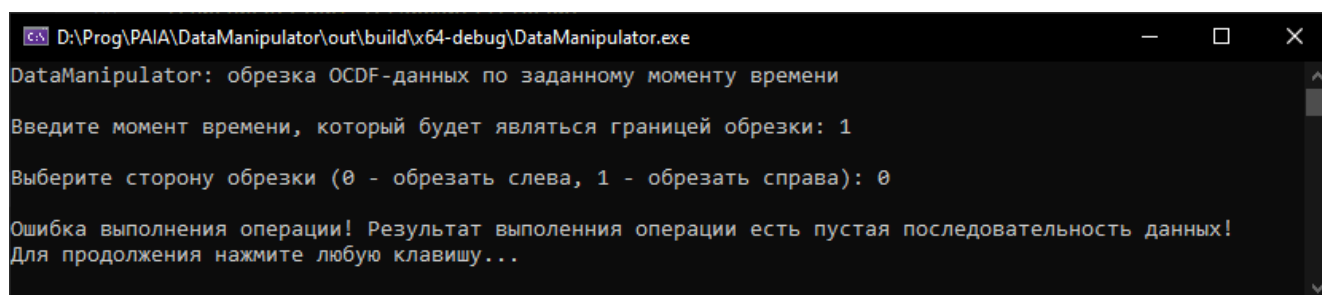
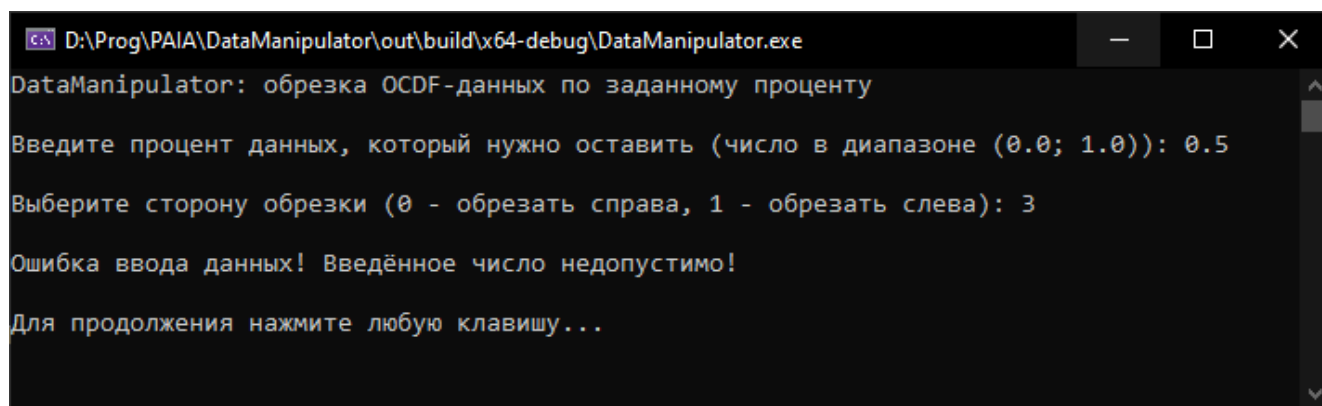


Рис. 23: пример ошибки при возврате пустых OCDF-данных.

Такая ошибка может возникать также при вводе 0,00 для обрезки данных по заданному проценту. В случае такой ошибки данные никак не изменятся и управление перейдёт к меню обработки OCDF-данных.

Последняя возможная ошибка относится к вводу недопустимых символов при выборе режимов обрезки. В таком случае возникает ошибка, которая перезапускает операцию заново. Пример такой ошибки продемонстрирован на рисунке 24.



```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: обрезка OCDF-данных по заданному проценту
Введите процент данных, который нужно оставить (число в диапазоне (0.0; 1.0)): 0.5
Выберите сторону обрезки (0 - обрезать справа, 1 - обрезать слева): 3
Ошибка ввода данных! Введённое число недопустимо!
Для продолжения нажмите любую клавишу...
```

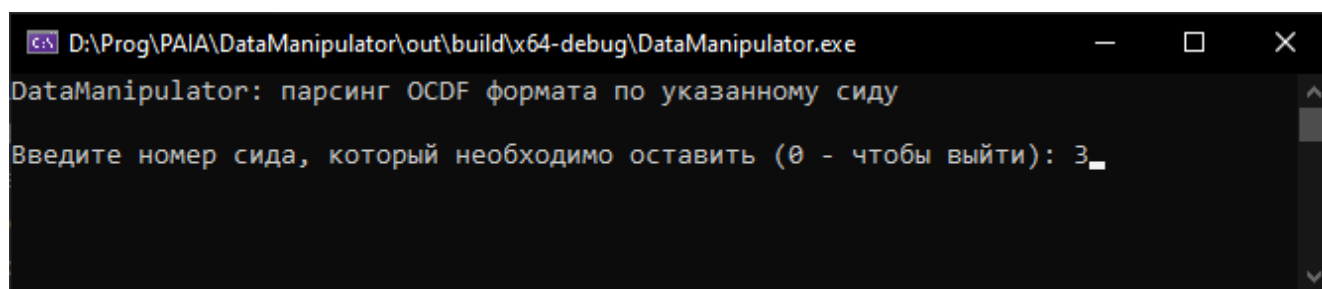
Рис. 24: пример ввода некорректных данных при выборе режима обрезки OCDF-данных.

3.5 Парсинг OCDF-данных.

Основной функцией парсинга данных является разделение данных относительно указанного сида для данных, где имеются значения двух и более сидов, с целью получения данных, содержащих информацию, относящуюся исключительно к указанному сиду.

3.5.1 Процесс парсинга OCDF-данных.

Процесс парсинга начинается с того, что программа запрашивает: данные какого сида необходимо получить? Это можно увидеть на рисунке 25.



```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: парсинг OCDF формата по указанному сиду
Введите номер сида, который необходимо оставить (0 - чтобы выйти): 3_
```

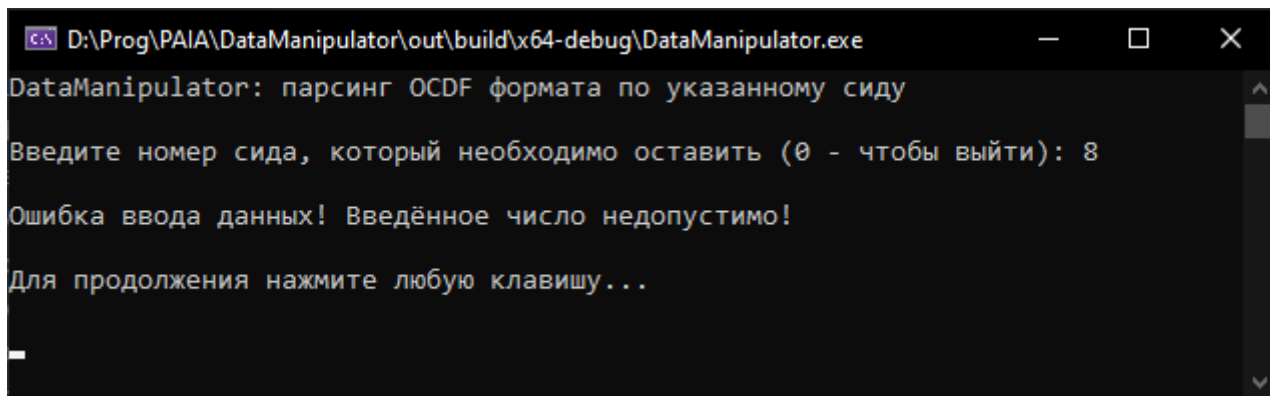
Рис. 25: ввод номера сида для парсинга OCDF-данных.

После ввода номера сида, который необходимо получить, выполняется сам процесс парсинга, результатом которого являются данные, содержащие информацию, касающиеся указанного сида. Данная операция весьма проста.

3.5.2 Возможные ошибки при парсинге OCDF-данных.

Несмотря на простоту операции парсинга, все же есть ситуации, при которых возникают ошибки или исключительные случаи.

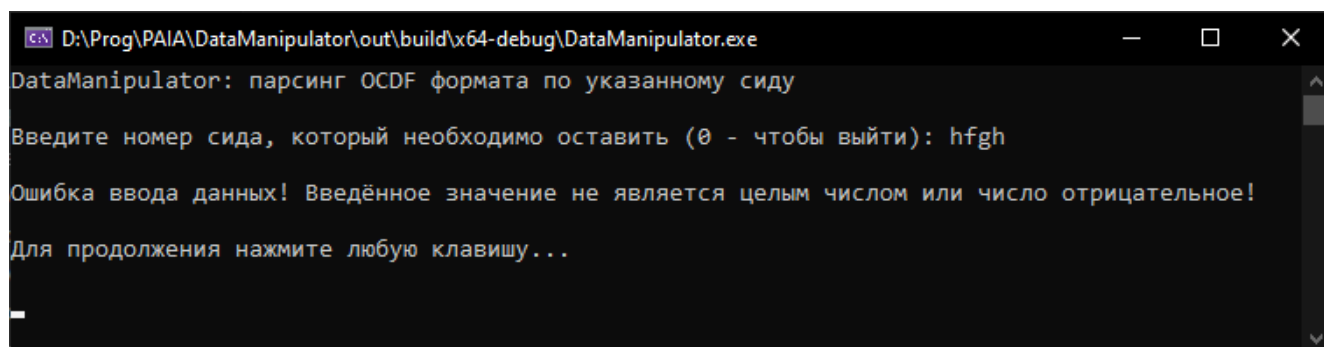
Ошибка может возникать, если ввести недопустимый сид. На данный момент программа считает допустимыми следующие сиды: 1, 2, 3, 4, 5, 6. Иные введённые числа вызывают ошибку, продемонстрированную на рисунке 26.



```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: парсинг OCDF формата по указанному сиду
Введите номер сида, который необходимо оставить (0 - чтобы выйти): 8
Ошибка ввода данных! Введённое число недопустимо!
Для продолжения нажмите любую клавишу...
```

Рис. 26: ошибка ввода несуществующего сида при парсинге OCDF-данных.

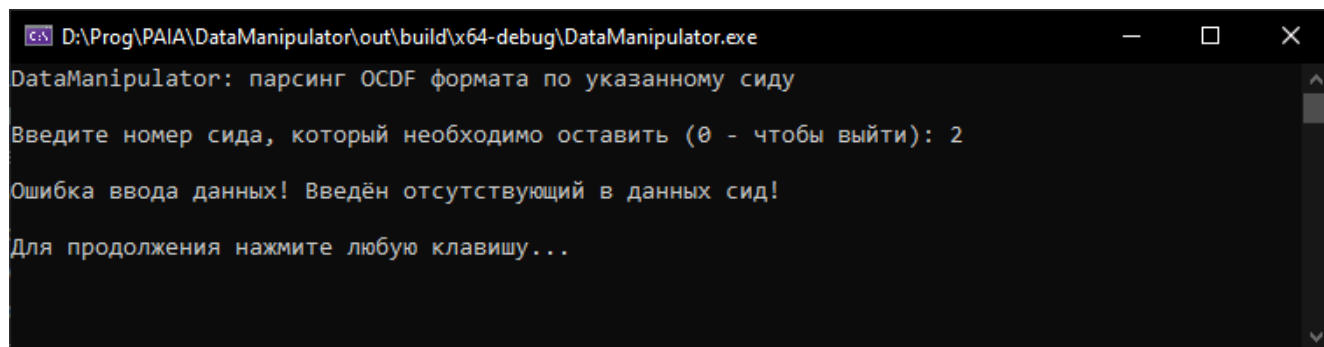
Ошибка также может возникать, если ввести недопустимые символы, т.е. символы отличные от цифр. Данная ошибка продемонстрирована на рисунке 27.



```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: парсинг OCDF формата по указанному сиду
Введите номер сида, который необходимо оставить (0 - чтобы выйти): hfgh
Ошибка ввода данных! Введённое значение не является целым числом или число отрицательное!
Для продолжения нажмите любую клавишу...
```

Рис. 27: ошибка ввода недопустимого символа при парсинге OCDF-данных.

Также ошибка может возникать в случае, если был введён сид, которого нет в данных, подвергаемых парсингу. Данная ошибка продемонстрирована на рисунке 28.



```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: парсинг OCDF формата по указанному сиду
Введите номер сида, который необходимо оставить (0 - чтобы выйти): 2
Ошибка ввода данных! Введён отсутствующий в данных сид!
Для продолжения нажмите любую клавишу...
```

Рис. 28: ошибка ввода отсутствующего сида в OCDF-данных.

Результатом выше перечисленных ошибок будет перезапуск операции парсинга.

3.6 Выравнивание диапазонов OCDF-данных.

Выравнивание диапазонов OCDF-данных по оси времени предполагает формирование равноинтервальных данных. Такие данные необходимы прежде всего для корректной работы модели, чтобы она могла представлять поведение графика в определённых моментах времени, иначе точность модели упадёт или возникнут трудности в определении моментов времени у предсказанных величин.

3.6.1 Математические основы выравнивания диапазонов OCDF-данных.

Равноинтервальные данные предполагают то, что расстояние или интервалы между соседними точками по оси абсцисс или по оси времени будут взаимно равноудалёнными друг от друга. В таком случае, момент времени каждой точки можно выразить с помощью арифметической прогрессии по формуле 3.

$$t_i = t_0 + i \cdot r \quad (3)$$

где t_0 – начальное значение времени;

r – необходимое расстояние между точками, представляющее собой некоторую константу, которую может задать пользователь;

i – номер точки, итерации, элемента прогрессии; t_i – i -ый элемент прогрессии;

t_i – i -ый элемент прогрессии.

Теперь необходимо определить значение ординаты для каждого нового рассчитанного момента времени. Для этого используем уравнение из аналитической геометрии, а именно уравнение прямой, проходящей через две точки, которое можно увидеть на формуле 4.

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} \quad (4)$$

где точка (x, y) – это точка, которую необходимо найти между точками (x_1, y_1) и (x_2, y_2) , взятых из реальных данных.

Про соотношение этих точек известно, что $x_1 \leq x \leq x_2$.

Для определения ординаты i -того элемента прогрессии необходимо взять две ближайших точки из реальных данных относительно оси абсцисс к искомой точке. Тогда для определения ординаты имеем формулу 5.

$$y = \left(\frac{(x - x_1) \cdot (y_2 - y_1)}{x_2 - x_1} \right) + y_1 \quad (5)$$

Таким образом можно получить аппроксимированные данные, где взаимное удаление между двумя соседними точками будет везде во всей цепочке данных постоянным.

3.6.2 Процесс выравнивания диапазонов OCDF-данных.

Для получения равноинтервальных OCDF-данных необходимо выбрать соответствующий пункт меню, после чего нас встретит сообщение, запрашивающее задать: каким будет интервал между точками? Это продемонстрировано на рисунке 29.

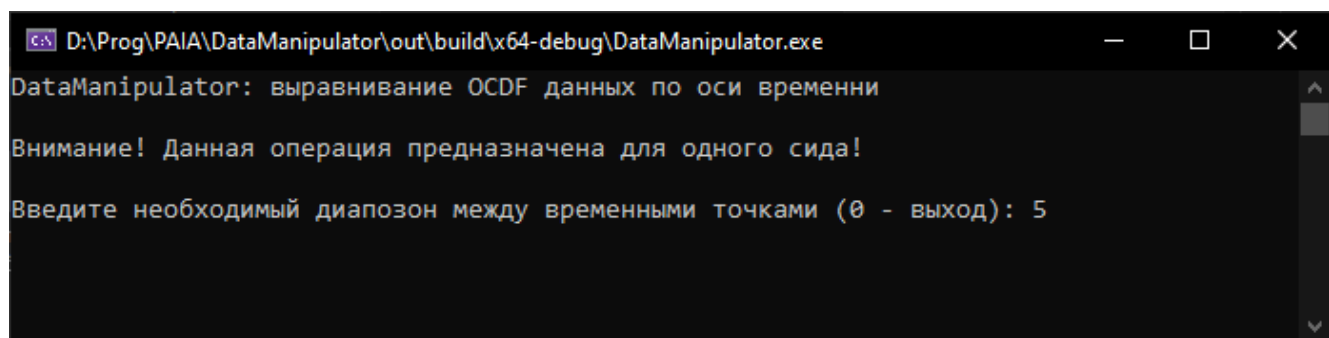


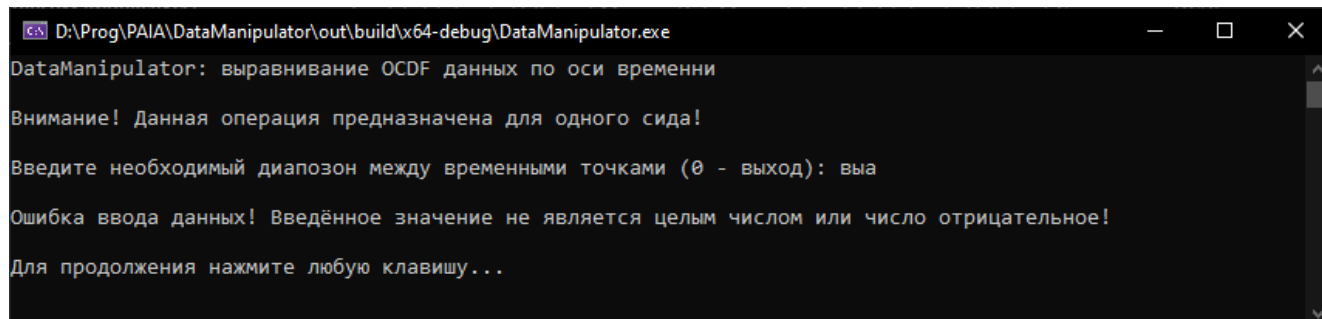
Рис. 29: ввод интервала для аппроксимации OCDF-данных.

Также на рисунке присутствует важное предупреждение, говорящее о том, что данная операция предназначена для данных, содержащих информацию всего по одному сиду.

После ввода начнётся процесс аппроксимации, в результате которого можно будет получить данные, расстояние между соседними точками которых равно введённому пользователем числу.

3.6.3 Возможные ошибки при выравнивании диапазонов OCDF-данных.

Также к знакомой нам уже ошибке относится и ввод отрицательных, не целых чисел и ввод отличных от цифр символов, продемонстрированный на рисунке 30.



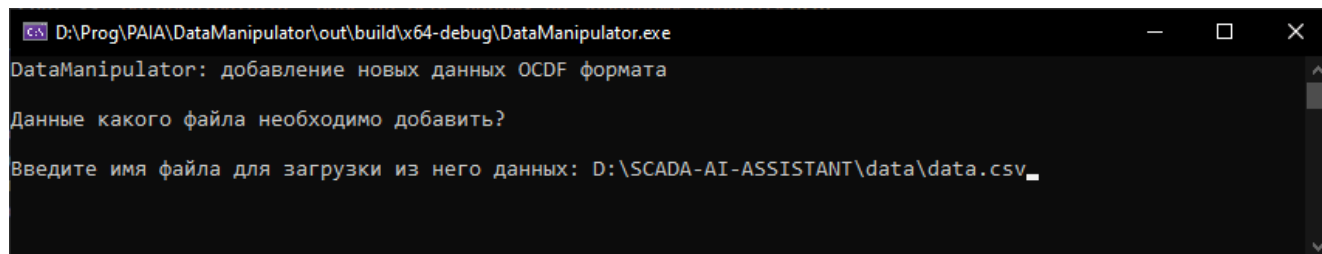
```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: выравнивание OCDF данных по оси времени
Внимание! Данная операция предназначена для одного сида!
Введите необходимый диапазон между временными точками (0 - выход): выа
Ошибка ввода данных! Введённое значение не является целым числом или число отрицательное!
Для продолжения нажмите любую клавишу...
```

Рис. 30: ввод некорректных значений при аппроксимации OCDF-данных.

По сути, при вводе будущего интервала нет никаких ограничений, однако при вводе больших чисел результат может быть неудовлетворительным. Этот момент остаётся на совести пользователя.

3.7 Добавление OCDF-данных.

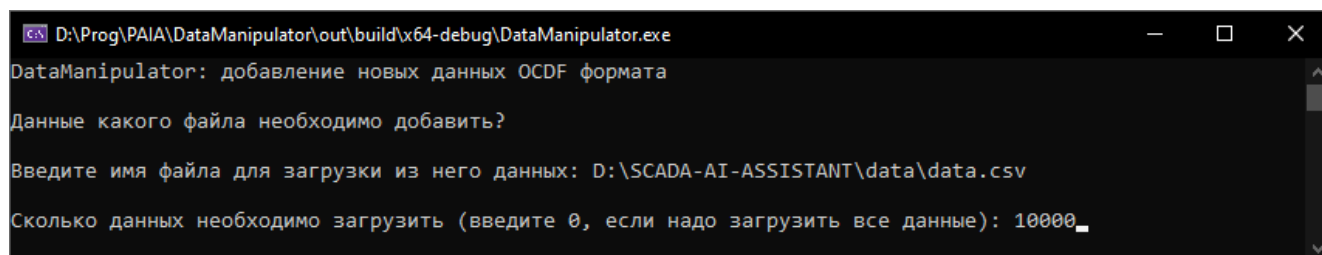
Имеется возможность добавление OCDF-данных к другим OCDF-данным. Это делается с помощью выбора соответствующего пункта меню. Для начала будет предложено ввести имя файла с данными, которые необходимо добавить. Это продемонстрировано на рисунке 31.



```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: добавление новых данных OCDF формата
Данные какого файла необходимо добавить?
Введите имя файла для загрузки из него данных: D:\SCADA-AI-ASSISTANT\data\data.csv_
```

Рис. 31: ввод файла для добавления OCDF-данных.

Далее будет предложено ввести количество данных, которых необходимо добавить. Это продемонстрировано на рисунке 32.



```
D:\Prog\PAIA\DataManipulator\out\build\x64-debug\DataManipulator.exe
DataManipulator: добавление новых данных OCDF формата
Данные какого файла необходимо добавить?
Введите имя файла для загрузки из него данных: D:\SCADA-AI-ASSISTANT\data\data.csv
Сколько данных необходимо загрузить (введите 0, если надо загрузить все данные): 10000_
```

Рис. 32: ввод количества OCDF-данных для добавления к другим OCDF-данным.

Стоит предупредить, что программа никак не обрабатывает добавляемые OCDF-данные, за исключением идентификации данных в файле как OCDF, поэтому добавление новых данных ложится на совесть пользователя. Он должен знать, что он добавляет, иначе операция может пройти корректно, но, например, одни данные могут быть равноинтервальными, а другие нет, разрыв, расстояние между добавленными и исходными данными может быть огромным и так далее.

Впрочем, при добавлении данных, новые и старые данные смешиваются и сортируются по хронологическому порядку. Стоит отметить также и то, что данная функция может нарушать основное правило временных рядов: все данные отсортированы в хронологическом порядке и нет данных с одинаковым временным моментом. Нарушается скорее вторая часть правила. При добавлении данных могут быть значения, относящиеся к моментам времени, которые уже определены в исходных данных. В таком случае данные всё равно добавятся. В одном временном моменте сначала будут идти исходные данные, а затем только добавленные.

К ошибкам во время процесса добавления можно отнести ошибки, описанные в разделе 3.3.1.

3.8 Визуализация OCDF-данных.

Для визуализации данных достаточно выбрать соответствующий пункт меню и дождаться появления окна с визуализированными данными. Вводить ничего не требуется.

Визуализация данных может проходить по-разному. Всё зависит от данных, которые требуется визуализировать. Если в данных содержится информация, касаемая исключительно одного сйда, то в окне визуализации появится только один график, а оси будут подписаны в соответствии с тем, что обозначает визуализированный сид. Это можно увидеть на рисунке 33.

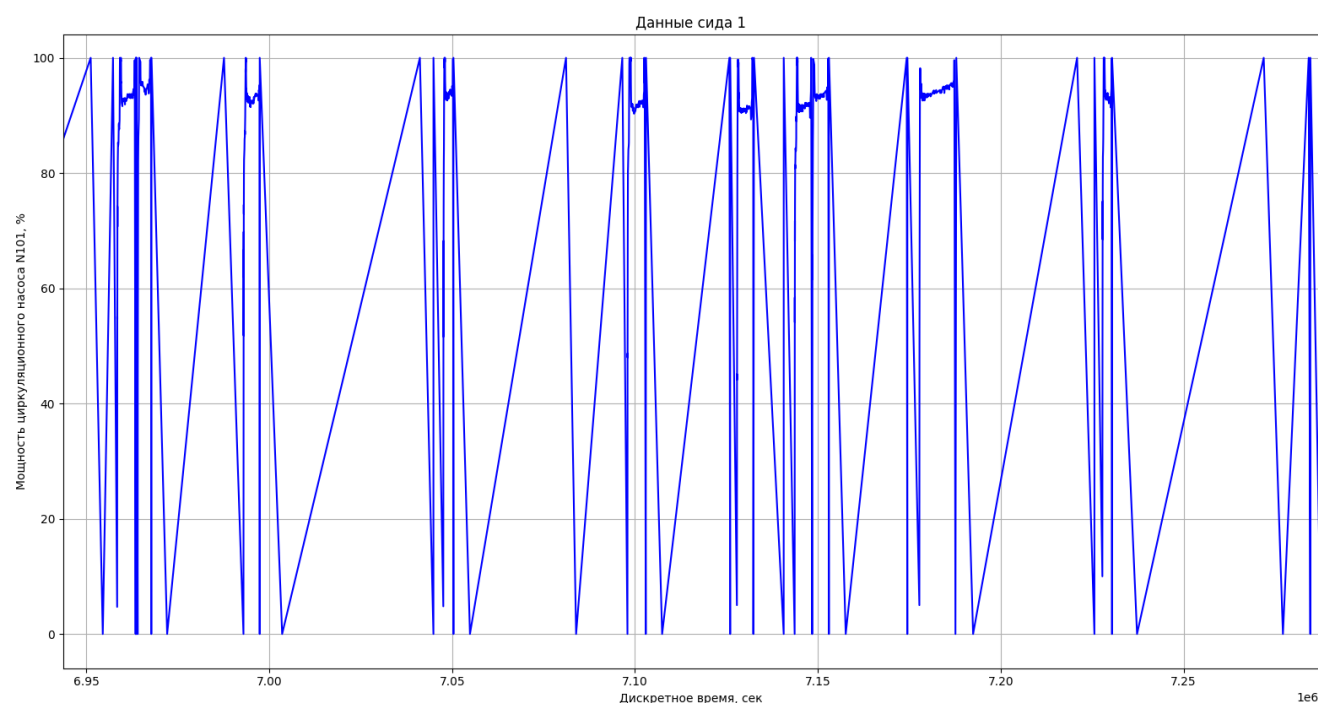


Рис. 33: визуализация OCDF-данных одного сйда

В случае, если данные содержат информацию более одного сйда, то в первую очередь будет построено столько графиков, сколько сидов содержится в данных, во-вторых, появится легенда, дающая информацию о визуализированных сйда, в-третьих ось ординат не будет

никак подписана, вся информация хранится в легенде. Это можно увидеть на рисунках 34 и 35.

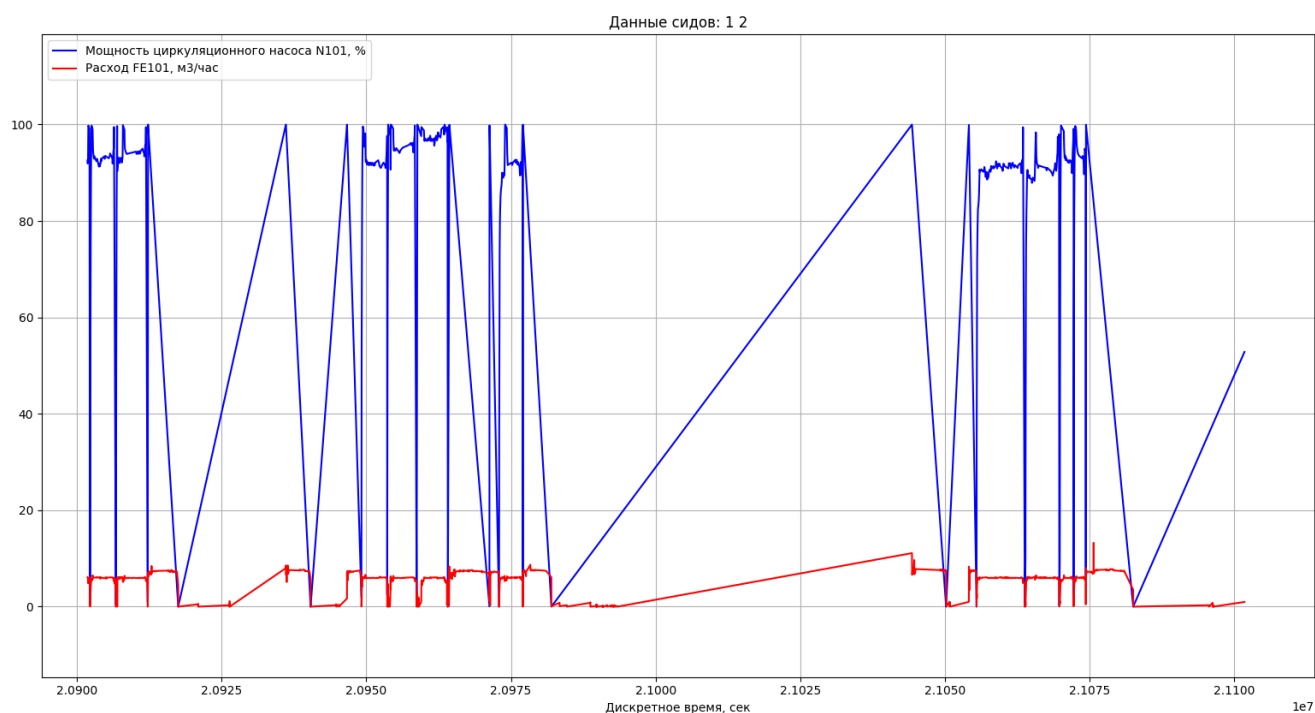


Рис. 34: пример визуализация OCDF-данных с двумя сидами

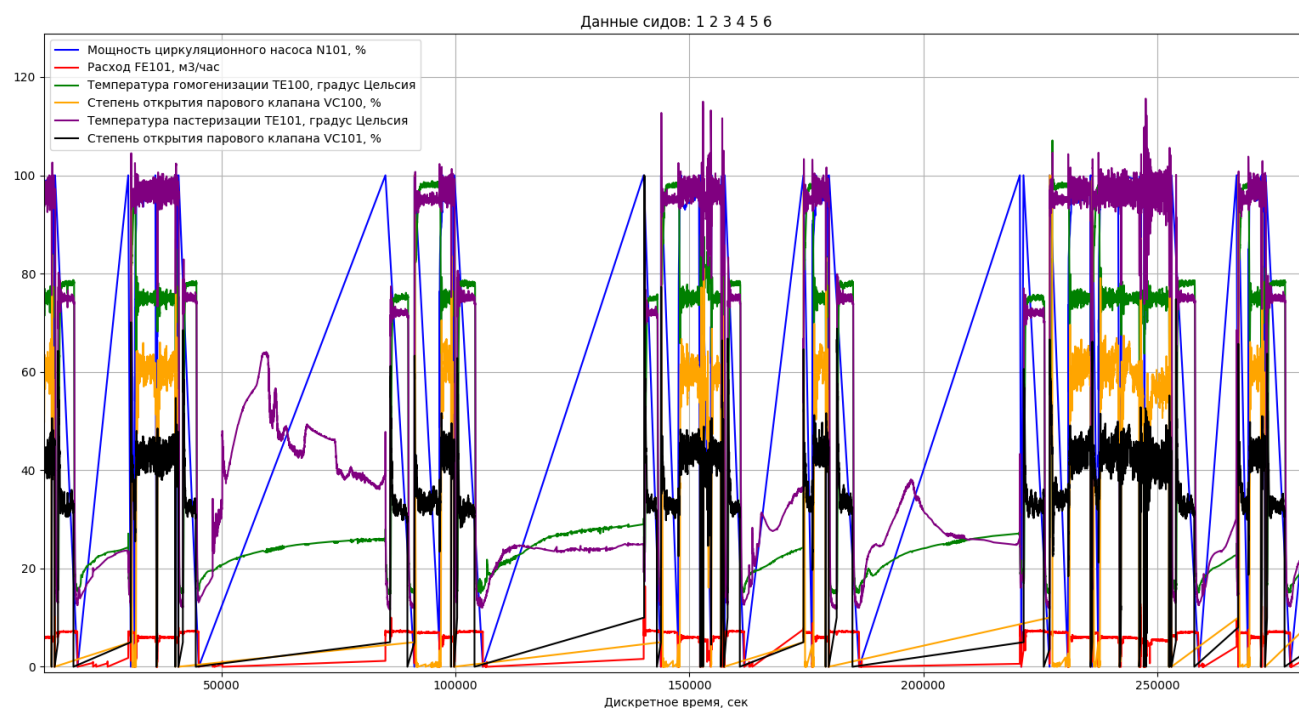


Рис. 35: пример визуализация OCDF-данных со всеми сидами

Управление перейдёт к основной программе только после закрытия окна визуализации.

3.8.1 Управление визуализацией OCDF-данных.

При визуализации данных предусмотрено масштабирование окна визуализации. Прежде чем об этом рассказать, определить два сокращения. Первое сокращение: вращение колёсика мышки вперёд – вкмВ. Второе сокращение: вращение колёсика мышки назад –

вкмН. Тогда имеем:

вкмВ – позволяет увеличить масштаб окна визуализации. Проще говоря, позволяет приблизить изображение.

вкмН – позволяет уменьшить масштаб окна визуализации. Проще говоря, позволяет отдалить изображение.

Shift + вкмВ – позволяет увеличить масштаб по оси абсцисс (оси времени, оси X, нижней оси) окна визуализации. Проще говоря, позволяет растянуть изображение по ширине.

Shift + вкмН – позволяет уменьшить масштаб по оси абсцисс (оси времени, оси X, нижней оси) окна визуализации. Проще говоря, позволяет сжать изображение по ширине.

CTRL + вкмВ – позволяет увеличить масштаб по оси ординат (оси значений, оси Y, боковой оси) окна визуализации. Проще говоря, позволяет растянуть изображение по высоте.

CTRL + вкмН – позволяет уменьшить масштаб по оси ординат (оси значений, оси Y, боковой оси) окна визуализации. Проще говоря, позволяет сжать изображение по высоте.

3.9 Сохранение OCDF-данных в файл.

После работы с данными конечно же появится надобность в сохранении результатов обработки данных. Для этого есть возможность сохранить данные в текстовом или бинарном режимах.

3.9.1 Сохранение OCDF-данных в файл формата csv.

Для сохранения данных в текстовом режиме достаточно выбрать соответствующий пункт меню и в появившемся запросе указать путь к файлу, в который будут сохранены данные. Это продемонстрировано на рисунке 36.

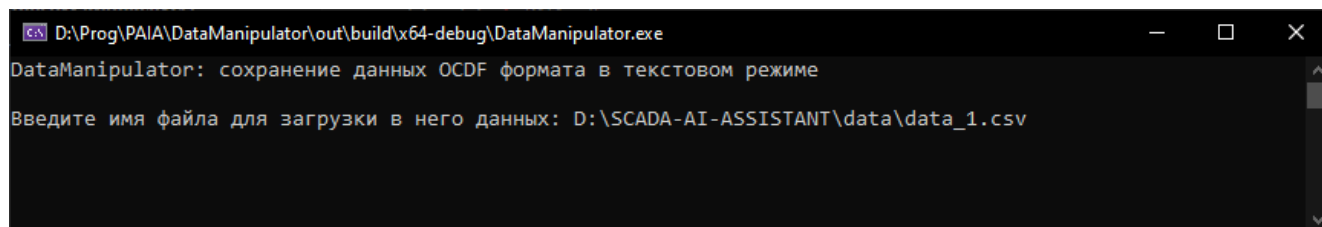


Рис. 36: сохранение OCDF-данных в текстовом режиме.

Из особенностей, хотелось бы отметить, что при сохранении в текстовом режиме требуется обязательно указывать расширение файла, поскольку текстово можно сохранить в любом формате, кроме зарезервированных форматов. Но всё же рекомендуется указывать расширение .csv.

3.9.2 Сохранение OCDF-данных в бинарный файл.

Для сохранения данных в бинарном режиме достаточно выбрать соответствующий пункт меню и в появившемся запросе указать путь к файлу, в который будут сохранены данные. Это продемонстрировано на рисунке 37.

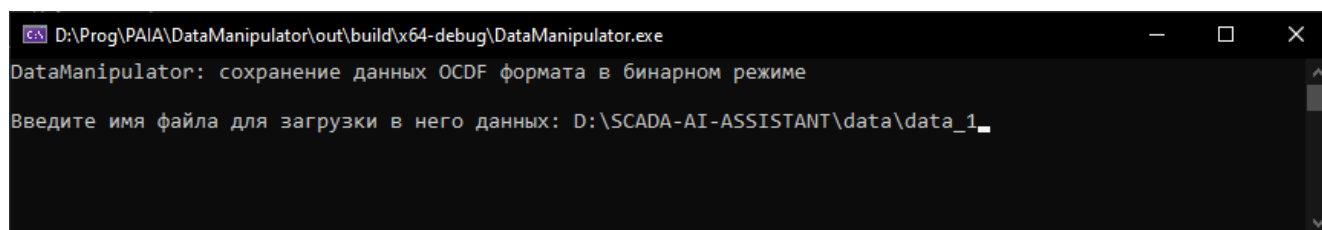


Рис. 37: сохранение OCDF-данных в бинарном режиме.

Из особенностей, хотелось бы отметить, что при сохранении в бинарном режиме не обязательно указывать расширение файла, поскольку расширение .bin будет добавлено автоматически, даже если расширение указать.

3.9.3 Возможные ошибки при сохранении OCDF-данных в файл.

В качестве ошибок может выступать ввод зарезервированных форматов при сохранении в текстовом режиме, таких как .t1stm, .vlstm, .lstm, .bin, что показано на рисунке 38.

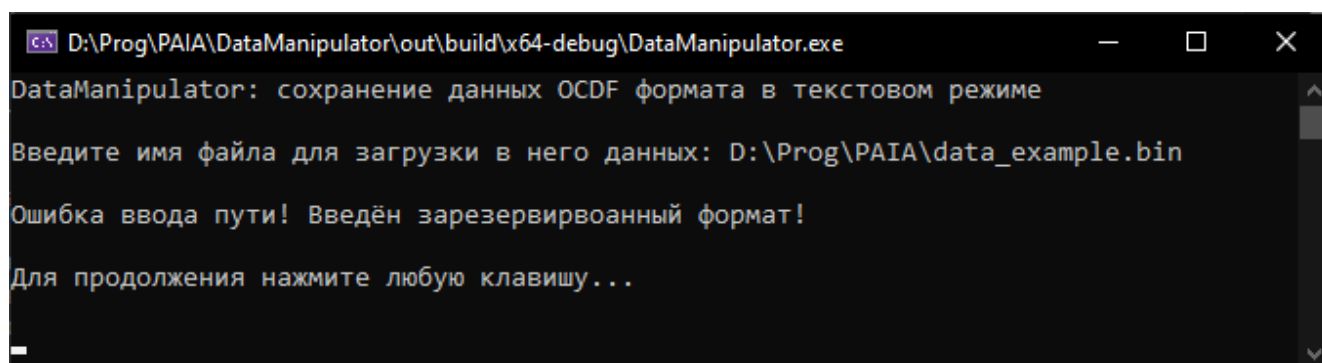


Рис. 38: сохранение OCDF-данных в бинарном режиме.

также к ошибкам можно отнести слишком маленькое имя файла. Имя файла должно иметь более 5 символов в своём названии включая расширение. Эта ошибка продемонстрирована на рисунке 39.

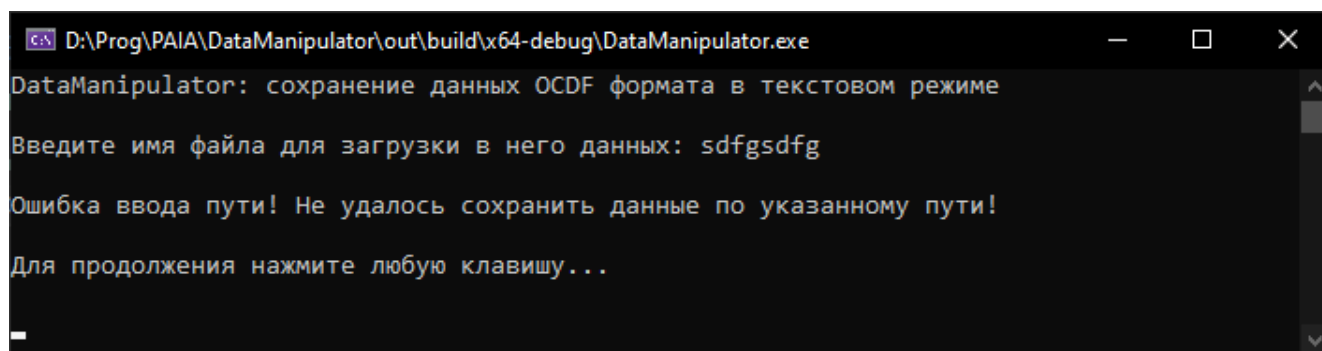


Рис. 39: сохранение OCDF-данных в бинарном режиме.

3.10 Выход из меню работы с OCDF-данными.

Для выхода из меню работы с OCDF-данными достаточно ввести 0 и тогда загруженные из файла OCDF-данные освободятся из памяти, поэтому перед выходом из меню не забудьте сохранить результаты обработки данных.

3.11 Сохранение TDF-данных в файл.

После работы с данными конечно же появится надобность в сохранении результатов обработки данных. Для этого есть возможность сохранить данные в текстовом или бинарном

режимах.

Возможные ошибки при сохранении данных, которые могут возникать, уже описаны в разделе 3.9.3.

3.11.1 Сохранение TDF-данных в файл формата csv.

Для сохранения TDF-данных в текстовом режиме достаточно выбрать соответствующий пункт меню и в появившемся запросе указать путь к файлу, в который будут сохранены данные. Это продемонстрировано на рисунке 40.

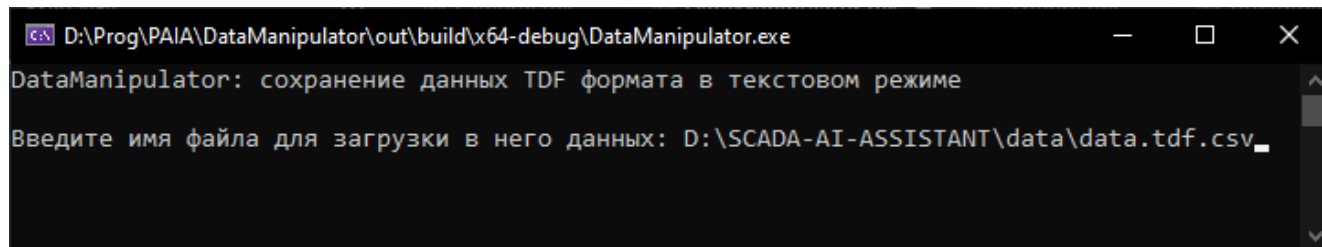


Рис. 40: сохранение TDF-данных в текстовом режиме.

Из особенностей, хотелось бы отметить, что при сохранении в текстовом режиме требуется обязательно указывать расширение файла, поскольку текстово можно сохранить в любом формате, кроме зарезервированных форматов. Но всё же рекомендуется указывать расширение .csv.

3.11.2 Сохранение TDF-данных в бинарный файл.

Для сохранения TDF-данных в бинарном режиме достаточно выбрать соответствующий пункт меню и в появившемся запросе указать путь к файлу, в который будут сохранены данные. Это продемонстрировано на рисунке 41.

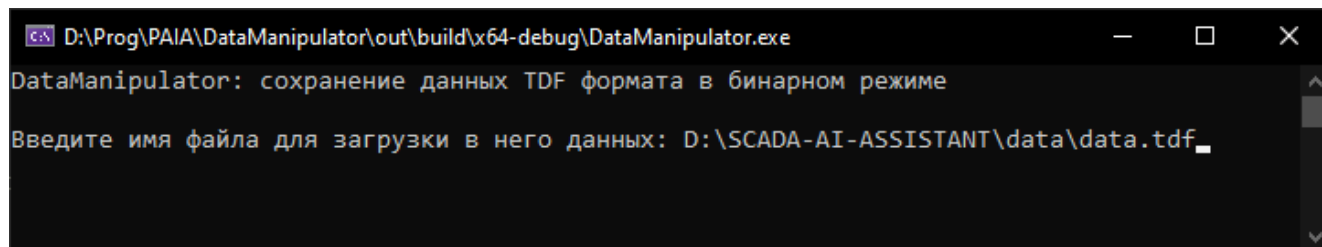


Рис. 41: сохранение TDF-данных в бинарном режиме.

Из особенностей, хотелось бы отметить, что при сохранении в бинарном режиме не обязательно указывать расширение файла, поскольку расширение .bin будет добавлено автоматически, даже если расширение указать.

3.12 Выход из меню работы с TDF-данными.

Для выхода из меню работы с TDF-данными достаточно ввести 0 и тогда загруженные из файла TDF-данные освободятся из памяти, поэтому перед выходом из меню не забудьте сохранить результаты обработки данных.