

Сглаживание финансовых временных рядов с помощью вейвлетов. Вейвлеты, что это такое и их реализация в языке программирования Python.

Воронкин Р.А., Богадунов В.И.

Постановка задачи: что такое вейвлет-преобразование и в каких областях находят применение вейвлет преобразования, а также какие плюсы и минусы данный подход имеет и привести пример реализации вейвлетов на языке программирования Python. **Цель работы:** изучить что такое вейвлет-преобразования и научиться взаимодействовать с ними в высокоуровневом языке программирования Python. **Используемые методы:** метод анализа. **Результат:** объяснение того, для чего необходимы вейвлет-преобразования, что они собой представляют, какие методы имеют в себе и показать пример реализации на высокоуровневом языке программирования Python. **Практическая значимость:** вейвлет-преобразования используются для анализа данных и имеют огромную значимость при разборе различного рода сигналов.

Ключевые слова: вейвлет-преобразования, анализ данных, вейвлеты, вейвлет, свойства вейвлетов, сигналы, разбор, декомпозиция, изображения.

Свою статью я позиционирую, как обобщённую обзорную статью на несколько статей взятых с habr.com и других сайтов, вся полученная информация была пропущена через себя и преобразована для дальнейшей работы с ней. Ссылки на статьи приведу в дополнительной литературе.

1) Что такое вейвлет-преобразование?

Если переводить данное слово дословно, от англ wavelet – небольшая волна, всплеск. Впервые вейвлеты были открыты ещё в 1910 году. А.Хаар опубликовал данную полную ортонормальную систему базисных функций с локальной областью определения, которую называли вейвлетами Хаара.

Самое первое упоминание вейвлетов в литературе было в 1984 году, в работах А.Гроссмана и Ж.Морле в статьях об цифровой обработке и анализу сейсмических сигналов.

По своей же сути, вейвлет преобразования – это математический инструмент, который используется для анализа сигналов и изображений. Он позволяет разложить сигнал или изображение на компоненты различных частот и разрешений, что позволяет извлекать информацию из сигнала на разных уровнях детализации.

Основными компонентами вейвлет-преобразования являются вейвлеты, которые представляют собой функции, которые могут быть использованы для анализа сигнала. Вейвлеты имеют свойства, которые позволяют им анализировать сигналы на различных уровнях разрешения и частот, что делает их более мощными, чем классические методы анализа сигналов.

Процесс вейвлет-преобразования включает следующие шаги:

- 1. Разбиение сигнала на компоненты различных масштабов и уровней разрешения.**
- 2. Анализ каждой компоненты с помощью выбранного вейвлета.**
- 3. Построение коэффициентов преобразования, которые отражают степень влияния каждой компоненты на исходный сигнал.**
- 4. Восстановление сигнала из полученных коэффициентов.**

Пример на языке программирования Python:

В качестве примера возьмём сигнал, состоящий из двух синусоид с частотами 10 и 50 Гц. Затем мы используем функцию `wavedec` библиотеки `wavelet` для разбиения сигнала на компоненты различных масштабов и уровней разрешения. В данном примере мы используем вейвлет Добеши 4-го порядка и разбиваем сигнал на 5 уровней разрешения:

```
import pywt
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
# Создание сигнала
```

```
x = np.linspace(0, 1, num=1024)
```

```
y = np.sin(2 * np.pi * 10 * x) + np.sin(2 * np.pi * 50 * x)
```

```
# Разбиение сигнала на компоненты различных масштабов и уровней разрешения
```

```
wavelet = 'db4'
```

```
levels = 5
```

```
coeffs = pywt.wavedec(y, wavelet, level=levels)
```

Дальше мы производим анализ каждой компоненты с помощью выбранного вейвлета и строим графическое представление исходного сигнала и его компонентов различных масштабов и уровней разрешения. Для этого мы используем функцию `subplots` библиотеки `matplotlib` для создания нескольких графиков в одной фигуре и функции `plot` для построения графиков.

```
# Анализ каждой компоненты с помощью выбранного вейвлета
```

```
fig, axs = plt.subplots(levels + 1, 1, figsize=(10, 15))
```

```
axs[0].plot(x, y, 'r')
```

```
axs[0].set_title('Original Signal')
```

```
for i in range(levels):
```

```
    axs[i + 1].plot(coeffs[i], 'g')
```

```
    axs[i + 1].set_title('Level ' + str(i + 1) + ' Detail Coefficients')
```

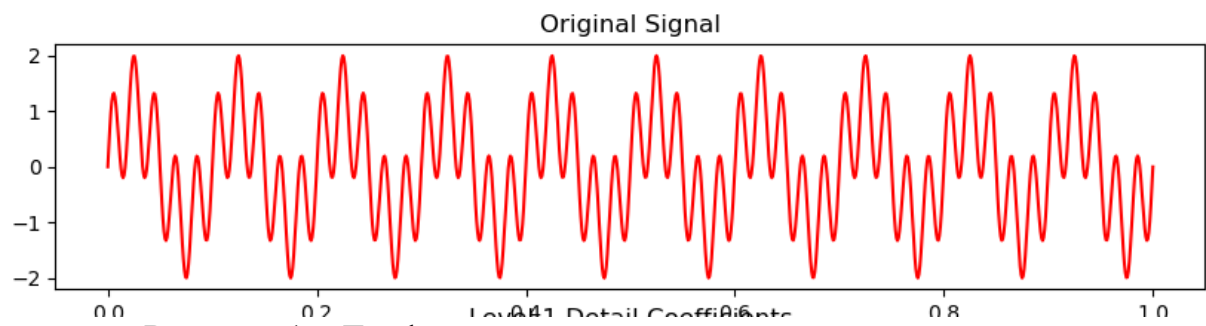


Рисунок 1 – Графическое представление исходного сигнала

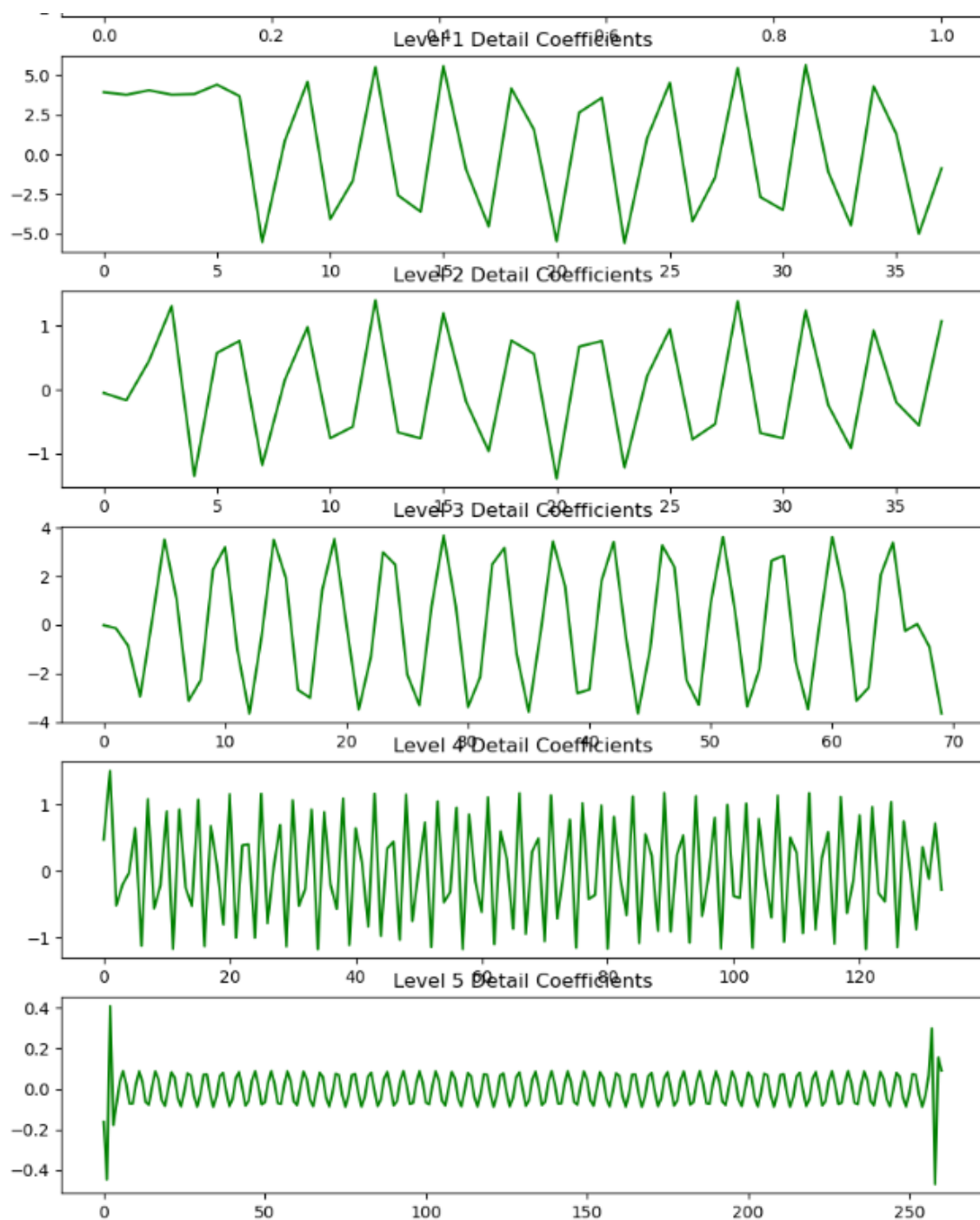


Рисунок 2 – Графическое представление компонентов сигнала различных масштабов и уровней разрешения.

Далее мы строим коэффициенты преобразования, которые отражают степень влияния каждой компоненты на исходный сигнал. Для этого мы используем функцию `waverec` библиотеки `wavelet`, которая производит обратное вейвлет-преобразование.

```
# Построение коэффициентов преобразования
```

```
reconstructed_signal = pywt.waverec(coeffs, wavelet)
```

Наконец, мы восстанавливаем сигнал из полученных коэффициентов с помощью функции `plot` библиотеки `matplotlib`. Результатом выполнения данного кода является графическое представление исходного и восстановленного сигналов.

```
# Восстановление сигнала из полученных коэффициентов
```

```
plt.figure(figsize=(10, 5))
```

```
plt.plot(x, y, 'r', label='Original Signal')
```

```
plt.plot(x, reconstructed_signal, 'b', label='Reconstructed Signal')
```

```
plt.legend()
```

```
plt.title('Original vs Reconstructed Signal')
```

```
plt.show()
```

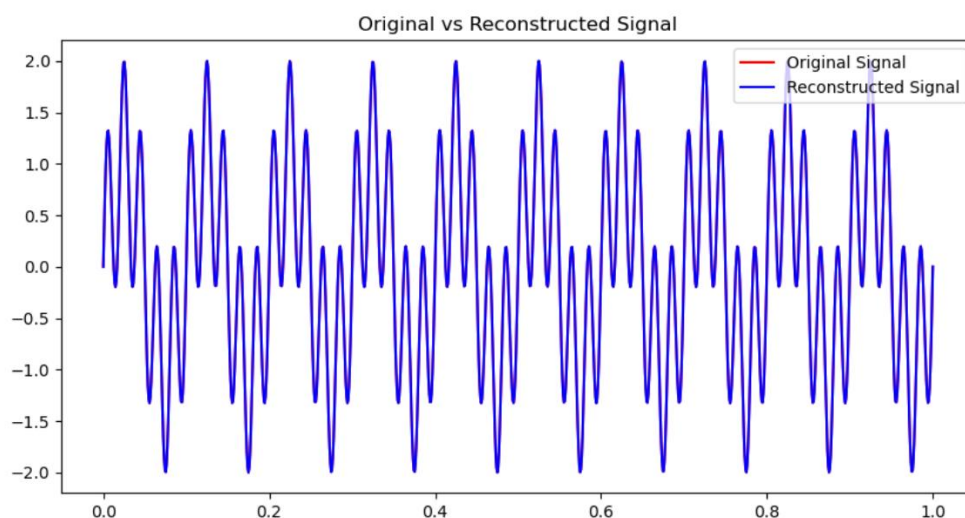


Рисунок 3 – Восстановленный сигнал

Как было написано выше, вейвлеты обладают рядом свойств, которые делают их полезными для анализа сигналов и изображений. Ниже приведены некоторые из основных свойств вейвлетов:

1. Компактность: вейвлеты являются компактными функциями, что означает, что они имеют компактную поддержку в пространстве и частотной области. Это позволяет использовать их для анализа сигналов на различных уровнях разрешения и частот.

2. Ортогональность: некоторые вейвлеты обладают свойством ортогональности, что означает, что они могут использоваться для ортогонального разложения сигнала на компоненты различных масштабов и уровней разрешения. Ортогональность вейвлетов позволяет эффективно использовать их для анализа и обработки сигналов.

3. Мультирезолюционность: вейвлеты могут использоваться для мультирезолюционного анализа сигналов, что значит, что они могут анализировать сигналы на разных уровнях разрешения. Это позволяет извлекать информацию из сигналов на различных уровнях детализации.

4. Локализация: вейвлеты имеют свойство локализации, что означает, что они могут точно определить местоположение особенностей сигнала. Это позволяет использовать их для обнаружения и извлечения особенностей сигнала, таких как ребра, контуры, текстуры и т.д.

5. Вариантность: вейвлеты могут быть вариантными или инвариантными к сдвигу и масштабированию. Вариантность к сдвигу означает, что вейвлеты могут обнаруживать и извлекать информацию из сигнала, даже если он сдвинут. Несмотря на это, вейвлеты могут быть нечувствительными к масштабированию, что означает, что они могут не обнаруживать изменения размера объектов на изображении.

6. Изменяемость: вейвлеты могут быть изменены и адаптированы для работы с различными типами сигналов и изображений. Это позволяет использовать их для решения различных задач обработки сигналов и изображений.

2) В каких областях обработки данных вейвлет-преобразования находят себя (используются)?

1) Обработка экспериментальных данных. Вейвлеты появились именно как механизм обработки экспериментальных данных, потому что по своей сути вейвлет-преобразование дает наиболее наглядную и информативную картинку результатов эксперимента, это позволяет очистить исходные данные от шумов и случайных искажений, и даже "на глаз" подметить некоторые особенности данных и направление их дальнейшей обработки и анализа.

2) Обработка изображений. В результате использования вейвлет-преобразований, можно сгладить неровности картинки, выделять некоторые детали в особые детали в изображениях, менять его размерность, а также улучшать качество картинки.

3) Сжатие данных. Ещё одной особенностью вейвлет преобразований является, что полученные данные при сжатии, могут быть легко обратно преобразованы (распакованы) при помощи тех же самых фильтров и средств. Кроме того, следует отметить, что вейвлет преобразования достаточно легко реализует сжатие с потерями, для этого достаточно просто отбросить детали на тех масштабах, где они не так важны. Как раз-таки изображения обработанные с помощью вейвлет преобразований можно сжать в 3-10 раз без больших потерь информации. В качестве примера отмечу, что вейвлет-преобразование является основой стандарта сжатия данных MPEG4.

4) Нейросети и другие механизмы анализа данных. Нам известно, что большие трудности при обучении нейросетей создает сильная зашумленность данных или наличие большого числа случайных выбросов, пропусков, нелинейных искажений и т.д.. Данные помехи или выбросы значительно искажают

восприятие полученной информации. Поэтому рекомендуется очистить данные, прежде чем анализировать их. По уже приведенным выше соображениям, а также благодаря наличию быстрых и эффективных алгоритмов реализации, вейвлеты представляются весьма удобным и перспективным механизмом очистки и предварительной обработки данных для использования их в статистических и бизнес-приложениях, системах искусственного интеллекта и т.п.

5) Системы передачи данных и цифровой обработки сигналов. Оно может использоваться для обнаружения различных особенностей в сигналах и таких как контуры, текстуры и т. д.

Несмотря на то что вейвлет преобразования используются во многих областях анализа данных, они также имеют свои минусы при работе и взаимодействии с ними.

3) Плюсы и минусы применения вейвлет преобразований:

Плюсы:

- Вейвлет преобразование может анализировать сигналы с различными частотами, что делает его более мощным, чем традиционные методы анализа сигналов. Это означает, что вейвлет преобразование может обнаруживать и извлекать информацию из сигнала, которую традиционные методы не могут обнаружить.

- Оно может быть использовано для декомпозиции сигналов на различные уровни разрешения, что позволяет более эффективно анализировать сигналы. Вейвлет преобразование позволяет разбить сигнал на компоненты, которые могут быть анализированы отдельно друг от друга. Это позволяет обнаруживать и анализировать детали сигнала на разных уровнях разрешения.

- Вейвлет преобразование может использоваться для компрессии изображений, что позволяет сократить объем данных, не ухудшая качество изображения. Это достигается путем удаления избыточной информации из изображения, сохраняя при этом все необходимые детали.

- Оно может использоваться для обнаружения различных особенностей в сигналах и изображениях, таких как ребра, контуры, текстуры и т.д. Это позволяет использовать вейвлет преобразование для решения многих задач обработки сигналов и изображений, таких как распознавание образов, сжатие данных и т.д.

Минусы:

- Вейвлет преобразование может быть сложным для понимания и использования для тех, кто не имеет опыта в математике и сигнальной обработке. Кроме того, выбор подходящего вейвлета для конкретной задачи может быть сложным.

- Оно может привести к искажению сигнала, если не используется правильно. Вейвлет преобразование может быть чувствительным к шуму и другим искажениям в сигнале, поэтому важно правильно выбирать параметры преобразования и методы обработки данных.

- Вейвлет преобразование может быть медленнее, чем другие методы анализа сигналов. Это связано с вычислительной сложностью преобразования и необходимостью анализировать сигнал на разных уровнях разрешения.

4) Реализация вейвлетов на языке программирования Python.

В Python для работы с вейвлет-преобразованием можно использовать модуль PyWavelets. Вот пример кода для применения вейвлет-преобразования на одномерном массиве сигнала:

```
import pywt
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
# Создаем пример одномерного сигнала
```

```
signal = np.array([1, 2, 3, 4, 5, 6, 7, 8])
```

```
# Задаем тип вейвлета и уровень декомпозиции
```

```
wavelet = 'db4'
```

```
level = 2
```

```
# Применяем вейвлет-преобразование
```

```
coeffs = pywt.wavedec(signal, wavelet, level=level)
```

```
print(coeffs)
```

В результате выполнения кода в переменной `coeffs` будет храниться список коэффициентов декомпозиции сигнала на различные уровни разрешения. Первый элемент списка - это коэффициенты наивысшего уровня разрешения (т.е. наименьшей частоты), а последний элемент - это коэффициенты низшего уровня разрешения (т.е. наибольшей частоты).

Для восстановления сигнала из коэффициентов декомпозиции можно использовать функцию `waverec`:

```
# Восстанавливаем сигнал из коэффициентов декомпозиции
```

```
reconstructed_signal = pywt.waverec(coeffs, wavelet)
```

```
print(reconstructed_signal)
```

В результате выполнения кода в переменной `reconstructed_signal` будет храниться восстановленный одномерный сигнал.

```
# Получаем коэффициенты на уровне 1 и 2
```

```
cA2, cD2, cD1 = coeffs
```

```
# Получаем сигналы на уровне 1 и 2
```

```
signal_level2 = pywt.upcoef('d', cD2, wavelet, level=level)[:len(signal)]
```

```
signal_level1 = pywt.upcoef('a', cA2, wavelet, level=level)[:len(signal)]
```

Здесь мы использовали функцию `pywt.upcoef` для получения сигналов на каждом уровне детализации и построили график с помощью функции `plot`, используя исходный сигнал, компоненты на уровне 2 и аппроксимационную компоненту на уровне 1.

Результат выполнения кода - график вейвлет-сигнала:

```
# Построение графика вейвлет-сигнала
```

```
fig, ax = plt.subplots(figsize=(12, 6))
```

```
ax.plot(signal, label='Original Signal', color='black')
```

```
ax.plot(signal_level2, label='Level 2 Detail Coefficients', color='red')
```

```
ax.plot(signal_level1, label='Level 1 Approximation Coefficients', color='blue')
```

```
ax.legend()
```

```
plt.show()
```

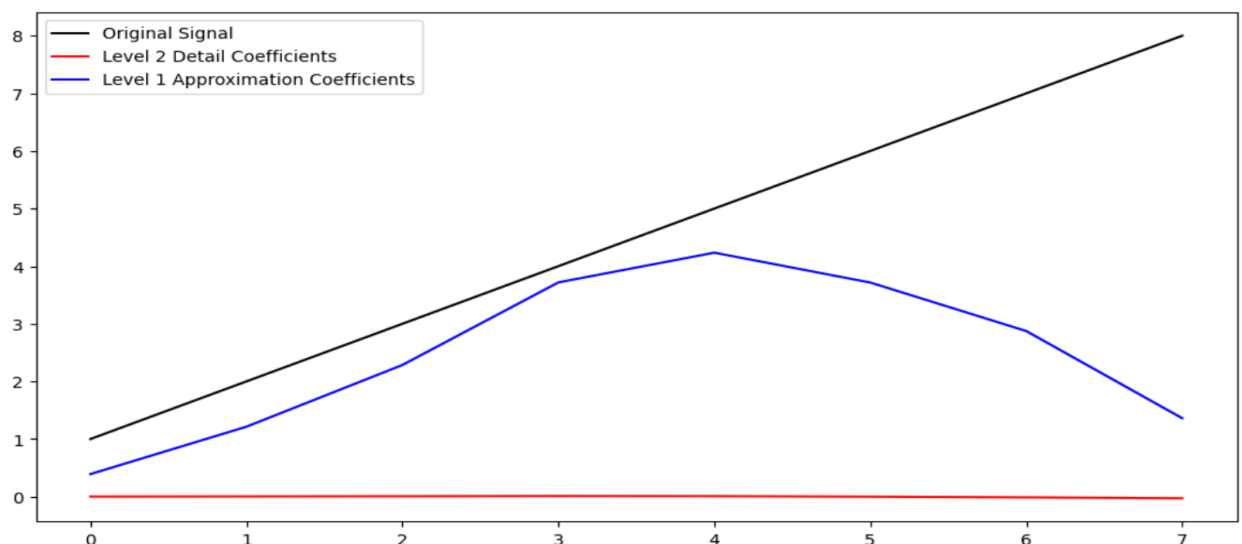


Рисунок 2 – Построение графика вейвлет сигнала

Из графика можно заметить, что вейвлет-преобразование позволяет разложить сигнал на компоненты различных частот и разрешений. Компоненты

на более высоких уровнях детализации соответствуют более высоким частотам, а компонента на уровне аппроксимации соответствует низким частотам.

Вывод: в результате ознакомления с данной статьёй можно сделать следующие выводы: вейвлет-преобразования (библиотека `pywt`) являются мощными средствами для обработки различных сигналов и их декомпозиции, на высокоуровневом языке программирования Python.

Список используемой литературы:

1. Основы теории вейвлет-преобразования. Url:<https://ba-segroup.ru/community/articles/intro-wavelets> (Дата обращения: 16.05.2023)
2. Н. М. Астафьева, Вейвлет-анализ: основы теории и примеры применения, УФН, 1996, том 166, номер 11, 1145–1170. Url:<https://www.math-net.ru/links/f6a79ef888705187c093cdc68bb65482/ufn1260.pdf> (Дата обращения: 16.05.2023)
3. Статья на тему: Вейвлет – анализ. Основы. Url:<https://habr.com/ru/articles/449646/> (Дата обращения: 16.05.2023)
4. Статья на тему: Вейвлет – анализ. Часть 1. <https://habr.com/ru/articles/451278/> (Дата обращения: 16.05.2023)
5. Статья на тему: Вейвлет-преобразование. Url:<http://gwyddion.net/documentation/user-guide-ru/wavelet-transform.html> (Дата обращения: 16.05.2023)
6. Статья на тему: Открытый курс машинного обучения. Тема 9. Анализ временных рядов с помощью Python. Url:<https://habr.com/ru/companies/ods/articles/327242/> (Дата обращения: 17.05.2023)
7. РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ДЛЯ АНАЛИЗА ГЕОМАГНИТНЫХ ВАРИАЦИЙ НА ОСНОВЕ ВЕЙВЛЕТ-ТЕХНОЛОГИЙ. Авторы: С.Н. Верзунов Н.М. Лыченко. Кыргызско-Российский славянский университет, г. Бишкек. Url:<http://msm.omsu.ru/jrns/jrn32/Verzunov.pdf> (Дата обращения 16.05.2023)

8. Вейвлет-преобразование (Wavelet Transform).

Url:<https://wiki.loginom.ru/articles/wavelet-transform.html> (Дата обращения:
16.05.2023)