Wavelet Transform in Practice

From Theory to Production-Ready Python Applications

VOLUME III-B

Financial, Security, and Digital Systems

Finance, Data Security, Signal Processing, and Anomaly Detection

Shouke Wei

Wavelet Transform in Practice

From Theory to Production-Ready Python Applications

Series

Wavelet Transform in Practice

From Theory to Production-Ready Python Applications

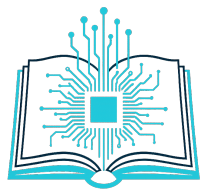
VOLUME III-B

Financial, Security, and Digital Systems

Finance, Data Security, Signal Processing, and Anomaly Detection

---

Shouke Wei



DEEPSIM PRESS

First Edition, 2026

For permission requests, contact the publisher:
Email: shouke.wei@deepsim.ca

ISBN 978-1-0699284-9-8 (eBook)
DOI 10.5281/zenodo.18933173

Published by
Deepsim Press
An independent imprint of Deepsim Intelligence Technology Inc.
Abbotsford, British Columbia, Canada
https://press.deepsim.ca

This book is intended for educational and professional readers.

For resources, updates, and companion code, visit:
https://press.deepsim.ca/wavelet-books

About the Author

Shouke Wei, Ph.D., is a researcher, scientist, and entrepreneur specializing in intelligent IoT systems, robotics, big data analytics, modeling and forecasting, early-warning systems, and edge computing. With academic and industry experience across Europe, North America, and Asia, Dr. Wei is recognized for bridging advanced theory with real-world, production-ready systems.

Dr. Wei earned his Ph.D. in Environmental and Resource Management from the Department of Environmental Informatics at Brandenburg University of Technology Cottbus–Senftenberg (Germany). He conducted postdoctoral research at the Swiss Federal Institute of Aquatic Science and Technology (Eawag), where he also served as a doctoral supervisor, and held research positions at the University of British Columbia (Canada).

Recognized as a National High-End Talent (Class A) in China, Dr. Wei has held distinguished and adjunct professorships at multiple institutions, including Yantai University, Ludong University, and Jining University. He has served as a graduate supervisor and distinguished professor in computer science, control engineering, and applied mathematics.

Dr. Wei currently serves as CEO and Chief Scientist of Deepsim Intelligent Technology Inc. (Canada), Chief Scientist at Canadian Sincerity Enterprises Inc., and Chief Scientist of Shandong Deepsim Intelligent Technology Co., Ltd. He is also a Postdoctoral Co-Supervisor at the Shandong Postdoctoral Innovation Practice Base and currently serves as Director of Qilu Artificial Intelligence and Digital Manufacturing Innovation at Shandong Deepsim Intelligent Technology Co., Ltd., China.

Dr. Wei has led or contributed to 19 major international research projects and the development of 19 intelligent systems, including autonomous water-quality monitoring vessels, AI-based environmental early-warning platforms, medical image diagnosis models, precision agriculture robots, and autonomous service robots.

His scholarly contributions include 40+ peer-reviewed publications and 500+ tutorial online articles, 6 patents, 30 software copyrights, and 2 China national scientific and technological achievements. Dr. Wei's work focuses on making advanced computational methods—particularly wavelet-based signal processing—accessible, practical, and impactful for researchers and practitioners worldwide.

For more info, visit: `https://shouke.deepsim.ca`

Contents

Preface

Wavelet methods are widely recognized for their ability to reveal structure in signals that evolve across multiple temporal and frequency scales. While these techniques are often introduced in mathematical or theoretical contexts, their practical significance emerges most clearly when they are applied within information-driven systems where signals must be interpreted, monitored, and acted upon in real time. This volume of *Wavelet Transform in Practice: From Theory to Production-Ready Python Applications* focuses on such systems, examining how wavelet-based analysis supports financial modeling, information security, digital signal processing, and anomaly detection in modern data environments.

Building upon the theoretical foundations established in **Volume I** and the methodological developments explored in **Volume II**, this volume, *Financial, Security, and Digital Systems: Finance, Data Security, Signal Processing, and Anomaly Detection* turns toward application domains in which signals arise primarily from digital infrastructure and economic systems. Financial markets, communication networks, and digital media streams produce time series that are noisy, nonstationary, and often shaped by complex human and technological interactions. Wavelet-based analysis offers a powerful framework for examining these signals because it enables localized multiscale representations that capture transient events, regime shifts, and hidden structural patterns.

The chapters in this volume examine several domains in which multiscale analysis provides meaningful analytical insight. These include wavelet-based approaches to data security and steganography, where transform-domain representations enable robust information embedding and detection; financial time-series analysis, where wavelet decompositions help reveal volatility

dynamics and cross-scale interactions in market behavior; digital audio signal analysis, where wavelets support efficient representation, denoising, and feature extraction; and multiscale anomaly detection, where wavelet-based features assist in identifying abnormal events within complex systems.

A recurring theme throughout this volume is the relationship between **multiscale representation and interpretability**. Signals generated by financial markets, communication systems, and digital media frequently exhibit behaviors that manifest differently across time scales. Wavelet transforms provide a means of separating these scales, but meaningful interpretation requires careful consideration of system context, domain knowledge, and the mechanisms that generate the observed data. For this reason, the emphasis in this volume extends beyond algorithmic implementation to include analytical reasoning and system-aware interpretation.

Consistent with the philosophy of the series, all examples are implemented using reproducible Python workflows. The objective is not to introduce novel algorithms, but to demonstrate how wavelet-based analysis can be integrated into real analytical pipelines used for monitoring, detection, and interpretation. Readers are encouraged to adapt the provided implementations to their own data sources and operational systems.

Who This Book Is For

This volume is intended for:

- Data scientists and analysts working with financial or digital signal data
- Engineers and researchers studying information security and signal processing systems
- Practitioners developing anomaly detection and monitoring systems
- Analysts interpreting complex multiscale behavior in financial markets or digital media

- Researchers seeking domain-oriented examples of wavelet-based workflows

Readers with a working understanding of wavelet theory who wish to explore how multiscale analysis operates within financial, security, and digital signal systems will find this volume particularly useful.

Organization of This Volume

This volume is organized into three parts reflecting major classes of information-driven systems.

Part I: Security and Information Systems
Examines wavelet-based techniques for data security and steganography, emphasizing robustness, detectability, and transform-domain information representation.

Part II: Financial Systems
Explores wavelet analysis of financial time series, focusing on volatility dynamics, multiscale market behavior, and regime changes.

Part III: Digital Signal and Monitoring Systems
Presents wavelet-based approaches to audio signal analysis and multiscale anomaly detection, highlighting applications in monitoring and diagnostic systems.

On Systems, Interpretation, and Reproducibility

Reproducibility remains a central principle throughout this volume. Python-based examples are designed to be transparent, modular, and adaptable. At the same time, equal emphasis is placed on interpretation and system context. Analytical results are evaluated not only in terms of numerical performance, but also with respect to their relevance to real-world systems and operational decision-making.

Concluding Perspective

Wavelet transforms provide a powerful bridge between mathematical signal processing and practical data analysis. In financial systems, information security, and digital media environments, multiscale representations reveal patterns that are difficult to detect using single-scale techniques. When combined with domain knowledge and careful interpretation, these representations can support more effective monitoring, analysis, and decision-making.

Together with the preceding volumes, this book completes a progression from foundational theory, through methodological development, to real-world analytical systems. It is my hope that readers will not only gain practical tools for working with multiscale signals, but also develop a deeper understanding of how wavelet-based reasoning can illuminate complex modern data environments.

Shouke Wei, PhD
Deepsim Intelligent Technology Inc.
Deepsim Academy
Abbotsford, Canada
March 18, 2026

Acknowledgments

This volume extends the *Wavelet Transform in Practice* series into application domains shaped by information systems, financial markets, and digital signal environments. Developing this part of the series required not only technical analysis, but also an understanding of how analytical tools interact with operational systems, data infrastructures, and decision processes. I am deeply grateful to the many individuals and communities whose insights and experiences contributed to this perspective.

I thank colleagues and collaborators working in financial analytics, signal processing, cybersecurity, and data-driven monitoring systems for sharing their experiences applying analytical techniques within complex operational environments. Their perspectives highlighted the practical challenges of interpreting multiscale signals in domains where uncertainty, noise, and evolving system dynamics are unavoidable.

I am especially grateful to my students and research collaborators for their thoughtful discussions and critical questions about real-world applications of wavelet-based methods. Their engagement continually shaped how many ideas in this volume were clarified and refined. I also thank my family for their patience, encouragement, and unwavering support throughout the writing of this trilogy.

This work builds upon decades of foundational research in wavelet theory and multiscale signal analysis. I gratefully acknowledge the contributions of pioneers such as Ingrid Daubechies, Stéphane Mallat, and many other researchers whose work continues to guide the development of modern multiscale methods.

I also extend my sincere appreciation to the open-source scientific computing community—particularly the contributors to **PyWavelets**, **NumPy**, **SciPy**, **Matplotlib**, **pandas**, **scikit-learn**, **streamlit**, and related projects—whose tools make transparent and reproducible data analysis possible.

This volume was prepared using **Python**, **Jupyter Notebook/Lab**, **Quarto**, **MyST Markdown**, and **LaTeX**. I thank the communities behind these platforms for enabling reliable and accessible technical communication.

Closing Acknowledgment to the Trilogy

This volume brings the *Wavelet Transform in Practice* trilogy to completion. Together, the three volumes reflect an intended progression—from theoretical foundations, to applied methodologies, to real-world systems and decision contexts. Any coherence achieved across this work is the result of a broader community of researchers, educators, practitioners, and open-source contributors who continue to advance multiscale analysis in both theory and practice. I am deeply grateful to all who made this journey possible.

Notation and Abbreviations

This document consolidates the most relevant mathematical symbols, wavelet notation, performance metrics, and domain-specific abbreviations appearing across **all provided chapters** (Anomaly Detection, Stock Analysis, Audio Denoising, Steganography, Data Security).

Core Mathematical & Wavelet Symbols

Symbol / Abbreviation	Meaning	Main Chapter(s)	Notes / Typical Usage
$x(n)$, $x(t)$	Discrete / continuous signal	All chapters	Input time series or signal
t	Time variable	All chapters	Continuous time
n	Sample index	All chapters	Discrete index
j	Decomposition level / scale index	All chapters	DWT / CWT level
k	Translation index	All chapters	Position in wavelet transform
$\psi(t)$, $\psi_{j,k}$	Mother wavelet / wavelet basis function	All chapters	Core wavelet function
$W(j,k)$	Wavelet coefficient	Anomaly, Stock, Security, Steganography	General coefficient notation

Symbol / Abbreviation	Meaning	Main Chapter(s)	Notes / Typical Usage
$W'(j,k)$	Modified wavelet coefficient	Security, Steganography	After embedding / encryption
A_j / cA_j	Approximation coefficients (level j)	Anomaly, Stock, Audio, Security	Low-frequency / smooth part
D_j / cD_j	Detail coefficients (level j)	Anomaly, Stock, Audio, Security	High-frequency / detail part
DWT	Discrete Wavelet Transform	All chapters	Most common transform
CWT	Continuous Wavelet Transform	Stock, Anomaly (scalogram)	Used for scalograms / time-frequency visualization
MRA	Multi-Resolution Analysis	Anomaly, Security	Wavelet decomposition framework

Performance & Quality Metrics

Symbol / Abbreviation	Meaning	Main Chapter(s)	Notes / Typical Usage
SNR	Signal-to-Noise Ratio	Audio, Anomaly	Denoising & anomaly evaluation (often in dB)
PSNR	Peak Signal-to-Noise Ratio	Steganography, Audio, Security	Image quality / imperceptibility metric
MSE	Mean Squared Error	Steganography, Audio	Reconstruction / distortion measure
SSIM	Structural Similarity Index	Steganography, Audio	Perceptual image quality metric

Domain-Specific Abbreviations & Concepts

Symbol / Abbreviation	Meaning	Main Chapter(s)	Notes / Typical Usage
ASR	Automatic Speech Recognition	Audio (speech denoising)	Downstream task quality measure
NAB	Numenta Anomaly Benchmark	Anomaly detection	Benchmark dataset & evaluation framework
HH, HL, LH, LL	Wavelet subbands (diagonal, vertical, horizontal, approximation)	Steganography, Security, Audio	Standard DWT subband notation
LSB	Least Significant Bit	Steganography	Classical embedding technique (comparison)
QIM	Quantization Index Modulation	Steganography	Quantization-based hiding method
(alpha)	Embedding strength / scaling factor	Steganography	Controls trade-off between capacity & imperceptibility

Common Wavelet Families Mentioned

Symbol / Abbreviation	Meaning	Main Chapter(s)	Notes / Typical Usage
db4	Daubechies wavelet (order 4)	Steganography, Audio, Anomaly	Very frequently used
sym5, sym3	Symlet wavelets	Steganography, Audio	Good symmetry properties
coif3	Coiflet wavelet	Steganography, Audio	—

Symbol / Abbreviation	Meaning	Main Chapter(s)	Notes / Typical Usage
morl	Morlet wavelet	Stock analysis	Classic choice for CWT / scalograms
gaus4	Gaussian wavelet (order 4)	Stock analysis	Used in financial scalograms
mexh	Mexican hat wavelet	Stock analysis	Common for CWT
bior	Biorthogonal wavelets	Audio, Steganography	Often used when perfect reconstruction needed

Other Frequently Used Terms

Symbol / Abbreviation	Meaning	Main Chapter(s)	Notes / Typical Usage
stego-image	Image with hidden data	Steganography	Output after embedding
cover image	Original / host image	Steganography	Image before embedding
scalogram	Time-frequency representation (CWT)	Stock, Anomaly	Visual time-frequency map

Setup Python Environment

This volume continues to use the Python environment established in the earlier books of the *Wavelet Transform in Practice* series.

If you have already completed the setup instructions from **Volume I**, **Volume II-A**, **Volume II-B**, or **Volume III-A**, your system should already contain most of the required packages.

Only a small number of additional libraries are introduced in this volume to support the practical application examples.

Core Scientific Python Stack

All chapters rely on the standard scientific Python ecosystem:

- **NumPy** — numerical computing
- **SciPy** — scientific and signal processing utilities
- **Pandas** — data manipulation and tabular structures
- **PyWavelets** — wavelet transforms
- **Matplotlib** — visualization

These libraries provide the computational foundation used throughout the examples.

Additional Packages for Application Chapters

Volume III-B focuses on **real-world applications of wavelet analysis**.

Some chapters introduce additional packages depending on the data domain.

Chapter	Main Packages
Anomaly detection	numpy, scipy, pandas, pywavelets, matplotlib, scikit-learn
Financial analysis	yfinance, pandas, duckdb, plotly
Speech denoising	librosa, soundfile, scipy
Image steganography	opencv-python
Data security examples	opencv-python, scipy

Not every chapter requires every library. Readers may install only the packages needed for the chapters they intend to explore.

Installing Packages

The easiest approach is to install all packages at once using `pip`.

```
pip install numpy scipy pandas pywavelets matplotlib \
            scikit-learn yfinance duckdb plotly \
            librosa soundfile pydub opencv-python streamlit
```

Some audio examples may optionally use real-time audio libraries:

```
pip install pyaudio
```

If `pyaudio` is difficult to install on your system, you may instead use:

```
pip install sounddevice
```

Installing from requirements.txt

For reproducibility, the project repository includes a `requirements.txt` file.

You may install the complete environment using:

```
pip install -r requirements.txt
```

This ensures the package versions match those used to generate the results shown in the book.

Optional Deep Learning Libraries

A few optional extensions and experiments may use deep learning frameworks.

These are **not required** for most chapters.

```
pip install torch
```

or

```
pip install tensorflow
```

Virtual Environment (Recommended)

To avoid conflicts with other Python projects, it is recommended to use a virtual environment.

```
python -m venv .venv
```

Activate the environment:

Linux / macOS

```
source .venv/bin/activate
```

Windows

```
.venv\Scripts\activate
```

Then install the packages described above.

Data Sources

Examples in this volume use a mixture of:

- built-in datasets
- downloaded financial data (via `yfinance`)
- audio files
- image files

Readers are encouraged to adapt the examples to their own datasets.

Wavelet methods are widely applicable across domains, and the workflows presented here can be easily extended to new applications.

Summary

Compared with the earlier volumes, the computational setup for **Volume III-B** remains lightweight.

The emphasis of this volume is not on new mathematical libraries, but on **applying wavelet analysis in practical computational workflows** across domains such as anomaly detection, financial analysis, audio processing, and information security.

Part I

Security and Information Systems

1 Wavelet Transform for Data Security

Data security has become one of the most critical challenges in the digital era, where vast amounts of information are continuously transmitted, stored, and shared. Protecting sensitive data from unauthorized access, manipulation, or theft requires advanced methods beyond traditional cryptography. The wavelet transform, a powerful mathematical tool for signal and image analysis, has emerged as a versatile technique in data security due to its multi-resolution and localization properties in both time and frequency domains (Daubechies 1992; Stéphane Mallat 1999; Vetterli and Kovačević 1995).

Wavelets enable effective techniques for encryption, steganography , watermarking , and secure transmission by providing multi-resolution analysis (MRA), energy compaction, and joint time-frequency localization. Unlike traditional Fourier-based methods, wavelets preserve both spatial and frequency information simultaneously (Strang and Nguyen 1996), making them particularly suitable for multimedia security applications (Barni, Bartolini, and Piva 2001; Kundur and Hatzinakos 1998).

The evolution of wavelet-based security has progressed through several key phases:

1. **Early Applications (1990s-2000s)**: Initial use in digital watermarking and basic steganography (Cox 2008; Piva et al. 1997).
2. **Hybrid Systems (2000s-2010s)**: Integration with traditional cryptographic methods (Subramanyam, Emmanuel, and Kankanhalli 2012; Lian 2008).

3. **Advanced Techniques (2010s-2020s)**: Machine learning integration and robust multi-domain approaches (Ramanathan et al. 2020; Raja et al. 2008).
4. **Modern Innovations (2020s-Present)**: AI-enhanced security with lifting transforms and advanced optimization (Thasleen et al. 2024; Mohamed et al. 2025).

Recent advances include secure image encryption algorithm using chaos-based block permutation and weighted bit planes chain diffusion (Wen et al. 2024), robust Image Watermarking Using Lifting Wavelet Transform and Convolutional Neural Network (Thasleen et al. 2024), and enhanced brain image security using a hybrid of lifting wavelet transform (LWT) and SVM (Mohamed et al. 2025).

1.1 Overview

This chapter explores the application of wavelet transforms to modern data security challenges, covering encryption, steganography, digital watermarking, and secure biometric communication. Building on the mathematical foundations established in earlier volumes, it presents practical implementations that leverage wavelets' unique multi-resolution, localization, and energy compaction properties to protect sensitive information across multimedia and signal-based systems.

Wavelet transforms offer a powerful and versatile framework for data security by enabling selective, efficient, and robust protection of signals and images — going beyond traditional cryptographic methods to preserve both spatial and frequency information simultaneously, making them especially well-suited for modern multimedia security applications including encryption, steganography, watermarking, and biometric template protection.

Key Objectives:

- Understand how wavelet properties — multi-resolution analysis (MRA), energy compaction, and time-frequency localization — underpin security applications

- Implement **selective encryption** strategies that target critical wavelet coefficients while reducing computational overhead
- Apply **steganography techniques** (LSB, QIM, adaptive embedding) to hide information imperceptibly within signals
- Design **robust digital watermarking** systems using SVD-DWT hybrid and multi-domain approaches
- Develop **biometric template protection** schemes and evaluate their resistance to impostor attacks
- Build **secure communication systems** using spread-spectrum techniques in the wavelet domain
- Evaluate security systems across computational, perceptual, statistical, and robustness dimensions

1.2 Why Data Security?

In the digital era, the sheer volume of sensitive information being transmitted, stored, and shared across networks has made data security one of the most pressing technological challenges of our time. Traditional cryptographic methods alone are no longer sufficient to protect against increasingly sophisticated threats — demanding advanced, mathematically robust approaches that can operate across multimedia, biometric, and communication systems.

1. **Explosive Growth of Digital Data** — Vast amounts of sensitive information are continuously transmitted, stored, and shared, creating an ever-expanding attack surface for unauthorized access, manipulation, and theft.
2. **Limitations of Traditional Cryptography** — Conventional methods lack the ability to simultaneously preserve spatial and frequency information, making them inadequate for multimedia security where images, audio, and video must be protected without perceptible degradation.
3. **Rise of Multimedia and Biometric Systems** — Modern applications demand security solutions that work across images, signals, and biometric templates — requiring techniques like watermarking, steganography, and cancelable biometrics that go beyond simple encryption.

4. **Evolving and Sophisticated Attacks** — Security systems must now withstand statistical analysis attacks, frequency-domain attacks, geometric transformations, and even emerging quantum computing threats — demanding multi-layered, mathematically rigorous defenses.
5. **Need for Computational Efficiency** — Real-world systems require security that is not only strong but practical — selectively protecting the most critical data components without overwhelming processing resources, especially in real-time applications like video encryption and secure communication.

1.3 Problem Statement and Challenges

Securing digital information in an era of increasingly complex and voluminous data requires methods that are simultaneously robust, efficient, and adaptable across diverse media types. Traditional cryptographic approaches fall short when applied to multimedia systems, biometric data, and real-time communications — where protecting content must not compromise perceptual quality, computational speed, or signal integrity.

1.3.1 Core Challenges

1. **Perceptual Transparency** — Security operations such as watermarking and steganography must remain invisible or inaudible to human perception while still embedding recoverable, tamper-resistant information within signals and images.
2. **Robustness vs. Capacity Trade-off** — Increasing the amount of hidden or protected data typically weakens resistance to attacks such as compression, noise, and geometric transformations — requiring careful balancing of embedding strength and payload size.
3. **Computational Overhead** — Encrypting entire signals or images is often impractical in real-time systems; selective protection of critical wavelet coefficients must achieve strong security without excessive processing demands.

4. **Resistance to Diverse Attack Vectors** — Systems must defend against statistical analysis, brute-force cryptanalysis, frequency-domain attacks, and emerging quantum computing threats simultaneously.
5. **Biometric Template Vulnerability** — Unlike passwords, biometric data cannot be reissued once compromised; template protection schemes must provide both security and revocability without degrading authentication accuracy.

1.3.2 Research Questions

This chapter is guided by the following core research questions, each addressing a fundamental challenge in the application of wavelet transforms to data security:

1. How can wavelet decomposition properties — particularly energy compaction and multi-resolution analysis — be leveraged to design selective encryption schemes that balance computational efficiency with cryptographic strength?
2. To what extent can wavelet-domain steganographic techniques embed information imperceptibly within signals and images while maintaining robustness against common attacks such as compression, noise addition, and geometric transformations?
3. How do hybrid wavelet-based watermarking approaches — combining DWT with SVD or multi-domain processing — perform in terms of imperceptibility, robustness, and detection accuracy compared to single-domain methods?
4. What level of security and revocability can wavelet-based biometric template protection schemes — including cancelable biometrics and fuzzy vault methods — provide while preserving authentication accuracy across genuine and impostor samples?
5. How effectively do spread-spectrum wavelet communication systems maintain signal integrity and message recovery under realistic channel conditions, including AWGN, fading, and interference?

1.4 Methodology

1.4.1 Wavelet Transform Properties for Security

The wavelet transform possesses several properties that make it particularly suitable for security applications:

Multi-resolution Analysis: Wavelets decompose signals into multiple scales, allowing selective manipulation of different frequency components. This property enables partial encryption where only critical low-frequency information is encrypted, reducing computational overhead.

Localization: Unlike Fourier transforms, wavelets provide both time and frequency localization, making them resistant to localized attacks and suitable for adaptive security schemes.

Energy Compaction: Most signal energy concentrates in approximation coefficients, enabling efficient selective encryption strategies.

Orthogonality and Perfect Reconstruction: These properties ensure that security operations can be perfectly reversed with the correct keys.

1.4.2 Security Analysis Framework

For any wavelet-based security system, we must evaluate:

1. **Computational Security**: Resistance to brute-force and cryptanalytic attacks
2. **Perceptual Security**: Invisibility of hidden information to human perception
3. **Statistical Security**: Resistance to statistical analysis attacks
4. **Robustness**: Survival of security features under signal processing operations

1.4.3 Mathematical Foundation

The discrete wavelet transform can be expressed as:

$$W(j,k) = \sum_{n} x(n)\,\psi_{j,k}(n) \tag{1.1}$$

where:

- $x(n)$ — input signal
- $\psi_{j,k}(n)$ — wavelet basis at scale j and position k
- $W(j,k)$ — wavelet coefficient
- $\sum$ — summation over all samples n

For security applications, wavelet coefficients can be manipulated to embed or protect information while preserving the overall signal structure. Typical approaches include:

- **Coefficient modification**

$$W'(j,k) = f(W(j,k), \text{key}) \tag{1.2}$$

- **Selective processing**
  Apply security operations only to specific wavelet subbands.
- **Multi-level processing** Use different security keys for different decomposition levels.

Where:

- $W(j,k)$ — original wavelet coefficient at scale j and position k
- $W'(j,k)$ — modified or protected coefficient after processing
- $f(\cdot)$ — transformation or encryption function
- key — secret parameter controlling the security operation

- j — decomposition level (scale)
- k — coefficient index within the subband

1.5 Dataset

1.5.1 Dataset Description

This chapter does not rely on a single external dataset but instead employs synthetically generated signals, images, and biometric data to demonstrate wavelet-based security techniques in a controlled and reproducible manner. All datasets are programmatically constructed within the Python computational environment, ensuring consistency across platforms and eliminating external data dependencies.

1.5.2 Data Types Used

1. **1D Signals** — Synthetically generated time-series signals used to demonstrate selective wavelet encryption and spread-spectrum secure communication techniques.
2. **2D Images** — Grayscale and color images constructed or loaded programmatically to illustrate steganographic embedding, digital watermarking, and SVD-DWT hybrid security schemes.
3. **Wavelet Coefficient Arrays** — Multi-level decomposition outputs produced via PyWavelets (`pywt`) across Haar, Daubechies, and Biorthogonal wavelet families, serving as the primary data structure for all security operations.
4. **Biometric Templates** — Simulated biometric feature vectors representing genuine and impostor samples, used to evaluate template encryption, cancelable biometrics, and fuzzy vault protection schemes.
5. **Communication Channel Data** — Synthetically modeled signal transmissions under AWGN, fading, and interference channel conditions, used to assess robustness of wavelet-based secure communication systems.

1.5.3 Data Generation and Reproducibility

All synthetic data is generated using NumPy and SciPy with fixed random seeds where applicable, ensuring that experimental results are fully reproducible across different computing environments. No proprietary or sensitive real-world data is required to execute the code examples presented in this chapter.

1.6 Wavelet-Based Encryption

1.6.1 Fundamentals of Wavelet Encryption

Wavelet-based encryption leverages the energy compaction property of wavelets, where most signal energy concentrates in approximation coefficients. This allows for selective encryption strategies that reduce computational cost while maintaining security.

Key Advantages:

- **Selective encryption**: Encrypt only critical components
- **Scalable security**: Different security levels for different applications
- **Format compliance**: Maintain file format compatibility
- **Computational efficiency**: Reduced encryption overhead

1.6.2 Encryption Algorithms

1. Selective Encryption

This implementation demonstrates selective encryption of signals using wavelet decomposition, where critical approximation coefficients are encrypted using standard cryptographic methods while detail coefficients are scrambled using simple permutations. The approach balances security with computational efficiency by focusing protection on the most perceptually important components.

```
import numpy as np
import pywt
from cryptography.fernet import Fernet
import matplotlib.pyplot as plt
from scipy import ndimage

class WaveletEncryption:
    def __init__(self, wavelet='haar', levels=3):
        self.wavelet = wavelet
        self.levels = levels
        self.key = Fernet.generate_key()
        self.cipher = Fernet(self.key)

    def encrypt_signal(self, signal):
        """Selective encryption of 1D signal using wavelet
        transform"""
        # Decompose signal
        coeffs = pywt.wavedec(signal, self.wavelet, level=self.levels)

        # Encrypt approximation coefficients (most critical)
        approx_bytes = coeffs[0].tobytes()
        encrypted_approx = self.cipher.encrypt(approx_bytes)

        # Scramble detail coefficients with simple permutation
        encrypted_coeffs = [encrypted_approx]
        for i, detail in enumerate(coeffs[1:], 1):
            # Use level-dependent scrambling
            scrambled = np.roll(detail, i * 3)
            encrypted_coeffs.append(scrambled)

        return encrypted_coeffs

    def decrypt_signal(self, encrypted_coeffs):
        """Decrypt and reconstruct signal"""
        # Decrypt approximation coefficients
        decrypted_approx_bytes = self.cipher.decrypt(
            encrypted_coeffs[0]
        )
        approx_coeffs = np.frombuffer(decrypted_approx_bytes,
            dtype=float)
```

```
        # Unscramble detail coefficients
        decrypted_coeffs = [approx_coeffs]
        for i, scrambled_detail in enumerate(encrypted_coeffs[1:], 1):
            unscrambled = np.roll(scrambled_detail, -i * 3)
            decrypted_coeffs.append(unscrambled)

        # Reconstruct signal
        return pywt.waverec(decrypted_coeffs, self.wavelet)

    def visualize_encryption_effect(self, signal):
        """Visualize the encryption effect"""
        encrypted_coeffs = self.encrypt_signal(signal)

        # For visualization, create a "partially encrypted" signal
        # by reconstructing with scrambled coefficients
        original_coeffs = pywt.wavedec(signal, self.wavelet,
            level=self.levels)
        # random approximation for visualization
        viz_coeffs = [np.random.random(len(original_coeffs[0]))]
        viz_coeffs.extend(encrypted_coeffs[1:])

        try:
            encrypted_signal = pywt.waverec(viz_coeffs, self.wavelet)
        except:
            encrypted_signal = np.random.random(len(signal))

        plt.figure(figsize=(12, 8))

        plt.subplot(3, 2, 1)
        plt.plot(signal)
        plt.title('Original Signal')
        plt.grid(True)

        plt.subplot(3, 2, 2)
        plt.plot(encrypted_signal)
        plt.title('Encrypted Signal')
        plt.grid(True)

        # Show wavelet decomposition
```

```
        original_coeffs = pywt.wavedec(signal, self.wavelet,
            level=self.levels)

        plt.subplot(3, 2, 3)
        plt.plot(original_coeffs[0])
        plt.title('Original Approximation Coefficients')
        plt.grid(True)

        plt.subplot(3, 2, 4)
        plt.plot(np.random.random(len(original_coeffs[0])))
        plt.title('Encrypted Approximation (Random for Display)')
        plt.grid(True)

        plt.subplot(3, 2, 5)
        if len(original_coeffs) > 1:
            plt.plot(original_coeffs[1])
            plt.title('Original Detail Coefficients (Level 1)')
        plt.grid(True)

        plt.subplot(3, 2, 6)
        if len(encrypted_coeffs) > 1:
            plt.plot(encrypted_coeffs[1])
            plt.title('Scrambled Detail Coefficients (Level 1)')
        plt.grid(True)

        plt.tight_layout()
        plt.savefig('./output/wavelet_encryption_effect.png')
        plt.show()

# Example usage
if __name__ == "__main__":
    # Generate test signal
    t = np.linspace(0, 2*np.pi, 256)
    signal = (
        np.sin(t)
        + 0.5*np.sin(3*t)
        + 0.25*np.sin(5*t)
        + 0.1*np.random.randn(256)
    )
```

```
    # Create encryptor
    encryptor = WaveletEncryption(wavelet='db4', levels=3)

    # Encrypt and decrypt
    encrypted_coeffs = encryptor.encrypt_signal(signal)
    decrypted_signal = encryptor.decrypt_signal(encrypted_coeffs)

    # Visualize results
    encryptor.visualize_encryption_effect(signal)

    # Verify perfect reconstruction
    reconstruction_error = (signal - decrypted_signal)**2
    mse = np.mean(reconstruction_error)
    max_error = np.max(np.abs(signal - decrypted_signal))
    print(f"Max Reconstruction Error: {max_error:.2e}")
    print(f"Mean Reconstruction Error (MSE): {mse:.2e}")

    # Plot Original, Decrypted, and Reconstruction Error
    plt.figure(figsize=(18, 6))

    plt.subplot(1, 2, 1)
    plt.plot(signal, label='Original')
    plt.plot(decrypted_signal, '--', label='Decrypted')
    plt.title('Original vs Decrypted')
    plt.legend()
    plt.grid(True)

    plt.subplot(1, 2, 2)
    plt.plot(reconstruction_error)
    plt.title('Reconstruction Error')
    plt.grid(True)

    plt.tight_layout()
    plt.savefig('./output/reconstruction_error.png')
    plt.show()
```

The reconstructed evaluation results:

```
Max Reconstruction Error: 8.88e-16
```

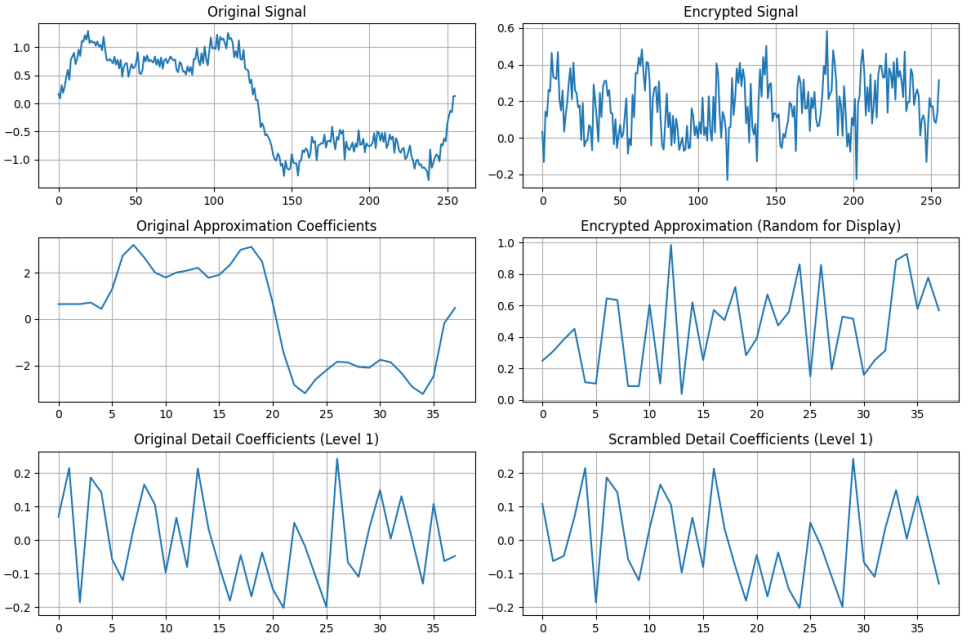


Figure 1.1: Visualization of the selective wavelet-based encryption process. The original signal is compared with the partially encrypted signal reconstructed from scrambled coefficients, alongside the corresponding approximation and detail coefficients before and after encryption. This demonstrates how encryption impacts different levels of wavelet decomposition.

```
Mean Reconstruction Error (MSE): 6.55e-32
```

The experimental results show that the proposed wavelet-based encryption and decryption scheme achieves almost lossless reconstruction. With a maximum reconstruction error of only 8.88×10^{-16} and a mean squared error (MSE) as low as 6.55×10^{-32}, the decrypted signal is virtually identical to the original.

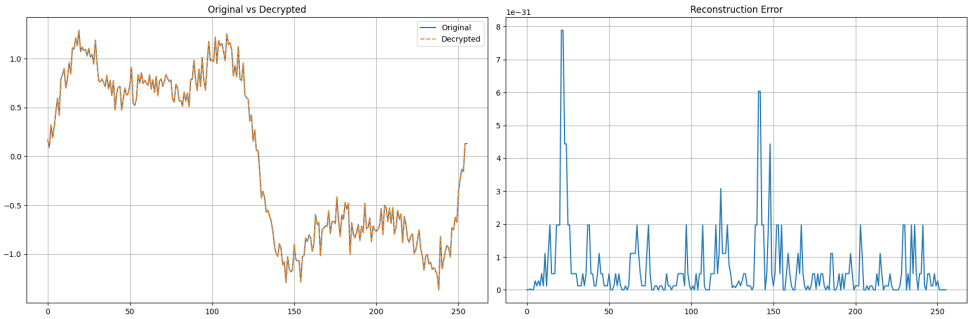


Figure 1.2: Comparison between the original and decrypted signals after full encryption–decryption. The overlay shows near-perfect reconstruction, while the error plots confirm that the residual distortion is negligible, validating the effectiveness of the proposed method.

2. Chaos-Based Wavelet Encryption

This implementation combines chaotic systems with wavelet transforms for image encryption. The Lorenz chaotic system generates pseudo-random sequences that are used to encrypt different wavelet subbands through XOR operations, providing enhanced security through the unpredictable nature of chaotic dynamics.

```
import numpy as np
import pywt
from scipy.integrate import odeint
import matplotlib.pyplot as plt
from skimage.metrics import structural_similarity as ssim
from skimage.io import imread
from skimage.color import rgb2gray

class ChaoticWaveletEncryption:
```

```
    plt.subplot(1,3,1); plt.imshow(
        img_c[zy:zy+patch, zx:zx+patch], cmap='gray');
    plt.title('Orig patch'); plt.axis('off')
    plt.subplot(1,3,2); plt.imshow(dec_c[zy:zy+patch, zx:zx+patch],
        cmap='gray');
    plt.title('Dec patch'); plt.axis('off')
    plt.subplot(1,3,3);
    patch_error = img_c[zy:zy+patch, zx:zx+patch] - dec_c[zy:zy+patch,
↪
        zx:zx+patch]
    plt.imshow(patch_error, cmap='seismic')
    plt.title('Patch error')
    plt.axis('off')
    plt.tight_layout()
    plt.savefig('./output/astronaut_patch_zoom.png')
    plt.show()
```

Output:

```
Decryption MSE: 5.16e-10
Decryption PSNR: 92.88 dB
Decryption SSIM: 0.9858
Correlation: 0.9998083584110499
```

The proposed chaotic wavelet encryption scheme achieves nearly lossless reconstruction. The decrypted image shows an extremely low MSE of 5.16×10^{-10}, a very high PSNR of 92.88 dB, and an SSIM of 0.986, indicating strong structural preservation. The correlation coefficient of 0.9998 further confirms that the decrypted image is almost identical to the original. The visulization results are as follows:

1.6.3 Security Analysis and Performance Evaluation

This comprehensive security analysis framework evaluates encryption schemes using multiple cryptographic metrics including entropy analysis, correlation analysis, histogram uniformity, and randomness tests. The implementation provides both quantitative security measures and visual analysis tools.

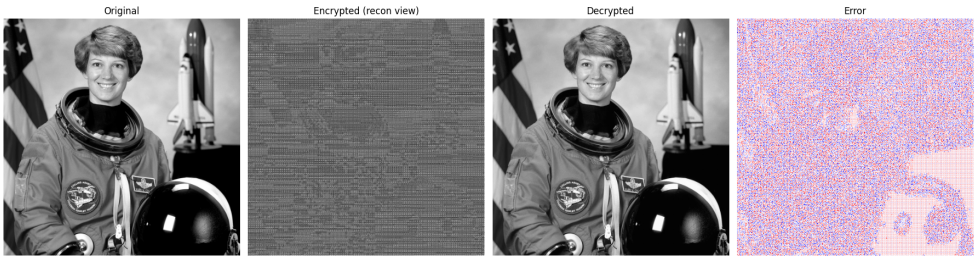


Figure 1.3: Chaos-based wavelet encryption demonstration of a grayscale real-world astronaut image. From left to right: (a) original test image, (b) encrypted image with chaotic scrambling applied to all wavelet subbands, (c) perfectly reconstructed decrypted image, and (d) Reconstruction error.

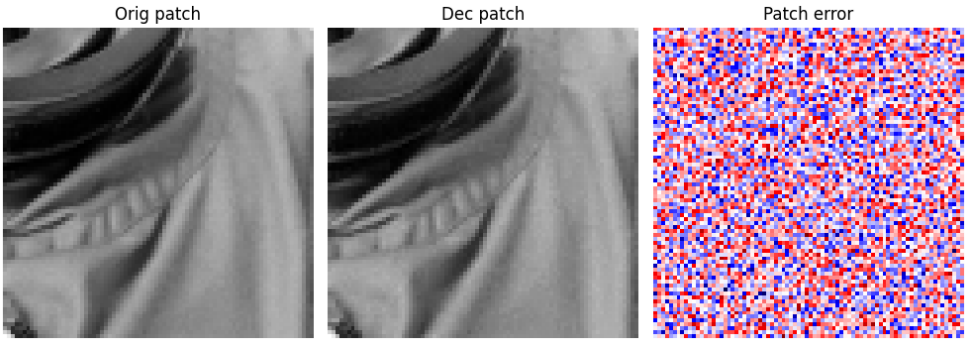


Figure 1.4: Zoomed-in comparison of the wavelet–chaos decryption. From left to right: (a) Original image patch, (b) Decrypted patch, (c) Patch-wise error map.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import entropy
import pywt
from imageio.v2 import imread


# ----------------------------
# Improved Chaotic Wavelet Encryption with CSPRNG Diffusion
# ----------------------------
class ImprovedChaoticWaveletEncryption:
    def __init__(self, key=0.12345):
        self.key = key  # Key for PRNG

    def _logistic_map(self, size, r=3.99):
        """Chaotic sequence for scrambling wavelet coefficients
        (small-scale)"""
        x = np.zeros(size)
        x[0] = self.key
        for i in range(1, size):
            x[i] = r * x[i-1] * (1 - x[i-1])
        return x

    def encrypt_image(self, image):
        """Encrypt image using wavelet + CSPRNG bit-level diffusion"""
        # 1. Wavelet decomposition
        coeffs = pywt.dwt2(image, 'haar')
        cA, (cH, cV, cD) = coeffs

        # 2. Chaotic scrambling of detail coefficients (still using
        # logistic map)
        shape = cH.shape
        size_detail = shape[0] * shape[1]
        chaos_seq_detail = self._logistic_map(
            size_detail).reshape(shape)

        cH_enc = np.mod(cH + chaos_seq_detail, 1)
        cV_enc = np.mod(cV + chaos_seq_detail, 1)
        cD_enc = np.mod(cD + chaos_seq_detail, 1)
```

5 Multiscale Anomaly Detection Using Wavelets

Anomaly detection is a fundamental task in modern data-driven systems that monitor temporal signals. Applications arise in a wide range of domains, including transportation demand monitoring, financial markets, industrial sensor networks, environmental observation systems, and large-scale digital services. In such systems, unusual patterns may indicate unexpected events, system failures, cyber-attacks, or rare external disturbances that require timely detection and interpretation (Chandola, Banerjee, and Kumar 2009; Aggarwal 2017)

Real-world time series typically contain multiple temporal structures simultaneously. A signal may exhibit long-term trends, periodic cycles, seasonal variations, and sudden transient events. Traditional anomaly detection techniques often operate at a single temporal scale and therefore may struggle when abnormal behavior occurs within overlapping patterns or when local deviations are masked by global variability (Laptev, Amizadeh, and Flint 2015).

Wavelet transforms provide a powerful framework for analyzing such signals because they decompose time series into multiple temporal scales while preserving time localization. This multiresolution capability allows abnormal behavior to be identified in different frequency bands independently, enabling analysts to capture both global trends and localized disturbances within the same representation (Stéphane Mallat 2009; Percival and Walden 2000). Wavelet-based techniques have been widely applied to anomaly detection in engineering systems, signal processing, and time-series monitoring because

```
    cmap="hot",
    interpolation="bilinear",
)
axes[1].set_ylabel("Scale")
axes[1].set_xlabel("Sample Index")
axes[1].set_title("CWT Scalogram (Morlet)", fontsize=12)
plt.colorbar(im, ax=axes[1], label="|CWT Coefficient|")

plt.tight_layout()
plt.savefig("cwt_scalogram.png", bbox_inches="tight")
plt.show()
```

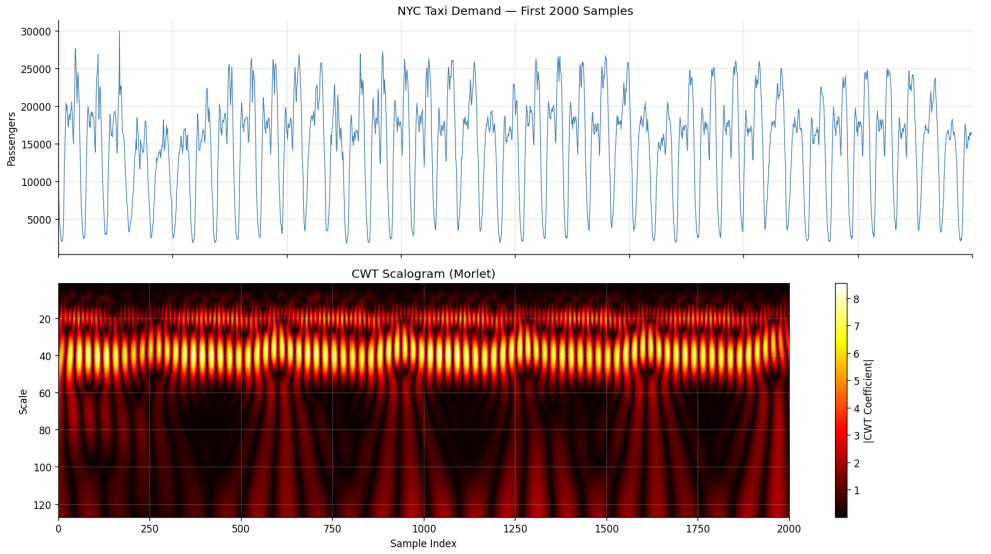


Figure 5.10: Continuous wavelet transform (CWT) scalogram of the NYC taxi demand signal. The top panel shows the first 2000 samples of the taxi demand time series. The bottom panel displays the CWT scalogram computed using the Morlet wavelet, where color intensity represents the magnitude of the wavelet coefficients across time and scale. This time–frequency representation reveals how oscillatory patterns and transient events evolve across multiple temporal scales in the signal.

5.14.1 Interpretation of the CWT Scalogram

The scalogram reveals which temporal scales carry the most energy and when these patterns occur in time. Bright regions at **finer scales (low scale values)** correspond to **short-duration, high-frequency events**, while broader bright regions at **coarser scales (high scale values) represent slower structural changes or long-term variations in the signal**.

The scalogram (Figure 5.10) reveals strong **periodic oscillations in the NYC taxi demand signal**, visible as repeating high-energy bands across time. The most prominent energy concentration occurs around **mid-level scales (approximately 20–50)**, indicating that the dominant dynamics of the signal occur at these temporal frequencies, which likely correspond to the **daily demand cycle**.

At **smaller scales (top of the scalogram)**, the energy appears more fragmented, reflecting short-term fluctuations and local variations in demand. In contrast, **larger scales (bottom region)** show smoother and broader patterns, representing slower, longer-term variations in the signal.

Overall, the CWT visualization confirms that the taxi demand time series contains a **stable multiscale structure dominated by regular periodic behavior**, with occasional localized bursts of energy that may correspond to transient demand events or irregular activity.

5.15 Summary and Discussion

This chapter demonstrated a complete workflow for wavelet-based multiscale anomaly detection applied to a real-world transportation demand dataset.

5.15.1 Key Findings

1. **Multiscale structure matters.** The NYC Taxi dataset exhibits anomalies at different temporal scales — sharp spikes at fine scales (New Year's Eve, Blizzard onset) and broader demand shifts at coarser

scales (Thanksgiving, Christmas). A single-scale detector misses some of these.

2. **Wavelet energy is a sensitive anomaly indicator.** Sliding-window wavelet energy effectively highlights periods of unusual signal behavior, with scores that correlate well with known event windows.
3. **Method comparison reveals trade-offs.** Z-score and moving-average detectors are computationally simple but rely on global or locally stationary statistics. Isolation Forest adds feature-based context. Wavelet methods capture non-stationary, scale-dependent behavior that the others may miss.
4. **Denoising and detection can be combined.** Soft thresholding removes high-frequency noise before analysis, reducing false positives without sacrificing sensitivity to genuine anomalies.
5. **CWT scalograms complement DWT scores.** The continuous scalogram provides an interpretable visualization of time-frequency structure, useful for exploratory analysis and communicating results to non-specialists.

5.15.2 Limitations and Future Work

- The sliding-window approach introduces a latency of approximately half the window size; real-time systems require causal (one-sided) windows.
- The universal threshold used in denoising may be too aggressive for signals with a low signal-to-noise ratio.
- Labeled anomaly data for the NYC Taxi dataset is sparse; more comprehensive evaluation would require a dataset with denser ground-truth labels.
- Future work could explore adaptive thresholding, wavelet packet decomposition, or deep learning architectures (e.g., wavelet-regularized autoencoders) for improved performance.

5.16 Conclusions

Wavelet-based multiscale anomaly detection provides a flexible and interpretable framework for monitoring real-world temporal systems. By decomposing a signal into multiple frequency bands, practitioners can localize unusual behavior in both time and scale — a capability that single-scale statistical methods lack.

The NYC Taxi demand dataset illustrates how wavelet transforms separate trend, seasonality, and transient spikes, and how energy metrics derived from detail coefficients effectively score anomalous windows. Comparison with z-score, moving-average, and Isolation Forest detectors demonstrates the complementary strengths of each approach, highlighting scenarios where wavelet methods provide a meaningful advantage.

5.17 Exercises and Projects

5.17.1 Exercises

1. Apply wavelet decomposition to the NYC Taxi dataset using `sym5` and `coif3`. Compare the resulting energy distributions with those obtained using `db4`. Which wavelet family best localizes the Thanksgiving anomaly?
2. Implement a **causal sliding-window** energy score that uses only past observations (no future samples). How does the detection delay change compared to the centered window?
3. Experiment with different anomaly thresholds ($\tau = 2.0, 2.5, 3.0, 3.5$). Plot precision and recall against the threshold and identify the optimal operating point.
4. Apply hard thresholding instead of soft thresholding in the denoising step. Compare the quality of the denoised signal and the effect on downstream anomaly detection.
5. Extend the Isolation Forest detector by adding wavelet energy features at each decomposition level to the feature matrix. Does performance improve?

5.17.2 Projects

1. Multiscale Transportation Demand Monitoring

Objective: Build a complete anomaly monitoring pipeline for the NYC Taxi dataset.

Tasks:

- Load and preprocess the NYC Taxi demand data
- Perform DWT decomposition at levels 1–6 and compute per-scale energy profiles
- Implement the sliding-window anomaly scorer and tune the threshold using the known event windows as soft labels
- Visualize detected anomalies alongside known events
- Write a short report summarizing the performance and scale at which each event is most visible

Expected Outcomes: End-to-end anomaly detection pipeline with quantitative evaluation.

Deliverables: Annotated Python notebook and anomaly analysis report.

2. Comparative Anomaly Detection Study

Objective: Rigorously compare wavelet, statistical, and machine learning detectors.

Tasks:

- Implement all four detectors (Wavelet, Z-score, Moving Average, Isolation Forest)
- Tune each detector's hyperparameters via cross-validation on a held-out portion of the dataset
- Compute AUC-ROC for all methods using the known event labels
- Analyze false positives: what do the non-event detections correspond to?
- Investigate combining detectors (e.g., voting ensemble) to reduce false positives

Expected Outcomes: Quantitative comparison table and ensemble detector.

Deliverables: Python notebook with ROC curves, performance table, and discussion.

3. Real-Time Streaming Anomaly Detector

Objective: Adapt the wavelet anomaly detector to an online, streaming setting.

Tasks:

- Implement a sliding-window DWT that processes one new observation at a time
- Maintain a buffer of recent wavelet coefficients and update the anomaly score incrementally
- Benchmark detection latency and compare causal vs. centered windows
- Simulate a streaming feed using the NYC Taxi dataset and visualize real-time detection

Expected Outcomes: A functional real-time detector with latency analysis.

Deliverables: Python module with streaming interface and performance benchmarks.

References

Abdelwahab, Ahmed A., and Lobna A. Hassaan. 2008. "A Discrete Wavelet Transform Based Technique for Image Data Hiding." *National Radio Science Conference, NRSC, Proceedings*. https://doi.org/10.1109/NRSC.2008.4542319.

Abry, P., and D. Veitch. 1998. "Wavelet Analysis of Long-Range-Dependent Traffic." *IEEE Transactions on Information Theory* 44 (1): 2–15. https://doi.org/10.1109/18.650984.

Acharya, U. Rajendra, Hamido Fujita, Oh Shu Lih, Muhammad Adam, Jen Hong Tan, and Chua Kuang Chua. 2017. "Automated Detection of Coronary Artery Disease Using Different Durations of ECG Segments with Convolutional Neural Network." *Knowledge-Based Systems* 132 (September): 62–71. https://doi.org/10.1016/J.KNOSYS.2017.06.003.

Addison, Paul S. 2017. *The Illustrated Wavelet Transform Handbook: Introductory Theory and Applications in Science, Engineering, Medicine and Finance*. CRC press.

Aggarwal, Charu C. 2017. *Outlier Analysis*. Springer. https://doi.org/10.1007/978-3-319-47578-3.

Antoni, Jérôme. 2006. "The Spectral Kurtosis: A Useful Tool for Characterising Non-Stationary Signals." *Mechanical Systems and Signal Processing* 20 (2): 282–307. https://doi.org/https://doi.org/10.1016/j.ymssp.2004.09.001.

Bahoura, Mohammed, and Jean Rouat. 2001. "Wavelet Speech Enhancement Based on the Teager Energy Operator." *IEEE Signal Processing Letters* 8: 10–12. https://doi.org/10.1109/97.889636.

Barni, Mauro, Franco Bartolini, and Alessandro Piva. 2001. "Improved Wavelet-Based Watermarking Through Pixel-Wise Masking." *IEEE Transactions on Image Processing* 10 (May): 783–91. https://doi.org/10.1109/83.918570.

Boll, Steven F. 1979. "Suppression of Acoustic Noise in Speech Using Spectral Subtraction." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 27: 113–20. https://doi.org/10.1109/TASSP.1979.1163209.

Chandola, Varun, Arindam Banerjee, and Vipin Kumar. 2009. "Anomaly Detection: A Survey." *ACM Comput. Surv.* 41 (3). https://doi.org/10.1145/1541880.1541882.

Chang, S. G., Bin Yu, and M. Vetterli. 2000. "Adaptive Wavelet Thresholding for Image Denoising and Compression." *IEEE Transactions on Image Processing* 9 (9): 1532–46. https://doi.org/10.1109/83.862633.

Cox, I. J.. 2008. "Digital Watermarking and Steganography," 593.

Daubechies, Ingrid. 1992. *Ten Lectures on Wavelets*. Society for Industrial. https://doi.org/10.1137/1.9781611970104.

Donoho, David L., and Jain M. Johnstone. 1994a. "Ideal Spatial Adaptation by Wavelet Shrinkage." *Biometrika* 81 (September): 425–55. https://doi.org/10.1093/BIOMET/81.3.425.

Donoho, David L, and Iain M Johnstone. 1994b. "Ideal Spatial Adaptation by Wavelet Shrinkage." *Biometrika* 81 (3): 425–55. https://doi.org/10.1093/biomet/81.3.425.

Gençay, Ramazan, Ramazan Gençay, Faruk Selçuk, and Brandon J. Whitcher. 2001. "An Introduction to Wavelets and Other Filtering Methods in Finance and Economics." *Elsevier Monographs*. https://ideas.repec.org/b/eee/monogr/9780122796708.html https://ideas.repec.org//b/eee/monogr/9780122796708.html.

Grinsted, A., J. C. Moore, and S. Jevrejeva. 2004. "Application of the Cross Wavelet Transform and Wavelet Coherence to Geophysical Time Series." *Nonlinear Processes in Geophysics* 11 (November): 561–66. https://doi.org/10.5194/NPG-11-561-2004.

Kamath, Sunil, and Philipos Loizou. 2002. "A Multi-Band Spectral Subtraction Method for Enhancing Speech Corrupted by Colored Noise." *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 4: 4164. https://doi.org/10.1109/ICASSP.2002.5745591.

Kundur, Deepa, and Dimitrios Hatzinakos. 1998. "Digital Watermarking Using Multiresolution Wavelet Decomposition." *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 5:

2969–72. https://doi.org/10.1109/ICASSP.1998.678149.

Laptev, Nikolay, Saeed Amizadeh, and Ian Flint. 2015. "Generic and Scalable Framework for Automated Time-Series Anomaly Detection." In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1939–47.

Lavin, Alexander, and Subutai Ahmad. 2015. "Evaluating Real-Time Anomaly Detection Algorithms – the Numenta Anomaly Benchmark." In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 38–44. https://doi.org/10.1109/ICMLA.2015.141.

Lian, Shiguo. 2008. *Multimedia Content Encryption: Techniques and Applications*. Auerbach Publications.

Loizou, Philipos C. 2013. "SPEECH ENHANCEMENT: Theory and Practice, Second Edition." *Speech Enhancement: Theory and Practice, Second Edition*, January, 1–677. https://doi.org/10.1201/B14529/SPEECH-ENHANCEMENT-PHILIPOS-LOIZOU/RIGHTS-AND-PERMISSIONS.

Mallat, S. 1989. "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11 (7): 674–93. https://doi.org/10.1109/34.192463.

Mallat, Stéphane. 1999. "A Wavelet Tour of Signal Processing," 637.

———. 2009. *A Wavelet Tour of Signal Processing (3rd Edition)*. Elsevier. https://doi.org/10.1016/B978-0-12-374370-1.X0001-8.

Mandelbrot, Benoit B. 1997. "The Variation of Certain Speculative Prices." In *Fractals and Scaling in Finance: Discontinuity, Concentration, Risk. Selecta Volume e*, 371–418. New York, NY: Springer New York. https://doi.org/10.1007/978-1-4757-2763-0_14.

Mohamed, Asmaa Fathallah, Ahmed S. Samra, Bedir Yousif, and Abeer Tawkol Khalil. 2025. "Enhanced Brain Image Security Using a Hybrid of Lifting Wavelet Transform and Support Vector Machine." *Scientific Reports* 15 (December): 9570. https://doi.org/10.1038/S41598-025-92580-X.

Percival, Donald B., and Andrew T. Walden. 2000. "Wavelet Methods for Time Series Analysis." *Wavelet Methods for Time SeriesAnalysis*. https://doi.org/10.1017/CBO9780511841040.

Piczak, Karol J. 2015. "ESC: Dataset for Environmental Sound Classification." *MM 2015 - Proceedings of the 2015 ACM Multimedia Conference*, October, 1015–18. https://doi.org/10.1145/2733373.2806390.

Piva, A., M. Barni, F. Bartolini, and V. Cappellini. 1997. "DCT-Based Watermark Recovering Without Resorting to the Uncorrupted Original Image." *IEEE International Conference on Image Processing* 1: 520–23. https://doi.org/10.1109/ICIP.1997.647964.

Raasveldt, Mark, and Hannes Mühleisen. 2019. "DuckDB: An Embeddable Analytical Database." *Proceedings of the ACM SIGMOD International Conference on Management of Data*, June, 1981–84. https://doi.org/10.1145/3299869.3320212.

Raja, K. B., S. Sindhu, T. D. Mahalakshmi, S. Akshatha, B. K. Nithin, M. Sarvajith, K. R. Venugopal, and L. M. Patnaik. 2008. "Robust Image Adaptive Steganography Using Integer Wavelets." *ICST International Conference on Communication System Software and Middleware*, 614–21. https://doi.org/10.1109/COMSWA.2008.4554484.

Ramanathan, Thirumalaimuthu T., J. Hossen, S. Sayeed, and J. Emerson Raja. 2020. "Survey on Computational Intelligence Based Image Encryption Techniques." *Indonesian Journal of Electrical Engineering and Computer Science* 19 (September): 1428–35. https://doi.org/10.11591/IJEECS.V19.I3.PP1428-1435.

Strang, Gilbert, and Truong Nguyen. 1996. *Wavelets and Filter Banks*. SIAM.

Subhedar, Mansi S., and Vijay H. Mankar. 2014. "Current Status and Key Issues in Image Steganography: A Survey." *Computer Science Review* 13-14 (November): 95–113. https://doi.org/10.1016/J.COSREV.2014.09.001.

Subramanyam, A. V., Sabu Emmanuel, and Mohan S. Kankanhalli. 2012. "Robust Watermarking of Compressed and Encrypted JPEG2000 Images." *IEEE Transactions on Multimedia* 14: 703–16. https://doi.org/10.1109/TMM.2011.2181342.

Thanki, Rohit, and Surekha Borra. 2018. "A Color Image Steganography in Hybrid FRT–DWT Domain." *Journal of Information Security and Applications* 40 (June): 92–102. https://doi.org/10.1016/J.JISA.2018.03.004.

Thasleen, J. Fathima, P. L. Lekshmy, S. Srinivas, and D. Jayakumari. 2024. "Robust Image Watermarking Using Lifting Wavelet Transform and Convolutional Neural Network." *International Journal of Intelligent Systems and Applications in Engineering* 12 (March): 01–10. https://www.ijisae.org/index.php/IJISAE/article/view/5217.

Vetterli, Martin, and Jelena Kovačević. 1995. "Wavelets and Subband Coding." *Book*, 1–519.

Wen, Heping, Yiting Lin, Shenghao Kang, Xiangyu Zhang, and Kun Zou. 2024. "Secure Image Encryption Algorithm Using Chaos-Based Block Permutation and Weighted Bit Planes Chain Diffusion." *iScience* 27 (January): 108610. https://doi.org/10.1016/J.ISCI.2023.108610.

Wiener, Norbert. 1949. "Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications." *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, August. https://doi.org/10.7551/MITPRESS/2946.001.0001.

Xia, Xiang-Gen, Charles G Boncelet, and Gonzalo R Arce. 1998. "Wavelet Transform Based Watermark for Digital Images." *Optics Express* 3 (12): 497–511.

Yousif, Adel Jalal. 2020. "Image Steganography Based on Wavelet Transform and Color Space Approach." *Diyala Journal of Engineering Sciences* 13 (September): 23–34. https://doi.org/10.24237/DJES.2020.13303.

Index

T

U

V

Z

VOLUME III-B

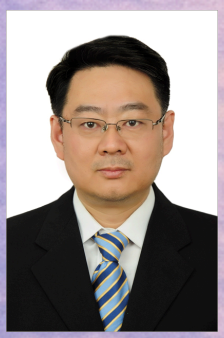
Wavelet Transform in Practice, Volume III-B explores how wavelet-based-analysis operates within financial, information, and digital signal systems.

Building on the foundations established in Volumes I and II, this volume examines multiscale behavior in complex data generated by markets, communication systems and digital media.

Topics include:

- Wavelet-based data security and steganography
- Multiscale analysis of financial time series
- Audio signal analysis
- Wavelet-based anomaly detection

About the Author:



Shouke Wei earned his Ph.D, from Brandenburg University of Technology Cottbus–Senftenberg (Cermany) and conducted postdoctoral research at Eawag (Switzerfand) He has held research positions at the University of British Columbia (Canada) and served as a distinguished and adjunct professor at multiple universities (China).

His work focuses an reproducible, deployable wavelet-based methods for analytics and signal processing.

https://press.deepsim.ca/

9 781069 928498