

# **ЗАХИСТ АУДІОФАЙЛІВ НА ОСНОВІ ДИСКРЕТНОГО ВЕЙВЛЕТ-ПЕРЕТВОРЕННЯ**


**Танасійчук Анна Олександрівна,  
учениця 11 класу  
комунального закладу  
«Центральноукраїнський науковий  
ліцей-інтернат Кіровоградської обласної ради»**

**Інформаційна війна з росією – важливий аспект сучасної війни, тому нам варто перемагати і на фронті захисту інформації і даних.**

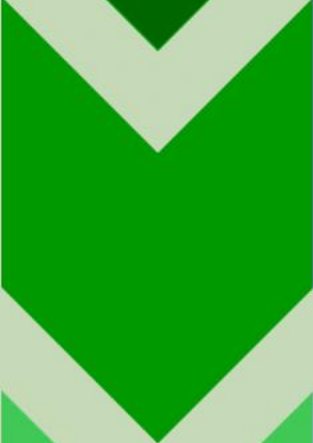
**Захист інформації та ліцензований контент – європейські цінності.**



Рис. 1. Інформаційна війна з росією (<https://chas.cv.ua/autor/58292-інформаційна-війна-рф-проти-україни.html>)



- **Актуальність теми:** Захисна система дозволить власникам аудіоконтенту або спецслужбам впроваджувати водяний знак у аудіофайли та створювати цифрові бібліотеки аудіо.



- Наукова новизна полягає в розробці двох алгоритмів, які використовують дискретні вейвлет-перетворення:
  - створення цифрових відбитків аудіофайлів;
  - вбудовування в аудіодоріжку «водяного знаку».



- Практичне значення роботи полягає у розробці методів інформаційної боротьби з неліцензійним аудіоконтентом та розробці алгоритмів шифрування даних в аудіо.



# Вейвлет-перетворення

Вейвлет-перетворення сигналів є узагальненням спектрального аналізу, типовий представник якого – класичне перетворення Фур'є.

Вейвлети – це узагальнена назва сімейств математичних функцій певної форми, які локальні в часі та за частотою, і в яких всі функції виходять з однієї базової (що породжує) за допомогою її зсувів та розтягувань по осі часу.

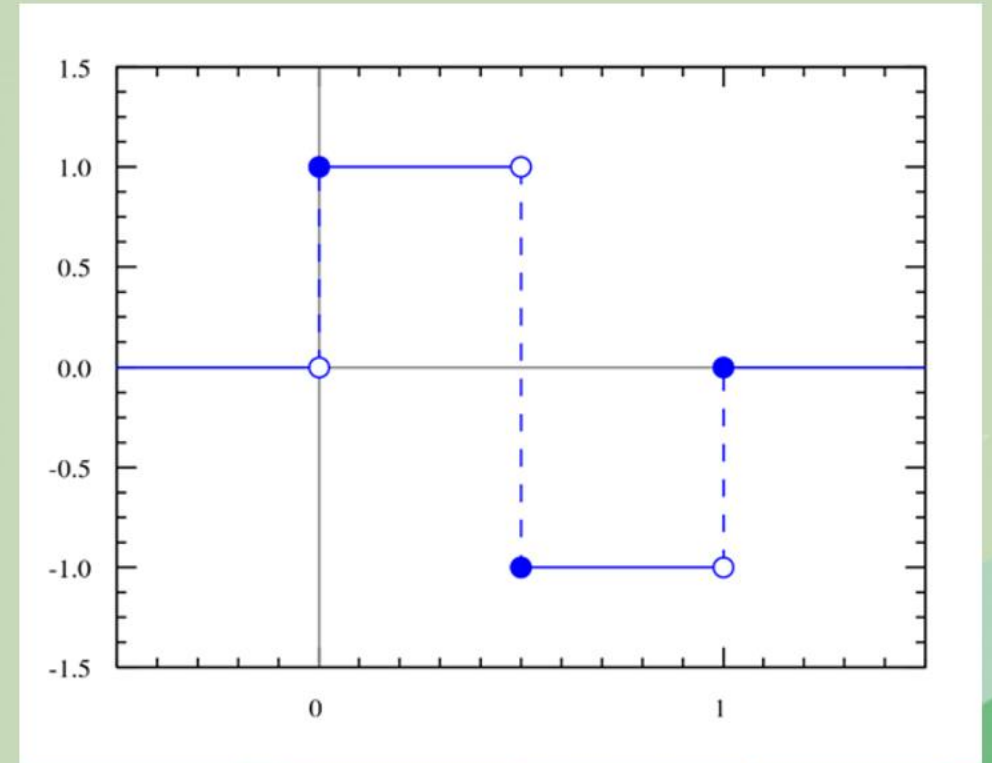


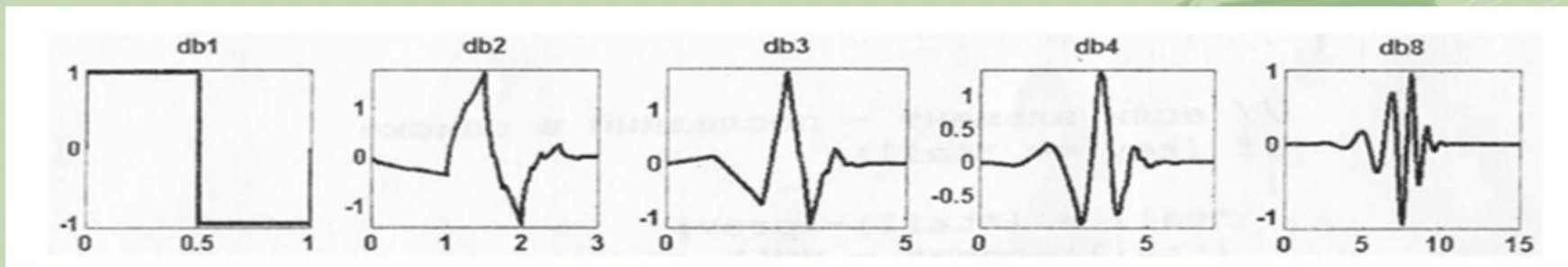
Рис. 2. Вейвлет Хаара – приклад вейвлет-перетворення ([https://ru.wikipedia.org/wiki/Вейвлет\\_Хаара](https://ru.wikipedia.org/wiki/Вейвлет_Хаара))

# Вейвлети Добеші

Вейвлети Добеші являють собою сімейство ортогональних вейвлетів, що визначають дискретне вейвлет-перетворення. Даного роду вейвлети обчислюються ітераційним шляхом.

**Основні властивості вейвлетів сімейства Добеші:**

- мають хороший локальний спектр в частотній області, характеризуються двома функціями: вейвлет-функцією і масштабуючою функцією;
- є вейвлетами ортогонального типу, зосереджені на кінцевому інтервалі часу і мають кінцеву кількість фільтруючих коефіцієнтів;
- здатні повністю відновити довільний локальний сигнал на нульовому рівні реконструкції;
- мають можливість виконувати дискретні перетворення із застосуванням алгоритмів швидкого вейвлет-перетворення (ШВП).



# Алгоритм створення бібліотеки цифрових знімків аудіофайлів



Рис. 2. Демонстрація відмінності кількості інформації у wav та mp3 файлах (Джерело: <https://www.off-the-beat.com/wav-vs-mp3/>)



# Алгоритм створення бібліотеки цифрових знімків аудіофайлів

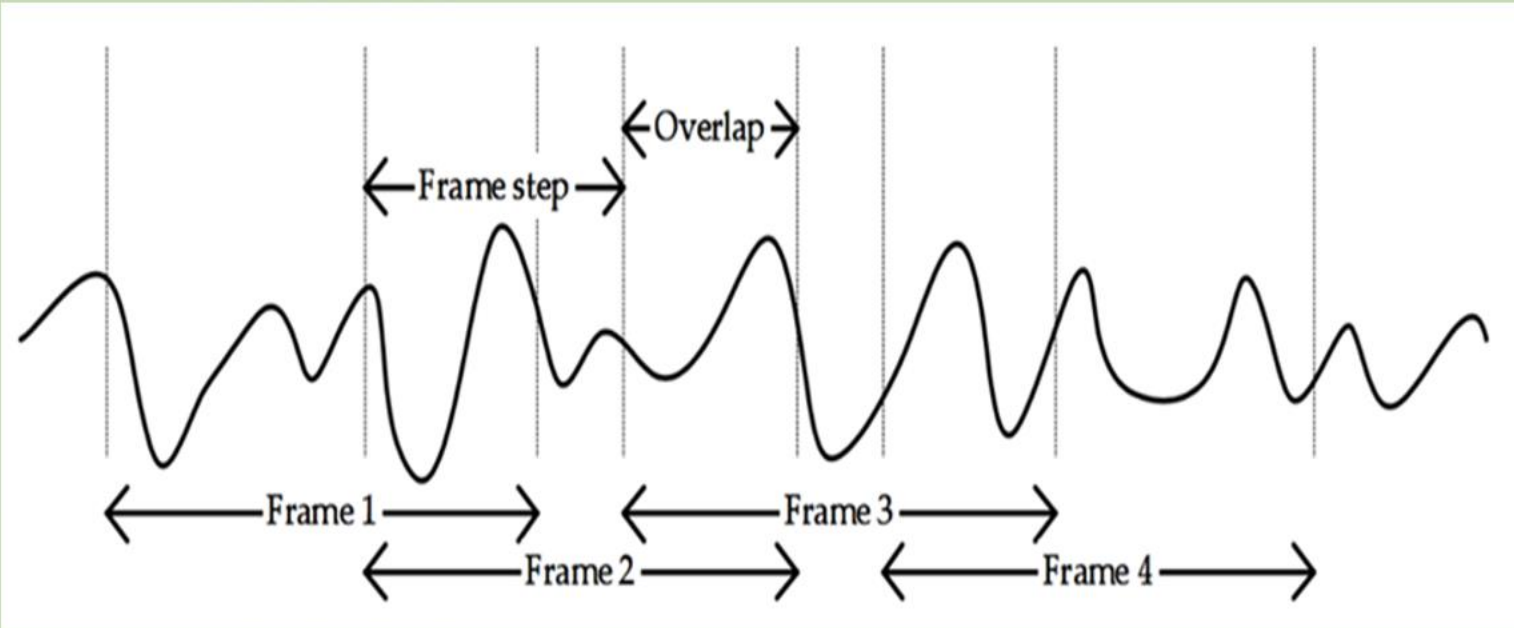
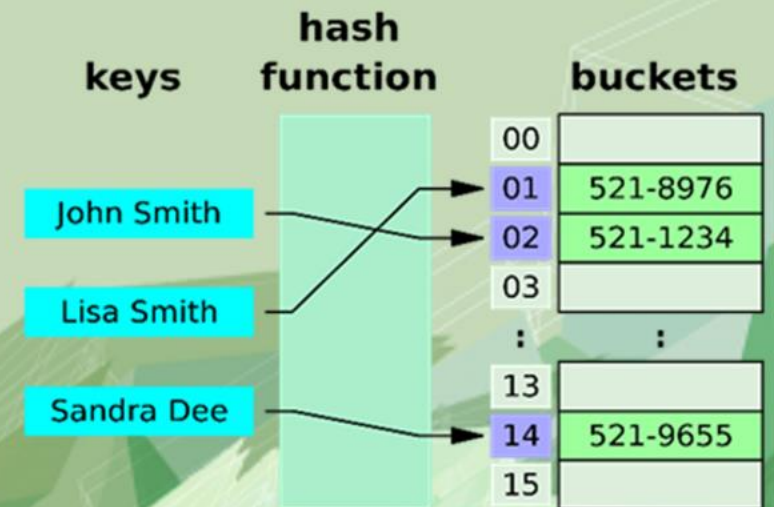


Рис. 5. Приклад використання геш-функції (Джерело: [https://en.wikipedia.org/wiki/Hash\\_table](https://en.wikipedia.org/wiki/Hash_table))

Рис. 3. Алгоритм розділення аудіофайлу на фрейми з перекриттям (Джерело: <https://i.stack.imgur.com/d7Rdx.png>)



# Алгоритм створення бібліотеки цифрових знімків аудіофайлів

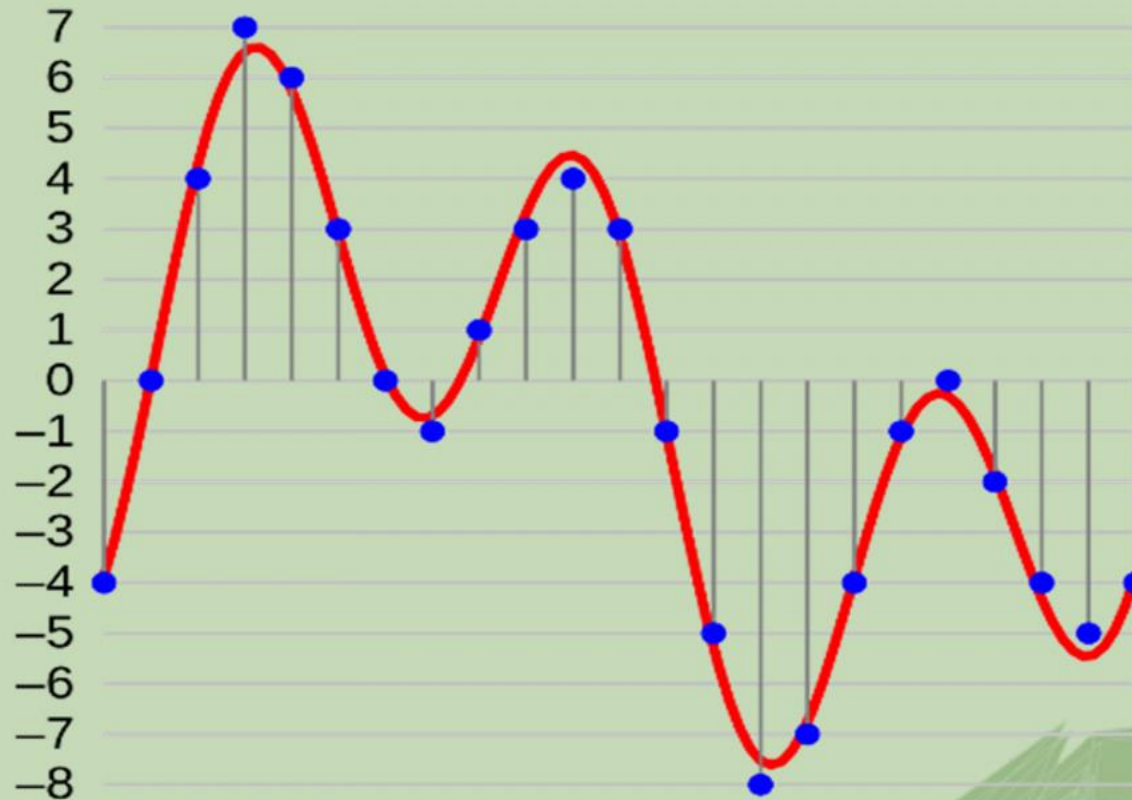


Рис. 4. Представлення акустичної інформації через цифрову

(Джерело:<https://upload.wikimedia.org/wikipedia/commons/thumb/2/21/4-bit-linear-PCM.svg/1200px-4-bit-linear-PCM.svg.png>)



# Стиснення аудіосигналу за допомогою вейвлет-перетворення Добеші db3

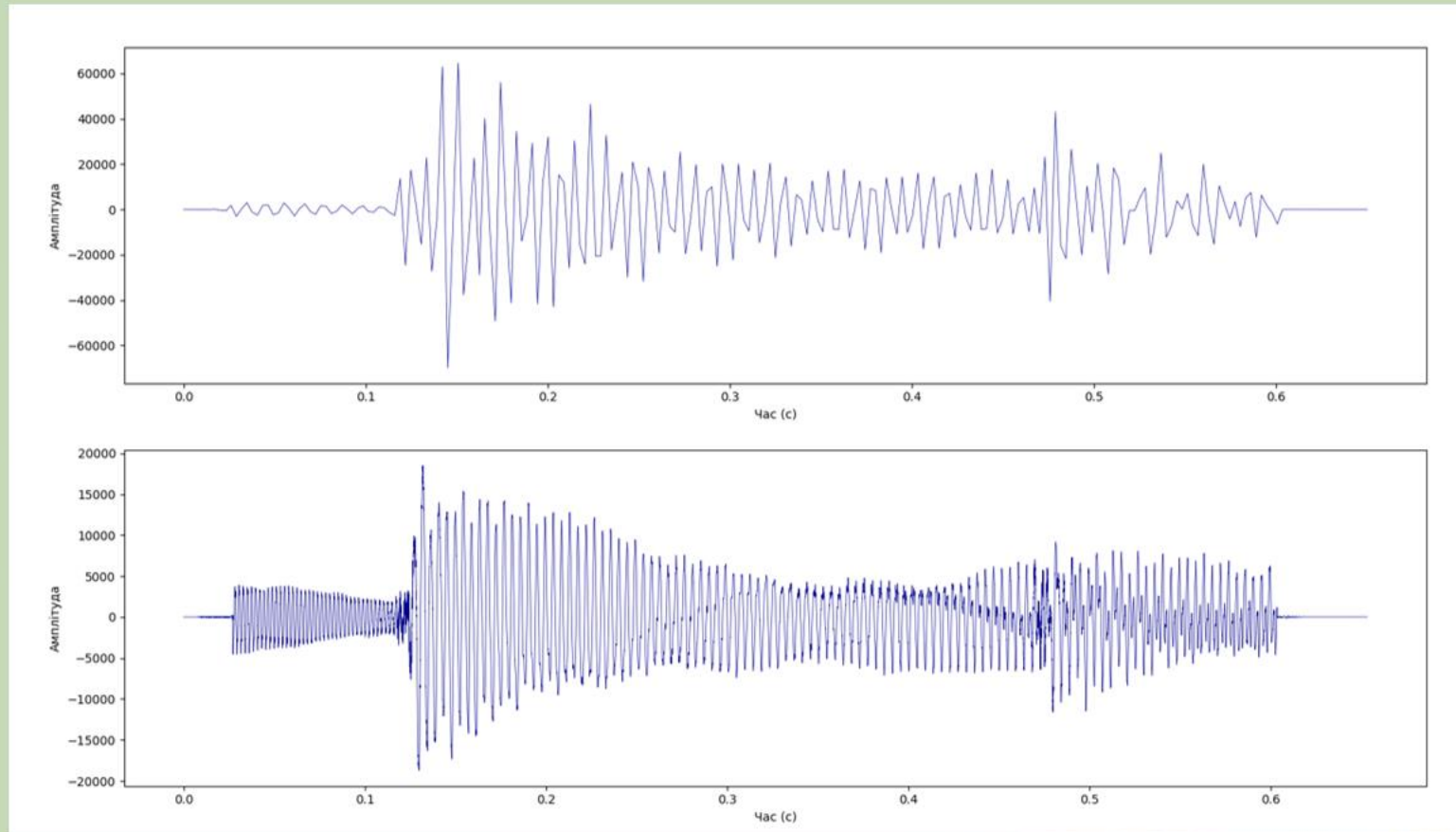


Рис. 5. Стиснення аудіосигналу за допомогою вейвлет-перетворення Добеші db3 7 разів (стиснення в 128 разів): знизу вхідний сигнал, зверху – стиснений сигнал (Власний скріншот)

# Алгоритм створення бібліотеки цифрових знімків аудіофайлів

Практичною спробою реалізувати цей алгоритм стала програма розроблена на мові програмування Python, яка перетворює частину mp3-файлу на цифровий відбиток аудіофайлу згідно цього алгоритму.

```
#number of dwt db3
cA1, cD1 = pywt.dwt(channel1, 'db3')
cA2, cD2 = pywt.dwt(cA1, 'db3')
cA3, cD3 = pywt.dwt(cA2, 'db3')
cA4, cD4 = pywt.dwt(cA3, 'db3')
cA5, cD5 = pywt.dwt(cA4, 'db3')
cA6, cD6 = pywt.dwt(cA5, 'db3')
cA7, cD7 = pywt.dwt(cA6, 'db3')

#compress time to cA7
time_compress = time[::128]

#prints
print("The total number of channel1:")
print(len(channel1))
print("The total number of 7 DWT:")
print(len(cA7))
print("Compression of channel1 after 7 dwt db3")
print(len(channel1)/len(cA7))

#plot
plt.figure(1)
plt.subplot(211)
plt.plot(time_compress, cA7[4:], linewidth=0.5, alpha=1, color='#ff7f00')
plt.xlabel('Час (с)')
plt.ylabel('Амплітуда')
plt.subplot(212)
plt.plot(time, channel1, linewidth=0.5, alpha=1, color='#ff7f00')
plt.xlabel('Час (с)')
plt.ylabel('Амплітуда')
plt.show()
```



Код на мові програмування Python створення бібліотеки цифрових знімків аудіофайлів.

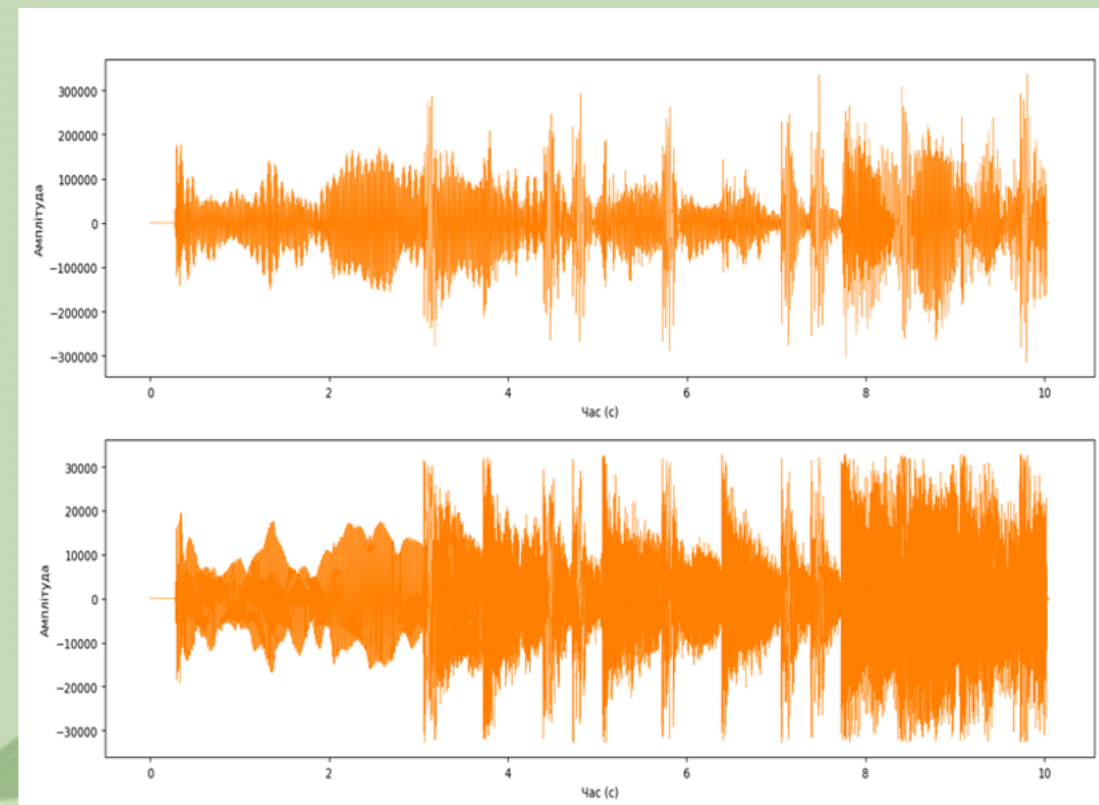


Рис. 6. Робота алгоритму в Python (Джерело: власний скріншот)

# Алгоритм ін'єктування водяного знаку в аудіофайл

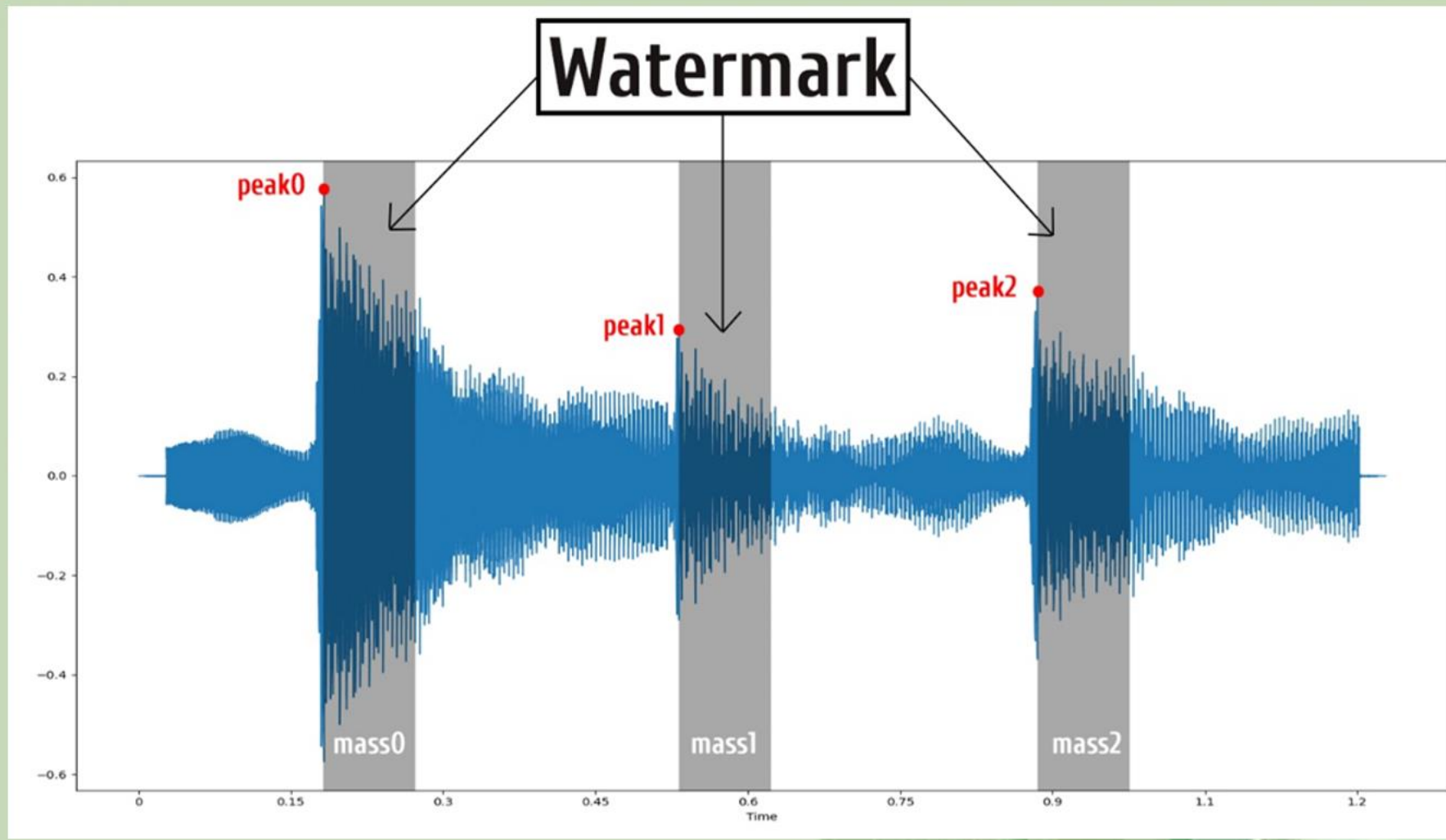


Рис. 7. Алгоритм визначення місць для масивів вбудовування водяних знаків (власна схема).



# Алгоритм ін'єктування водяного знаку в аудіофайл

Виконуємо послідовно ДВП (найкраще 5-7), при цьому отримуємо стиснений масив і масиви уточнюючих коефіцієнтів (рис.7).

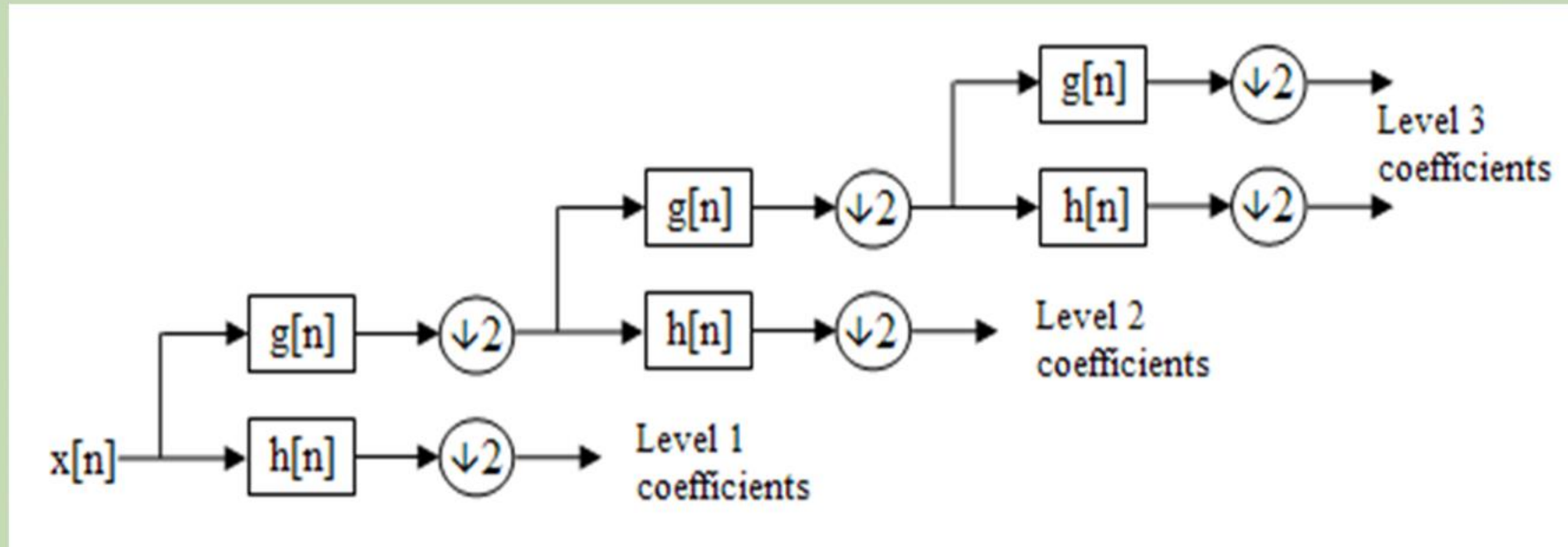


Рис. 8. Приклад трьох послідовних ДВП, де  $g(n)$  стиснений масив  $x(n)$ , а  $h(n)$  відповідні уточнюючі коефіцієнти для кожного рівня

(Джерело: ([https://ru.wikipedia.org/wiki/Дискретное\\_вейвлет-преобразование](https://ru.wikipedia.org/wiki/Дискретное_вейвлет-преобразование))).

# Алгоритм ін'єктування водяного знаку в аудіофайл

```
#dwt with mass0
cA1, cD1 = pywt.dwt(dic_mass["mass0"], 'db2')
cA2, cD2 = pywt.dwt(cA1, 'db2')
cA3, cD3 = pywt.dwt(cA2, 'db2')
cA4, cD4 = pywt.dwt(cA3, 'db2')
cA5, cD5 = pywt.dwt(cA4, 'db2')
cA6, cD6 = pywt.dwt(cA5, 'db2')
cA7, cD7 = pywt.dwt(cA6, 'db2')

#print(cA7)
print("Massive after 7 dwt with mass0 - "+str(cD7))

#massive of watermark
watermark = np.linspace(10, 10, num=len(cD7))

#add watermark and cD7
cD7 = cD7+watermark


#decomposition
cA6 = pywt.idwt(cA7, cD7, 'db2')
cA5 = pywt.idwt(cA6, cD6, 'db2')
cA4 = pywt.idwt(cA5, cD5, 'db2')
cA3 = pywt.idwt(cA4, cD4, 'db2')
cA2 = pywt.idwt(cA3, cD3, 'db2')
cA1 = pywt.idwt(cA2, cD2, 'db2')
mass_decompoz = pywt.idwt(cA1[:-1], cD1, 'db2')
mass0 = dic_mass["mass0"]
```

Рис. 9. Виконання ДВП, вбудовування водяного знаку та виконання декомпозиції.

Перетворюємо цифрові семпли на аудіофайл wav, mp3 або інші в залежності від потреби.

```
#return audiodata
audData[:,0]=channel1

#save new data as watermark_audio.wav
scipy.io.wavfile.write("watermark_audio.wav", rate, audData)
```

 watermark\_audio.wav

Створюємо новий аудіофайл з водяним знаком



Код на мові програмування Python ін'єктування водяного знаку в аудіофайл



# Висновки

---

- проаналізовано спеціальну літературу, узагальнено сучасні уявлення про ДВП, зокрема про одновимірне та двовимірне перетворення Хаара, ДВП Добеші та інші ДВП;
- наведено відомості про сучасні алгоритми захисту інформації: цифрові відбитки аудіофайлів та водяні знаки, їх види, методи їх впровадження, особливості їх вбудовування та способи виявлення;
- описано сучасні технології створення цифрових відбитків аудіофайлів;
- розроблено два алгоритми захисту аудіофайлів засобами дискретного вейвлет-перетворення: створення цифрової бібліотеки знімків, алгоритм вбудовування водяного знаку в аудіофайл, на основі яких написані відповідні програми мовою програмування Python.



# **ЗАХИСТ АУДІОФАЙЛІВ НА ОСНОВІ ДИСКРЕТНОГО ВЕЙВЛЕТ-ПЕРЕТВОРЕННЯ**

**Танасійчук Анна Олександрівна,  
учениця 11 класу  
комунального закладу  
«Центральноукраїнський науковий  
ліцей-інтернат Кіровоградської обласної ради»**