AGH UNIVERSITY OF KRAKOW

FIELD OF SCIENCE ENGINEERING AND TECHNOLOGY

SCIENTIFIC DISCIPLINE AUTOMATION, ELECTRONIC, ELECTRICAL ENGINEERING AND SPACE TECHNOLOGIES

DOCTORAL THESIS

Edge Implementation of an Adaptive Wavelet Transform for Improved Smart Grid Sensor Data Compression

Author: Kamil Prokop

First supervisor: Professor Andrzej Bień
Assisting supervisor: Dr Szymon Barczentewicz

Completed in: AGH University of Krakow, Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering

Kraków, 2025

This thesis could not have been created without the invaluable and constant support and patience of my beloved wife Kamila.

I would also like to thank our son Teodor and our unborn child for being an endless source of motivation.

I want to express my gratitude to my parents and sisters for their support.

I would like to thank my supervisor - professor Andrzej Bień, for being a mentor and supporting me with his experience.

I would also like to thank my co-supervisor - doctor Szymon Barczentewicz for sharing his knowledge and advice.

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

DZIEDZINA NAUK INŻYNIERYJNO-TECHNICZNYCH

DYSCYPLINA AUTOMATYKA, ELEKTRONIKA, ELEKTROTECHNIKA I TECHNOLOGIE KOSMICZNE

ROZPRAWA DOKTORSKA

Implementacja brzegowa adaptacyjnej transformaty falkowej w celu efektywniejszej kompresji danych z czujników w Smart Grid

Autor: Kamil Prokop

Promotor rozprawy: dr hab. inż. Andrzej Bień prof. AGH
Promotor pomocniczy: dr inż. Szymon Barczentewicz

Praca wykonana: Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

Kraków, 2025

Ta praca nie mogłaby powstać bez nieocenionego i ciągłego wsparcia oraz cierpliwości mojej ukochanej żony Kamili.

Chciałbym podziękować również naszemu synowi Teodorowi oraz naszemu jeszcze nienarodzonemu dziecku za bycie nieskończonym źródłem motywacji.

Chciałbym wyrazić wdzięczność dla moich rodziców oraz sióstr za ich wsparcie.

Dziękuję mojemu promotorowi - profesorowi Andrzejowi Bienowi, za bycie mentorem oraz wspieranie mnie swoim doświadczeniem.

Dziękuję również mojemu promotorowi pomocniczemu doktorowi Szymonowi Barczentewiczowi za podzielenie się swoją wiedzą oraz radami.

Abstract

This thesis addresses the problem of extensive use of data storage and transmission by measurement systems in Internet of Things devices used in Smart Grids. The increased demand for efficient and effective operation of modern power grids requires precise control. The quality of control strategies depends on the quality and quantity of the input data. The cost and complexity of data acquisition systems in power grids and other systems is directly correlated with the infrastructure utilized, such as communication networks and memory. The improvement of the control system in the power grid can be achieved by increasing the efficiency of data handling processes. The way to optimize the use of storage and transmission is data compression. The research presented in this thesis focuses on the creation of a new way of Discrete Wavelet Transform parameterization that will improve present algorithms.

This work outlines the most important problems with the handling of measurement data in power grids. The signals observed in power grids are described in detail. The most important qualities of the Smart Grids signals are pointed out, along with the difference between power transmission signals and different domains, like communication or audio. The roots of the problem laying in the communication theory are also pinpointed. The most popular communication technologies used in Smart Grids are described, with the definitions of modern problems that are to be solved using extensive exchange of information between different nodes in the grid.

The thesis contains a detailed review of the data compression methods utilized in Smart Grids, pointing out trends in the field, challenges, and most popular use cases. The architecture of the proposed solution is based on the insights gained during this stage of research. The review focuses on the most relevant, innovative and impactful work in the field. The proposed method leverages wavelet compression and Bayesian optimization to achieve significant reductions in data size while maintaining signal integrity, crucial for the efficient operation of Smart Grids. Wavelet compression, with its ability to decompose signals into different frequency components, is particularly suited to handle the varied and complex nature of power grid signals. By applying Bayesian optimization, the parameters of the wavelet transform can be fine-tuned to maximize compression efficiency.

The architecture of the proposed system is designed to be scalable and adaptable, making it well suited for deployment in the heterogeneous environments typical of Internet of Things and Smart Grid applications. The system's adaptability allows it to handle various types of signals and data conditions, typical for power grids. Additionally, the use of neural network for compression parameterization further augments the system's capabilities, providing an intelligent layer that learns and adapts to the specific characteristics of the data.

The intention of the research was the creation of the new Discrete Wavelet Parameterization framework that would reach beyond proof-of-concept and be implementation ready. To achieve that, a part for embedded systems was created and documented. Embedded code is intended to be used by endpoint devices in data acquisition systems. It was designed to be reliable and efficient and to fulfill the constraints of popular embedded systems used in metrology. The embedded part of the project is compatible with the parameterization component, and can change the compression algorithm used in the runtime, to match the changes in the signal.

In terms of contributions, this thesis presents a comprehensive study of signal characteristics in power grids, identifying key challenges and opportunities for data compression. The integration of wavelet compression and Bayesian optimization represents a novel approach that advances the state-of-the-art in Smart Grid data management. The architecture and implementation of the system are thoroughly detailed, providing a blueprint for future research and development in this field.

The main contribution of this work is creation of the method for dynamic Discrete Wavelet Transform parameterization, that improves the performance of compression algorithms both in terms of compression ratio and compression loss. The parameterization can be done in parallel to the normal operation of the compression system, which permits implementation that will add zero timing overhead, which is beneficial for real time systems. The method is useful in particular in systems analyzing signals for Power Systems protection and Power Quality monitoring, since Discrete Wavelet Transform performs well for compression of local transients.

The experimental results demonstrate the effectiveness of the proposed method, showing substantial improvements in compression ratios, on average providing 50% smaller data size. The use of a neural network resulted in a significant (over 3000 times) reduction of the computational effort needed for the parameterization. These results highlight the potential for significant cost savings and performance enhancements in Smart Grid operations through optimized data handling.

Finally, the thesis outlines several directions for further research, including the exploration of advanced neural network architectures for compression, the integration of real-time data analytics, and the application of the proposed methods to other domains within the Internet of Things. These future research directions aim to build on the foundation laid by this work, pushing the boundaries of what is possible in data compression and Smart Grid technology.

This thesis not only addresses a critical need in the field of Smart Grids but also provides a solid framework for ongoing innovation and improvement in data compression and transmission. Through detailed analysis, innovative methodology, and practical application, this work contributes to the advancement of efficient, effective, and intelligent Smart Grid systems.

Streszczenie

Niniejsza praca podejmuje problem rozległego wykorzystania pamięci masowej i transmisji danych przez systemy pomiarowe w urządzeniach Internetu Rzeczy stosowanych w inteligentnych sieciach elektroenergetycznych. Rosnące zapotrzebowanie na wydajne i efektywne działanie nowoczesnych sieci elektroenergetycznych wymaga precyzyjnej kontroli. Jakość strategii sterowania zależy od jakości i ilości danych wejściowych. Koszt i złożoność systemów akwizycji danych w sieciach elektroenergetycznych i innych systemach jest bezpośrednio skorelowana z wykorzystywaną infrastrukturą, taką jak sieci komunikacyjne i pamięć. Poprawę systemu sterowania w sieci elektroenergetycznej można osiągnąć poprzez zwiększenie wydajności procesów przetwarzania danych. Sposobem na optymalizację wykorzystania pamięci masowej i transmisji jest kompresja danych. Badania przedstawione w niniejszej pracy koncentrują się na stworzeniu nowego sposobu parametryzacji dyskretnej transformacji falkowej, który ulepszy obecne algorytmy.

Niniejsza praca przedstawia najważniejsze problemy związane z przetwarzaniem danych pomiarowych w sieciach elektroenergetycznych. Szczegółowo opisano sygnały obserwowane w sieciach elektroenergetycznych. Wskazano najważniejsze cechy sygnałów inteligentnych sieci elektroenergetycznych, a także różnice między sygnałami transmisji mocy a różnymi domenami, takimi jak komunikacja lub dźwięk. Wskazano również źródła problemu leżące w teorii komunikacji. Opisano najpopularniejsze technologie komunikacyjne stosowane w inteligentnych sieciach energetycznych, a także zdefiniowano współczesne problemy, które mają zostać rozwiązane przy użyciu szerokiej wymiany informacji między różnymi węzłami w sieci.

Rozprawa zawiera szczegółowy przegląd metod kompresji danych stosowanych w inteligentnych sieciach energetycznych, wskazując trendy w tej dziedzinie, wyzwania i najpopularniejsze przypadki użycia. Architektura proponowanego rozwiązania opiera się na spostrzeżeniach uzyskanych na tym etapie badań. Przegląd koncentruje się na najbardziej istotnych, innowacyjnych i wpływowych pracach w tej dziedzinie.

Proponowana metoda wykorzystuje kompresję falkową i optymalizację bayesowską w celu osiągnięcia znacznej redukcji rozmiaru danych przy jednoczesnym zachowaniu integralności sygnału, co jest kluczowe dla wydajnej pracy inteligentnych sieci energetycznych. Kompresja falkowa, dzięki swojej zdolności do rozkładania sygnałów na różne składowe częstotliwości, jest szczególnie odpowiednia do obsługi zróżnicowanej i złożonej natury sygnałów sieci energetycznych. Poprzez zastosowanie optymalizacji bayesowskiej parametry transformacji falkowej można precyzyjnie dostroić w celu maksymalizacji wydajności kompresji.

Architektura proponowanego systemu została zaprojektowana tak, aby była skalowalna i adaptowalna, dzięki czemu dobrze nadaje się do wdrażania w heterogenicznych środowiskach typowych dla aplikacji

Internetu Rzeczy i Smart Grid. Adaptowalność systemu pozwala mu obsługiwać różne typy sygnałów i warunki danych, typowe dla sieci energetycznych. Ponadto wykorzystanie sieci neuronowej do parametryzacji kompresji dodatkowo zwiększa możliwości systemu, zapewniając inteligentną warstwę, która uczy się i dostosowuje do specyficznych cech danych.

Intencją badań było stworzenie nowego frameworka parametryzacji falki dyskretnej, który wykraczałby poza proof-of-concept i byłby gotowy do wdrożenia. Aby to osiągnąć, stworzono i udokumentowano część dla systemów wbudowanych. Kod wbudowany ma być używany przez urządzenia końcowe w systemach akwizycji danych. Został zaprojektowany tak, aby był niezawodny i wydajny oraz spełniał ograniczenia popularnych systemów wbudowanych stosowanych w metrologii. Wbudowana część projektu jest zgodna ze składnikiem parametryzacji i może zmieniać algorytm kompresji używany w czasie wykonywania, aby dopasować zmiany w sygnale.

W zakresie wkładu, ta rozprawa przedstawia kompleksowe badanie charakterystyk sygnału w sieciach energetycznych, identyfikując kluczowe wyzwania i możliwości kompresji danych. Integracja kompresji falkowej i optymalizacji bayesowskiej stanowi nowatorskie podejście, które rozwija najnowocześniejsze rozwiązania w zakresie zarządzania danymi Smart Grid. Architektura i implementacja systemu są szczegółowo opisane, zapewniając plan przyszłych badań i rozwoju w tej dziedzinie.

Głównym wkładem tej pracy jest stworzenie metody dynamicznej parametryzacji dyskretnej transformacji falkowej, która poprawia wydajność algorytmów kompresji zarówno pod względem współczynnika kompresji, jak i strat kompresji. Parametryzacja może być wykonywana równolegle do normalnej pracy systemu kompresji, co umożliwia implementację, która doda zerowy narzut czasowy, co jest korzystne dla systemów czasu rzeczywistego. Metoda jest przydatna w szczególności w systemach analizujących sygnały w celu ochrony systemów energetycznych i monitorowania jakości energii, ponieważ dyskretna transformacja falkowa dobrze sprawdza się w kompresji lokalnych stanów przejściowych.

Wyniki eksperymentów dowodzą skuteczności proponowanej metody, wykazując znaczną poprawę współczynników kompresji, zapewniając średnio o 50% mniejszy współczynnik kompresji. Zastosowanie sieci neuronowej spowodowało znaczną (ponad 3000-krotną) redukcję wysiłku obliczeniowego potrzebnego do parametryzacji. Wyniki te podkreślają potencjał znacznych oszczędności kosztów i ulepszeń wydajności w operacjach Smart Grid dzięki zoptymalizowanej obsłudze danych.

Na koniec rozprawa przedstawia kilka kierunków dalszych badań, w tym eksplorację zaawansowanych architektur sieci neuronowych do kompresji, integrację analizy danych w czasie rzeczywistym i zastosowanie proponowanych metod w innych domenach w ramach Internetu rzeczy. Te przyszłe kierunki badań mają na celu budowanie na fundamencie położonym przez tę pracę, przesuwając granice tego, co jest możliwe w kompresji danych i technologii Smart Grid.

Ta rozprawa nie tylko odpowiada na krytyczną potrzebę w dziedzinie Smart Grids, ale także zapewnia solidne ramy dla ciągłych innowacji i ulepszeń w kompresji i transmisji danych. Dzięki szczegółowej analizie, innowacyjnej metodologii i praktycznemu zastosowaniu, praca ta przyczynia się do rozwoju wydajnych, efektywnych i inteligentnych systemów Smart Grid.

Contents

List of Abbreviations

A-XDR Adapted eXtended Data Representation

AAC Advanced Audio Coding

AC Alternating Current

AE Autoencoder

ALAC Apple Lossless Audio Codec

ALS MPEG-4 Audio Lossless Coding

AM Amplitude Modulation

AMCH Adaptive Markov Chain Huffman Coding

AMDC Adaptive Multivariate Data Compression

AMI Advanced Metering Infrastructure

AMMMO Adaptive Multi-Model Middle-Out

AMPds Almanac of Minutely Power dataset

ANSI American National Standards Institute

APE Monkey's Audio

ARIMA Autoregressive Integrated Moving Average

ASDU Application Service Data Units

ATH Adaptive Trimmed Huffman

AV1 AOMedia Video 1

AVC Advanced Video Coding

bps Bits Per Second

BREACH Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext

BWT Burrows–Wheeler Transform

bzip2 Burrows-Wheeler zip version 2

CD Campus Dataset

CF Compression Factor

CNN Convolutional Neural Network

CPU Central Processing Unit

CR Compression Ratio

CRIME Compression Ratio Info-leak Made Easy

CS Compressed Sensing

CSV Comma-separated values

CT Cosine Transform

CUDA Compute Unified Device Architecture

DC Data Compression

DCT Discrete Cosine Transform

DE Difference Encoding

DEGA Differential + Entropy/Golomb/Arithmetic

DER Distributed Energy Resource

DFT Discrete Fourier Transform

DRED Dutch Residential Energy Dataset

DSP Digital Signal Processor

DTCWT Dual Tree Complex Wavelet Transform

DWT Discrete Wavelet Transform

EBCOT Embedded Block Coding with Optimized Truncation

EC Exception Compression

ECO Electricity Consumption and Occupancy

EI Expected Improvement

EMI Electromagnetic Interference

EZWT Embedded Zerotree Wavelet Transform

FL Federated Learning

FLAC Free Lossless Audio Codec

FM Frequency Modulation

FPGA Field-programmable Gate Array

FT Fuzzy Transform

GIF Graphics Interchange Format

GMDH Group Method of Data Handling

GP Gaussian Process

GPU Graphics Processing Unit

GSL GNU Scientific Library

Gzip GNU zip

HEIF High Efficiency Image File Format

HEVC High Efficiency Video Coding

HTTP Hypertext Transfer Protocol

IEC International Electrotechnical Commission

IEEE Institute of Electrical and Electronics Engineers

ILWT Integer Lifting Wavelet Transform

INAE Integrated Normalized Absolute Error

IoT Internet of Things

IP Internet Protocol

ISAAC Intelligent Synchrophasor Data Real-Time Compression Framework for WAMS

JICE Joint Data Compression and Encryption

JPEG Joint Photographic Experts Group

JPEG2000 Joint Photographic Experts Group 2000

K-RLE Run-Length Encoding with a K-Precision

K-SVD K-Singular Value Decomposition

KB kilobytes

LCP Lossless Coding considering Precision

LFSRs Linear Feedback Shift Registers

LIFTED Labelled hIgh Frequency daTaset for Electricity Disaggregation

LoRa Long Range

LSTM Long Short-Term Memory

LZ77 Lempel-Ziv ’77

LZ78 Lempel-Ziv ’78

LZMA Lempel–Ziv–Markov Chain Algorithm

LZMH Lempel Ziv-Markov Chain-Huffman

LZW Lempel–Ziv–Welch

MADE Maximum Absolute Deviation Error

MB megabytes

MCU Microcontroller Unit

MDCT Modified Discrete Cosine Transform

MIAE Mean Integrated Absolute Error

MIT Massachusetts Institute of Technology

ML Machine Learning

MLP Multi-layer Perceptron

MP3 MPEG-1 Audio Layer III or MPEG-2 Audio Layer III

MP4 MPEG-4 Part 14

MPEG Moving Picture Experts Group

MQ-coder Multiplication/Quotient coder

MQTT MQ Telemetry Transport

MRA Multi-resolution Analysis

MSE Mean Squared Error

NN Neural Network

NRMSE Normalized Root Mean Square Error

OD Office Dataset

ONNX Open Neural Network Exchange

OpenCL Open Computing Language

openMP Open Multi-Processing

PC Personal Computer

PCA Principal Component Analysis

PF Power Factor

PG Power Grid

PI Probability of Improvement

PLA Piecewise Linear Approximation

PLC Power Line Communication

PMU Phasor Measurement Unit

PNG Portable Network Graphics

PPMd Prediction by Partial Matching variant d

PQ Power Quality

PS Power System

PWM Pulse Width Modulation

QAM Quadrature Amplitude Modulation

RAR Roshal Archive

RDC Resumable Data Compression

REDD Reference Energy Disaggregation Data Set

RIFF Resource Interchange File Format

RLE Run-length Encoding

RMS Root Mean Square

RMSE Root Mean Square Error

RT Real Time

SAX Symbolic Aggregation Approximation

SDT Swing Door Trending

SG Smart Grid

SHD Smart Home Dataset

SM Smart Meter

SME Small and Medium-sized Enterprises

SNR Signal-to-Noise Ratio

SoC System on a Chip

SPECK Set Partitioned Embedded Block Coder

SPIHT Set Partitioning in Hierarchical Trees

SV Sampled Value

SVD Singular Value Decomposition

TBB Intel Threading Building Blocks

THD Total Harmonic Distortion

TPE Tree-structured Parzen Estimator

TTA The True Audio

TUD TU Darmstadt

UCB Upper Confidence Bound

UK-DALE UK recording Domestic Appliance-Level Electricity

UPS Uninterruptible Power Supplies

UTF-8 8-bit Unicode Transformation Format

VP9 Video Processor 9

WAMS Wide-area Measurement System

WAV Waveform Audio File Format

WPD Wavelet Packet Decomposition

WSN Wireless Sensor Networks

WT Wavelet Transform

List of Figures

Chapter 1

Introduction

The advent of the Internet of Things (IoT) has transformed various domains, particularly the Smart Grid (SG), by enabling extensive data collection and RT monitoring through a network of interconnected sensors [7]. However, this exponential increase in data volume presents significant challenges related to data storage and transmission efficiency [8]. To address these challenges, it is essential to develop advanced Data Compression (DC) techniques that not only reduce data volume but also preserve data integrity and enable time-sensitive processing. This doctoral thesis presents a comprehensive study and development of a NN-based adaptive DC system specifically designed for IoT sensor networks in SG, utilizing Discrete Wavelet Transform (DWT) to compress time-series data.

1.1 Research Thesis

Compression algorithms using DWT to compress time-series data with distortion in SG can be improved by adapting WT parameters—wavelet function, decomposition level, and threshold value—to the sampled signal. This parameterization can be implemented in a way that does not introduce runtime overhead.

1.2 Methodology

To systematically address these objectives, the study followed these methodological steps:

- **Literature Review** — An exhaustive review of existing DC techniques relevant to SG and other IoT sensor networks was conducted. This included both lossless and lossy methods, evaluating their efficiency, computational complexity, and suitability for resource-constrained IoT environments.

- **System Development and Design** — Based on insights from the literature review, a novel adaptive DC system architecture was designed. This phase involved developing key components: data preparation software, a NN for DWT parameterization, embedded C++ software, and a robust validation framework.

- **NN Design and Training** — A NN was trained using a comprehensive dataset representing diverse types of IoT sensor data. The training focused on enabling the network to generalize effectively to unseen data, thereby improving compression efficiency and minimizing computational overhead during runtime.

- **Software Implementation and Testing** — The embedded software was implemented in C++ for direct deployment on IoT devices. Extensive testing—including unit, integration, and field tests—was conducted to verify the software's reliability and performance under a range of conditions.

- **Validation Framework Development** — A comprehensive validation framework was created to assess the effectiveness of the compression system using data typical of SG. In addition to simulated data, real-world measurements from various microgrid configurations were also used to verify system performance.

1.3 Expected Contributions

This research is expected to make several significant contributions to the field of DC for SG systems:

- **Adaptive DC System** — Development of an adaptive DC system that achieves superior compression ratios compared to existing DWT-based solutions and is fully implementation-ready for deployment in real-world applications.

- **System Architecture** — A detailed design and presentation of the proposed architecture, facilitating understanding and enabling replication in other IoT applications. A key focus of the design is on avoiding computational overhead during runtime; any additional steps are performed during development, deployment, or in parallel with the application.

- **Embedded Software** — Development of high-performance embedded C++ software compatible with user devices, ensuring broad interoperability and robustness across various IoT platforms.

- **NN-Based Parameterization** — Introduction of a NN-based method that significantly reduces the time required for DWT parameter selection, improving compression efficiency and enhancing suitability for RT applications.

1.4 Thesis Structure

The thesis is organized into 15 chapters. It provides a background in Power Grid (PG) signals and communication, outlines the core problem, reviews compression methods used in SG, and describes the architecture, design, and implementation of the proposed solution. The final chapters analyze results, compare them with alternative approaches, and suggest directions for future research.

The purpose of each chapter is as follows:

1. **Introduction** — Defines the research problem and provides an overview of the proposed solution.
2. **Signals in Power Grids** — Provides a description of signals in PG and highlights key features used for PQ analysis.
3. **Communication in Smart Grids** — Describes communication methods used in modern PGs and presents fundamental principles of communication theory.
4. **Theoretical Background and Literature Review on Data Compression in Smart Grids** — Offers an in-depth review of current research on DC in SG, with supporting background in information theory and comparisons to compression techniques in mature domains such as audio and video.
5. **Different Approaches to Data Compression in Electrical Signals** — Analyzes commonly used compression methods in SG and justifies the selection of a DWT-based approach.
6. **Wavelet Compression** — Presents a detailed examination of WT-based compression techniques.
7. **Parameterization of the Compression System** — Describes methods for parameterizing DWT-based compression and introduces the parameterization strategy adopted in this thesis.
8. **Proposed Solution** — Details the architecture and core concepts of the dynamically parameterized DWT-based compression system.
9. **Dataset** — Provides information about the data used to train the NN and validate the system.
10. **Neural Network** — Describes the core component parameterizing the compression system.
11. **Embedded Software Component** — Discusses the design of C++ code compatible with the parameterization system and proposes strategies for efficiently executing compression and inference using parallel computing.
12. **Results** — Presents the achieved CR, MSE improvements, and reductions in computational effort due to the NN.
13. **Contributions** — Summarizes the key contributions of the thesis.
14. **Summary** — Reflects on the research findings, methods, and overall contribution of the work.
15. **Directions for Future Research** — Proposes avenues for further development of data compression techniques in SG.

Chapter 2

Signals in power grids

Signals in PG are fundamental to the operation, monitoring, and control of electrical PS. These signals exhibit unique characteristics that distinguish them from signals in other domains such as telecommunications or audio processing. Understanding the unique properties of PG signals is essential for selecting proper DC methods.

2.1 Basic Characteristics of Power Grid Signals

PG signals are primarily composed of voltage and current waveforms, which can be represented mathematically as sinusoidal functions [9]. These signals are the backbone of power delivery and distribution systems and exhibit several distinctive features.

2.1.1 Sinusoidal Nature of Voltage and Current

The fundamental representation of voltage and current in PG is sinusoidal:

$$V(t) = V_m \sin(\omega t + \phi) \tag{2.1}$$

$$I(t) = I_m \sin(\omega t + \theta) \tag{2.2}$$

where V_m and I_m are the peak values of voltage and current in volts (V) and amperes (A), ω is the angular frequency in radians per second (rad/s) (related to the PS's nominal frequency, f, in hertz (Hz) by $\omega = 2\pi f$), and ϕ and θ are the phase angles of voltage and current in radians (rad), respectively.

The sinusoidal nature of these signals arises from the need for efficient energy transfer. Alternating Current (AC) systems, which operate at sinusoidal waveforms, minimize energy losses and allow for the efficient transformation of voltage levels via transformers. This is essential for long-distance power transmission [10].

2.1.2 Frequency and Harmonics

The nominal frequency of PG signals is typically 50 Hz or 60 Hz, depending on the region. However, non-linear loads such as rectifiers, variable-speed drives, and electronic devices introduce harmonics into the system. These harmonics are integer multiples of the fundamental frequency and can be represented as:

$$V(t) = \sum_{n=1}^{\infty} V_n \sin(n\omega t + \phi_n) \tag{2.3}$$

$$I(t) = \sum_{n=1}^{\infty} I_n \sin(n\omega t + \theta_n) \tag{2.4}$$

where n is the harmonic number, and V_n and I_n are the amplitudes of the n-th harmonic in volts (V) and amperes (A), respectively. Harmonics can cause various problems, such as increased heating in equipment, interference with communication lines, and misoperation of protection devices [11].

Harmonic distortion is a critical aspect of PQ. The most common harmonics found in PSs are the third, fifth, seventh, and ninth harmonics. These can be caused by various types of equipment, such as electric-arc furnaces, other arcing loads, variable frequency drives for motors, fluorescent lighting, and other electronic devices that draw non-linear currents. The presence of harmonics can be analyzed using Fourier series, which decompose the signal into its fundamental and harmonic components.

2.1.3 Phase Angle

The phase angle between voltage and current is crucial in PS, as it determines the Power Factor (PF). The PF is a measure of how effectively electrical power is being used:

$$\cos(\phi - \theta) = \text{Power Factor (unitless)} \tag{2.5}$$

A PF close to 1 indicates efficient power usage, where most of the power is being converted into useful work. In contrast, a lower PF indicates inefficiency, where more power is wasted in the form of reactive power. This can lead to higher energy costs and the need for larger capacity equipment.

The phase angle can also affect the synchronization of the generators in the grid. Generators must operate in phase with each other to ensure stable and reliable power supply. Any significant deviation in the phase angle can cause instability and potential outages.

2.1.4 Active, Reactive, and Apparent Power

Power in an electrical circuit is defined as the rate at which energy is transferred or converted by the components in the circuit. It represents how much work is done by electrical charges as they move through the circuit over time. The instantaneous power $p(t)$ at any moment in time is given by the product of the instantaneous voltage $v(t)$ and the instantaneous current $i(t)$:

$$p(t) = v(t) \times i(t) \tag{2.6}$$

where $p(t)$ is instantaneous power in watts (W), $v(t)$ is instantaneous voltage in volts (V), and $i(t)$ is instantaneous current in amperes (A).

Instantaneous Power in AC Circuits

In AC circuits, voltage and current are sinusoidal and vary with time. They can be represented as:

$$v(t) = V_m \sin(\omega t + \theta_v) \tag{2.7}$$

$$i(t) = I_m \sin(\omega t + \theta_i) \tag{2.8}$$

Here, V_m and I_m are the peak values of voltage and current in volts (V) and amperes (A), ω is the angular frequency in radians per second (rad/s), and θ_v and θ_i are the phase angles of voltage and current in radians (rad), respectively. The instantaneous power at any time t is then given by:

$$p(t) = V_m I_m \sin(\omega t + \theta_v) \sin(\omega t + \theta_i) \tag{2.9}$$

By applying trigonometric identities, this can be simplified to:

$$p(t) = \frac{V_m I_m}{2} [\cos(\theta_v - \theta_i) - \cos(2\omega t + \theta_v + \theta_i)] \tag{2.10}$$

This equation reveals that instantaneous power consists of two terms: a constant term $\frac{V_m I_m}{2} \cos(\theta_v - \theta_i)$, which represents the average power, and a time-varying term $\frac{V_m I_m}{2} \cos(2\omega t + \theta_v + \theta_i)$, which represents the oscillatory component of power in AC circuits.

Average Power

In AC circuits, we are typically interested in the **average power** over a complete cycle, rather than the instantaneous power at a given moment. The average power, P_{avg}, is given by:

$$P_{\text{avg}} = \frac{V_m I_m}{2} \cos(\theta_v - \theta_i) \tag{2.11}$$

The term $\cos(\theta_v - \theta_i)$ is known as the **PF**, denoted by $\cos\phi$, where ϕ is the phase difference between the voltage and current. Therefore, the average power can also be written as:

$$P_{\text{avg}} = V_{\text{rms}} I_{\text{rms}} \cos\phi \tag{2.12}$$

Here, V_{rms} and I_{rms} are the Root Mean Square (RMS) values of voltage and current in volts (V) and amperes (A), respectively.

Reactive and Apparent Power

The power in AC circuits can be divided into active power (P, watts, W), reactive power (Q, volt-ampere reactive, var), and apparent power (S, volt-amperes, VA):

$$P = V_m I_m \cos(\phi - \theta) \tag{2.13}$$

$$Q = V_m I_m \sin(\phi - \theta) \tag{2.14}$$

$$S = V_m I_m \tag{2.15}$$

Active power represents the real power consumed by resistive loads to perform work (e.g., heating, lighting). Reactive power represents the power oscillating between the source and reactive components (inductive and capacitive loads) in the system. Apparent power is the vector sum of active and reactive power and represents the total power flow in the system.

The relationships between these power components are often illustrated using a power triangle, where:

$$S = \sqrt{P^2 + Q^2} \tag{2.16}$$

Complex Power

In addition to the basic power components, the concept of complex power can be introduced. Complex power ($\mathbf{S}$) is represented as:

$$\mathbf{S} = P + jQ \tag{2.17}$$

where $\mathbf{S}$ is the complex power in volt-amperes (VA), P is the active power in watts (W), and Q is the reactive power in volt-ampere reactive (var).

Power in Three-Phase Systems

Three-phase systems are commonly used in industrial settings because they provide a more efficient and balanced power supply. The total power in a balanced three-phase system is the sum of the power in each phase. For three-phase systems, the total active power is given by:

$$P_{\text{total}} = \sqrt{3} V_{\text{line}} I_{\text{line}} \cos\phi \tag{2.18}$$

where V_{line} is the line-to-line voltage in volts (V), I_{line} is the line current in amperes (A), and P_{total} is the total active power in watts (W). Similarly, the total reactive and apparent power in a three-phase system are:

$$Q_{\text{total}} = \sqrt{3} V_{\text{line}} I_{\text{line}} \sin\phi \tag{2.19}$$

$$S_{\text{total}} = \sqrt{3} V_{\text{line}} I_{\text{line}} \tag{2.20}$$

where Q_{total} is in volt-ampere reactive (var) and S_{total} is in volt-amperes (VA).

2.2 Signal Quality and Power Grid Stability

PG stability and signal quality are essential for reliable operation. Several indicators and phenomena are critical to assessing and maintaining these aspects, such as transients and harmonics [12]. Different PQ parameters provide information about different components of the system and must be measured in a different way 2.1.

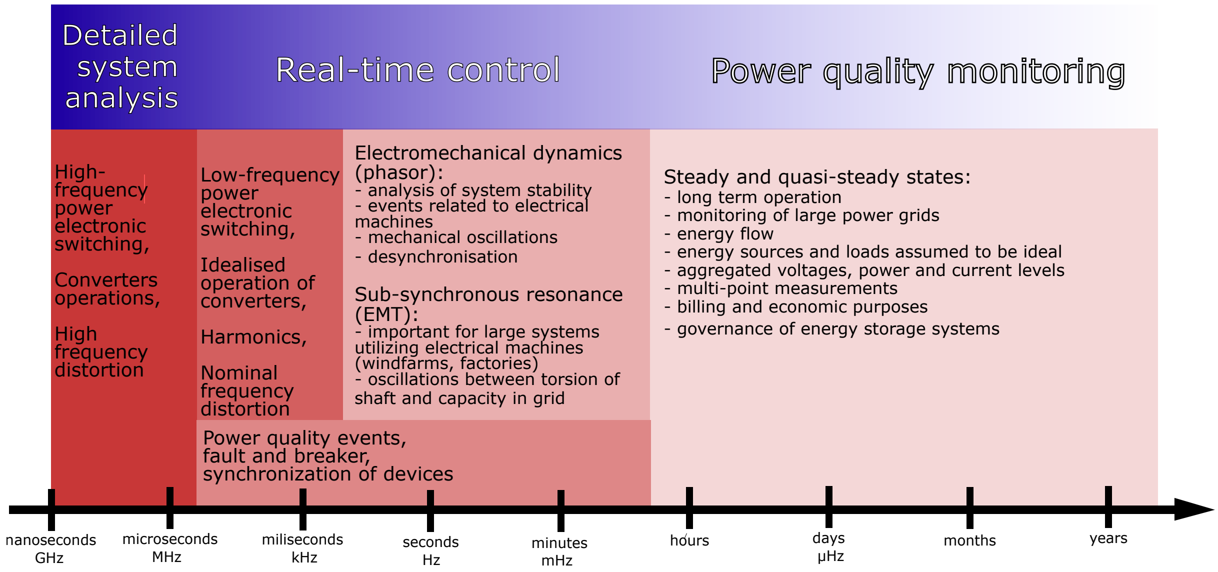


Figure 2.1: Data measured and information gathered in power systems in time and frequency domain. Blue stripe on top presents the main reasons for measurement of PSs properties in listed time domains. Red stripe in the middle presents crucial information extracted from measured data. Axis presenents time-frames of described phenomena and probing frequencies required to register them [1].

2.2.1 Voltage Sag, Swell, and Interruptions

Voltage sags are short-duration decreases in the RMS of the voltage, typically caused by faults, sudden load changes, or large motor starts. Voltage swells are short-duration increases in the RMS of the voltage, often resulting from switching operations or the sudden removal of large loads. Interruptions occur when the voltage falls to zero for a short period of time.

Voltage disturbances can be mathematically characterized by their magnitude and duration. For example, a voltage sag can be expressed as:

$$V_{\text{sag}}(t) = V_m \left(1 - \Delta V\right) \sin(\omega t + \phi) \tag{2.21}$$

where ΔV represents the percentage reduction in voltage magnitude (dimensionless).

The impact of voltage sags and swells on equipment and processes can be significant. Sensitive electronic devices and industrial processes may experience malfunctions or shutdowns during these disturbances [13]. Therefore, monitoring and mitigating voltage disturbances are critical to maintaining PQ. The loss of information about transients during the compression process may hinder the functionality of protection systems.

2.2.2 Total Harmonic Distortion

Total Harmonic Distortion (THD) is a measure of the distortion in a signal due to harmonics [14]. It quantifies the extent to which the waveform deviates from a pure sinusoidal shape:

$$\text{THD} = \frac{\sqrt{\sum_{n=2}^{\infty} V_n^2}}{V_1} \tag{2.22}$$

where V_1 is the amplitude of the fundamental frequency component in volts (V), and V_n are the amplitudes of the harmonic components in volts (V). THD is typically expressed as a percentage (dimensionless). High THD levels indicate significant waveform distortion, which can lead to overheating of electrical equipment, increased losses, and potential malfunction of sensitive electronic devices.

The effects of THD on power systems include increased heating in transformers and motors, reduced efficiency, and potential interference with communication systems [15]. To mitigate THD, various filtering techniques and harmonic mitigation strategies are employed in power systems.

2.2.3 Voltage and Frequency Stability

Voltage stability refers to the ability of the power system to maintain acceptable voltage levels on all buses of the system under normal operating conditions and after being subjected to a disturbance [16]. Frequency stability involves maintaining the frequency of the system within specified limits after a disturbance, ensuring the balance between generation and load.

Voltage instability can lead to voltage collapse, where the system cannot maintain voltage levels, resulting in blackouts. Frequency instability can cause generators to lose synchronization, leading to widespread outages [17].

To ensure voltage and frequency stability, power systems employ various control mechanisms, such as automatic voltage regulators, load shedding schemes, and frequency control reserves.

2.2.4 Flicker

Flicker in electrical power systems refers to the rapid, repeated fluctuations in voltage, which can cause noticeable variations in the intensity of lighting and other sensitive equipment. Mathematically, flicker is often represented as periodic voltage fluctuations superimposed on the nominal voltage waveform [18]. The fluctuating voltage, $V(t)$, can be described as a time-varying sinusoidal waveform with a modulation component:

$$V(t) = V_0\,[1 + m(t)]\sin(\omega t) \tag{2.23}$$

Here, V_0 is the nominal voltage in volts (V), ω is the angular frequency in radians per second (rad/s), and $m(t)$ is the modulation index (dimensionless) representing the flicker magnitude, which varies over time.

The severity of flicker can be quantified using the RMS of the voltage fluctuations. If the voltage fluctuates periodically, with a modulation index m, the RMS voltage V_{rms} is given by:

$$V_{\text{rms}} = V_0\sqrt{1 + \frac{m^2}{2}} \tag{2.24}$$

where V_{rms} is in volts (V) and m is dimensionless.

When flicker is present, this formula shows that the RMS voltage changes proportionally with the magnitude of m. Flicker is often evaluated through indices such as the short-term flicker severity index P_{st} and the long-term flicker severity index P_{lt}. The short-term index P_{st} is typically calculated over a 10-minute period, and the long-term index P_{lt} over several hours. Both indices are derived based on statistical analysis of the time series data for the modulation function $m(t)$.

The effects of flicker on the power system are not limited to lighting disturbances; they can also impact other electrical equipment. The power transferred to a load in an AC system is given by:

$$P(t) = V(t)I(t) = V_0 I_0 \left[1 + m(t)\right] \sin(\omega t) \sin(\omega t + \theta) \tag{2.25}$$

where V_0 is the nominal voltage in volts (V), I_0 is the nominal current in amperes (A), $I(t)$ is the instantaneous current in amperes (A), and θ is the phase difference between voltage and current in radians (rad). The time-varying term $m(t)$ affects the active power $P(t)$ in watts (W), causing periodic variations in the load. These variations can stress equipment, lead to power quality issues, and reduce system stability. Flicker mitigation involves controlling the sources of voltage fluctuations, such as large fluctuating loads or power electronic devices, through techniques like voltage regulation, reactive power compensation, or harmonic filtering [19].

2.3 Comparison with Other Domains

Signals in power grids differ from those in different domains in various ways.

2.3.1 Frequency Range

Main components of PG signals operate at low frequencies (50-60 Hz) compared to telecommunications (kHz to GHz) and audio signals (20 Hz to 20 kHz) [20, 21]. PS contain also components of higher and lower frequencies, but typically of much smaller power than main component. The lower frequency range in PS is primarily due to the need for efficient energy transmission over long distances and the historical development of the electrical grid.

The frequency range of the power grid signals is specifically chosen to balance the trade-off between energy efficiency and equipment design. Higher frequencies would result in increased energy losses due to skin effects and other phenomena, while lower frequencies would require larger and more costly equipment.

2.3.2 Signal Purpose

Power grid signals are primarily concerned with the transmission and distribution of electrical energy, ensuring that power is delivered efficiently and reliably from the generation sources to end users [22]. In contrast, telecommunication signals are used for data transmission, focusing on high-speed and high-frequency communication, and audio signals are designed for sound reproduction, emphasizing fidelity and clarity within the human hearing range.

The primary purpose of PG signals is to ensure the continuous and reliable supply of electrical power. This involves maintaining voltage levels, synchronizing generators, and managing power flows. Telecommunications signals, on the other hand, focus on encoding and transmitting information with minimal delay and error. The purpose of audio signals is to accurately reproduce sound waves for human perception.

2.3.3 Modulation Techniques

In telecommunications, various modulation techniques (Amplitude Modulation (AM), Frequency Modulation (FM), Quadrature Amplitude Modulation (QAM)) are used to encode information into carrier waves. In power systems, modulation is less common, but techniques such as Pulse Width Modulation (PWM) are used in power electronics to control the voltage and current supplied to loads [23]. In non-nominal conditions, the phasor is a signal with simultaneous Amplitude Modulation and Phase Modulation. In the standard, it is calculated using a quadrature demodulator, exactly as in communication systems.

PWM is widely used in power converters and inverters to regulate output voltage and current. It works by varying the duty cycle of a switching signal, effectively controlling the average voltage and current delivered to the load. This is essential for applications such as motor drives, renewable energy systems, and Uninterruptible Power Supplies (UPS).

2.3.4 Signal Integrity and Noise Immunity

Power grid signals must maintain high signal integrity and noise immunity to ensure stable power system operation [24]. This includes minimizing the impact of Electromagnetic Interference (EMI) and maintaining signal quality despite the presence of harmonics and other distortions. In telecommunications, signal integrity and noise immunity are also crucial, but are addressed through different means, such as error correction codes and advanced filtering techniques.

In PSs, ensuring signal integrity involves the use of robust insulation, adequate grounding, and shielding practices. Noise immunity is achieved through the use of filters, surge protectors, and other protective devices. In telecommunications, the integrity of the signal is maintained through the use of digital modulation techniques, error correction algorithms, and advanced signal processing methods.

Chapter 3

Communication in Smart Grids

Information theory provides the foundation for quantifying information. Claude Shannon's seminal work introduced the concept of *channel capacity*, which defines the maximum rate at which information can be transmitted over a communication channel without error:

$$C = B \log_2 \left(1 + \frac{S}{N}\right) \tag{3.1}$$

where C is the channel capacity in Bits Per Second (bps), B is the bandwidth in hertz (Hz), S is the signal power in watts (W), and N is the noise power in watts (W) [25, 26].

Throughput

Throughput is a measure of how much information a system can process in a given amount of time. It is often expressed in bps and is an important metric in evaluating the performance of communication systems. Unlike capacity, which is a theoretical maximum, throughput is an empirical measure that can be affected by network congestion, protocol overhead, and transmission errors. Mathematically, throughput (T) can be defined as

$$T = \frac{M}{t}, \tag{3.2}$$

where M is the total amount of data successfully transmitted in bits (b) and t is the total transmission time in seconds (s) [27].

Data compression reduces the redundant data sent through the channel by decreasing the amount of data required to store the same information (see Figure 3.1).

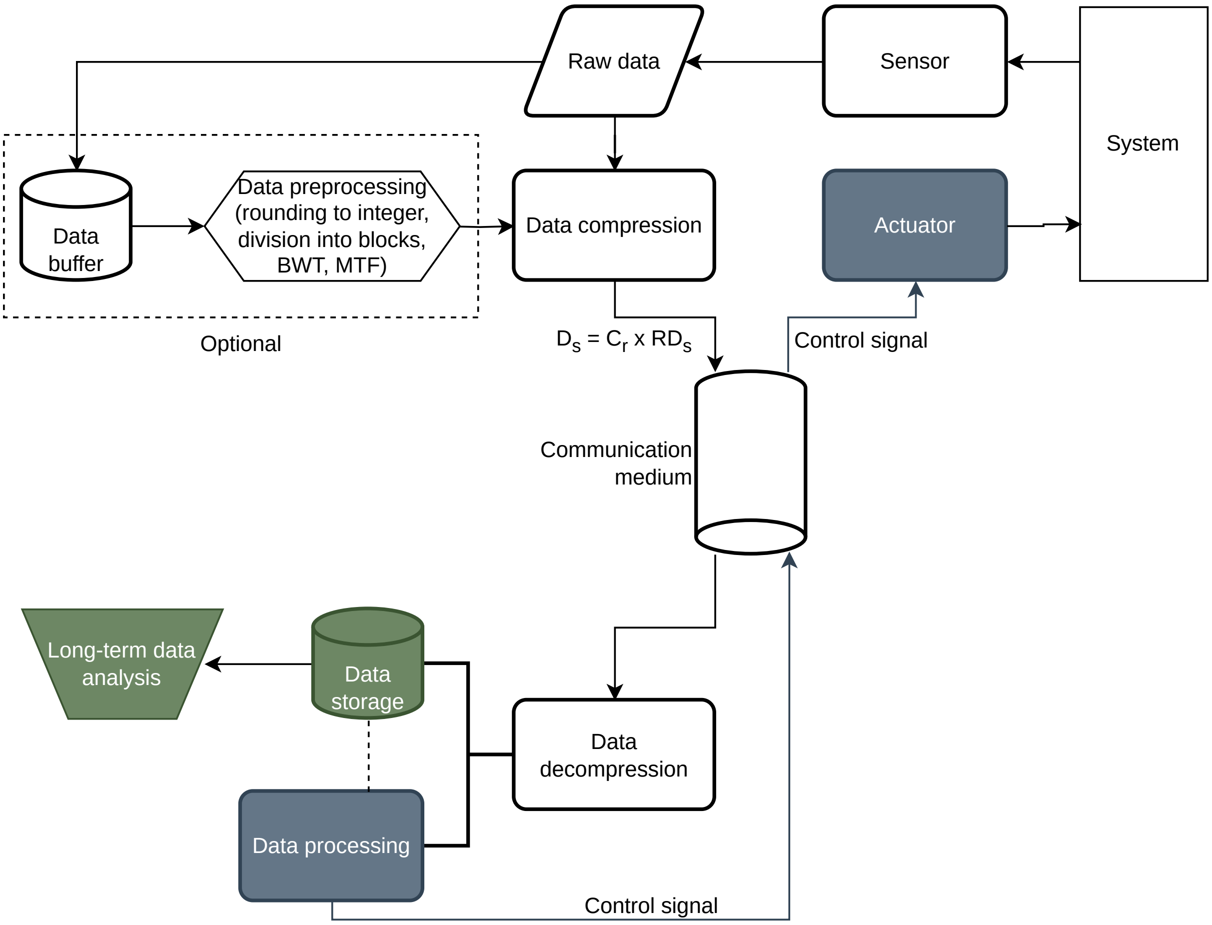


Figure 3.1: Role of data compression in a distributed measurement and control system in power grids. Components in green are typical parts of large-data transmission systems, whose sensor side often employs gigabyte- or terabyte-sized buffers. Components in blue are typical parts of real-time control systems, which use smaller buffers or pure streaming. The dotted line symbolizes an optional data flow. D_s—data size, C_r—compression ratio, RD_s—raw data size. Adapted from [1].

3.0.1 Network Protocols

Network protocols are essential for managing communication in SG. They define the rules for data exchange and ensure interoperability between different devices and systems. Common protocols in SG include the Internet Protocol (IP), Zigbee, and IEC 61850. Their performance can be evaluated using metrics such as latency, throughput, and reliability [28]. Ensuring that compression methods are compatible with these transmission protocols is a major challenge in modern power systems (see Figure 3.2).

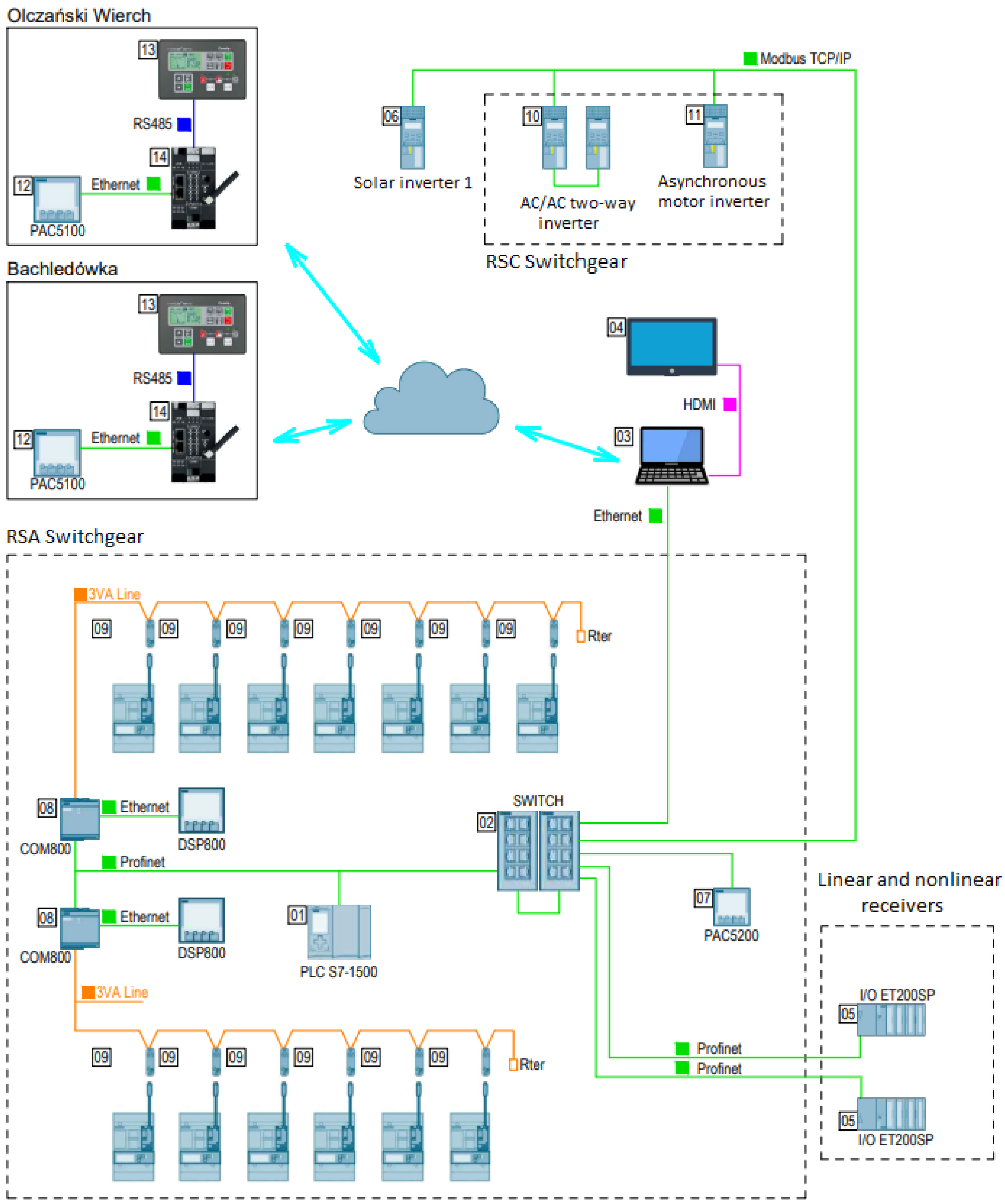


Figure 3.2: Communication architecture in a distributed power-grid laboratory. Sending a message requires compliance with multiple protocol layers, including widely used industrial standards such as Ethernet, PROFINET, and Modbus TCP/IP, as well as proprietary protocols such as the 3VA line [2].

3.1 Communication Technologies in Smart Grids

The deployment of various communication technologies in SG enables efficient data exchange and supports a wide range of applications. Key technologies include wireless communication, Power Line Communication (PLC), and fiber-optic links, although some specific applications may require less common solutions.

3.1.1 Wireless Communication

Wireless communication technologies such as Wi-Fi, cellular networks, and Wireless Sensor Networks (WSN)s are widely used in SG because of their flexibility and ease of deployment. The performance of a wireless link is often analyzed via the bit-error rate (BER) as a function of the signal-to-noise ratio (SNR):

$$\mathrm{BER} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right), \tag{3.3}$$

where $Q(x)$ is the Q-function, E_b is the energy per bit in joules (J), and N_0 is the noise power spectral density in watts per hertz (W/Hz) [29].

3.1.2 Power Line Communication

PLC uses existing electrical power lines to transmit data, effectively turning the power grid into a communication network. Data signals are modulated onto a carrier frequency that travels alongside the standard electrical current without interference. Smart Meter (SM)s and other devices equipped with PLC transceivers can therefore exchange information with the utility, enabling remote metering, device control, and rapid fault localization. The ability to reuse existing wiring over long distances makes PLC a cost-effective enhancement to SG. A drawback is its vulnerability to jamming from the high-frequency components present in power-grid signals.

3.1.3 Fiber Optics

Fiber-optic communication provides high-speed, high-capacity data transmission, making it well suited to critical SG applications. Attenuation of an optical signal can be modeled by

$$P(L) = P_0 e^{-\alpha L}, \tag{3.4}$$

where $P(L)$ is the power at distance L in watts (W), P_0 is the launch power in watts, and α is the attenuation coefficient in inverse meters (m^{-1}) [30]. A key benefit of fiber is its resilience to electromagnetic-compatibility (EMC) interference, which is typically significant near power-transmission lines.

3.2 Applications and Challenges

Integrating advanced communication technologies into SG supports applications such as demand response, distributed-generation management, and fault detection. At the same time, issues including cybersecurity, interoperability, capacity, and scalability must be addressed to ensure reliable grid operation [31].

3.2.1 Demand Response

Demand-response programs adjust power demand rather than supply. Two-way communication enables utilities and consumers to manage consumption in real time. Demand-response optimization can be cast as a minimization problem:

$$\min_x \left\{ \sum_{i=1}^{N} C_i(x_i) \;\middle|\; \sum_{i=1}^{N} x_i = D \right\}, \tag{3.5}$$

where x_i is the power consumption of the i-th consumer in watts, $C_i(x_i)$ is the monetary cost function (e.g., in USD), and D is the total demand in watts [32].

3.2.2 Distributed-Generation Management

Integrating DERs, such as photovoltaic panels and wind turbines, requires sophisticated communication for monitoring and control. Power flow in a distribution network with DERs can be analyzed through the load-flow equations:

$$P_i - V_i \sum_{j=1}^{N} V_j \big(G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij} \big) = 0, \tag{3.6}$$

where P_i is the active power in watts, V_i is the voltage magnitude in volts, G_{ij} and B_{ij} are the conductance and susceptance in siemens (S), respectively, and θ_{ij} is the phase-angle difference between nodes i and j in radians [33].

3.2.3 Fault Detection

Accurate and timely fault detection is crucial to maintaining SG reliability. Communication networks collect data from numerous sensors, which are then analyzed to detect and locate faults. The classification task can be expressed as

$$\hat{y} = \arg\max_y P(y \mid x), \tag{3.7}$$

where x is the observed feature vector and y is the fault class [34]. Data-compression (DC) methods must be chosen so that fault information is preserved; otherwise, protection systems may lose the ability to respond promptly.

Chapter 4

Theoretical Background and Literature Review on Data Compression in Smart Grids

Data exchange is a crucial part of SG management systems. To implement efficient lossy DC method means to filter out the information from the data that is irrelevant for the application. To do so, it is important to understand the properties of signal, and properties of compression algorithms. This chapter explains basis of information theory, which are critical for evaluation of compression techniques.

DC is not a mature field in the domain of power systems; however, it is much more advanced in other domains. Most popular applications of data compression are: filesystems, audio, video, and image data. Some of the properties of these domains can be found in PS. Audio compression algorithms frequently focus on retaining frequencies that are important for human hearing; this approach can be beneficial for PSs, since not all frequencies contained in the signal are equally important for control. Video compression algorithms often leverage the fact that frames are sequential. This property can be also found in PSs, where similarities can be found between consecutive samples, mutually dependent parameters or adjacent measurement points. Image compression focuses on filtering the noise and retaining rapid changes, which is important for PSs, because transients often carry important information about the state of the system. Filesystems usually use lossless compression and focus on computational efficiency, which is very important for PSs, as IoT systems used for data acquisition and control tend to have limited resources. This chapter provides a brief description of notorious compression algorithms in those fields and their properties that can be leveraged in PGs domain.

To define the problem and prove that it is relevant, a review of scientific work is also provided. The review is focused on recent and high-impact work in the field of DC in SG. Literature review lists out the trends in modern research, which helps positioning the work introduced in this thesis in the landscape of DC in PG.

4.1 Information theory

4.1.1 Entropy

Entropy is a fundamental concept in information theory that measures the uncertainty or randomness of a random variable. It provides a lower bound on the average number of bits needed to encode symbols drawn from a source. For a discrete random variable X with possible values $\{x_1, x_2, \ldots, x_n\}$ occurring with probabilities $\{p(x_1), p(x_2), \ldots, p(x_n)\}$, for an information source without memory, the entropy $H(X)$ (bits) is defined as:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i). \tag{4.1}$$

For continuous random variables, the differential entropy $h(X)$ (bits) is defined as:

$$h(X) = -\int_{-\infty}^{\infty} f(x) \log_2 f(x)\, dx, \tag{4.2}$$

where $f(x)$ is the probability density function of X (unitless).

The concept of entropy can be extended to measure the amount of information shared between variables using mutual information $I(X;Y)$ (bits):

$$I(X;Y) = H(X) + H(Y) - H(X,Y), \tag{4.3}$$

where $H(X,Y)$ is the joint entropy of X and Y (bits). Mutual information quantifies the amount of information obtained about one random variable through the other.

4.1.2 Data Compression

Information theory, founded by Claude Shannon in 1948, is a branch of applied mathematics and electrical engineering that involves the quantification of information. The primary objective of information theory is to determine the limits on signal processing operations such as DC and reliable data storage and transmission. The central concept in information theory is the measure of information, which is quantified using entropy. This process can be lossy or lossless. Lossless compression ensures that the original data can be perfectly reconstructed from the compressed data, while lossy compression permits some loss of information, typically in exchange for higher compression ratios.

Mathematically, the efficiency of a compression algorithm can be evaluated using the average code length L (bits per symbol), which for an optimal prefix-free code is given by the entropy $H(X)$ of the source:

$$L \geq H(X), \tag{4.4}$$

where X is a random variable representing the data source, x_i are the possible outcomes, and $p(x_i)$ is the probability of outcome x_i. The entropy $H(X)$ (bits) represents the average amount of information produced by the stochastic data source.

4.2 Data compression across domains

The problem of reducing the storage and transmission burden occurred in many domains where large files are being handled. Efficient streaming of music, video, and other types of media has fueled the development of various data compression methods. Many codecs algorithms are designed specifically to fulfill the requirements of one domain - for example, archiviation of files, video-specific or image-specific algorithms. Each of the domains takes advantage of different properties of the data. Some of the leveraged properties are common with time series signals in PG, and approaches from different domains might be used to build more efficient data compression algorithms specialized for SG.

4.2.1 General-purpose algorithms

General-purpose algorithms are used to archive files to reduce transfer and storage. By nature, they have to be lossless, since even a loss of one bit may corrupt the useful functions of files such as computer programs or text. General-purpose data compression algorithms reduce file size by exploiting redundancies and patterns within the data. These algorithms are most effective when the input contains repeated sequences, predictable structures, or common substrings that can be stored more succinctly than if each occurrence were written out in full. For example, textual data often contain repeated words and characters, while binary data may have recurring bit sequences or repeated blocks. By analyzing the frequency of symbols or rearranging data to expose patterns, compression routines can store repeated information with fewer bits. Some methods also apply transforms that redistribute the data in a way that highlights regularities, making them more amenable to statistical or dictionary-based encoding. Ultimately, by leveraging these repeating elements and predictable features, they significantly reduce storage requirements without losing the essential information.

Common algorithmic primitives used in general-purpose data compression include Run-length Encoding (RLE), dictionary-based techniques (such as Lempel-Ziv '77 (LZ77) and Lempel-Ziv '78 (LZ78)), and entropy encoding (such as Huffman coding or arithmetic coding). RLE condenses consecutive instances of the same symbol into a shorter representation, while dictionary methods replace repeated substrings with references to previous occurrences. Entropy-based schemes assign shorter bit patterns to frequently occurring symbols and longer patterns to rarer ones, thus reducing the average code length. The most popular general-purpose compression algorithms that employ these techniques include Roshal Archive (RAR), GNU zip (Gzip), Lempel–Ziv–Markov Chain Algorithm (LZMA), Lempel–Ziv–Welch (LZW), Burrows-Wheeler zip version 2 (bzip2), and ZIP.

In the domains of electrical signals measurements, such methods may be beneficial for compressing the crucial signals, that cannot afford any loss of information. They can also be used to compress the signal that was initially pre-processed by lossy algorithms, such as WT with thresholding. Many of general purpose algorithms have big compression time, because they are designed for high CR and fast decompression, in such cases, they might add large time overhead for streaming data, but might be used for archivization of signal data files.

4.2.2 Image-specific data compression algorithms

Image compression algorithms often take advantage of the fact that adjacent pixels in an image tend to have similar color or intensity values, leading to a spatial redundancy that can be minimized. Large uniform or slowly varying areas within the image can be represented more compactly by storing a single characteristic value plus a measure of variation, rather than detailing every single pixel. Many methods also consider the variable sensitivity of the human visual system to color and luminance, removing or reducing detail in portions less perceptible to the eye. By converting pixel data into a frequency domain (via transformations) and assigning coarser precision to frequencies less significant to human vision, algorithms further reduce storage without significant perceived quality loss. Additionally, segmentation of color channels allows compression schemes to allocate bits more effectively across different components of the image. Collectively, these techniques enable both lossless and lossy forms of compression, balancing quality and file size.

At the core of image compression lie several fundamental primitives, such as the Discrete Cosine Transform (DCT) and DWT, which help to represent images in a more compressible form by separating out frequencies. Quantization then reduces precision in frequency components that are less critical to the human eye, while lossless compression is applied to minimize the bit costs of frequent patterns. Other common operations include color-space conversion and filtering that reveal more compressible patterns. Some of the most widely used image compression algorithms and formats include Joint Photographic Experts Group (JPEG), Portable Network Graphics (PNG), Graphics Interchange Format (GIF), WebP, High Efficiency Image File Format (HEIF), and Joint Photographic Experts Group 2000 (JPEG2000).

In the domain of electrical signal compression the approach of using transforms such as WT, Cosine Transform (CT), or Burrows–Wheeler Transform (BWT) to express the information contained in the signal in a different domain with more redundancy that can be efficiently compressed by general-purpose algorithms. Focusing on changes between subsequent data points is crucial for image quality, but the same methodology can be used in the compression of electrical signals to preserve data regarding transients, which is crucial for protection systems and PQ analysis.

4.2.3 Video-specific data compression algorithms

Video compression algorithms exploit spatial and temporal redundancies across frames to reduce data size. Consecutive frames in a video sequence often share large regions with minimal change, so instead of encoding each frame independently, these algorithms encode differences (or predicted motion) between them. Image-based compression principles, such as block-based transforms, are applied within individual frames (intra-frame) to capture spatial patterns. Motion estimation and compensation techniques are then used to represent interframe redundancies, reducing the need to store repeated information. In addition, certain parts of moving images can be quantized more aggressively based on how sensitive the human visual system is to changes in those areas. By combining these approaches, video compression ensures both efficient storage and preservation of visual quality.

Core primitives used in video compression include block-based transform coding (like the DCT) for spatial data, quantization to reduce precision where the human eye is less sensitive, then lossless compression is added. Motion estimation and motion compensation are key components that identify and exploit temporal redundancies by predicting how objects move from one frame to the next. In addition, advanced rate control and filtering techniques help balance bit rate with perceived video quality. The most popular video compression standards and codecs include H.264/Moving Picture Experts Group (MPEG)-4 Advanced Video Coding (AVC), H.265/High Efficiency Video Coding (HEVC), Video Processor 9 (VP9), and AOMedia Video 1 (AV1).

Besides appliaction of filtering techniques and transforms typical for intra-frame processing, inter-frame techniques might be particulary interesting in the domain of PGs. Electrical signals can also be treated as a sequence of values, especially with a constant sampling frequency. That property might be exploited by applying methods such as differential encoding that focus on difference between current and previous data points, instead of absolute value of the function.

4.2.4 Audio-specific data compression algorithms

Audio data compression takes advantage of both temporal and spectral redundancies present in sound waves. Consecutive audio samples often exhibit correlation over time, allowing algorithms to store changes between samples rather than full representations of every point. In the frequency domain, many techniques rely on transforms such as the Modified Discrete Cosine Transform (MDCT) to separate signals into bands that can be quantized differently. These compression schemes also incorporate psychoacoustic models, which analyze the limits of human hearing and mask frequencies that are less perceptible, thus reducing bit usage without significantly affecting perceived quality. Entropy coding, such as Huffman or arithmetic coding, is then used to encode more frequent sound patterns with fewer bits. By combining these measures, audio codecs balance fidelity and efficiency, making formats such as MPEG-1 Audio Layer III or MPEG-2 Audio Layer III (MP3), Advanced Audio Coding (AAC), Free Lossless Audio Codec (FLAC), and Opus widely popular for streaming and storage.

The fundamental methods behind audio compression can also be applied to electrical signals of various types. Similarly to how audio data leverages correlations in time and frequency, instrumentation or sensor signals often contain predictable patterns over time, making them prime candidates for transform-based or predictive coding. Electical signal compression can also leverage the approach of selecting the most important frequencies in the signal and discarding the rest, since not all frequencies are equally needed for every purpose. For example higher order harmonics can be ignored while analyzing signal for transients by protection system.

4.3 Data compression algorithms used in Smart Grids

Table 4.1: Examples of DC methods used in SG

Ref.	Year	Algorithms and methods	Use case	Comment
[35]	2010	WT, spline interpolation	Impulsive transient, oscillatory transient and voltage sag events	A guideline on selecting appropriate wavelet transform parameters based specific event. Paper indicates that wavelet transform of certain events in PG is more efficient if it has been selected specifically for one type of event. An examplary use case can be design of algorithm used for compression of short circuit or islanding related transients.
[4]	2011	Wavelet based	WT applied to signals in 39-bus system to reduce amount of data needed for describing events and disturbances	Research focused on non-periodic events data
[5]	2012	Integer lifting WT, Adaptive thresholding, Huffman coding	Communication of PS data in RT measurement systems	Authors separate stationary and non-stationary components of the signal and process them independently for better efficiency
[36]	2012	Principal Component Analysis (PCA), Data Sub-selection, Predictive Algorithm	RT measurement architecture for PG	Authors propose an algorithm based on prediction of the next measurement and encoding of the difference between predicted and actual value, reducing data amount by 40-50% for Phasor Measurement Unit (PMU) data. Authors also describe connection between all parts of Smart Metering architecture, including DC.

Continued on next page

Continuation of Table 4.1

Ref.	Year	Algorithms and methods	Use case	Comment
[37]	2014	Lempel-Ziv, Delta-modulation Huffman Coding, JPEG2000, various DWTs, Wavelet Packet, Damped Sinusoids Modeling	Simulated PS, comparison mainly focused on harmonics, sags and swells, inter-harmonics, transients	Authors compare lossy and lossless methods to reduce amount of data needed to monitor simulated PG
[38]	2014	Proprietary algorithm, based on Newton-Raphson algorithm	Authors prove that phasor data of complex systems can be compressed with efficiency up to 50%	Described method can be used for RT control of SG
[39]	2014	Normalization, differential coding, variable length coding, code word concatenation, and entropy coding	Massachusetts Institute of Technology (MIT) Reference Energy Disaggregation Data Set (REDD) and TU Darmstadt (TUD) data sets	proposed algorithm has very low complexity, meaning it is perfect for systems with limited resources, that are frequently used in PGs
[40, 41]	2013, 2015	Wavelet and zerotree based	PMUs and general PSs data	Authors underline the usage of their algorithm not only to compress data but also to denoise reconstructed signal
[42]	2015	Joint Data Compression and Encryption (JICE), Machine Learning (ML)-based	Measuring signal in one point, authors claim that method is able to support 144 measurement point to maintain distortion not larger that 5%	Authors present an approach of combining data security and data encryption, which reduces computation time, it is worth noticing that these two data properties are corelated
[43]	2015	Exception Compression (EC) with Swing Door Trending (SDT)	PMUs of a hydropower plant	Solution was tested and proved to show disturbances in RT for Wide-area Measurement System (WAMS), which makes it usefull for low-latency applications in SG

Continued on next page

Continuation of Table 4.1

Ref.	Year	Algorithms and methods	Use case	Comment
[44]	2015	Piecewise regression, RLE, Huffman coding, Differential coding	Among others REDD, Office Dataset (OD), Smart Home Dataset (SHD), Campus Dataset (CD) and other	Presented method has been implemented in database management system, which led to better data storage and indexing
[45]	2015	Piecewise linear approximation for time series data with maximum error guarantees	Streaming of time-series data	Authors present an algorithm that can guarantee maximum error for approximation of each datapoint in time-series. Algorithm is capable of reducing amount of data by 15%. Algorithm is designed to work with streaming data, which is a beneficial property for RT systems.
[6]	2016	Differential encoding, Variable-length integer representation, Removal of redundant information	PQ monitoring, fault recorders, and PMUs	Approach useful mainly in WAMS
[46]	2016	Spatial and temporal redundancies in PMU data, PCA, DCT, statistical change detection	Solution created for storing extensive amounts of PMU data, however can be adapted to encode PMU data exchanged in IoT systems	Method validated against WAMS
[47]	2016	K-Singular Value Decomposition (K-SVD)	SM measurements dataset of 100 households over 100 days	Authors propose decomposition of signal into several usage patterns for better compression, this approach migh be useful for looking for typical patterns for specific application

Continued on next page

Continuation of Table 4.1

Ref.	Year	Algorithms and methods	Use case	Comment
[48]	2016	Dual Tree Complex Wavelet Transform (DTCWT), RLE	Simulation of voltage sags, authors claim that method is useful also for other types of disturbances	Method based on decomposition of signal into real and imaginary part and computing DWT separately on both of them in order to achieve better performance
[49]	2017	Singular Value Decomposition (SVD), DWT, bzip2, LZMA	Aggregartion of daily measurements	Authors discuss rate of compression ratio to the loss of information for popular algorithms
[50]	2017	Adapted eXtended Data Representation (A-XDR), Lempel Ziv-Markov Chain-Huffman (LZMH), and Differential + Entropy/Golomb/Arithmetic (DEGA) coding	Voltage and current measurements of household-consumers from REDD and UK recording Domestic Appliance-Level Electricity (UK-DALE) data sets	Authors focus on loseless algorithms, which is not a popular approach, but useful in systems where potential errors migh have great severity, and increasing load on communication will be justified
[51]	2017	PCA, DCT	Real synchrophasor data	Algorithm pre-qualifies data prior to compression and selects compression method based on RT requirements
[52]	2017	K-SVD sparse representation technique	Real measurements of PQ for residential cusotmers, small and medium enterprises	Authors present a method leveraging the content of partial usage patterns in PG signals, underlining the fact that decomposition of a signal to its components will result in better DC performance

Continued on next page

Continuation of Table 4.1

Ref.	Year	Algorithms and methods	Use case	Comment
[3]	2018	Apple Lossless Audio Codec (ALAC), MPEG-4 Audio Lossless Coding (ALS), Monkey's Audio (APE), FLAC, TrueAudio, LZMA, Delfate, Prediction by Partial Matching variant d (PPMd), bzip2, Gzip	Overview and evaluation of general archiving and video specific compression algorithms using open source datasets - UK-DALE, MIT-REDD, EDR	Authors focus on exploiting quasi-periodic behaviour of signals in PGs
[53]	2018	Huffman coding	Sample graphs with varying sizes	Authors use an approach to divide information based on its frequency of occurence, using less data for more frequent events. A solution based on reduction of data needed to encode more frequent (or, in RT control systems, more severe) events can be beneficial for PGs functioning in a repetetive way.
[54]	2019	Compressed Sensing (CS), Linear Feedback Shift Registers (LFSRs)	Large scale Advanced Metering Infrastructure (AMI)	Authors presnt a way to compress the data and authenticate it at the same time, which is a very beneficial approach to SM design, since it will reduce computation needed for efficient communication, while addressing one of the most important CyberSecurity threats in that field
[55]	2019	Proprietary algorithm - Group Method of Data Handling	Prediction of PS variables verified on the example of Belgian PG	Paper contributes presenting framework for PS data prediction, which is being used to compress data in systems with repetitive character of operation

Continued on next page

Continuation of Table 4.1

Ref.	Year	Algorithms and methods	Use case	Comment
[56]	2020	Adaptive Trimmed Huffman (ATH), Adaptive Markov Chain Huffman Coding (AMCH), LZMA, LZMH, Run-Length Encoding with a K-Precision (K-RLE)	Communication of power consumption data of distributed metering system using MQ Telemetry Transport (MQTT) over Institute of Electrical and Electronics Engineers (IEEE) 802.11	Authors highly focus on reducing transmission rate of communication architecture and unification of all it's elements
[57]	2020	CS	Cryptographical attacks on compressed signal	Authors present successfull attacks resulting in deciphering information encrypted by CS method. Paper is underlining a significant influence of cybersecurity properties on compression algorithms
[58]	2020	Cross correlation exploitation, Adaptive Multivariate Data Compression (AMDC)	Real smart metering setup	Algorithm shows a great performance considering time of execution and amount of preserved information, making it promising for RT application
[59]	2020	Concept of generalized deduplication, lossless scheme with random access	Time-series data	Authors present a concept for compression a time-series data in the domain of PG. Main contribution is ability to random access desired part of data without the need to unpack all of the packet. This approach is very beneficial to swift communication of event data, since only samples describing the event can be decoded while all other, less critical data might stay compressed. A promising approach for improving efficiency of RT communication.

Continued on next page

Continuation of Table 4.1

Ref.	Year	Algorithms and methods	Use case	Comment
[60]	2020	Adaptive Multi-Model Middle-Out (AMMMO) framework, using reinforcement learning, including transform primitives (e.g., delta, xor) and encoding primitives (offset coding, bitmask coding)	IoT sensor data, other types of business data	Authors propose processing of the data on two levels - which offers a possibility to use different algorithm to encode any given data point. An approach that is useful for selecting crucial data for RT control, while saving on computation and communication resources on data gathered for analytical purposes.
[61]	2020	Delta encoding, splitting, zigzag encoding, bit conversion, Window-Bit-Block Compression, deduplication	RT data processing in IoT systems	Authors propose indexing of compressed data with timestamps. This approach is usefull for decoding only data important for the task and integration of SG system components that were developed independently.
[62]	2020	Autoencoder based on NN	Power consumption measurements, validated using Electricity Consumption and Occupancy (ECO), Dutch Residential Energy Dataset (DRED), UK-DALE and REDD	Authors propose NN based compression method. Important take-away is the fact that model fed with the data from one location can operate in another location, since pattern of power consumption tend to be similar.
[63]	2021	Lossless Coding considering Precision - consisting of: differential coding, XOR coding, and variable length coding	Time-critical communication, validated on low voltage consumer grid	Method fits well time series with long steady periods

Continued on next page

Continuation of Table 4.1

Ref.	Year	Algorithms and methods	Use case	Comment
[64]	2021	RLE, Huffman coding, Differential coding	SG AMI networks, validated on REDD	Light-weight lossless compression algorithm designed for AMI networks, connection of various lossless methods
[65]	2021	Gradient compression mechanism based on Top-k selection	Industrial IoT systems, proposed model validated on real-life IoT system data	Authors prove that selecting only the gradients with the largest absolute values (Top-k selection), it exploits the observation that a large portion of the gradients can be pruned without significantly impacting the accuracy of the model.
[66]	2022	Wavelet compression with hybrid thresholding	Communication of events over IEC61850-compliant communication	Hybrid thresholding method introduced by authors significanltly improves CR
[67]	2022	WT based algorithms	Comparison of various wavelets using simulated signals (sags, swells, interruptions, notching)	A deep dive on WT and guide of selecting right parameters for it
[68]	2022	Multi-layer Perceptron (MLP), integrated NN model, WT, PCA, SVD, and dimension reduction methods	compression of data from Almanac of Minutely Power dataset (AMPds)	authors describe the architecture of an edge computing system, which is gaining popularity in PG, proposed solution shall help balancing loads in such system
[69]	2022	Fuzzy Transform (FT)	Real PS signals	A novel approach of using fuzzy logic to compress data, described performance of an algorithm is promising, however many PG applications require usage of deterministic algorithms for time-critical data, which might be a challange for the development of methods based on fuzzy logic

Continued on next page

Continuation of Table 4.1

Ref.	Year	Algorithms and methods	Use case	Comment
[70]	2022	8-bit Unicode Transformation Format (UTF-8) encoding, Similarity matching	Spectrum data in the field of radio management	Spectrum encoding techinques presented in paper will also be usefull for compressing frequency-related events in SG
[71]	2022	Long Short-Term Memory (LSTM) model	Simulation of long range edge computing system for SG	Authors evaluate the energy usage of a proposed system, which is an important parameter for the scalability, but is often ommited during evaluation
[72]	2023	WT, multi-resolution analysis, and thresholding techniques	Data transmission in SG over IEC61850 Protocol	Combined binary regression wavelet-surrogate tree and a hybrid thresholding method were introduced by authors as a part of the whole SG control system.
[73]	2023	Divide-and-conquer, K-SVD	REDD, Residential and utility datasets	Authors present an approach based on pre-analizing the signal and using different compression methods for steady-state, fluctution and event parts of the time-series. This approach shall be used in future IoT systems in PGs in order to balance the CR and significance of information provided by signal in a given moment.

The authors of [3] evaluated 14 audio and general-purpose compression schemes on high-resolution PG time-series data. They found The True Audio (TTA) 2.3 and MPEG-4 Part 14 (MP4) ALS most effective for energy waveform compression, while FLAC 1.3.2 allows fast decompression but with performance variations. PPMd achieves a high CR without requiring Resource Interchange File Format (RIFF)-Waveform Audio File Format (WAV) conversion. General-purpose algorithms like DEFLATE, bzip2, and Gzip were unsuitable for time-series data 4.1. Longer signal representations help exploit the quasiperiodic behavior of PG. Future research may explore phasor-like data representation to leverage inter-channel similarities.

The artilce [4] proposes a wavelet-based approach 4.2 for SG time-series data compression to address communication congestion. The study identifies Order 2 Daubechies wavelet and scale 5 as the best choice for disturbance signals. Using IEEE New England 39-bus system simulations, the method effectively compresses and denoises data while preserving transients. The results highlight wavelet-based MRA as a nonredundant, noise-suppressing, and efficient compression method.

The authors of [49] present a MATLAB-implemented data compression methodology tested on real UK utility company data. The study compares SVD and DWT, finding that SVD offers a better trade-off at high

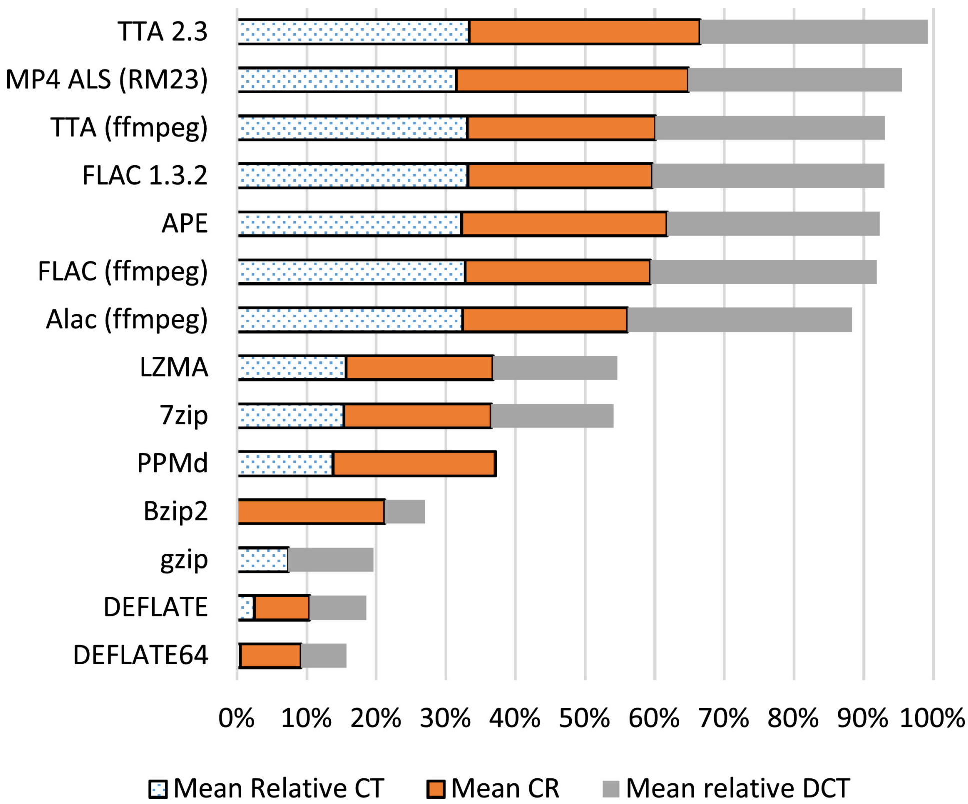


Figure 4.1: Ranking of codecs based on normalized relative compression time, decompression time, and CR results [3].

CR. Combining SVD and DWT further improves results, balancing data reduction and information retention. The computational efficiency of SVD was discussed, but the encoding complexity was not extensively analyzed.

[37] explores SG signal compression methods, summarizing various approaches and their evaluation metrics. The paper identifies the lack of standardized evaluation criteria for lossy compression as a key issue. The challenges of modern PSs—renewable integration, regulatory constraints, and automation—necessitate reliable data storage and transmission. Compression plays a crucial role in mitigating storage and bandwidth constraints while supporting fast fault detection and system maintenance.

Compression for electric signals differs from media data, prioritizing minimal distortion for fault analysis. Future research should focus on refining evaluation parameters and exploring cognitive and cooperative compression techniques.

The paper [38] investigates RT SG state estimation using compressed power measurements. The study highlights correlated voltage phasors in distributed generation, leveraging CS for state estimation. Two methods, indirect and direct, are proposed. The indirect method reconstructs injected power values before estimating voltage phasors via Newton-Raphson iteration, while the direct method incorporates compressed measurements into the iteration process.

Both methods achieve accurate state estimation with 50% compressed measurements, reducing data storage and communication overhead. Performance is evaluated using Integrated Normalized Absolute Error

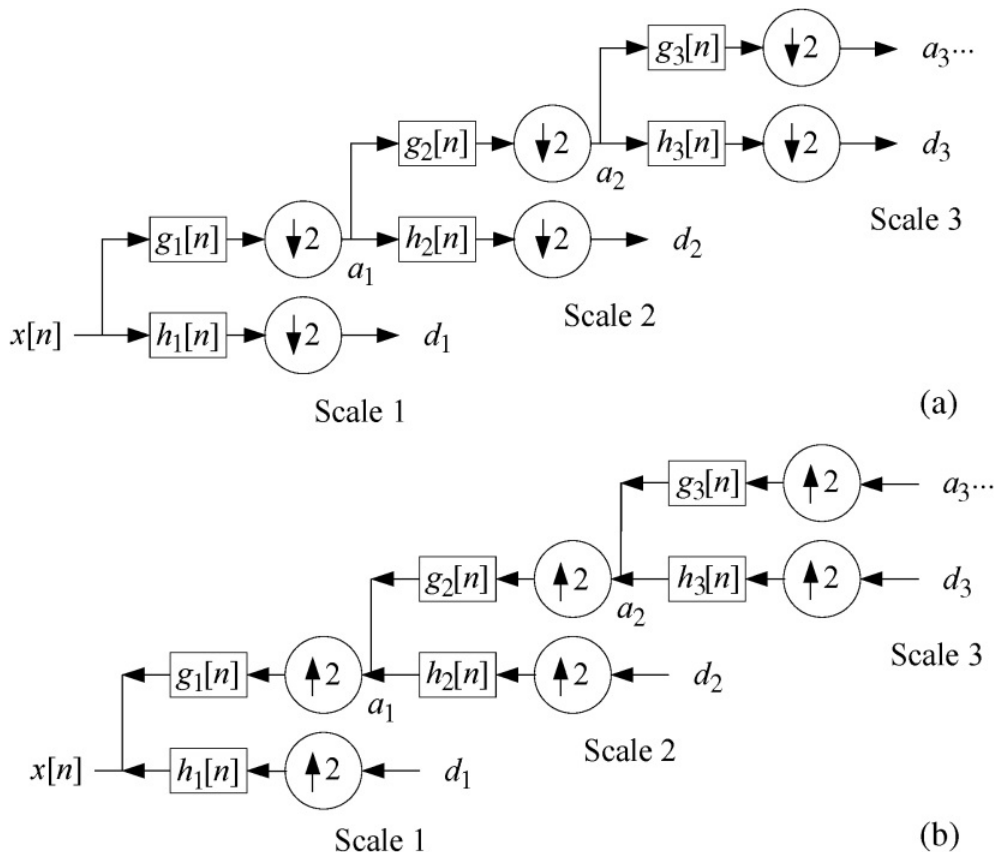


Figure 4.2: Procedure of WT-based MRA: (a) Decomposition, (b) Reconstruction [4].

(INAE) and Mean Integrated Absolute Error (MIAE). Future work may address topology changes, noise, and bad data in estimation processes.

[5] presents a RT electrical waveform compression algorithm for PQ monitoring. The algorithm achieves high compression while preserving key waveform features, operating on integer data to simplify computations on low-cost Digital Signal Processor (DSP)s.

The algorithm separates stationary and non-stationary waveform components. A lobe slope detection algorithm extracts predictable stationary elements, while Integer Lifting Wavelet Transform (ILWT) compresses nonstationary elements 4.3. Adaptive thresholding preserves significant wavelet coefficients and Huffman coding reduces the data entropy. Tested on a prototype SM using ZigBee communication, the method ensures high compression, low cost, and efficient storage.

Articles [40, 41] propose an Embedded Zerotree Wavelet Transform (EZWT) for SG DC and denoising. EZWT, which requires no pretraining, encodes significant coefficients first, with less important ones stored in a zerotree structure. The algorithm applies a biorthogonal wavelet to extract key coefficients and reconstructs signals with high fidelity. Evaluated on electrical and PMU signals, EZWT achieves a CR of 69.04% with a Normalized Root Mean Square Error (NRMSE) of 1.6110^{-2}, outperforming conventional WT methods.

The algorithm provides variable CRs, allowing for adaptability to transmission rate and storage constraints. By eliminating nonsignificant coefficients, it also enhances Signal-to-Noise Ratio (SNR). Simu-

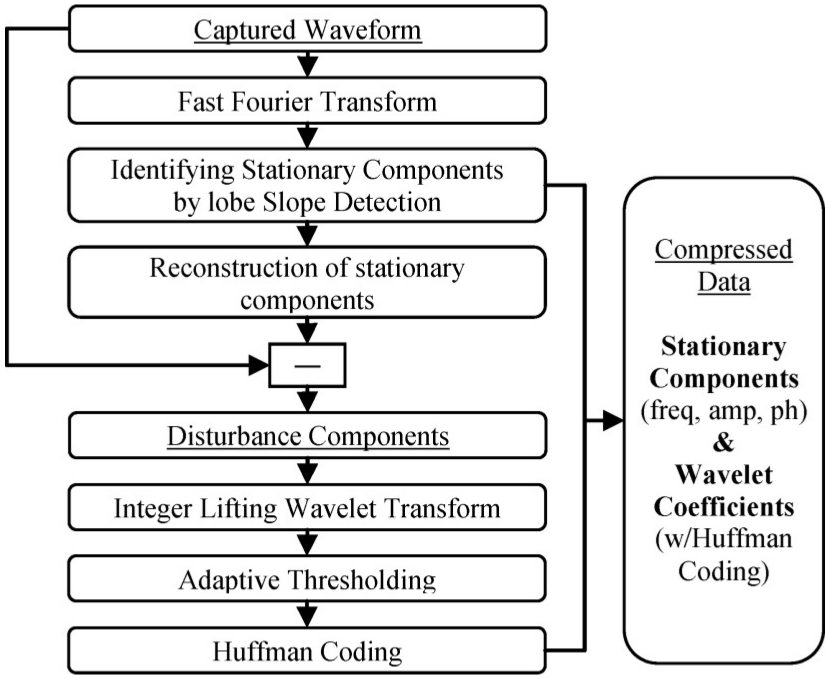


Figure 4.3: Data flow of the RT PQ monitoring compression algorithm [5].

lated noise tests confirm substantial improvements in compression and denoising, making EZWT suitable for time-critical embedded SG systems.

The paper [46] presents a PMU data compression method leveraging spatial and temporal redundancies. Using PCA and DCT, the method efficiently compresses WAMS data while maintaining signal integrity. A statistical change detection technique dynamically adjusts compression parameters based on detected disturbances.

The algorithm applies PCA to minimize spatial redundancy among PMU data streams and DCT for temporal redundancy reduction. Performance is measured using CR, Root Mean Square Error (RMSE), and Maximum Absolute Deviation Error (MADE). Field data from four U.S. PMUs demonstrate that PCA significantly improves compression performance as the number of PMUs increases. Compared to DWT, DCT achieves superior compression, preserving critical disturbance information with reduced storage demands.

[63] introduces the Lossless Coding considering Precision (LCP) method for RT SG DC. The model-free method uses differential, XOR, and variable-length coding to transmit encoded data without prior knowledge of time series dynamics. LCP efficiently compresses steady-period high-resolution time series by encoding each value based on the previous data point.

Tested on REDD, Labelled hIgh Frequency daTaset for Electricity Disaggregation (LIFTED), AMPds, and PMU datasets, LCP achieves superior CRs and lower latency compared to Resumable Data Compression (RDC) and DEFLATE. It adapts to data patterns and precision requirements, following the American National Standards Institute (ANSI) C12.1 standards for voltage and current signals. With $O(n)$ time complexity and constant space complexity, LCP enables RT data transmission while outperforming conventional compression techniques.

The authors of [66] focus on DC within the International Electrotechnical Commission (IEC)61850 SG communication protocol. A wavelet-based compression method applies predictor importance to determine relevant detail levels. A hybrid thresholding technique, combining hard and soft thresholding, improves compression efficiency.

The method integrates wavelets with regression trees, ranking wavelet details based on predictor importance derived from MSE. Evaluated via RT simulation in IEC61850 systems, the approach significantly reduces message sizes while ensuring high-quality signal reconstruction.

The article [56] examines adaptive data compression for SG WSNs. The study assesses two-way communication infrastructure using IEEE 802.11 and MQTT, evaluating network performance in scenarios with and without compression. Implementing an adaptive mechanism reduces congestion, latency, and packet loss, enhancing RT decision-making.

A bidirectional model that incorporates edge, fog and cloud computing layers reduces data flow from SMs to the PG operator. Various compression techniques are analyzed for preventing network overload. The results highlight improved efficiency and stability in SG communication networks.

The authors of [6] propose a RT Sampled Value (SV) data compression method based on the IEC 61869-9 recommendations. The approach reduces data size by more than 50%, improves encoding/decoding speed, and minimizes network jitter 4.4. By reducing SV transmission, the method supports low-latency applications such as PMUs and wide-area protection.

The algorithm applies differential encoding to remove redundant data, reducing Ethernet transmission times. Compared to conventional SV encoding, the approach achieves a 60-byte reduction per Application Service Data Units (ASDU), significantly decreasing network load.

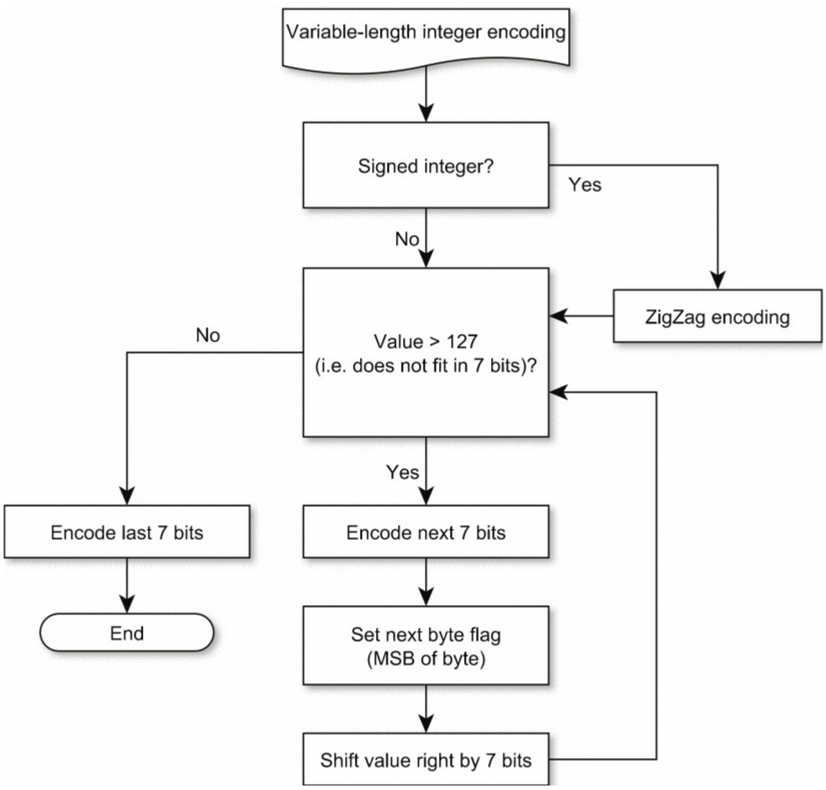


Figure 4.4: Variable-length integer encoding [6].

This compression method improves SG bandwidth efficiency, improving RT PQ monitoring, fault detection, and protection systems. It provides a cost-effective solution to improve communication in critical grid applications.

The article [42] presents JICE, a CS-based approach for wireless energy auditing networks. It compresses and encrypts data simultaneously, improving transmission and storage while adapting to power pattern variations using ML. Implemented on a smart plug platform, JICE increases the meter support by 50% while keeping the distortion below 5%, outperforming other methods in data delivery and accuracy.

Researchers in [58] introduce AMDC for smart IoT measurement, using correlations between power, current, voltage, and frequency to reduce redundancy. It applies Autoregressive Integrated Moving Average (ARIMA) for temporal compression and a multivariate normal distribution for residual modeling. The algorithm is designed for RT transmission, balancing bandwidth savings, and reconstruction accuracy. Applied to electricity billing, AMDC preserves measurement integrity while minimizing data overhead.

The authors of [67] focus on compression of PQ events using the WT. The study evaluates 80 wavelet functions and different decomposition levels in 12 PQ disturbances, identifying symlet 4 with three decomposition levels as the best combination. These findings refine wavelet-based PQ compression techniques and enhance signal storage efficiency.

Lossless compression for high-frequency voltage and current data is explored in [50]. The study evaluates A-XDR, LZMH and DEGA coding using MIT REDD and UK-DALE datasets, confirming that the feasibility of compression varies with the resolution and characteristics of the dataset. At 16 and 50 kHz sampling rates, compression proves practical, offering recommendations for RT SM data transmission.

A novel edge computing-based compression method is introduced in [68]. By integrating an MLP classifier, the approach improves data classification speed while maintaining at least 80% accuracy. Edge computing reduces cloud storage needs, minimizing transmission bandwidth and latency. This method offers an efficient solution for handling increasing grid data loads.

The authors of [47] explore K-SVD-based compression for SMs, decomposing load profiles into partial usage patterns. Compared to PCA and DWT, K-SVD achieves superior compression and pattern extraction. This method improves the analysis of energy consumption while preserving essential data features.

In [69], the FT is applied to the compression of data from the SG. It maps signals into a reduced-dimensional domain while preserving integrity. FT efficiently compresses SG data compared to traditional methods, making it well-suited for RT transmission.

A sparse representation approach for SM DC is proposed in [52]. Using K-SVD, the method extracts meaningful consumption patterns, enhancing the classification accuracy for residential and Small and Medium-sized Enterprises (SME) customers. Allows for a more precise analysis of the trends in electricity usage.

The study in [39] presents a five-step lossless compression method for load profile data. It exploits small consecutive value differences to achieve efficient storage with low complexity. The algorithm is particularly effective for handling large-scale SM data.

RT data compression for WAMS is addressed in [43]. The proposed method integrates EC and the Straight-line Deviation Technique, improving compression ratios while preserving signal integrity. This technique significantly reduces WAMSs data traffic.

A unified approach to compression and authentication for SM is proposed in [54]. Using CS, it secures data transmission via a measurement matrix-based authentication approach. The method improves efficiency and security in AMI networks.

The article [51] introduces Intelligent Synchrophasor Data Real-Time Compression Framework for WAMS (ISAAC), a RT WAMS compression framework that integrates PCA and DCT. A statistical change detection mechanism ensures adaptive compression. The authors claim that this technique significantly improves data storage and transmission efficiency.

Another framework for RT data analysis in SG is explored in [36]. The system integrates PCA-based data reduction and predictive compression, achieving 40-50% compression for PMU data. By minimizing data transmission delays, according to the authors, it enhances grid automation.

The authors of [72] propose a binary regression wavelet-surrogate tree with hybrid thresholding to reduce SG message exchanges. Their approach enhances communication efficiency while preserving signal fidelity, improving RT system performance.

Spectrum data compression using UTF-8 character encoding and similarity matching is explored in [70]. The method reduces redundant spectral data, lowering storage requirements while maintaining accurate signal analysis.

A compression strategy for SG AMI networks is introduced in [64]. By combining RLE, Huffman coding, and differential coding, the method achieves up to 10% compression ratio while preserving data accuracy.

The authors of [44] present a time series compression technique based on piecewise regression. Their approach ensures that user-defined maximum deviations are not exceeded, making it ideal for high-precision forecasting and analytics.

An LSTM-based Compression-Decompression model for smart metering in Long Range (LoRa) networks is proposed in [71]. It effectively reduces data size while maintaining accuracy, improving energy efficiency in SG communications.

The security vulnerabilities in CS-based encryption for IoT devices are examined in [57]. The study highlights risks associated with ciphertext-only attacks, revealing critical weaknesses in existing CS-based encryption schemes.

Renewable energy forecasting using Group Method of Data Handling (GMDH) NNs is explored in [55]. Although not directly a compression technique, predictive modeling supports efficient data handling by reducing redundant information in SG communications.

A lossless compression scheme for time-series data is introduced in [59]. The approach enables low-cost random access without full decompression, using generalized deduplication to enhance compression while preserving efficient data retrieval.

The article [35] applies WT and spline interpolation for PQ disturbance compression. By optimizing Daubechies scaling functions at four decomposition levels, the approach achieves high compression ratios while controlling signal distortion.

The authors of [48] propose a method combining DTCWT and RLE for PQ monitoring. Their technique efficiently compresses and reconstructs power disturbances, improving SG signal analysis.

The article [65] addresses the challenge of anomaly detection in Industrial IoT by introducing a deep anomaly detection framework based on Federated Learning (FL). This approach enables edge devices to collaboratively train a detection model while maintaining privacy and reducing communication overhead.

The proposed framework consists of three components. First, FL allows decentralized devices to improve anomaly detection through shared training. Second, an attention-based Convolutional Neural Network (CNN)-LSTM model improves accuracy by capturing fine-grained time series features while mitigating memory loss. Finally, a Top-k gradient compression mechanism reduces communication overhead by 50%, accelerating anomaly detection.

Extensive experiments on real-world datasets (space shuttle, power demand, engine) validate the framework's effectiveness, demonstrating accurate detection with minimal communication costs. The study also highlights future research directions, including privacy-enhanced FL frameworks and robust models for diverse IoT environments.

The authors of [45] examine the Piecewise Linear Approximation (PLA) for time series data with maximum error guarantees. The goal is to construct an efficient linear function approximation within a set error bound. A novel algorithm processes streaming data in RT, decreasing representation size while preserving accuracy.

Compared to previous methods, this approach adaptively uses a mix of joint and disjoint knots, minimizing storage while maintaining precision. Experimental results show a 15% reduction in representation size over existing techniques, making it highly effective for resource-constrained environments.

The study in [61] explores storage reduction for RT IoT data. According to authors, existing compression techniques largely focus on integer values, neglecting retrieval efficiency. To address this, the authors introduce a lossless compression framework for floating-point time-series data that enables retrieval without full decompression.

The compression process includes delta encoding, zigzag encoding, and bit conversion, ensuring full reversibility. A novel indexing method, attaching time-stamps to compressed data, further enhances retrieval speed. Experiments demonstrate a 97.88% reduction in storage space, surpassing conventional techniques in efficiency.

A divide-and-conquer compression method for SM data is introduced in [73]. The method classifies data into three segments, event, fluctuation, and steady state, each processed using specialized techniques.

Event segments, which capture appliance state transitions, retain original values due to their complexity. Fluctuation segments undergo CS-based compression, allowing pre-reconstruction on edge devices before transmission. Steady-state segments, featuring smooth power variations, use Symbolic Aggregation Approximation (SAX) for data reduction.

Comparative experiments across datasets from North America and China confirm higher reconstruction accuracy and compression efficiency than existing methods. The approach facilitates scalable SG data transmission and analytics.

The authors of [62] propose NeuralCompression, an Autoencoder (AE)-based method for compressing high-frequency SG data. Unlike traditional techniques, AE learns nonlinear transformations to improve compression ratios while maintaining accuracy.

Tested on ECO, DRED, UK-DALE, and REDD datasets, the AE model outperforms CS in both compression efficiency and computational speed. Transfer learning experiments confirm the model's adaptability across different geographical datasets, making it a robust solution for large-scale SG data management.

A Huffman coding-based compression technique for IoT devices is presented in [53]. The exponential increase in IoT-generated data requires efficient storage and transmission. The proposed method compresses graph-based data structures, reducing memory usage while enabling efficient analytics.

Using adjacency matrices and identifying common 32-bit patterns, the algorithm achieves compression of up to 80%. Compared to previous graph compression methods, it is better suited for resource-constrained IoT environments, ensuring seamless data transmission.

The article [60] explores a two-level data compression strategy for time-series databases. Many existing methods focus on global patterns, leading to inefficiencies in handling local variations. To address this, the AMMMO framework dynamically selects compression schemes for each data point based on reinforcement learning.

The framework consists of major mode selection (evaluating data segments) and sub-mode selection (choosing the compression scheme). Experimental results indicate up to a 120% improvement in compression ratios compared to traditional techniques, making it a promising solution for diverse time-series data.

4.3.1 Conclussion

The rise of articles covering the topic of DC techniques in SG indicates the importance of reducing the amount of data transmitted and processed. Some authors ([71], [68]) claim that the future of SG (as a subset of IoT systems) is edge computing. The architecture of edge computing systems (4.5) requires the balance of loads between locally available processing units that are expensive but have low latency, and remotely available cloud computing centers that are cheaper to use, but communication with them requires more time. Some systems might also feature an interim step between local Central Processing Unit (CPU)s and remote cloud computing, that will be able to operate as a local control center, connecting few IoT devices, that would offer less computing power that cloud, but more than CPU of embedded system and less latency than cloud but more than local processing unit.

The authors try to improve both computing paradigms, local and remote processing, with DC. Some of the crucial aspects mentioned across the papers are as follows.

4.3.2 Criteria for Evaluation of Data Compression Methods

Evaluation of data compression methods is a complex task, as it is highly dependent on the specific application. The ability to tolerate some loss of data during compression can be a debatable matter. According

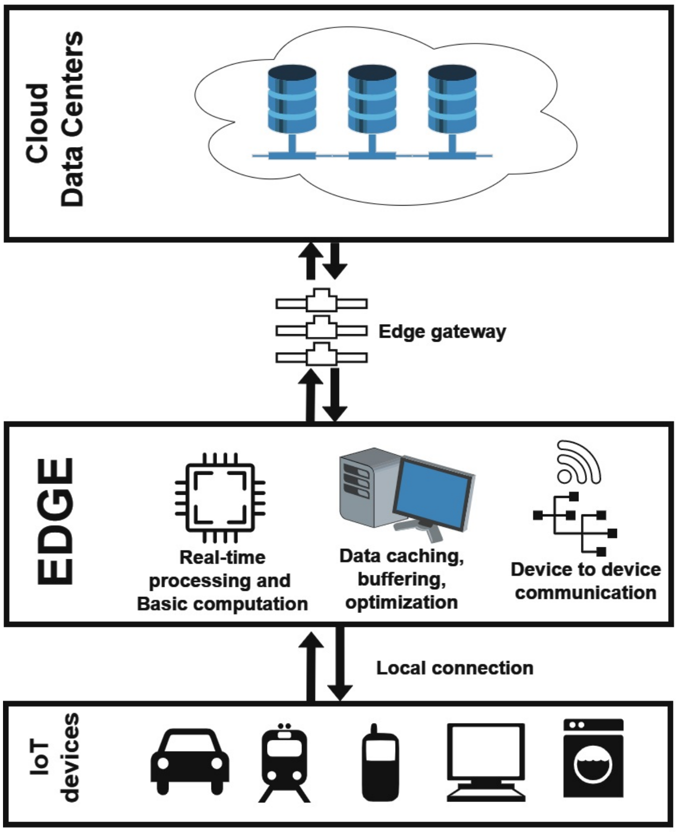


Figure 4.5: Architecture of edge computing system.

to the literature, the criteria for evaluating DC techniques vary, largely depending on whether the data is intended for use in RT systems.

Compression Ratio

The compression ratio is a fundamental metric used to evaluate the effectiveness of DC algorithms. It quantifies the level of compression achieved by comparing the original size of the data to the size of the compressed data. The compression ratio is typically expressed as a ratio or percentage (%). It is one of the primary indicators of the benefit gained from the compression process and is directly related to the amount of memory savings achieved.

- **Original Data Size**: the size of the data before compression, usually measured in bytes (B), kilobytes (KB), megabytes (MB), or other appropriate units.
- **Compressed Data Size**: the size of the data after compression, measured in the same units (B, KB, MB, etc.).

In cases where additional data such as dictionaries, headers, or index structures must be included with the compressed data, these overheads should be accounted for when calculating the effective compression ratio, as shown in Equation 4.5:

$$CR = \frac{data_{\text{additional}} + \sum_{i=1}^{n} data_{\text{piece}i}}{data_{\text{original}}} \times 100\% \tag{4.5}$$

where CR is the compression ratio (percentage, %), $data_{\text{additional}}$ is the size of auxiliary structures in bytes (B), $data_{\text{piece}i}$ are the sizes of the i-th compressed data segments in bytes (B), and $data_{\text{original}}$ is the size of the original uncompressed data in bytes (B).

The memory savings MS (percentage, %) is the amount of memory freed during the DC process. It can be calculated by subtracting the CR from 100 %, as presented in Equation 4.6:

$$MS = 100\% - CR \tag{4.6}$$

The goal of the compression process is to achieve the CR as small as possible, while still preserving critical information necessary for the intended application. It is important to note that the achievable CR depends heavily on the nature of the data and the characteristics of the compression algorithm. Therefore, different algorithms may yield significantly different performance across various types of data.

In some publications, the Compression Factor (CF) is used instead of the CR. It is defined as the inverse of the CR:

$$CF = \frac{data_{\text{original}}}{data_{\text{compressed}}} \tag{4.7}$$

where CF is the compression factor (dimensionless ratio), $data_{\text{original}}$ is the size of the original data in bytes (B), and $data_{\text{compressed}}$ is the size of the compressed data in bytes (B). For example, a value of $CF = 2$ means that the compressed bitstream is twice as small as the original data. The use of CF can sometimes simplify interpretation, especially in fields like information theory and codec evaluations, where it directly reflects the relative reduction.

Computation complexity

Computational complexity is an essential consideration for DC algorithms implemented in RT scenarios due to several reasons:

1. Real-time Processing Demands: The algorithms must process the incoming data quickly to ensure that the compressed data is available for transmission or storage without noticeable delays. High computational complexity can lead to significant processing times, especially for large datasets. Not meeting the deadlines might render the compression method useless for RT application.

2. Data Throughput and Bandwidth Usage: RT applications often operate under specific data throughput and bandwidth constraints. Compression algorithms with low computational complexity can efficiently reduce the size of data, enabling faster transmission or storage. This is an important issue taking into account the limited bandwith in SG communication systems.

3. Energy Efficiency: Many RT scenarios involve devices with limited power resources, such as mobile devices or IoT sensors. High computational complexity can drain the device's battery quickly or require more expensive power supply and consume more energy.

4. Scalability: RT applications often deal with varying data sizes and processing loads. Adaptive compression algorithms are more scalable as they can handle different data sizes without a significant increase in processing time. This ability is crucial for WAMS, especially in the matter of commercialization.

5. Latency Reduction: In time-critical applications like PG control, low-latency DC is essential to avoid delayed responses, which can create disturbances that might be even more severe.

6. Embedded Systems: In RT embedded systems, such as those found in PG, automotive, aerospace, or medical devices, computational resources are often limited. DC algorithms with low computational complexity are more suitable for implementation in such environments. This property is especially important for two of the most popular architectures present in PG - edge computing architecture where embedded systems do part of the processing and distributed computing (local architecture), where they are responsible for all of the processing.

7. Adaptability to Changing Data: Some RT scenarios involve continuously changing data streams. This situation may not be relevant for every PS, however there are systems with variable power usage, depending on time of the week, year or some events. Ability to adapt will also future-proofs the solutions by avoiding errors when the configuration of measurement system changes.

In RT applications, computational complexity is a crucial parameter of an algorithm. If computing the compressed pattern is too complex, the system may fail to meet RT requirements.

Table 4.2: Types of RT systems

Type of RT system	**Properties**
Hard	Miss of the deadline will result in failure of the system. Failure often results in large-scale loss or threat to human life or health. Data becomes useless after missing the deadline.
Firm	Not reacting within defined time period will cause loss, but can be tolerated if it is infrequent.
Soft	Delayed reaction can be tolerated, usually with some methods to recover after failure. Usually, soft real-time systems may tolerate small latency.

Most power systems will qualify as firm real-life systems 4.2, since they can handle lack of reaction from the controller by employing other ways such as different methods or protection, redundancy or simply ability to tolerate some amount of disturbances. Some of the most crucial systems (for example high voltage lines) might be qualified as hard real time systems due to a small tolerance for disturbance and severe consequences of a failure.

Size and Type of Data Input

DC algorithms process data in different sizes and formats, either as blocks or streams. Block-based algorithms operate on fixed-size data chunks, while stream-based algorithms process data continuously. The choice between these methods affects latency, efficiency, and adaptability, which are critical for RT PG applications.

Block-based Algorithms: These divide data into independent blocks, influencing compression efficiency and system performance.

- *Latency and Processing Time:* Larger blocks require more processing before compression, increasing latency. Minimizing delays is essential in RT PG applications.
- *Compression Ratio:* Bigger blocks often yield better compression but increase computational complexity, whereas smaller blocks process faster but may achieve lower compression ratios.
- *Adaptability:* Data patterns vary in PGs, and block size selection must balance efficiency and adaptability.

Stream-based Algorithms: These handle data continuously without fixed blocks, making them more flexible.

- *Dynamic Data Size:* They adjust to different data rates but may struggle with scaling if not designed properly.
- *Memory and Buffering:* Some buffering is required before compression, which can be a limitation in embedded systems.
- *Real-time Throughput:* Fast processing is necessary to prevent bottlenecks and ensure efficient data flow.

The choice between block- and stream-based compression depends on latency needs, memory constraints, adaptability, and throughput. Compatibility with encryption algorithms is also crucial in system design.

Integration with other systems

Some authors present an approach based on integration with other components of the system. An important issue is data indexing [61]. Indexing can be used to organize data between different components of the system. This property is not critical for efficient DC; however, it helps with further data handling and storage. Data indexation can also improve system relaibility and avoid errors in data reconstruction in the case of a delayed or missed packet. This property gives the system information about the continuity of frames.

Some authors also investigate the compliance of compressed data with certain communication protocols [6], [66]. Compliance with communication protocols guarantees that the compressed data maintain their integrity throughout the transmission process. It ensures that DC algorithms can be easily integrated into existing infrastructure and compatible with future upgrades. This standardization facilitates scalability and simplifies the integration of new compression solutions into the PG ecosystem. PGs are subject to various regulatory requirements and industry standards. Compliance with communication protocols can be a prerequisite for meeting these regulations. Using compression algorithms that adhere to communication standards helps PG operators demonstrate adherence to relevant guidelines and compliance frameworks.

Reliance on the repetitiveness of data

Repetitiveness of data in most cases is beneficial for the compression of data, since it allows the selection of frequent patterns and encoding them using data-consuming symbols. It is beneficial for both lossy and lossless compression methodologies; however, some methods may perform better for data with more statistical similarity. Most lossless methods are based on the detection of the same pattern in the data set, which is the basis for compression primitives such as RLE or Huffman Coding.

Most data compression methods apply some lossless compression algorithms during later steps (in most cases selecting RLE or Huffman Coding, often using Adaptive Huffman coding, which is well-suited for streaming data, as a last step), since they do not introduce any additional error and the benefit from further compression outweighs additional computation steps.

In general - repetetive data is easier to compress; however, the performance of some algorithms is more reliant on repetitiveness of data, so this property shall be taken into account during selection of compression algorithm for a certain application.

Signal reconstruction

Data integrity is paramount in PG, as accurate data is crucial to making informed decisions, maintaining grid stability, and ensuring the safety and reliability of the PS [74]. The error value quantifies the discrepancy between the original data and the data after compression and decompression [75]. A low error value indicates that the compression algorithm has effectively preserved the fidelity of the original data, reducing the risk of misinterpretations or erroneous decisions based on inaccurate data [76].

RT PG applications often involve automated decision-making processes that rely on sensor data and measurements. If compressed data introduces significant errors, it could lead to incorrect decisions, potentially jeopardizing the stability and security of the PG [77].

In addition, PGs use accurate data for grid stability and control. For example, PS control applications require precise data on voltage levels, frequency, and load conditions. High error values could lead to imprecise control actions, which could affect PQ and grid stability [78].

Accurate data are also crucial for fault detection and diagnosis in PG. Errors introduced during DC could obscure or distort critical information related to faults or anomalies, making it difficult to identify and address issues promptly [79].

Furthermore, PG are subject to various regulatory and reporting requirements. Accurate data representation is necessary to comply with these regulations and standards. High error values may raise concerns about the accuracy of the data and the compliance with industry guidelines [80].

DC algorithms aim to reduce the size of the data while preserving important information. The error value serves as a measure of the efficiency with which the algorithm achieves this goal. Algorithms with lower error values are more effective in compressing data while maintaining data integrity [81].

RT PG applications operate in dynamic and sometimes noisy environments. Compression algorithms must be robust and resistant to noise and disturbances. The error value reflects how well the algorithm can handle noisy data and preserve critical information amidst environmental fluctuations [82].

Ability to process compressed or partially decompressed data

Most of the algorithms require full decompression of the data prior to any further processing. There are methods to perform some calculations on compressed data. This approach, while not popular, is beneficial for reducing the time delay between the event and reaction. Main strategies used to speed up execucion of a control algorithm by feeding it compressed data are:

1. aggregation of data into numerical interval, which will result in loss of resolution, but migh be enough to examine properties such as rate of change
2. indexation of compressed data, which permits for unpacking only the data related to some event, while saving computation resources by not running additional calculations on samples registered prior to or after the event

Some authors take this property into account [59], however, most of the available algorithms do not introduce any additional steps to allow processing of encoded data.

4.4 Data compression and cybersecurity

DC plays a significant role in the efficiency and performance of data management systems. However, its influence on cybersecurity is complex and multifaceted. Although compression can improve security by reducing data exposure and facilitating encryption, it can also introduce vulnerabilities that compromise data integrity, confidentiality, and system robustness. Therefore, it is crucial to carefully select and implement compression algorithms considering their potential cybersecurity implications.

4.4.1 Data Compression and Data Integrity

Data integrity refers to the accuracy and consistency of data throughout its lifecycle. Compression can introduce vulnerabilities that compromise integrity, particularly with lossy compression, where data is irreversibly altered. For example, lossless compression methods ensure that the original data can be perfectly reconstructed, maintaining integrity [83]. However, lossy compression methods, which discard some data to achieve higher compression ratios, may lead to degradation in data quality, potentially affecting the integrity of sensitive information [84]. Lossy compression algorithms should be selected very carefully and should always be checked for compliance with security standards applicable for the system. Moreover, hacker may try to parameterize lossy compression component in a way that will not allow for the useful reconstruction of data.

4.4.2 Data Compression and Confidentiality

Confidentiality is the assurance that information is accessible only to authorized individuals. Compression can impact confidentiality in several ways. On the one hand, compressed data is more compact, making it easier to encrypt and less likely to be intercepted during transmission [85]. On the other hand, certain compression algorithms might inadvertently expose patterns that could be exploited by attackers. For instance, specific sequences in compressed data might reveal underlying information about the original data structure, potentially aiding cryptanalysis [86].

4.4.3 Data Compression and System Robustness

System robustness refers to the ability of a system to withstand and recover from adverse conditions, including cyber attacks. Compressed data can improve robustness by reducing the amount of data that must be transmitted, thus reducing the risk of data interception and loss [87]. However, the additional computational overhead required for the compression and decompression processes can introduce vulnerabilities. Attackers may exploit these processes, injecting malicious code during compression, or taking advantage of decompression flaws to execute attacks [88].

4.4.4 Case Studies and Applications

Several case studies highlight the practical implications of data compression in cybersecurity. For example, the use of compression in web traffic (for example, Hypertext Transfer Protocol (HTTP) compression) has been shown to improve performance while also reducing the attack surface by minimizing data exposure [89]. In contrast, vulnerabilities such as the Compression Ratio Info-leak Made Easy (CRIME) and Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext (BREACH) attacks demonstrate how compression techniques can be exploited to leak sensitive information from encrypted web traffic [90].

Chapter 5

Different approaches to data compression in electrical signals

Most signal compression systems are constructed using a combination of lossy and lossless compression algorithms. Over time, many approaches were developed to solve the problem of data redundancy in electrical signals. Some of them are exploiting the correlation of data within a signal, others try to find similarities between different measurement points. Most of the solutions described in the literature are based on one or more of the following methods: WT, phasor, CT, CS, SVD, predictive or model-based coding. Choosing the best methods depends mostly on the knowledge of the grid architecture, availability of historical data, computational resources, timing constraints, reconstruction quality requirements, and precise needs of the user.

5.1 Phasors-based methods

Phasors are an approach to describing sinusoidal electric signals using only amplitude and phase shift. The cosinusoidal and phasor representations of the signal will look like this:

$$x(t) = X_m \cos(\omega t + \theta), \tag{5.1}$$

$$X = X_m e^{j\theta}. \tag{5.2}$$

where X_m is the amplitude in volts (V), ω is the angular frequency in radians per second (rad/s), and θ is the phase angle in radians (rad).

PMUs provide measurements of synchronized current and voltage phasors. The sparsity and periodical nature of the phasor data allow efficient compression techniques to be applied. One of the simplest DC methods for phasors is Difference Encoding (DE). Since changes in the phasors over short time intervals tend to be small, it is efficient to save and transmit just the differences between consecutive values of the phasors instead of actual values:

$$\Delta X_k = X_k - X_{k-1}. \tag{5.3}$$

By sending just ΔX_k instead of the full phasor X_k, many data are minimized, particularly in stable operation when the phasor changes are small. Phasor measurements across various locations have high correlation, allowing for dimensionality reduction using PCA. Frequently, predictive models are also used. Predictive models estimate the evolution of the phasors, reducing the amount of data to be transmitted.

Phasors over time are sparse in the frequency domain, and this sparsity is exploitable for compression. With the DCT or Discrete Fourier Transform (DFT), we get:

$$X_f = \sum_{n=0}^{N-1} X_n e^{-j\frac{2\pi}{N}nf}, \tag{5.4}$$

where X_f is the frequency-domain coefficient in volts (V). It is possible to reconstruct a compressed version of the phasor data with minimal loss by retaining only the largest coefficients. Quantization decreases the number of bits required to represent the data of a phasor. Uniform quantization employs a fixed step size:

$$X_q = \text{round}\left(\frac{X}{\Delta}\right) \cdot \Delta, \tag{5.5}$$

where Δ is the quantization step in volts (V). More complex variable-bit quantization techniques allocate dynamically based on signal variation, optimizing compression efficiency.

5.2 Cosine Transform-Based Methods

Voltage and current waveforms exhibit periodic and slowly varying behavior, the DCT provides an effective means of transforming time-domain signals into a frequency-domain representation with minimal loss of information. The one-dimensional DCT of a signal x_n of length N is defined as:

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right), \quad k = 0, 1, \ldots, N-1, \tag{5.6}$$

where X_k denotes the DCT coefficient corresponding to the k-th frequency component in volts (V). Due to the nature of typical grid signals, most of the energy tends to be concentrated in a small number of low-frequency components, enabling efficient DC.

For streaming or RT signals in SG, it is common to divide the signal into non-overlapping blocks of fixed length. The DCT is applied to each block independently:

$$X^{(b)} = \text{DCT}(x^{(b)}), \tag{5.7}$$

where $x^{(b)}$ denotes the b-th signal block. Following transformation, only a subset of the largest-magnitude coefficients is retained, and the remaining are either discarded or zeroed. This truncation leads to a substantial reduction in data size. Coefficient thresholding and quantization are done similarly to the wavelet compression.

In dynamic grid environments, predictive models can be used in conjunction with the DCT to compress the residual signal. Let $\hat{x}_n$ be the predicted value of x_n, the residual r_n is then:

$$r_n = x_n - \hat{x}_n, \tag{5.8}$$

and the DCT is applied to r_n instead of x_n. Since r_n typically contains lower energy, its DCT representation is even sparser, enabling higher compression ratios.

For multidimensional data streams from multiple PMUs, the DCT can be applied along the temporal axis of each measurement channel. The resulting DCT matrix $\mathbf{X}_{\text{DCT}}$ often exhibits low-rank structure. A truncated low-rank approximation using SVD is given by:

$$\mathbf{X}_{\text{DCT}} \approx \sum_{i=1}^{r} \lambda_i \mathbf{a}_i \mathbf{b}_i^T, \tag{5.9}$$

where $r \ll \min(m, n)$, λ_i are the dominant singular values in volts (V), and $\mathbf{a}_i, \mathbf{b}_i$ are the corresponding singular vectors. This factorization achieves high compression efficiency, especially for spatially correlated measurements.

5.3 Compressed Sensing-Based Methods

CS is a signal processing framework that enables the reconstruction of sparse or compressible signals from a reduced number of linear measurements. The central premise of CS is that many real-world signals, including those observed in SG, exhibit sparsity in some transform domain, such as the Fourier or wavelet domain. This sparsity can be exploited to recover the original signal from far fewer samples than traditional Nyquist sampling would require, thus enabling efficient DC and reduced communication overhead in PS monitoring.

Let $\mathbf{x} \in \mathbb{R}^N$ be the original signal in volts (V), which is assumed to be sparse or compressible in some basis $\mathbf{\Psi}$, such that $\mathbf{x} = \mathbf{\Psi}\mathbf{s}$, where $\mathbf{s} \in \mathbb{R}^N$ has only $K \ll N$ non-zero or significant entries. Instead of sampling $\mathbf{x}$ directly, CS obtains a lower-dimensional measurement vector $\mathbf{y} \in \mathbb{R}^M$ (with $M \ll N$) using a sensing matrix $\mathbf{\Phi}$:

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} = \mathbf{\Phi}\mathbf{\Psi}\mathbf{s}. \tag{5.10}$$

Under certain conditions on $\mathbf{\Phi}$ and the sparsity of $\mathbf{s}$, it is possible to recover $\mathbf{x}$ from $\mathbf{y}$ by solving an optimization problem.

The original sparse coefficient vector $\mathbf{s}$ can be recovered by solving the convex optimization problem:

$$\min_{\mathbf{s}} \|\mathbf{s}\|_1 \quad \text{subject to} \quad \mathbf{y} = \mathbf{\Phi}\mathbf{\Psi}\mathbf{s}, \tag{5.11}$$

where $\|\cdot\|_1$ denotes the ℓ_1-norm. This approach promotes sparsity while ensuring that the recovered signal matches the observed measurements.

For geographically distributed measurements with joint sparsity (e.g., correlated PMU streams), Distributed CS allows the simultaneous recovery of multiple signals using joint sparsity models. If $\mathbf{X} = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(P)}]$ shares a common support in their sparse representations, recovery can be further improved via group sparsity formulations.

5.4 Singular Value Decomposition - Based Methods

SVD is a matrix factorization technique widely used for data compression, dimensionality reduction, and feature extraction in various domains. In the context of SG monitoring, where vast volumes of data are generated by PMUs and other sensors, SVD provides an effective means to exploit the inherent low-rank structure and correlation among measurements. This enables high-fidelity compression while preserving the essential dynamics of the system.

Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ denote a data matrix representing measurements from m sensors (e.g., PMUs) over n time instances. The SVD of $\mathbf{X}$ is given by:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{5.12}$$

where $\mathbf{U} \in \mathbb{R}^{m \times m}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ are orthogonal matrices containing the left and right singular vectors, respectively, and $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ is a diagonal matrix with non-negative real numbers $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ on the diagonal, known as singular values in volts (V).

In practice, many SG datasets are approximately low-rank due to high spatial and temporal correlations. A rank-r approximation of the original matrix $\mathbf{X}$ can be constructed by retaining only the first r largest singular values:

$$\mathbf{X} \approx \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^T, \tag{5.13}$$

where $\mathbf{U}_r \in \mathbb{R}^{m \times r}$, $\mathbf{\Sigma}_r \in \mathbb{R}^{r \times r}$, and $\mathbf{V}_r \in \mathbb{R}^{n \times r}$ contain the truncated components, and σ_i are the singular values in volts (V). This low-rank representation significantly reduces storage and transmission requirements while maintaining a high approximation accuracy.

PMU measurements collected over multiple substations and time periods form naturally structured matrices. Applying SVD to such data reveals that a small number of singular values capture the majority of the variance, making SVD-based compression highly effective. For example, let $\mathbf{X}$ represent the synchronized phasor magnitudes or angles across m buses over n time steps. The compressed representation consists only of the top r singular values and their associated vectors, which can be stored or transmitted efficiently.

In hybrid approaches, SVD is applied not directly to the raw time-series data but to its transform-domain representation. For instance, a DCT or DWT is first applied to $\mathbf{X}$, and SVD is then used to compress the transformed matrix:

$$\mathbf{X}_{\text{trans}} = \text{Transform}(\mathbf{X}), \quad \mathbf{X}_{\text{trans}} \approx \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^T. \tag{5.14}$$

This further enhances sparsity and compression efficiency by decorrelating the data before decomposition.

5.5 Predictive and Model-Based methods

Predictive and model-based compression methods leverage the temporal correlation and predictable dynamics inherent in electrical signals within SG. Rather than transmitting raw data, these methods aim to estimate future signal values based on historical patterns and transmit only the model parameters or residual prediction errors.

Let x_k denote the value of a monitored signal (e.g., voltage) in volts (V) at discrete time k. A predictive model estimates x_k using a function $f(\cdot)$ applied to a window of past values:

$$\hat{x}_k = f(x_{k-1}, x_{k-2}, \ldots, x_{k-p}), \tag{5.15}$$

where p is the model order. The residual or prediction error is defined as:

$$e_k = x_k - \hat{x}_k, \tag{5.16}$$

Compression is achieved by transmitting e_k and, if necessary, the model parameters, rather than the full-resolution signal x_k. To accommodate changes in grid dynamics, model parameters may be adaptively updated using algorithms such as least squares, recursive least squares, or Kalman filtering. This ensures that the predictive model remains accurate even in the presence of load fluctuations or disturbances. The updated model is periodically transmitted if significant deviation from previous parameters is detected.

The predictive coding architecture includes a predictor, an error encoder, and a decoder. At the transmitter, the predictor generates $\hat{x}_k$ and computes the error e_k. A quantized version of e_k is transmitted:

$$e_k^q = \text{Quantize}(e_k), \tag{5.17}$$

and the decoder reconstructs the signal as:

$$\tilde{x}_k = \hat{x}_k + e_k^q. \tag{5.18}$$

Due to the typically small magnitude of e_k, fewer bits are needed to encode it compared to x_k, leading to efficient compression.

Predictive and model-based compression techniques offer low computational complexity, high compression ratios, and adaptability to dynamic grid behavior. Their implementation in embedded systems (e.g., within PMUs or SMs) allows for local pre-processing and transmission of only essential information, reducing the communication burden on central monitoring systems.

5.6 Conclusion

Several distinct approaches to data compression for electrical signals have been developed, each with specific strengths and limitations depending on the nature of the data and application constraints. Phasor-based methods and predictive techniques are computationally lightweight and efficient for steady-state conditions but may struggle with capturing sharp transients and dynamic phenomena critical for PS protection and monitoring. DCT-based techniques offer effective compression for slowly varying periodic signals; however, their block-based nature may introduce artifacts and reduced flexibility in representing localized events. CS leverages sparsity for significant data reduction but typically requires complex reconstruction algorithms, making RT application challenging. SVD efficiently exploits spatial and temporal correlations but may be sensitive to noise and computationally intensive for high-resolution streaming data.

WT-based approaches provide an attractive compromise between compression efficiency, computational complexity, and signal reconstruction quality. The multi-resolution nature of wavelets enables representation of low-frequency trends as well as high-frequency transient events, which are crucial in power grid monitoring and control. Wavelet-based compression can adapt to the nonstationary characteristics of electrical signals, achieving high compression ratios while maintaining high fidelity. Moreover, the hierarchical organization of wavelet coefficients enables integration with efficient entropy coding techniques. This research focuses on the development of efficient DC method for signals with distortion, mainly for the purpose of protection and PQ analysis. Therefore, based on the analysis presented, a DWT-based compression approach was selected as the foundation for the system developed in this thesis.

5.7 Reasons for choosing Discrete Wavelet Transform

Analysis of most popular approaches in the literature shown that each of them has its benefits in certain situations. The comparison between methods was is shown in 5.1. DCT was chosen due to the following reasons.

1. It does not require the knowledge of grid architecture (like SVD and Model-Based methods).
2. It does not require intervention in the data acquisition component (like CS), the measured signal alone is sufficient.
3. It is suitable to compress large datasets ($O(n)$ computational complexity).
4. It offers high noise robustness and adaptability to the requirements (tradeoff between compression ratio and compression loss can be calibrated).
5. It can compress transients efficiently (which is a drawback of phasor and DCT based methods).
6. It is suitable for real-time systems.

Table 5.1: Comparison of Data Compression Methods for Smart Grids

Criteria	DWT Methods	Phasor-Based Methods	DCT Methods	CS	PCA/SVD-Based Methods	Predictive & Model-Based Compression
Domain of Operation	Time-Frequency	Time/Phasor	Frequency (DCT)	Measurement / Transform	Measurement Matrix	Time-Domain
Sparsity Exploitation	High (multi-scale)	Moderate (periodicity)	High (low-frequency energy compaction)	High (sparse basis)	High (low-rank structure)	Moderate to High (temporal correlation)
Compression Efficiency	High (especially for non-stationary signals)	Moderate to High	High (especially in steady-state)	Very High	High	Moderate to High
Adaptability	Good with multiresolution analysis	Static or semi-dynamic	Static block-wise	Moderate (randomized)	Low unless updated adaptively	High (model updated online)
Computational Complexity	Moderate to High	Low	Moderate to High	Moderate to High (especially recovery)	High (SVD computation)	Low to Moderate
Noise Robustness	High	Moderate	Moderate	High (with relaxed recovery)	High (via denoising)	High (model filters noise)
Real-Time Feasibility	Feasible with efficient implementation	Very Feasible	Highly Feasible	Challenging (reconstruction cost)	Moderate (incremental SVD needed)	Highly Feasible
Spatial Correlation Handling	Limited (unless extended)	Yes (PMU network)	Limited	Yes (via joint sparsity models)	Excellent (matrix-based)	Possible
Temporal Correlation Handling	Yes (via scale decomposition)	Yes (phasor dynamics)	No (requires hybrid approach)	Indirectly	No (requires hybrid approach)	Yes (model-driven)
Typical Use Cases	Transient analysis, localized faults	Steady-state monitoring	Voltage/ frequency streams	Energy-constrained sensors	Large-scale grid compression	Streaming, forecasting, anomaly detection

Chapter 6

Wavelet compression

The WT is particularly well-suited for the compression of time-series data, such as sinusoidal waveforms, due to its ability to retain essential transient information and key features relevant to PS analysis, including harmonics [91]. Unlike traditional Fourier Transform or Cosine Transform-based methods, which provide a global frequency representation, the WT offers a localized analysis in both time and frequency domains [92]. This characteristic enables it to effectively capture transient phenomena such as faults, voltage sags, and surges, which are critical for system diagnostics. Moreover, the WT efficiently encodes the periodic components of sinusoidal signals while accurately preserving harmonics, which are fundamental in assessing PQ and detecting distortions. Although the computational complexity of the WT is higher than that of simpler techniques, advancements in algorithmic efficiency have made it feasible for RT applications [93, 94], providing a better trade-off between compression efficiency and computational demand. As a result, the WT is highly effective for the compression and analysis of electrical signals [95], maintaining the integrity of critical information while reducing data redundancy.

The DWT decomposes a signal into a set of orthogonal basis functions known as wavelets 6.1. These wavelets are generated from a mother wavelet $\psi(t)$ through translation and scaling:

$$\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k) \tag{6.1}$$

where j and k are integers representing the scale and translation parameters, respectively; t is time in seconds (s); and $\psi_{j,k}(t)$ has the same physical units as the signal amplitude (e.g. volts, V) [96]. The DWT of a continuous-time signal $x(t)$ is given by the inner products:

$$W_{j,k} = \langle x(t), \psi_{j,k}(t) \rangle = \int_{-\infty}^{\infty} x(t)\psi_{j,k}(t)\, dt \tag{6.2}$$

These coefficients $W_{j,k}$ (volts, V) capture the signal's details at various scales and positions [97].

6.1 Signal Decomposition and Reconstruction

The DWT involves the decomposition of a signal into approximation and detail coefficients [98]. This is typically done using a pair of filters: a low-pass filter G and a high-pass filter H [99]. The decomposition process can be represented as:

$$a_{j+1}[n] = \sum_k g[k - 2n]a_j[k] \tag{6.3}$$

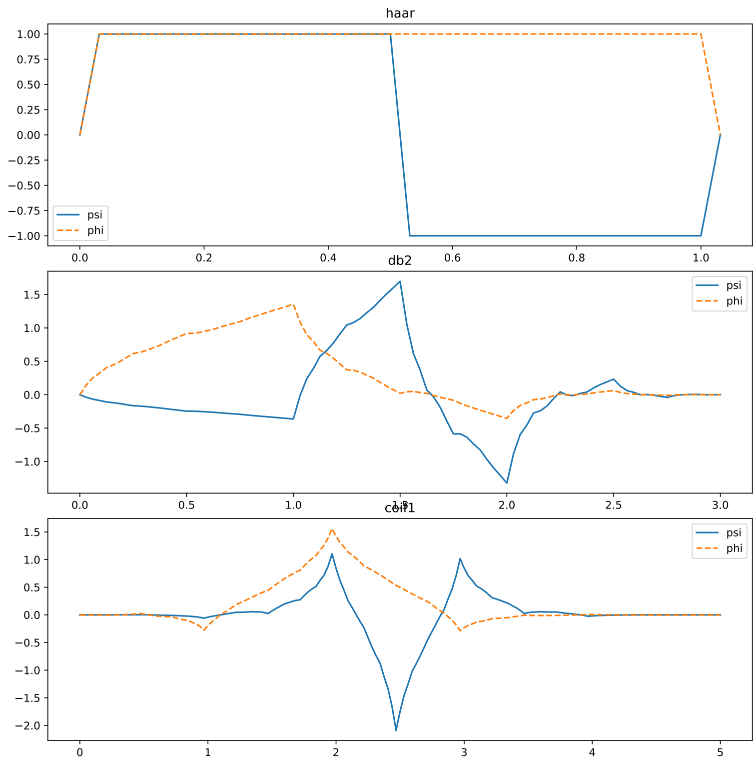


Figure 6.1: Wavelets from Haar, Daubechies and Coiflets families. For each wavelet, the scaling function φ and the wavelet function ψ are presented. [1]

$$d_{j+1}[n] = \sum_k h[k-2n]a_j[k] \tag{6.4}$$

where $a_j[n]$ and $d_j[n]$ are the approximation and detail coefficients at scale j (volts, V), respectively. The original signal can be reconstructed by reversing the decomposition process, ensuring no loss of information.

6.2 Application in Data Compression

The goal of data compression is to reduce the amount of data required to represent a signal without a significant loss of quality. The DWT is highly effective in this regard as a result of its ability to concentrate energy into a few significant coefficients [100, 101]. This property allows for efficient thresholding and quantization [102].

6.2.1 Thresholding

Thresholding involves setting the small wavelet coefficients to zero, as they typically represent noise or insignificant details [103]. A common approach is hard thresholding, defined as:

$$W_{j,k} = \begin{cases} W_{j,k} & \text{if } |W_{j,k}| \geq \lambda \\ 0 & \text{if } |W_{j,k}| < \lambda \end{cases} \tag{6.5}$$

where λ is a chosen threshold value in volts (V). Soft thresholding is another technique, given by:

$$W_{j,k} = \begin{cases} \text{sign}(W_{j,k})(|W_{j,k}| - \lambda) & \text{if } |W_{j,k}| \geq \lambda \\ 0 & \text{if } |W_{j,k}| < \lambda \end{cases} \tag{6.6}$$

6.2.2 Quantization and Encoding

After thresholding, the remaining significant coefficients are quantized and encoded. Quantization involves mapping the continuous range of coefficient values to a finite set of levels, while encoding compresses these levels into a binary format [104, 105]. Several specialized compression algorithms have been developed to efficiently encode coefficients produced by the discrete wavelet transform (DWT), exploiting their inherent sparsity, clustering, and hierarchical relationships. Popular encoding methods include Embedded Block Coding with Optimized Truncation (EBCOT), Set Partitioning in Hierarchical Trees (SPIHT), Set Partitioned Embedded Block Coder (SPECK), RLE and Huffman Coding [106].

The **SPIHT** algorithm, introduced by Said and Pearlman [107], organizes wavelet coefficients into hierarchical trees based on spatial orientation. It exploits the principle that if a wavelet coefficient is insignificant with respect to a given threshold, its descendants are also likely to be insignificant. SPIHT operates by progressively refining the significance of coefficients through successive approximations. The algorithm partitions the set of coefficients into three lists: the List of Significant Pixels, the List of Insignificant Pixels, and the List of Insignificant Sets. At each bit-plane level, it compares coefficients against a threshold $T = 2^n$, where n is the current bit-plane, and encodes significant coefficients using a refinement pass while splitting sets otherwise. This hierarchical structure enables efficient embedded coding and offers a high CR even for lossless cases when the quantization step size is unity.

The **EBCOT** algorithm, proposed by Taubman [108], forms the core of the JPEG2000 standard and operates in two tiers. In Tier-1, wavelet coefficients are quantized and encoded independently in small rectangular blocks (code-blocks) using bit-plane coding with context modelling. Specifically, each coefficient is encoded bit-plane-by-bit-plane, with contexts conditioned on the significance of neighbouring coefficients, improving entropy coding efficiency. For lossless compression, EBCOT employs a reversible integer wavelet transform (e.g. the (5,3) lifting scheme) and omits quantization, ensuring exact reconstruction. The compression efficiency of EBCOT can be further improved by applying an arithmetic coder such as the Multiplication/Quotient coder (MQ-coder) on bit-plane outputs. Formally, the coding rate R after compression can be approximated as:

$$R = \sum_i r_i \tag{6.7}$$

where R and each r_i are measured in bits.

The **SPECK** algorithm, developed by Islam and Pearlman [109], also targets efficient wavelet coefficient compression by partitioning the wavelet domain into progressively smaller rectangular sets. SPECK encodes the most significant coefficients first by recursively subdividing sets into smaller regions based on their significance relative to a threshold. Unlike SPIHT, which tracks significance using trees, SPECK uses spatially contiguous blocks, resulting in simpler data structures and faster implementation. As with SPIHT, a bit-plane thresholding approach is used, with the same threshold $T = 2^n$ at bit-plane n, and coefficients are classified into significant or insignificant sets accordingly. SPECK is well suited for both lossless and near-lossless compression scenarios where computational efficiency is critical.

6.3 Advantages of Discrete Wavelet Transform in Data Compression

The DWT offers several advantages in data compression [110, 111]:

- **Multi-resolution Analysis**: DWT provides both time and frequency localization, making it suitable for non-stationary signals.
- **Energy Compaction**: The DWT can concentrate the signal energy into a few large coefficients, facilitating efficient compression.
- **Adaptability**: Different mother wavelets can be chosen to match the signal characteristics, enhancing compression performance.

6.4 Filter Bank vs. Lifting-Scheme Implementation of the DWT

The DWT may be realised through two mathematically equivalent algorithms: the classical **filter-bank** (convolution) scheme and the **lifting** scheme. Both produce identical wavelet coefficients when the same underlying wavelet is employed; the difference lies solely in the computational pathway.

6.4.1 Implementation Differences

Filter-bank DWT At each decomposition level, the input signal $x[n]$ (volts, V) is passed in parallel through a low-pass analysis filter $h[n]$ and a high-pass analysis filter $g[n]$; the outputs are subsequently down-sampled by 2:

$$c_a[k] = \sum_n x[n]\, h[n-2k], \qquad c_d[k] = \sum_n x[n]\, g[n-2k], \tag{6.8}$$

where $c_a[k]$ and $c_d[k]$ are the approximation and detail coefficients (volts, V), respectively. Recursive application to the approximation branch yields the familiar multiresolution hierarchy.

The operation can be expressed in the *polyphase* domain by splitting the filters into even (H_0) and odd (H_1) parts and writing

$$\begin{pmatrix} C_A(z) \\ C_D(z) \end{pmatrix} = \underbrace{\begin{pmatrix} H_0(z) & H_1(z) \\ G_0(z) & G_1(z) \end{pmatrix}}_{\mathbf{P}(z)} \begin{pmatrix} X_e(z) \\ X_o(z) \end{pmatrix}, \tag{6.9}$$

where $X_e(z)$ and $X_o(z)$ are the z-transforms of the even and odd sample streams (volts, V). The matrix $\mathbf{P}(z)$ is invertible for any wavelet basis that guarantees perfect reconstruction.

Lifting scheme. The lifting algorithm factorises the same polyphase matrix into a cascade of upper- and lower-triangular matrices, each corresponding to a *predict* or *update* step:

$$\mathbf{P}(z) = \begin{pmatrix} 1 & S(z) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ T(z) & 1 \end{pmatrix} \begin{pmatrix} K & 0 \\ 0 & 1/K \end{pmatrix}, \tag{6.10}$$

where $S(z)$ and $T(z)$ are dimensionless predict and update filters and K is an optional dimensionless scaling factor. All signal samples processed in the lifting steps retain the units of volts (V). Integer-to-integer variants of the lifting scheme permit exactly reversible transforms in finite-precision arithmetic.

6.4.2 Equivalence of Outputs

Because (6.10) factorizes the same polyphase matrix as (6.9), both algorithms produce identical coefficients (up to roundoff) when supplied with the same wavelet filters $h[n],\ g[n]$. Consequently, measurable compression performance — CR, MSE, SNR, *etc.* — depends only on the wavelet basis, the depth of decomposition and the coefficient thresholding strategy, not on the computational execution.

6.4.3 Justification for the Filter-Bank Realisation

The filter-bank implementation has been adopted throughout the presented work for the following reasons:

- **Conceptual clarity and prototyping flexibility.** The convolution-and-downsample view in (6.8) aligns directly with the multiresolution analysis framework and allows immediate visual interpretation of sub-band content. Python's `PyWavelets` (`pywt`) library exposes a concise interface—for example, `pywt.wavedec`—that facilitates rapid substitution of wavelet families, decomposition levels, and threshold policies while analysing highly transient electrical signals.

- **Compatibility with scientific evaluation.** Explicit filter impulse responses enable straightforward examination of frequency selectivity, vanishing-moment characteristics, and sub-band energy distributions. Such transparency simplifies the systematic assessment of compression performance as a function of wavelet type, transform depth, and threshold value.

- **Alignment with deployment libraries and DSP hardware.** The target embedded platform executes a C++ wavelet packet decomposition library that itself relies on finite impulse response filter convolutions. Many IoT-class DSP cores include hardware multiply-accumulate units and optimised finite impulse response kernels; a filter-bank DWT therefore maps efficiently to the available instruction set, whereas the data-dependent memory access pattern of lifting may offer no practical speed advantage. Using the same conceptual realisation at both prototyping and deployment stages eases verification: transform coefficients produced offline match those obtained on the embedded device.

In summary, the filter-bank approach provides transparent, readily verifiable signal analysis, straightforward parameter exploration, and direct compatibility with both high-level scientific software and the intended embedded hardware, without any loss in compression efficacy relative to the lifting scheme.

Chapter 7

Parameterization of the Compression System

7.1 Parameterization of the Discrete Wavelet Transform

The DWT is highly configurable, allowing for adjustments based on the specific needs of the application or the characteristics of the input signal. The key parameters of the DWT that can be adjusted include the choice of wavelet, decomposition level, and thresholding methods. These parameters play a crucial role in determining the effectiveness of the DWT in signal analysis and processing.

Choice of Wavelet

The choice of wavelet is fundamental to the DWT [112]. Common wavelets include the Haar wavelet, Daubechies wavelets, Symlets, and Coiflets [113]. Each wavelet has unique properties that make it well-suited for different types of signals.

- **Haar Wavelet**: Simple and computationally efficient, suitable for signals with sharp changes.
- **Daubechies Wavelets**: Provide a balance between time and frequency localization, ideal for a wide range of signals.
- **Symlets**: Symmetrical wavelets that reduce phase distortion, useful for signals requiring minimal phase distortion.
- **Coiflets**: Provide better frequency resolution, suitable for signals requiring detailed frequency analysis.

Mathematically, a wavelet function $\psi(t)$ must satisfy the admissibility condition:

$$\int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|}\, d\omega < \infty, \tag{7.1}$$

where $\hat{\psi}(\omega)$ is the Fourier transform of $\psi(t)$ (unitless), and ω is the angular frequency in radians per second (rad/s).

Decomposition Level

The decomposition level determines how many times the signal is decomposed into approximation and detail coefficients [114]. It is typically denoted as J and can be chosen based on the desired resolution and the length of the signal.

$$J = \log_2 \left(\frac{N}{L} \right), \tag{7.2}$$

where N is the length of the signal (samples) and L is the length of the filter (samples).

Threshold Parameterization

The thresholding strategy needs to be decided by choosing between hard thresholding 6.5 and soft thresholding 6.6, based on the desired compression ratio and the permissible loss of information. The primary goal is to remove noise and irrelevant details while preserving the important features of the data. Hard thresholding operates by setting all coefficients below a certain threshold to zero, effectively eliminating noise and minor details below that threshold [115]. However, this method can result in abrupt changes and sometimes introduce artifacts into the data, potentially leading to a loss of important subtle features. Soft thresholding modifies the coefficients by reducing their magnitude proportionally to their distance from the threshold [116]. This method is often preferred in practice because it tends to produce smoother results and fewer artifacts in the compressed data. Smoother transitions help maintain a more natural appearance in the data after compression, which is desirable in applications such as image and audio compression, where fidelity is critical [117].

Additionally, adaptive thresholding can enhance the performance of the compression process. Adaptive thresholding involves dynamically adjusting the threshold value based on specific criteria, such as the level of noise present in the data or other relevant characteristics [118, 119]. By adapting the threshold in this manner, it is possible to achieve a better trade-off between the degree of compression and the preservation of data fidelity. This adaptive approach ensures that the important features of the data are maintained while effectively reducing the data size by removing noise and less significant details [120].

7.1.1 Parameter Dependence on Input Signal Characteristics

The parameterization of the DWT should be adapted to the specific characteristics of the input signal, such as the distortion levels, harmonic content, and transient behavior.

Distortion Levels

For signals with high distortion, wavelets with strong time localization, such as the Haar wavelet, may be preferable. Conversely, for signals with low distortion, wavelets with better frequency resolution, such as the Daubechies or Coiflets, might be more suitable.

Harmonic Content

Signals with high harmonic content benefit from wavelets that provide good frequency localization. For example, the Coiflets are well-suited for analyzing harmonic content due to their high number of vanishing moments and better frequency resolution.

Transient Behavior

Signals with high transients require wavelets that can accurately capture sudden changes. The Haar wavelet, with its simple and discontinuous nature, is particularly effective for this purpose. On the other hand, for signals with low transients, smoother wavelets such as the Symlets can be more appropriate.

7.2 Bayesian Optimization

Bayesian Optimization efficiently searches for a near-optimal solution by balancing exploration and exploitation within a limited number of function evaluations. As described previously, selecting the right wavelet parameters is crucial for effective signal compression [121]. Traditional methods for parameter selection often involve an exhaustive search, which is computationally expensive. Bayesian Optimization offers a powerful alternative by intelligently exploring the parameter space to find the best set of parameters with fewer evaluations [122]. Using Bayesian Optimization to select the wavelet parameters for a given signal enhances the efficiency and performance of the compression process [123], returning compression parameters faster than brute-force search.

Bayesian Optimization is a sequential design strategy for finding the global near-optimum of black-box functions that are expensive to evaluate. It consists of two main components: a surrogate model and an acquisition function [124].

7.2.1 Surrogate Model

A surrogate model, typically a Gaussian Process (GP), is used to model the objective function [125]. A GP is defined by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{7.3}$$

where $f(\mathbf{x})$ is the objective function value (dimensionless), $m(\mathbf{x})$ is its mean (dimensionless), and $k(\mathbf{x}, \mathbf{x}')$ is the covariance (dimensionless^2). The mean function $m(\mathbf{x})$ represents the expected value of the function (dimensionless), while the covariance function $k(\mathbf{x}, \mathbf{x}')$ describes the correlation between different points in the input space (dimensionless^2) [126].

7.2.2 Acquisition Function

The acquisition function guides the selection of the next point to evaluate by balancing exploration and exploitation [127, 128]. Common acquisition functions include Expected Improvement (EI), Probability of

Improvement (PI), and Upper Confidence Bound (UCB) [129]. The EI is given by:

$$EI(\mathbf{x}) = \mathbb{E}[\max(0, f(\mathbf{x}) - f(\mathbf{x}^+))] \quad (7.4)$$

where $EI(\mathbf{x})$ is the expected improvement (dimensionless), $f(\mathbf{x})$ is the objective function value (dimensionless), and $f(\mathbf{x}^+)$ is the current best observation (dimensionless). The acquisition function is maximized to determine the next point $\mathbf{x}_{\text{next}}$ to evaluate.

7.3 Application in Wavelet Parameter Optimization

This method can be used to select wavelet compression parameters, including wavelet type, decomposition level, and thresholding values, to maximize compression performance [130]. The objective function can be defined in terms of a compression metric, such as the compression ratio or the reconstruction error [**?**].

7.3.1 Formulating the Optimization Problem

Let $\mathbf{x} = (w, l, \lambda)$ represent the wavelet parameters, where w is the wavelet type, l is the decomposition level, and λ is the threshold value. The objective function $f(\mathbf{x})$ measures the performance of the wavelet parameters in terms of compression quality:

$$f(\mathbf{x}) = \text{CompressionMetric}(w, l, \lambda) \quad (7.5)$$

where $f(\mathbf{x})$ is the compression metric (dimensionless). Each parameter set is then evaluated against the maximum MSE requirement. If the solution does not meet this requirement, it is discarded by the algorithm.

7.3.2 Implementing Bayesian Optimization

The Bayesian Optimization process for wavelet parameter selection involves the following steps [131]:

1. **Initialization**: Evaluate the objective function at several initial points.
2. **Model Fitting**: Fit a Gaussian Process model to the observed data.
3. **Acquisition Function Maximization**: Maximize the acquisition function to select the next point for evaluation.
4. **Objective Function Evaluation**: Evaluate the objective function at the selected point.
5. **Iteration**: Update the Gaussian Process model with the new observation and repeat the process until convergence.

7.4 Advantages of Bayesian Optimization in Wavelet Parameter Selection

Bayesian Optimization offers several advantages for selecting wavelet parameters:

- **Efficiency**: It reduces the number of function evaluations required to find near-optimal parameters, making it suitable for expensive objective functions.

- **Flexibility**: It can handle different types of wavelet parameters and accommodate various performance metrics.

- **Adaptability**: It can adapt to the shape of the objective function, efficiently exploring and exploiting the parameter space.

Bayesian Optimization can significantly reduce the time required for parameterization. The algorithm is computationally demanding, making it unsuitable for most RT systems used in PG, but it can be used to classify the data used to train the NN.

Chapter 8

Proposed Solution

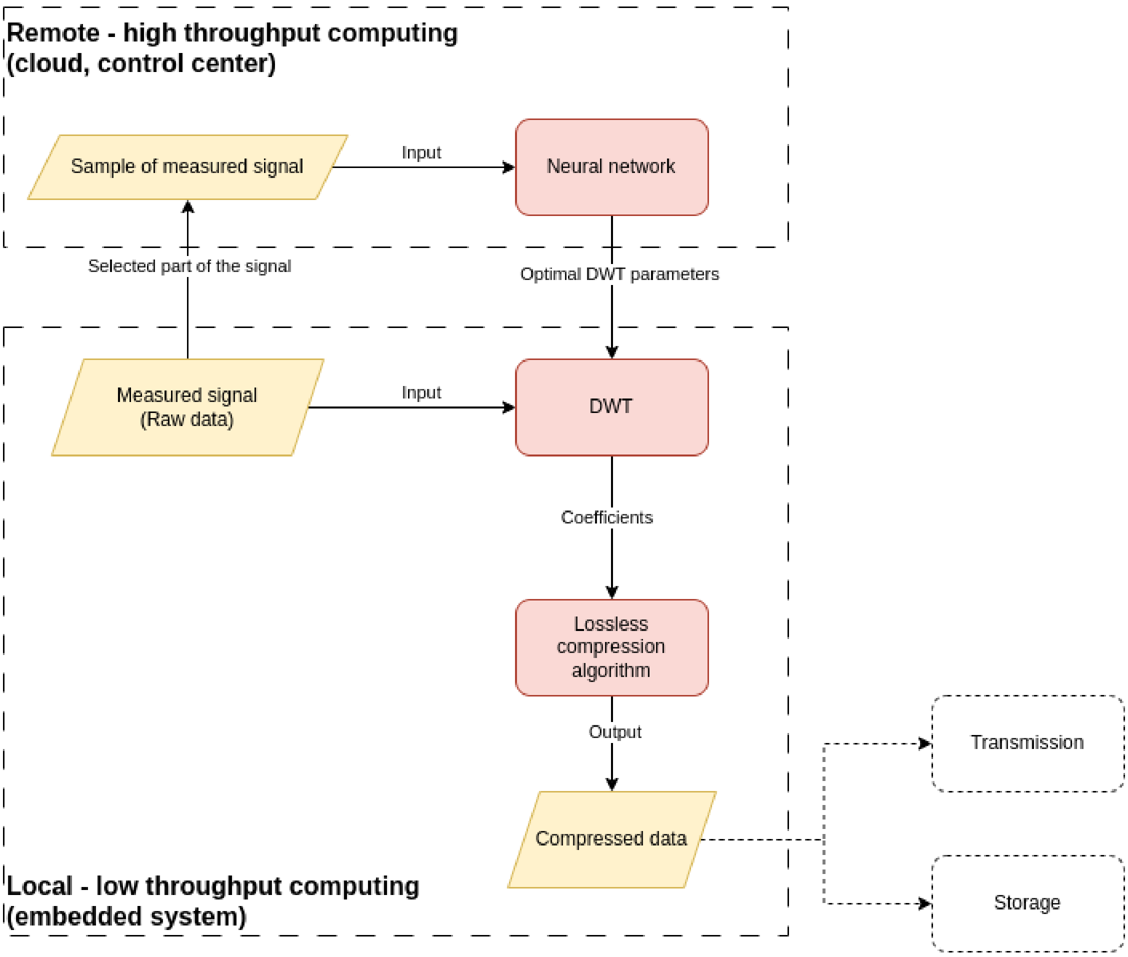


Figure 8.1: Logical architecture of the proposed solution from the user's point of view. Elements in yellow are data blocks provided by the system; elements in red are instruction blocks provided by the implementer of the compression system.

In real-world wide-area measurement systems, signals measured by sensors at different points will vary. The goal of the proposed system is to dynamically parameterize the DWT separately for each sensor in the network. To be scalable, the computational effort required for parameterization must be relatively low. As shown in 8.1, most of the computational burden can be processed in parallel. This load can be offloaded to a centralized computing cluster (such as a cloud platform or high-throughput supervisory computer) or to a different core, thereby avoiding any run-time overhead for the compression task.

8.1 Architecture of the Proposed System

The architecture depicted in 8.2 encompasses three primary stages: signal generation, Bayesian optimization, and NN training and deployment. The system is designed to generate distorted signals, select compression parameters, label signals, train a NN, and use the trained network in embedded sensing systems. Each stage of the process is described below.

8.1.1 Signal Generation

The signal generation process involves creating signals with specific characteristics defined by harmonics and transients.

Harmonics

Harmonics are defined by their minimum and maximum values. These parameters control the fundamental frequency components of the generated signals.

Transients

Transients are specified by their number and their minimum and maximum values. These parameters introduce non-stationary components into the signals, simulating real-world distortions.

8.1.2 Bayesian Optimization

The distorted signals are then passed to a Bayesian Optimization module. This module selects the compression parameters that minimize the size of the compressed signal while keeping the MSE below a predefined threshold.

Compression and Decompression

The Bayesian Optimization module includes a loop of compression and decompression operations. The signal is compressed and then decompressed, followed by an MSE check to evaluate the fidelity of the compression process.

Compression Ratio Calculation

The compression ratio is calculated to assess the efficiency of the algorithm. The parameter selection process iterates to find the parameters that yield the lowest compression ratio while maintaining acceptable MSE values.

8.1.3 Signal Labeling

Signals are labeled with the selected parameters using a specific naming scheme:

```
signal<No>_<wavelet_function>_<decomposition_level>_<threshold>.csv
```

The dataset consists of 4000 labeled signal files for training and 1000 randomly selected signals for validation.

8.1.4 Neural Network Training

A NN is trained using the labeled signal files. The goal is for the NN to learn to predict the optimal DWT parameters for various types of signals.

Training Data

The training dataset consists first of 547 real signals, followed by 12,000 artificially generated and labeled signal files. Each file is associated with near-optimal parameters obtained through Bayesian Optimization.

Validation Data

Validation is performed using 50 real signals and 1000 synthetic signals to ensure that the NN generalizes well to unseen data.

Neural Network Deployment

The trained NN is deployed in embedded sensing systems, where it is used to determine the appropriate DWT parameters for unknown signals.

Selecting DWT Coefficients

The NN processes incoming signals and outputs the corresponding DWT parameters. These parameters are then used in embedded systems for efficient signal processing.

8.2 Advantages in IoT Environments

The integration of this advanced signal compression system into SG applications offers several benefits, especially in terms of reducing data volume relative to resource usage.

8.2.1 Scalability

One of the main advantages of the proposed system is its scalability. As the SG expands, the number of sensors and data points grows significantly [132]. The architecture described here supports this growth efficiently for the following reasons:

- **NN Adaptability:** The NN can be retrained with new data, allowing it to adapt to an increasing variety of signals without requiring extensive manual recalibration.
- **Modular Design:** The modular nature of the system facilitates the easy addition of new sensors and monitoring devices. The NN's ability to generalize from existing data ensures seamless integration of new components.

8.2.2 Adaptability

The dynamic nature of SGs, characterized by fluctuating power demands and the integration of renewable energy sources, necessitates a highly adaptable system. The described signal processing architecture is particularly well-suited for such environments:

- **Bayesian Optimization:** This component can adjust compression parameters based on newly acquired data, improving performance in response to real-time signal characteristics. This ensures consistently high compression efficiency and data fidelity.
- **Dynamic Parameterization:** Sensors can transmit a sample of the measured signal to a supervisory unit and, at runtime, receive new calibration parameters that are better suited to current conditions.

8.2.3 Enhanced Data Management

Efficient DC has a direct impact on data handling and communication within the SG:

- **Reduced Bandwidth Requirements:** Effective compression reduces the bandwidth needed for data transmission, which is especially important in large-scale deployments where communication robustness is critical [133].
- **Improved Storage Efficiency:** Compressed data occupies less space, enabling longer retention periods and more extensive historical analysis, which in turn supports trend analysis and predictive maintenance.

8.2.4 Cost Efficiency

The system's compression capabilities yield substantial cost savings [134]:

- **Lower Operational Costs:** Efficient DC reduces the volume of transmitted and stored data, lowering associated costs—particularly beneficial in large-scale SG infrastructures.
- **Energy Efficiency:** Improved signal processing reduces the computational and energy requirements of data handling. This is especially important in embedded sensing systems that rely on limited power sources.

The proposed signal processing system offers significant advantages for DC in SG environments. Its scalability and adaptability make it capable of handling the increasing complexity and dynamic nature of modern PGs. By improving data management, reducing costs, and maintaining high data fidelity, the system contributes meaningfully to the efficiency, reliability, and sustainability of SG operations.

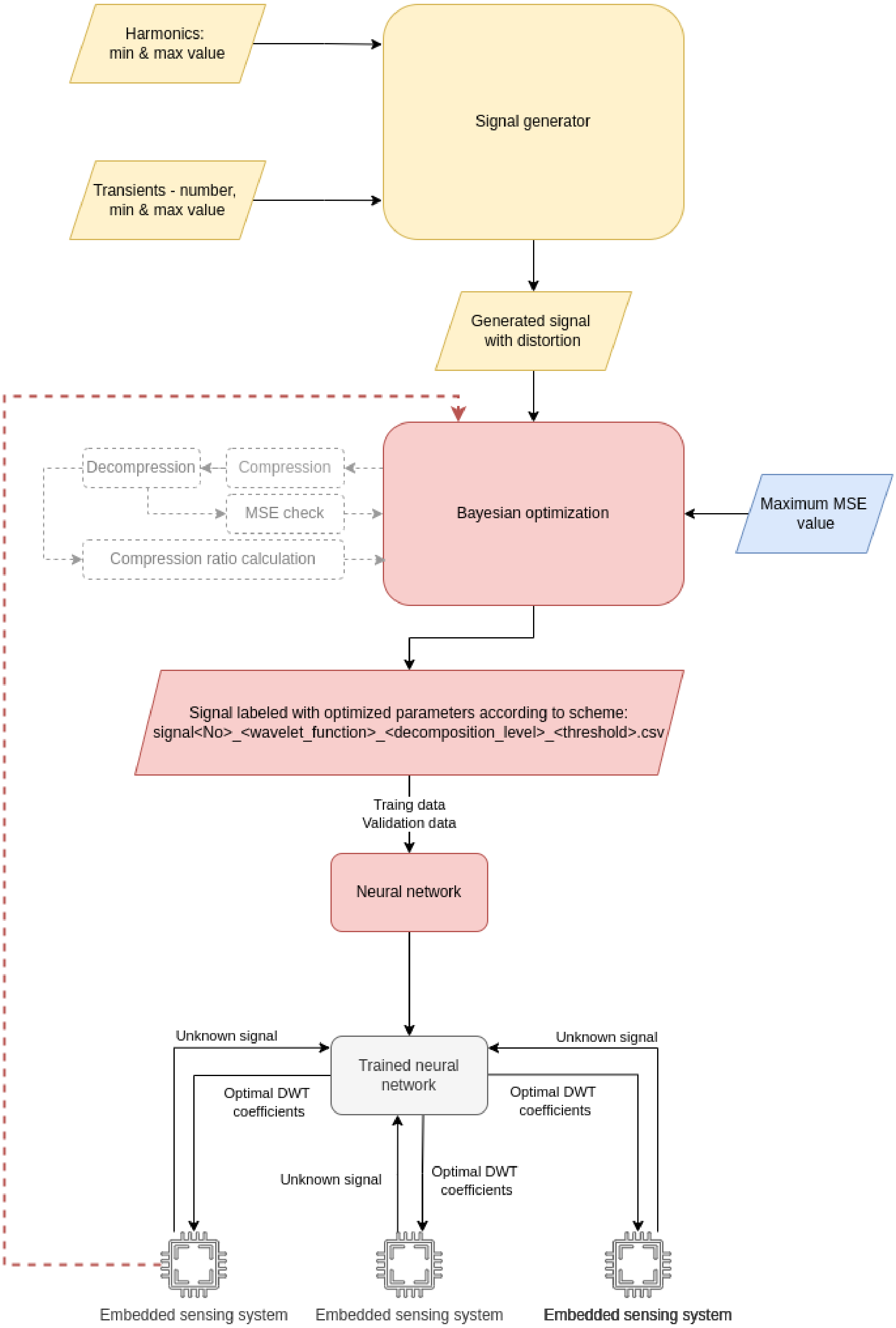


Figure 8.2: Detailed architecture of the proposed solution. Elements marked in red represent the core system, which uses real data for training. Elements in yellow are part of synthetic data preparation, which can substitute for real data. The element marked in blue (maximum MSE) is set by the user. Bayesian Optimization can use parameters either generated according to the use case or obtained from real signal data from the grid (indicated by an optional red arrow). Details of the algorithm are shown in transparent blocks. The deployed part of the system is shown in gray.

Chapter 9

Dataset

9.1 Data Generation

The data used for neural-network training are generated by a Python script to facilitate development and testing. The script accepts input parameters specifying the minimum and maximum distortion values in the signal and then interpolates linearly between them for each generated file. To simplify dataset creation further, the script also assigns labels for the NN. It was run with different parameter sets to create 5,000 files as training input for the NN. The observed training and validation losses indicated that the dataset was of sufficient quality and quantity. Details of the script's functionality and the data-generation process are provided in this chapter.

9.1.1 Signal Generation

The core of the script is the `generate_signal` function, which creates synthetic signals from the following parameters:

- `num_samples`: number of samples in the signal,
- `base_freq`: base frequency in hertz,
- `sample_rate`: sampling rate in hertz,
- `amplitude`: amplitude of the base signal,
- *Harmonics parameters* (`harmonic3`, `harmonic5`, `harmonic7`, `harmonic9`, `harmonic11`): amplitude multipliers for the 3rd, 5th, 7th, 9th, and 11th harmonics,
- `transient_amount`: number of transient spikes,
- `transient_max_value`: maximum magnitude of those transients.

A time vector `t` is created with `numpy`'s `linspace`; a sine wave at the base frequency forms the carrier, after which harmonics and random transient spikes are added to emulate real-world anomalies.

9.1.2 Data Saving

The generated signals are saved as Comma-separated values (CSV) files with the `save_to_csv` function, which builds a Pandas DataFrame from the time vector and signal values and then exports it for later analysis.

9.1.3 Signal Plotting

The `plot_signals` function locates files that match a user-supplied pattern (via `glob`). If none are found, a message is printed; otherwise, the first, middle, and last files are plotted with `matplotlib`. This visualization helps verify the integrity of the generated signals and the effects of the inserted transients and harmonics.

9.1.4 Signal Compression

Compression is handled by several functions:

- `dwt_compress`: applies DWT and coefficient thresholding,
- `dwt_reconstruct`: rebuilds the signal from the retained coefficients,
- `run_length_encoding`: performs RLE on the coefficients,
- `build_huffman_tree` and `huffman_code_tree`: construct a Huffman tree and codebook,
- `huffman_encoding`: applies the Huffman codes.

Thresholding reduces coefficient magnitudes, RLE removes consecutive redundancies, and Huffman encoding assigns variable-length codes according to symbol frequency.

9.1.5 Compression and Evaluation

The `compress_and_evaluate` function executes the selected compression steps and computes the MSE between the original and reconstructed signals. If the MSE exceeds a preset threshold, the compression is rejected and a CR of zero is returned. Otherwise, the CR is calculated as the ratio of the original size (stored as 32-bit floats) to the size of the compressed data.

9.1.6 Bayesian Optimization

Compression parameters are chosen with Bayesian optimization in `select_parameters`, which relies on the `hyperopt` library. The search space includes

- `wavelet`: index of the candidate wavelet,
- `level`: decomposition level,
- `threshold`: coefficient-threshold value.

The objective function maximizes CR while keeping the MSE below the specified limit; 's Tree-structured Parzen Estimator (TPE) algorithm explores the space.

9.1.7 Main Function

The `main` routine coordinates the workflow. It defines the number of files, signal parameters, and ranges for harmonics and transients, then creates an output directory. For each file, harmonic and transient values are interpolated within their ranges.

1. If an input signal exists, the script selects parameters that yield the best CR.
2. Otherwise, it generates a signal with interpolated parameters and optimizes the compression parameters for that signal.

The chosen parameters are embedded in each filename. Plots provide visual monitoring of the script's execution.

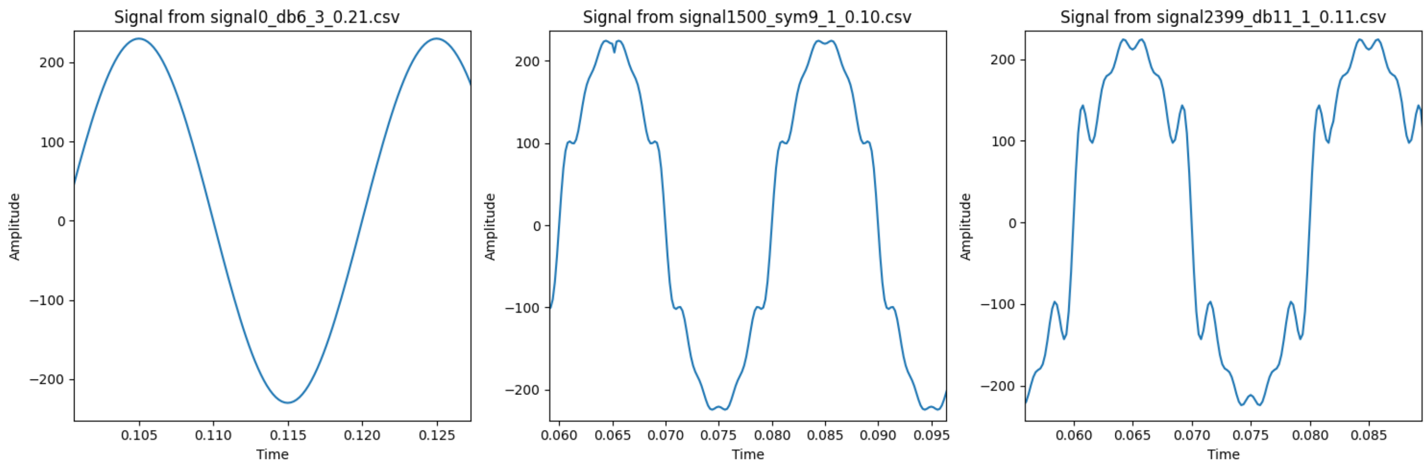


Figure 9.1: Example signals and labels generated by the script. The increasing harmonic distortion and the linear interpolation between distortion levels across consecutive signals are visible.

9.2 Training Data for the Neural Network

The data generated by the script are used for training and validating the neural network. A total of 5,000 labeled signal files were created with varying parameters. Table 9.1 lists the exact parameter ranges. The dataset was designed to span a broad range of harmonic and transient distortions so that the network encounters scenarios likely to occur in real systems.

9.3 Test Data for System Validation

To evaluate the system on unseen signals, a separate test set of 100 files was created. The files were split evenly into four categories:

- low harmonic distortion and low transient distortion,
- high harmonic distortion and low transient distortion,

- low harmonic distortion and high transient distortion,
- medium harmonic distortion and medium transient distortion.

Number of signal files	**Parameters**
2500	harmonic3_min, harmonic3_max = 0, 0.3 harmonic5_min, harmonic5_max = 0, 0.3 harmonic7_min, harmonic7_max = 0, 0.3 harmonic9_min, harmonic9_max = 0, 0.3 harmonic11_min, harmonic11_max = 0, 0.3 transient_amount_min, transient_amount_max = 0, 0 transient_max_value_min, transient_max_value_max = 0, 0
2500	harmonic3_min, harmonic3_max = 0, 0.08 harmonic5_min, harmonic5_max = 0, 0.1 harmonic7_min, harmonic7_max = 0, 0.12 harmonic9_min, harmonic9_max = 0, 0.1 harmonic11_min, harmonic11_max = 0, 0.08 transient_amount_min, transient_amount_max = 0, 10 transient_max_value_min, transient_max_value_max = 0, 30
2500	harmonic3_min, harmonic3_max = 0, 0.02 harmonic5_min, harmonic5_max = 0, 0.03 harmonic7_min, harmonic7_max = 0, 0.03 harmonic9_min, harmonic9_max = 0, 0.02 harmonic11_min, harmonic11_max = 0, 0.01 transient_amount_min, transient_amount_max = 1, 200 transient_max_value_min, transient_max_value_max = 2, 80
2500	harmonic3_min, harmonic3_max = 0, 0.12 harmonic5_min, harmonic5_max = 0, 0.12 harmonic7_min, harmonic7_max = 0, 0.12 harmonic9_min, harmonic9_max = 0, 0.12 harmonic11_min, harmonic11_max = 0, 0.12 transient_amount_min, transient_amount_max = 1, 200 transient_max_value_min, transient_max_value_max = 2, 80
2000	harmonic3_min, harmonic3_max = 0, 0.03 harmonic5_min, harmonic5_max = 0, 0.04 harmonic7_min, harmonic7_max = 0, 0.02 harmonic9_min, harmonic9_max = 0, 0.01 harmonic11_min, harmonic11_max = 0, 0.02 transient_amount_min, transient_amount_max = 0, 5 transient_max_value_min, transient_max_value_max = 2, 10

Table 9.1: Dataset configuration. Parameters common to every subset: `num_samples = 1000`, `base_freq = 50`, `sample_rate = 5000`, `amplitude = 230`, `mse_threshold = 0.01`.

Chapter 10

Neural network

10.1 Justification of the neural network usage

The goal of this system is to increase the amount of information transferred through the same channels or information stored in the same memory in IoT measurement systems in the SG. In order to do that, data compression was utilized, but the choice of compression parameters associated with the signal is computationally demanding. On the desktop Personal Computer (PC) it was taking about 2-3 seconds to choose compression for the signal with 1000 datum points. The timing of execution was an obstacle to the scalability of the system. To accelerate the response of the parameterization component, a NN was used [135, 136].

10.2 Architecture of the neural network

The primary goal of this network is to learn the mapping from input signals to wavelet parameters, including the wavelet function, the level of decomposition, and threshold value [137]. The dataset consists of signals stored in CSV files, and the parameters are extracted from the filenames. This NN architecture is implemented using PyTorch [138]and involves several key components, including data preprocessing, NN layers, and training strategies [139].

10.3 Data Preprocessing

The preprocessing of data is crucial for the performance of the NN. In this implementation, the data is preprocessed as follows:

10.3.1 Loading Data

The signals and their corresponding labels are loaded from CSV files. Each signal is normalized to have zero mean and unit standard deviation. The labels, which include the wavelet function, level of decomposition, and threshold value, are extracted from the filenames.

Listing 10.1: Loading and Normalizing Data

```
def load_data_from_folder(folder_path):
```

```
    signals = []
    labels = []
    for file_name in os.listdir(folder_path):
        if file_name.endswith('.csv'):
            file_path = os.path.join(folder_path, file_name)
            data = pd.read_csv(file_path)
            signal = data['Signal'].values
            signal_tensor = torch.tensor(signal, dtype=torch.float32)
            signal_tensor = (signal_tensor - torch.mean(signal_tensor)) / torch.st
            signals.append(signal_tensor)
            labels.append(extract_params_from_filename(file_name))
    return signals, labels
```

10.3.2 Extracting Parameters

The parameters are extracted from the filenames using a predefined format. The wavelet function is mapped to an integer and the level of decomposition and threshold value are stored as integers and floats, respectively.

Listing 10.2: Extracting Parameters from Filenames

```
def extract_params_from_filename(file_name):
    parts = file_name.replace('.csv', '').split('_')
    wavelet_function = parts[1]
    level_of_decomposition = int(parts[2])
    threshold_value = float(parts[3])
    return [wavelet_function, level_of_decomposition, threshold_value]
```

10.3.3 Data Splitting

The dataset is split into training and validation sets using an 80/20 ratio to ensure randomness and improve generalization. This is a standard approach in many cases of NN training.

Listing 10.3: Splitting Data into Training and Validation Sets

```
split_ratio = 0.8
train_files = file_names[:int(len(file_names) * split_ratio)]
val_files = file_names[int(len(file_names) * split_ratio):]
```

The NN architecture consists of multiple fully connected layers with batch normalization and dropout for regularization [140]. The architecture is designed to predict three outputs: the wavelet function, the level of decomposition, and threshold value.

The network is defined using PyTorch's `nn.Module` class. It consists of four fully connected layers 10.1, each followed by batch normalization and dropout layers to prevent overfitting [141]. All other tested configurations resulted in training and validation loss higher than 200, in most cases about 260-270. This design allows achieving the loss below 1.

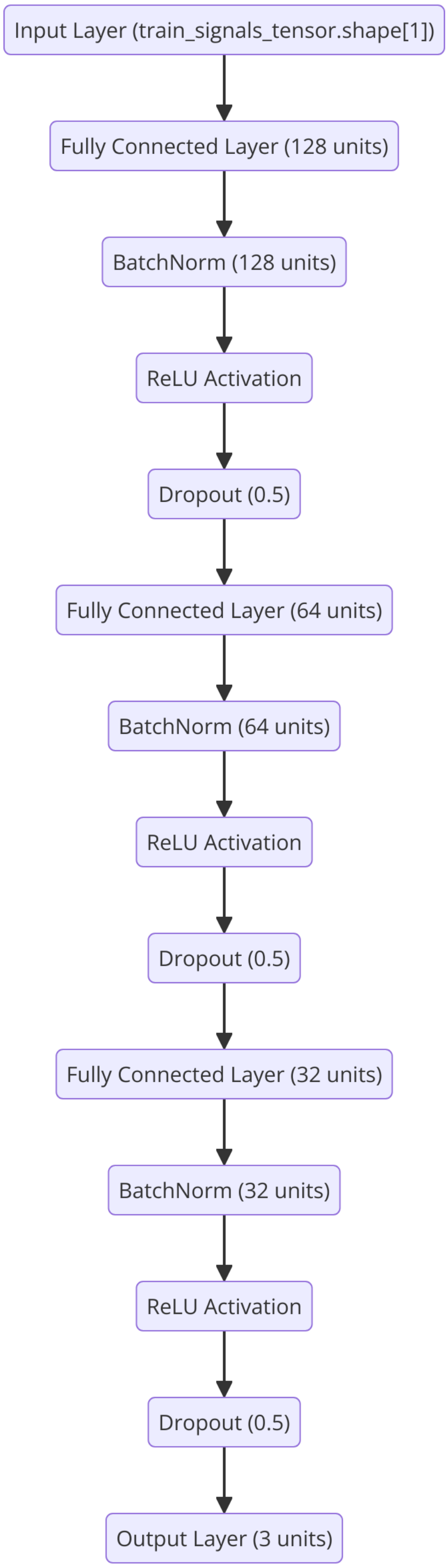


Figure 10.1: Detailed outline of NN's layers.

10.4 Training the Neural Network

The training process involves defining the loss function, the optimizer, and the learning rate scheduler. Early stopping is also implemented to prevent overfitting [142].

10.4.1 Loss Function and Optimizer

MSE loss is used as the loss function. The Adam optimizer is used with L2 regularization to minimize the loss [143].

Listing 10.4: Loss Function and Optimizer

```
criterion = nn.MSELoss()
optimizer = torch.optim.Adam(model.parameters(), lr=0.0005, weight_decay=0.005)
```

10.4.2 Learning Rate Scheduler

A learning rate scheduler is used to adjust the learning rate during training. The scheduler reduces the learning rate by a factor of 0.5 every 5 epochs.

Listing 10.5: Learning Rate Scheduler

```
scheduler = torch.optim.lr_scheduler.StepLR(optimizer, step_size=5, gamma=0.5)
```

10.4.3 Early Stopping

Early stopping is implemented to stop training when validation loss is stopped improving for a specified number of epochs. In the presented example of usage, the function was disabled, to observe the behavior of network over 100 epochs, but in the real use case, early stopping should be considered to prevent overfitting and reduce computational effort [144].

Listing 10.6: Early Stopping Implementation

```
early_stopping_patience = 10
early_stopping_counter = 0
best_val_loss = float('inf')

for epoch in range(num_epochs):
    if val_loss < best_val_loss:
        best_val_loss = val_loss
        early_stopping_counter = 0
        torch.save(model.state_dict(), 'best_model.pth')
    else:
        early_stopping_counter += 1
```

```
    if early_stopping_counter >= early_stopping_patience:
        print('Early␣stopping')
        break

    scheduler.step()
```

10.5 Evaluation and Results

The performance of the network is evaluated using the validation set. The best model is saved during training and the final model is loaded for evaluation [145].The best model saved during training is loaded for evaluation.

Listing 10.7: Loading the Best Model

```
model.load_state_dict(torch.load('best_model.pth'))
```

10.6 Neural network training results

The behavior of loss functions over epochs is crucial for evaluating the model's performance and its ability to generalize to new, unseen data. Training data is a subset of the dataset used to train the model. The model learns by adjusting its parameters (weights and biases) to minimize the error between its predictions and the actual labels. Validation data, on the other hand, is a separate subset of the dataset used to evaluate the performance of the model [146]. It helps in assessing the model's ability to generalize to unseen data and in tuning hyperparameters to prevent overfitting. The loss function measures the error between the predicted outputs and the actual labels, quantifying how well or poorly the model's predictions match the actual data. Common loss functions include MSE, Cross-Entropy Loss, and Hinge Loss. In this case, MSE was used, since it is a frequent choice to measure performance of lossy compression. The objective of training is to minimize this loss [139].

The graph 10.2 presents two curves: the training loss (blue line) and the validation loss (orange line). Both curves plot the loss values against the number of epochs. At epoch 0, the training loss starts at a high value (around 1.20), indicating a significant error in the model's initial predictions. The training loss decreases rapidly within the first 10 epochs, showing that the model is quickly learning from the training data. The training loss continues to decrease and stabilizes around a value just below 1.00, indicating that the model error on the training data has been significantly reduced and the model fits the training data well [141].

The validation loss starts at a high value and decreases rapidly within the first few epochs. It stabilizes at a similar value to the training loss, just below 1.00, and shows minor fluctuations. The convergence of training and validation losses to similar values suggests that the model generalizes well to the validation data. This indicates that the model does not overfit the training data. The sharp decline in both training and validation loss in the initial epochs is typical, as the model quickly adjusts from its random initial state to

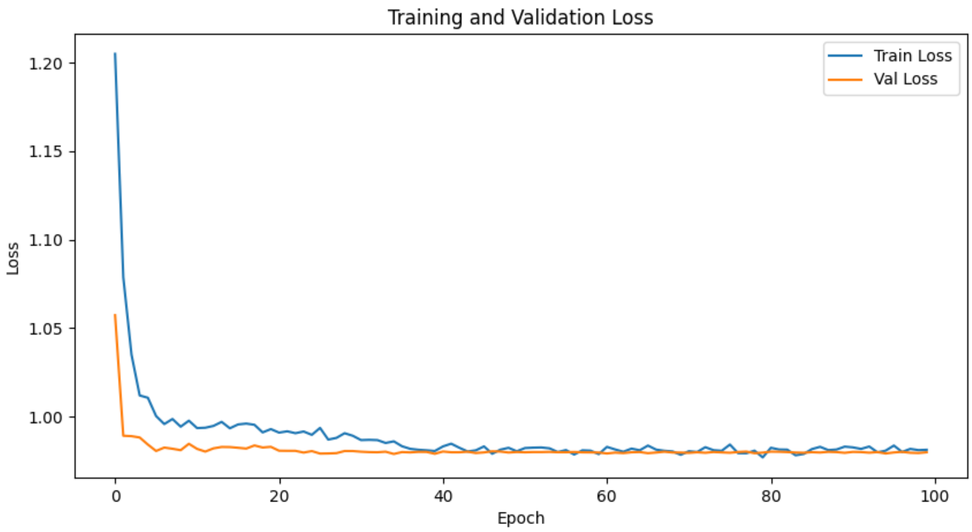


Figure 10.2: Training and validation loss of used NN.

a more stable state. The closer convergence of the training and validation losses is desirable as it indicates that the model performs consistently on both the training and validation data [147]. The stabilization of both losses suggests that the model has reached an equilibrium in which further training does not significantly improve performance, indicating that the model has effectively learned the patterns in the data without overfitting [45].

Although the model shows good generalization in this instance, it is crucial to continue monitoring for overfitting in future training runs. Techniques such as dropout or early stopping can be used if overfitting becomes evident. Continuing to explore different hyperparameter configurations can further improve model performance, including adjusting learning rates, batch sizes, and network architectures [61]. Implementing k-fold cross-validation can provide a more robust evaluation of model performance and ensure that the model generalizes well across different subsets of data. Regular evaluation of the model on validation data during training helps identify potential overfitting and underfitting early, allowing for timely adjustments to the training process.

Chapter 11

Embedded software component

11.1 Implementation Overview

The embedded system is designed to operate within the resource constraints typical of PG monitoring devices [148]. The C++ code provided implements the core functionalities of the embedded compression system. The code is structured to ensure efficient processing capabilities 11.1. The description focuses on crucial functionalities of the code, omitting auxiliary functions such as getters or setters.

11.1.1 Run Length Encoding

This file contains the implementation of RLE compression, which is used as a secondary compression step after applying DWT [149]. The function is consistent with the rest of the signal compression framework. It processes complex floating point numbers, which is a container that can hold any type of numerical data present in signal processing in the domain of PGs [150]. This module is designed to take the direct output of Huffman encoding in order to accelerate the execution of the algorithm.

Listing 11.1: RLE Compression and Decompression

```
void RLECompression::compressData(const std::vector<std::complex<double>>& data) {
    mCompressedData.clear();
    mCompressedString.clear();
    if (data.empty()) return;

    std::complex<double> lastElement = data[0];
    int count = 1;

    for (size_t i = 1; i < data.size(); ++i) {
        if (data[i] == lastElement) {
            count++;
        } else {
            mCompressedData.emplace_back(count, lastElement);
            lastElement = data[i];
            count = 1;
        }
    }
```

```
    mCompressedData.emplace_back(count, lastElement);

    std::stringstream ss;
    for (const auto& pair : mCompressedData) {
        ss << pair.first << "␣" << pair.second.real() << "␣" << pair.second.imag() << "␣";
    }
    mCompressedString = ss.str();
}

void RLECompression::decompressData() {
    mDecompressedData.clear();
    std::stringstream ss(mCompressedString);
    int count;
    double real, imag;

    while (ss >> count >> real >> imag) {
        std::complex<double> value(real, imag);
        for (int i = 0; i < count; ++i) {
            mDecompressedData.push_back(value);
        }
    }
}
```

11.1.2 Huffman encoding

This file contains the implementation of Huffman coding, which is another compression technique used in conjunction with DWT to achieve higher compression ratios [151, 152]. Most approaches operate on the bit representation of the data. In this case, the direct operation on complex floating point values was chosen, this class is intended to be used in applications specific to PG signals data compression, and omitting conversion to raw bit values simplifies the execution, making it more suitable for time-sensitive applications.

Listing 11.2: Huffman.cpp

```
\\HuffmanEncoding.hpp

struct ComplexHash {
    size_t operator()(const std::complex<double>& c) const {
        return std::hash<double>()(c.real()) ^ std::hash<double>()(c.imag());
    }
};

//HuffmanEncoding.cpp

void HuffmanEncoding::compressData(const std::vector<std::pair<int, std::complex<double>>>& data) {
    mCompressedData.clear();
    mCompressedString.clear();
    mDictionary.clear();
    if (data.empty()) return;
```

```

    std::unordered_map<std::complex<double>, int, ComplexHash> frequencyMap;
    for (const auto& pair : data) {
        frequencyMap[pair.second] += pair.first;
    }

    buildHuffmanTree(frequencyMap);

    for (const auto& pair : data) {
        for (int i = 0; i < pair.first; ++i) {
            mCompressedString += mDictionary[pair.second];
        }
    }

    for (char c : mCompressedString) {
        mCompressedData.push_back(c - '0');
    }
}

void HuffmanEncoding::buildHuffmanTree(const std::unordered_map<std::complex<double>, int, Comple
    std::priority_queue<HuffmanNode*, std::vector<HuffmanNode*>, Compare> pq;
    for (const auto& pair : frequencyMap) {
        pq.push(new HuffmanNode(pair.first, pair.second));
    }

    while (pq.size() > 1) {
        HuffmanNode* left = pq.top(); pq.pop();
        HuffmanNode* right = pq.top(); pq.pop();

        HuffmanNode* newNode = new HuffmanNode({0, 0}, left->frequency + right->frequency);
        newNode->left.reset(left);
        newNode->right.reset(right);

        pq.push(newNode);
    }

    root.reset(pq.top());

    generateCodes(root.get(), "");
}

void HuffmanEncoding::generateCodes(const HuffmanNode* node, const std::string& code) {
    if (!node->left && !node->right) {
        mDictionary[node->data] = code;
        return;
    }

    if (node->left) {
        generateCodes(node->left.get(), code + "0");
```

```
    }

    if (node->right) {
        generateCodes(node->right.get(), code + "1");
    }
}
```

11.1.3 `main()` file

This file integrates the DWT, RLE, and Huffman compression algorithms to achieve improved compression of power grid voltage signals. A crucial part of this framework is the selection of the DWT library used 11.1. Among the available options, the choice was made based on the trade-off between availability of wavelet types, performance, and ease of integration into larger systems. For that purpose, the Wavelet Packet Decomposition (WPD) was chosen, with the possibility of extension with GNU Scientific Library (GSL) and Wavelib in case of missing functions in WPD.

Listing 11.3: dataCompression.cpp

```
void DWTCompression::compressData(const std::vector<std::complex<double>>& data) {
    // Perform DWT using Wavelet2D library
    std::vector<double> realPart(data.size()), imagPart(data.size());
    for (size_t i = 0; i < data.size(); ++i) {
        realPart[i] = data[i].real();
        imagPart[i] = data[i].imag();
    }

    Wavelet2D wavelet;
    std::vector<double> compressedReal = wavelet.dwt(realPart, waveletType, decompositionLevel);
    std::vector<double> compressedImag = wavelet.dwt(imagPart, waveletType, decompositionLevel);

    compressedData.resize(compressedReal.size());
    for (size_t i = 0; i < compressedReal.size(); ++i) {
        compressedData[i] = {compressedReal[i], compressedImag[i]};
    }
}
```

11.2 Efficiency Considerations

The embedded compression system is designed to ensure efficiency in both computation and memory usage, making it suitable for time-sensitive applications to monitor the PG [157]. It processes electrical signals and transmits data with minimal delay. This is crucial for monitoring applications where timely data transmission is essential to maintain grid stability [158]. The use of lightweight algorithms ensures that the computational overhead remains low [159]. The C++ implementation is designed for performance and reduced resource consumption, enabling it to run efficiently on embedded devices with limited processing power [160]. The compression algorithms are designed to use minimal memory, which makes them suitable for deployment in resource-constrained environments typical of IoT devices [161, 162].

11.3 Parallelisation of Compression Algorithms and Neural Network Inference

Embedded devices increasingly ship with multi-core CPUs and on-board GPUs (e.g. NVIDIA Jetson, AMD Kria, Intel Movidius), making it possible to accelerate both the wavelet–based compression pipeline and any downstream neural inference without sacrificing the system's RT budget.

11.3.1 CPU Multithreading

For the **filter-bank DWT**, each decomposition level can be processed independently across signal frames: a simple `std::thread` or `std::async` pool (or Open Multi-Processing (openMP) `#pragma omp parallel for`) maps blocks of samples to different cores:

```
// Parallel 1-D DWT over frames
#pragma omp parallel for
for (size_t frame = 0; frame < nFrames; ++frame) {
    dwtForward(signalFrame[frame], coeff[frame]);
}
```

Run-length decoding and Huffman symbol lookup are largely serial, but the *build* phase (frequency histogram and tree construction) can be executed in parallel over data chunks, followed by a reduction step to merge histograms.

11.3.2 GPU Off-load with CUDA / OpenCL

Wavelet filtering is a pair of 1-D convolutions—an archetypal Graphics Processing Unit (GPU) workload. Porting the low- and high-pass stages to Compute Unified Device Architecture (CUDA) or Open Computing Language (OpenCL) involves:

1. Uploading frames to global GPU memory (using page-locked host buffers for zero-copy on Jetson-class devices).

2. Launching a kernel where each thread computes a single output sample:

```
// CUDA kernel: low-pass
__global__ void lpKernel(const double* __restrict__ x,
                         const double* __restrict__ h,
                         double* __restrict__ y, int N, int F) {
    int idx = blockIdx.x * blockDim.x + threadIdx.x;
    if (idx < N/2) {
        double acc = 0.0;
        #pragma unroll
        for (int k = 0; k < F; ++k)
            acc += x[2*idx-k] * h[k];
        y[idx] = acc;
```

```
        }
    }
```

3. Applying the same pattern to the high-pass branch and to inverse reconstruction.

Huffman bit-plane coding benefits from GPU warp-wide ballot instructions (`ballot_sync`) to pack bits quickly, while RLE can be implemented with the standard CUDA `cub::DeviceRunLengthEncode` primitive.

11.3.3 Parallel Neural-Network Inference

If compressed data feed a lightweight classifier or anomaly detector, the same hardware can be re-used for inference:

- **CUDA/cuDNN / TensorRT** on NVIDIA System on a Chip (SoC): export the trained network (Open Neural Network Exchange (ONNX)) and build an INT8 or FP16 engine that streams directly from the GPU wavelet workspace, eliminating extra copies.
- **OpenCL / SYCL** on AMD or Intel Field-programmable Gate Array (FPGA)s: map each layer to an out-of-order command queue so convolution, activation and memory transfers overlap.
- **CPU thread pools** (std::jthread, Intel Threading Building Blocks (TBB)) for Microcontroller Unit (MCU)s without a GPU: pin one core to decompression and the remaining to a batched matrix-multiplication kernel (e.g. CMSIS-NN or `Eigen::Tensor`).

This hybrid strategy—multi-core CPU for entropy coding, GPU for wavelet filtering, and a unified accelerator path for inference—yields a $2-8\times$ throughput gain versus the scalar baseline while staying within the power envelope of typical field-deployable PG monitors.

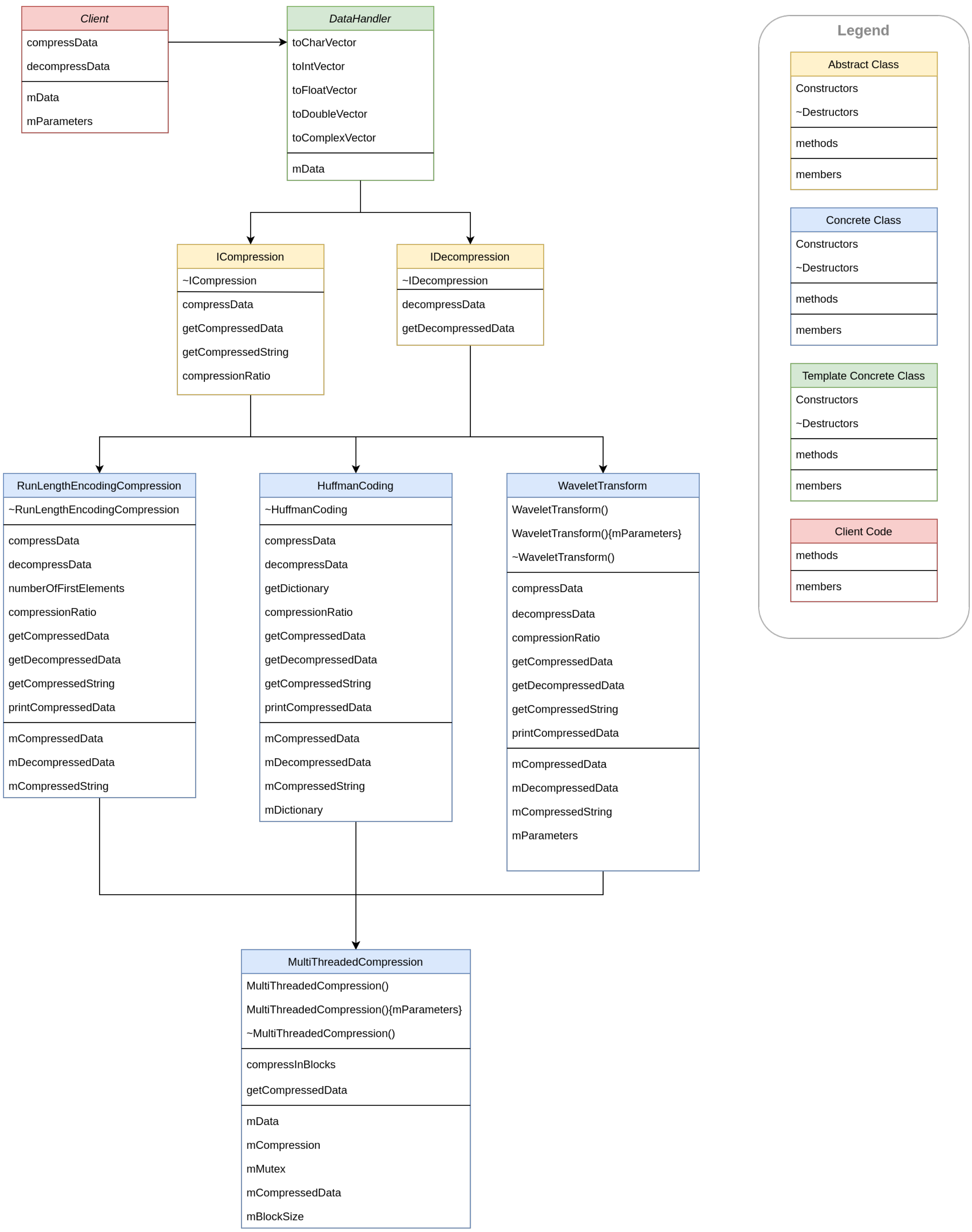


Figure 11.1: Detailed structure of embedded part of the system. The client can interact with the code by calling compression or decompression function. If the data is provided in an unknown format, DataHandler class can be used as an adapter between compression functions and client interface. Compression and decompression algorithms are accessible through interface classes. Implementation allows multi-threaded compression.

Library Name	**Strengths**	**Weaknesses**
WPD [153]	1. Comprehensive DWT and wavelet packet implementation. 2. Supports various wavelet types. 3.Good documentation and community support.	1. May have a steep learning curve. 2. Performance could be improved for large datasets.
GSL [154]	1. Highly efficient. 2. Extensive set of mathematical routines beyond wavelet transforms.	1. Requires understanding of GSL's general framework. 2. Limited to certain types of wavelets.
OpenCV [155]	1. Well-known and widely used library. 2. Provides additional functionalities for image processing. 3. Strong community and extensive documentation.	1. Primarily focused on computer vision, so wavelet support is limited. 2. Might be overkill for applications solely focused on wavelet transforms.
Wavelib [156]	1. Robust library for wavelet transforms. 2. Supports a wide range of wavelet functions.	1. Documentation can be sparse and difficult to navigate. 2. Less active development and community support compared to other libraries.

Table 11.1: Overview of C++ DWT Libraries

Chapter 12

Results

The proposed system was evaluated in three critical performance categories: CR, compression quality, and time overhead. The empirical results aligned closely with the initial hypotheses. In addition, an analysis was conducted to compare the proposed algorithm with the top performing methods presented in the recent literature [163, 67, 43, 4, 37, 158]. The main focus of the research is to validate the algorithm with signals that contain distortion with harmonics and transients, since preservation of these qualities will render the algorithm useful for protection and PQ monitoring systems. Evaluating compression loss is a difficult task, and no universal metrics has been set. In presented research MSE is used as a basis of quality comparison. However, this metric has its drawbacks, since in some cases it can yeld misleading results. For example, if the long part of the signal is a perfect sinusoid, but contains a transient, a MSE might be low if the reconstructed signal losses the transient it still may have very low MSE, but the reconstructed signal might be useless for protection or PQ analysis. For this reason, comparing the loss of signals based on different signal processing methods is a difficult task. Results are compared only with other DWT-based methods, since methods based on other primitves such as phasor or DCT might give misleading methods, since these methods are specialized in compression of different types of signals. This thesis focuses in DWT, since primary intention of proposed algorithm is to be useful for signals distorted with transients, which are crucial for applications such as PQ analysis or protection.

After running the algorithm with real data, an improvement in balance between CR and compression loss was observed. Due to the limited availability of real data, a trial with synthetic data was conducted to further examine the capabilities of the algorithm. Supplying algorithm with large quantity data with high quality resulted in both - CR and compression loss improvement.

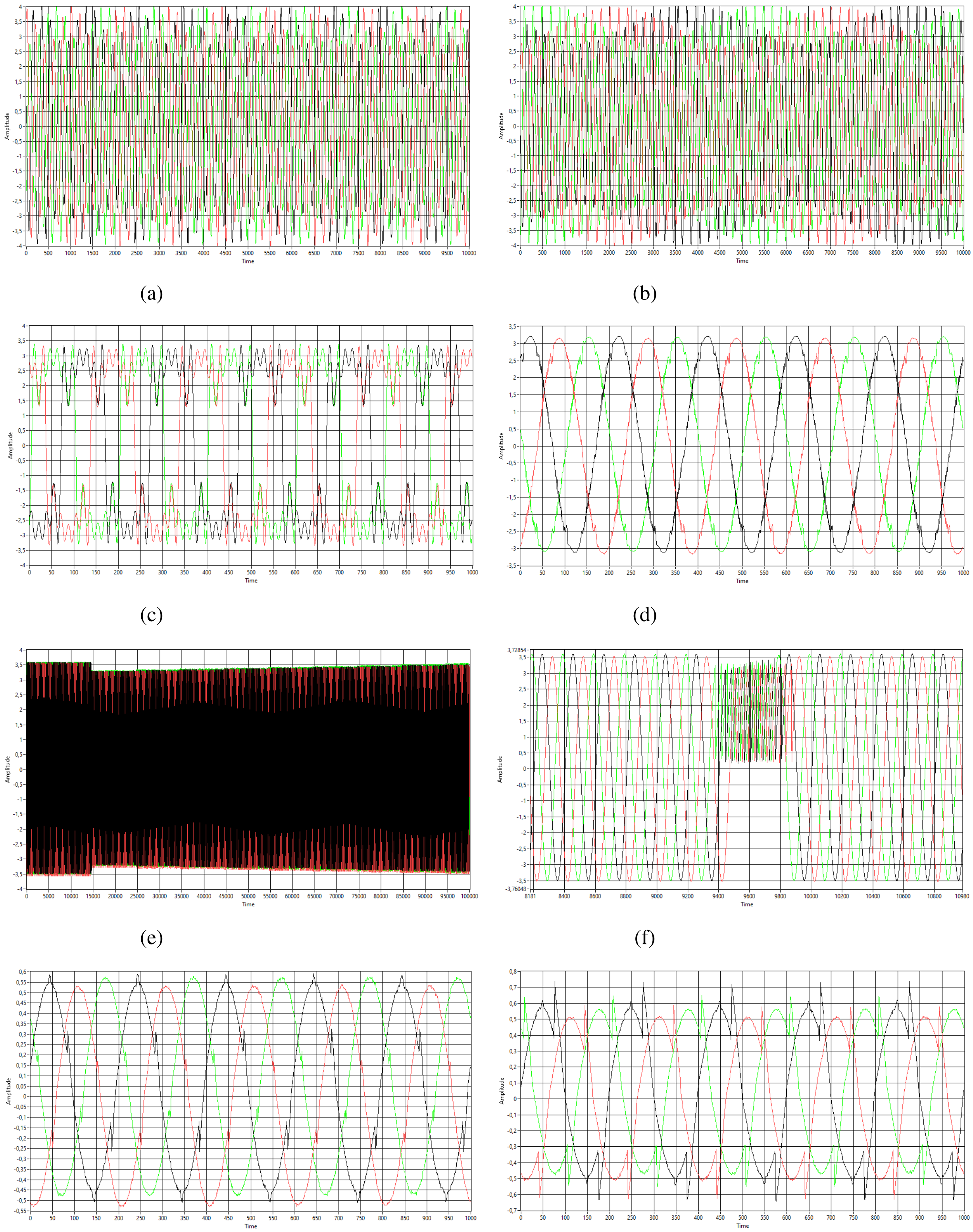
Results shown below should not be compared to different compression methods directly, since usually compression algorithms consists of variety of steps. This research aims to improve DWT component that can be used to replace presently used statically parameterized DWT in DC algorithms, rather than use it as a standalone compression method. The adaptive method was compared to the methods of DWT parameterization presented in modern literature.

12.1 Real data

To validate the algorithm, measurements were made to obtain data from various scenarios that occur in PG. Measurements were made in a laboratory simulating real wordl scenario, with different loads and sources of energy. Scenarios for voltage measurements included: harmonic distortion, interharmonic distortion, transient distortion, island powered with generator (unstable grid with significant noise), linear and non-linear loads. Scenarios for current measurement included commutation of non-linear load with different commutation points 12.1. Examples of the signals are presented on 12.1 After division of the obtained signals into chunks of 1000 datapoints, 547 signals were obtained. Data augumentation was performed with denoising the data with the Savitzky-Golay filter, denoising with the Kalman filter, and adding noise, resulting in 2188 files. After all of the signals were labeled, they were used for NN training.

Scenario	**Ranges of Distortion**	**Measured Value**
Linear load supplied from the grid	N/A	Voltage
Generator powered island with no load	N/A	Voltage
Island with non-linear load	N/A	Voltage
Island with linear load	N/A	Voltage
Harmonics	3rd, 5th, 7th, 9th, 11th harmonic, set to: 4.6 V / 9.2 V / 13.8 V / 18.4 V / 23 V / 27 V / 32.2 V / 36.8 V / 41.4 V / 46 V	Voltage
Interharmonics	52 Hz / 55 Hz / 59 Hz each with 5%, 10%, 20% of grid voltage	Voltage
Transients	Overvoltage, undervoltage, power loss	Voltage
Transients	Commutation of thyristors	Current

Table 12.1: Scenarios for data acquisition



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 12.1: Registered signals illustrating different types of distortion: (a) and (b) flicker, (c) harmonic distortion, (d) island with non-linear load, (e) and (f) voltage transients, (g) and (h) thyristor commutation.

12.1.1 Compression ratio

The CR was examined by running the algorithm on 100 randomly selected signals that were excluded from the learning data set. The CR for the DWT parameterized with the proposed method is on a level similar to other metods, meaning that this metod offers similar size of the encoded data to the best methods available 12.2.

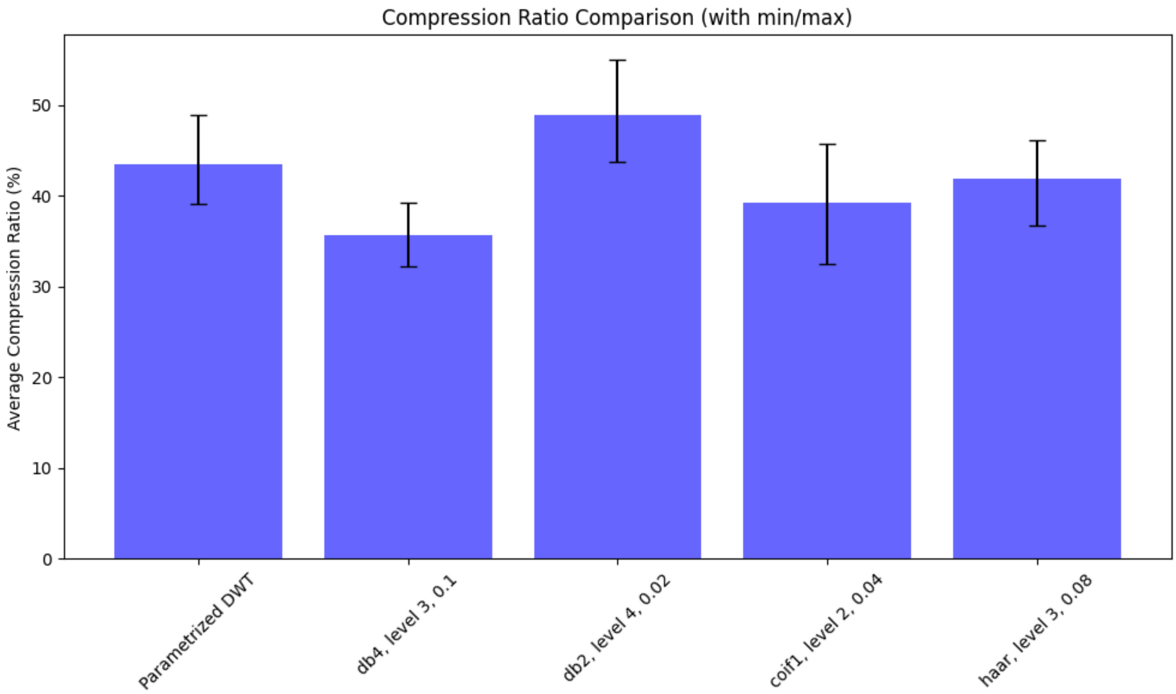


Figure 12.2: Comparison of CRns between all tested algorithms.

12.1.2 Compression loss

MSE between original signal and decompressed one was calculated as a baseline for asserting quality of the compression. Compared to other algorithms, this benchmark showed significant improvement (two-fold reduction, compared to the second-best algorithm in the category) 12.3. Considering similar CR, parametrized DWT offer better quality of the restored signal, while taking up a similar amount of storage space or transmission medium.

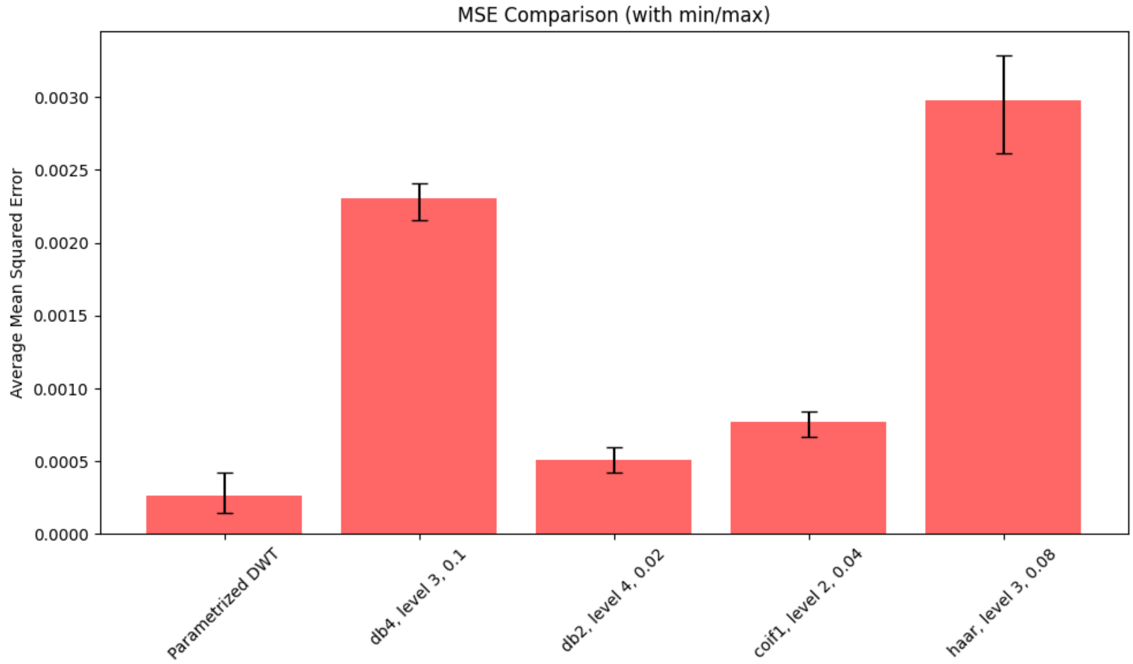


Figure 12.3: MSE statistics for all compared methods.

12.1.3 Conclussion

Although proposed parametrization method doesn't beat other algorithms in all metrics, it shows that similar CR can be achieved with smaller MSE. Presented method offer improved tradeoff between compressed data size, and reconstructed signal quality than algorithms proposed in current literature. NN was trained with limited dataset, due to high cost of obtaining high-quality laboratory measurements. About 500 signals, augmented to 2000 enabled algorithm to compress data with better reconstruction quality, while maintaining similar CR.

12.2 Synthetic data

Testing algorithm with real data brought promising results; however, the quantity of the data seemed to be a bottleneck for the method. To verify that, tests with more signals are needed. The structure of PG signals permits for reliable generation. Synthetic data also offer a very high degree of control over quantity and quality of distortion. Using synthetic data will help in researching the capabilities of an algorithm that supplies a larger amount of data. It will also provide a deeper analysis of the correlation between distortion and compression performance.

12000 samples of signals were generated to train the NN. Signals contain various distortions including transients and harmonics, which are described in more detail in chapter 9 - Dataset.

12.2.1 Compression ratio

Comparing to other algorithms, average CR for proposed solution was about 2 times better. This proves, that supplying more high quality data in the traing phase, further improves the performance of the algorithm. Achieving better CR shows that this algorithm may bring significant resource savings proportional to the scale of the system.

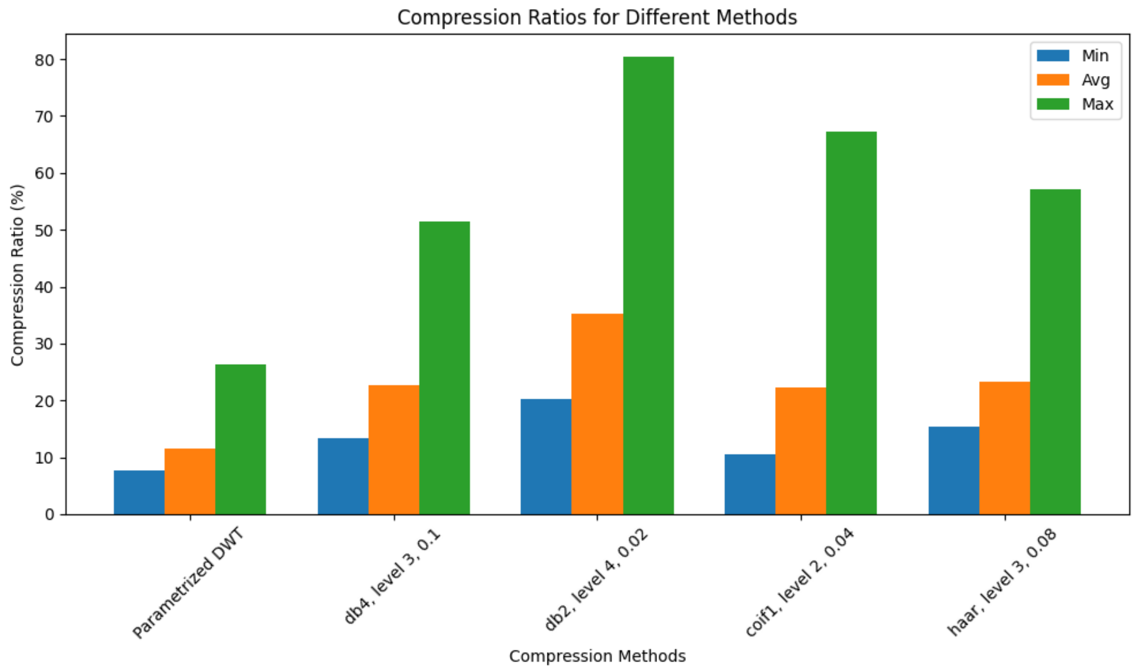


Figure 12.4: Comparison of CRns between all tested algorithms.

The data summarized in Table 12.2 indicates significant improvements in the CR. The proposed algorithm achieved a minimum CR improvement of 50.33%, a maximum of 60.18%, and an average improvement of 50.53% when compared to the second-best algorithm in this category.

Parameter	Value difference
Minimum CR	-50.33%
Maximum CR	-60.18%
Average CR	-50.53%

Table 12.2: Comparison of CR of proposed and second best algorith in category.

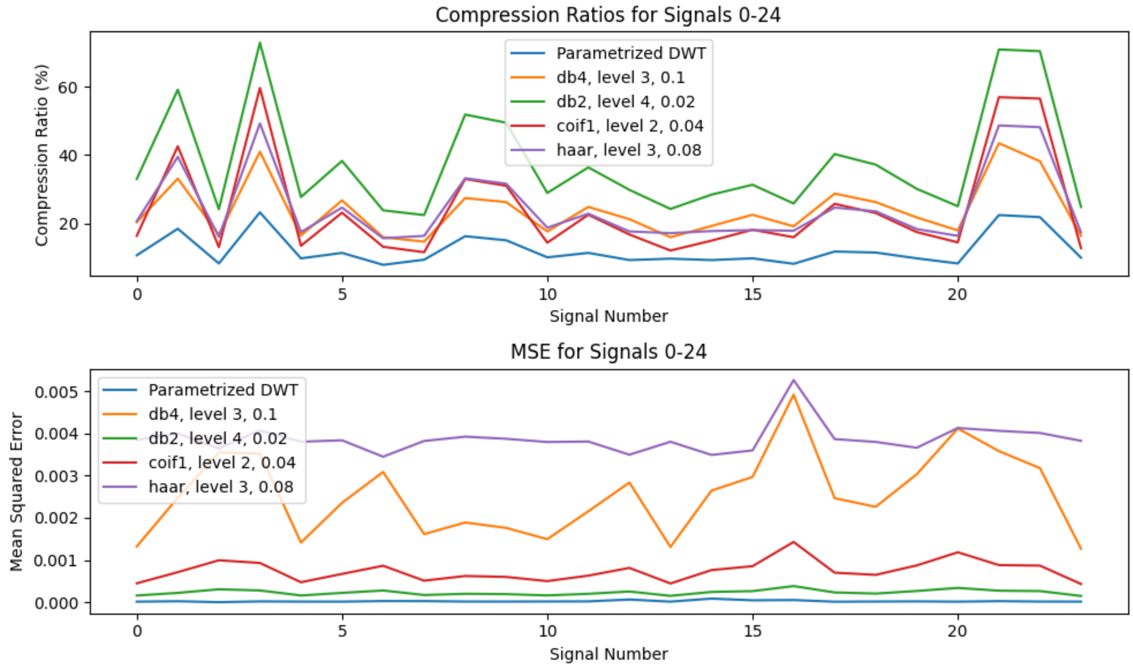


Figure 12.5: Average CR and MSE for each signal in the first group. The level of disturbance is rising with each consecutive signal.

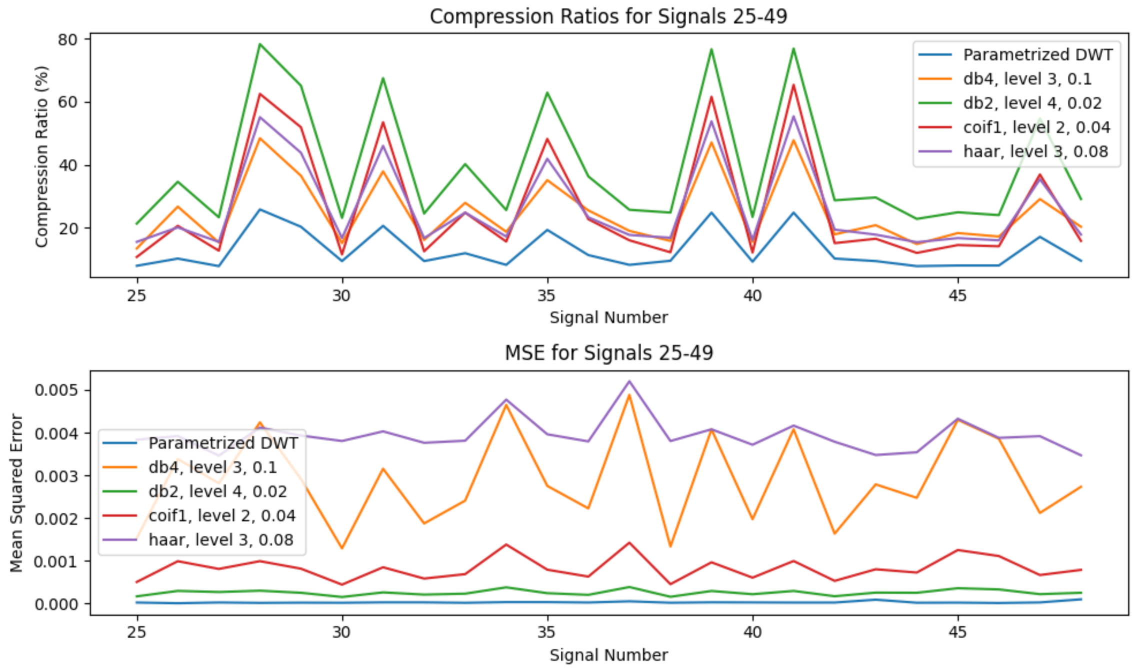


Figure 12.6: Average CR and MSE for each signal in the second group. The level of disturbance is rising with each consecutive signal.

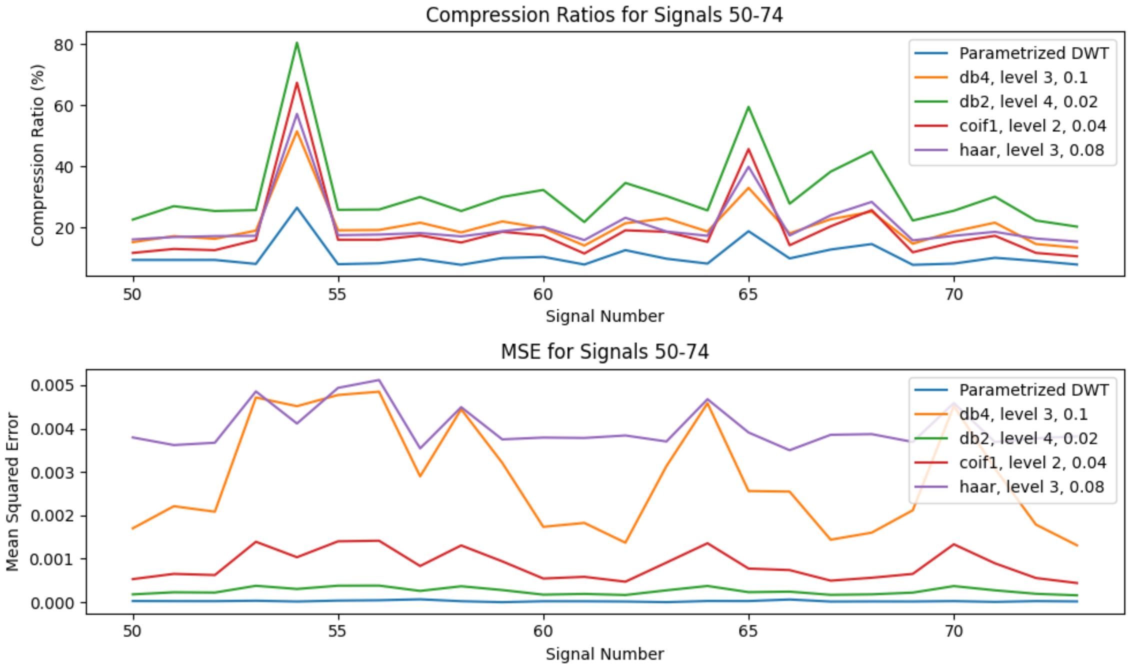


Figure 12.7: Average CR and MSE for each signal in the third group. The level of disturbance is rising with each consecutive signal.

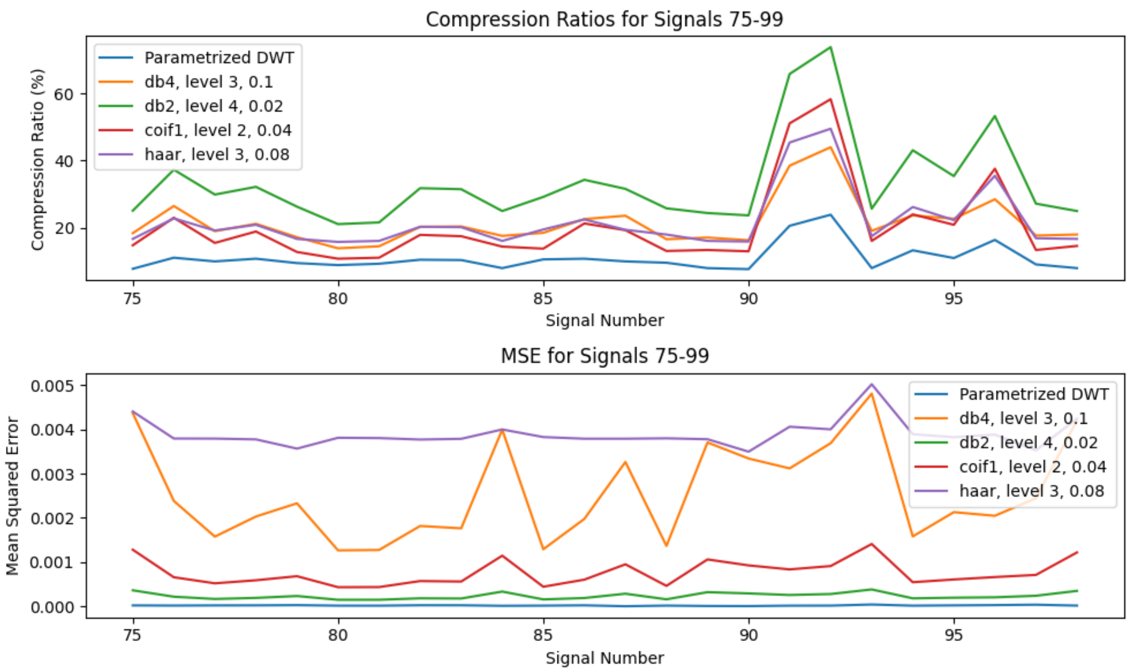


Figure 12.8: Average CR and MSE for each signal in the fourth group. The level of disturbance is rising with each consecutive signal.

As presented in the graphs 12.5, 12.6, 12.7 and 12.8, the proposed solution offered the best CR for each sample in the dataset. The average CR was significantly better than that obtained by using a wavelet transform of constant parameters. This compression algorithm can adapt to the changes in the signal, which is an

Signal group	Distortion type	Transient range	THD range
0-24	High harmonics Low transients	N/A	2.2%–56%
25-49	Low harmonics High transients	Quantity: 1–100 Magnitude: 1–60V	0.3%–10.9%
50-74	Medium harmonics Medium transients	Quantity: 1–20 Magnitude: 1–20V	0.9%–21.7%
75-99	Low harmonics Low transients	Quantity: 1–10 Magnitude: 1–10V	0.3%]–6.9%

Table 12.3: Comparison of CR of proposed and second best algorith in category.

advantage in modern SG, where environment of the system may change frequently. Each of the signal groups had different distortion types 12.3, to test the algorithm in various scenarios. The amount of harmonics and transients was growing linearly in all signal groups, however CR and compression loss are not following the linear trend, which shows that parametrization of wavelet transform cannot be done using simpler methods like a decision tree.

12.2.2 Compression loss

Depending on the parameterization of the NN's training data, the level of MSE may vary 12.12. In this case, the MSE produced by the compression method was higher than by the statically parametrized methods, but was still within an acceptable range. In most cases, the reconstructed signal has shown that changes of the value were preserved. The algorithm proves to be very effective in the preservation of transients, which is shown in 12.9. Signals containing harmonic distortion do not lose that information during the process 12.11. The algorithm also offers high reconstruction quality for signals with minor distortion, while offering a very high CR for such signals 12.10.

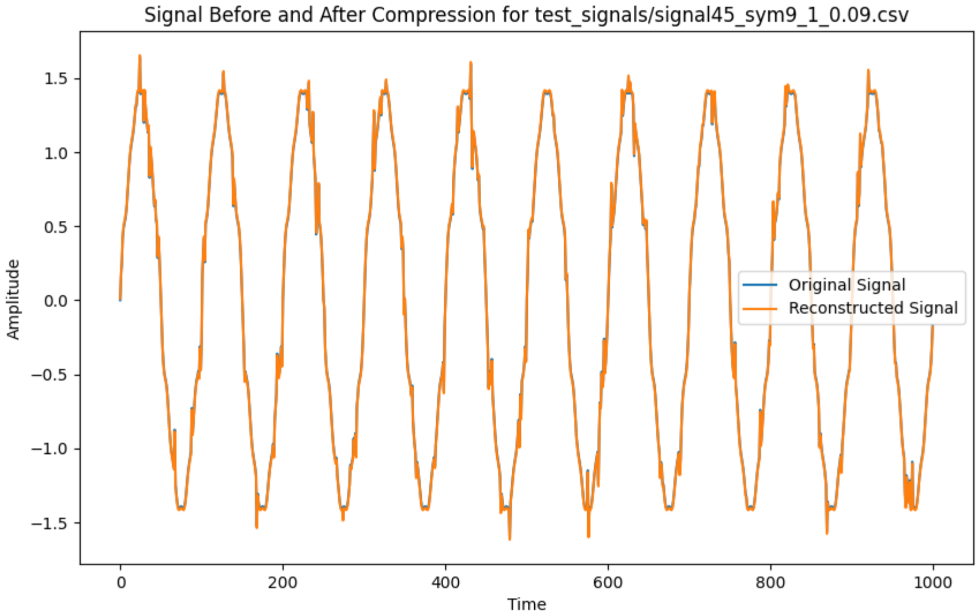


Figure 12.9: Compression loss for signal with significant transient distortion.

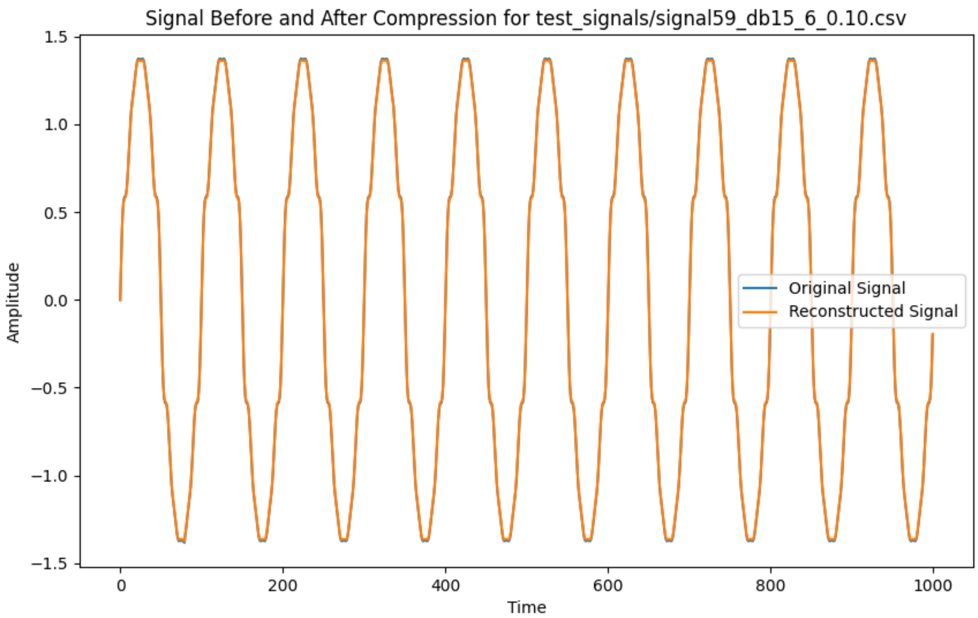


Figure 12.10: Compression loss for signal with minor harmonics distortion and no transient distortion.

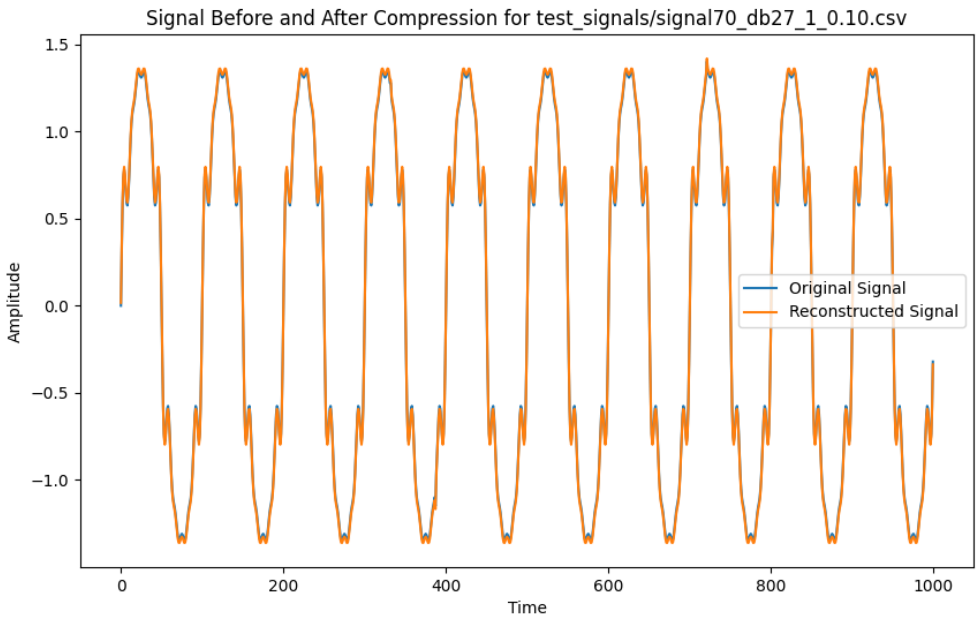


Figure 12.11: Compression loss for signal with significant harmonics distortion.

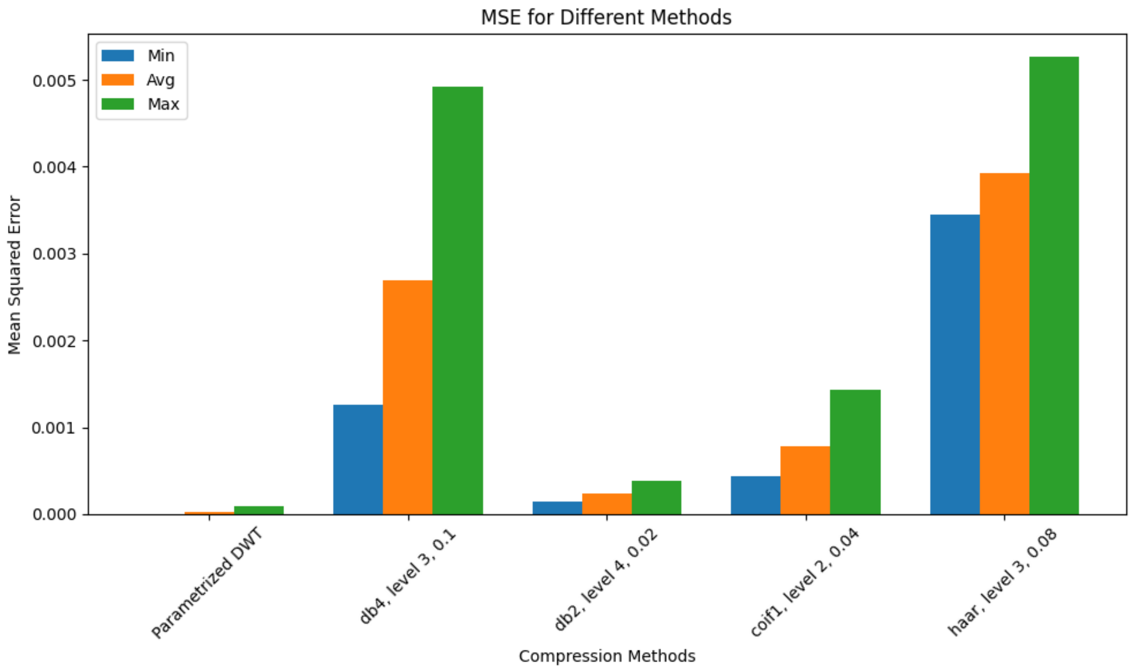


Figure 12.12: MSE statistics for all compared methods.

12.2.3 Conclussion

After using more signal files, with confidence that each contains predefined distortion, the algorithm showed much better performance, toping state-of-the-art solutions in both categories, CR, and compression loss. This shows that the performance of the algorithm improves proportionally to the amount and quality of data that is used to train the NN. Comparison of CR and quality against series of signal with linear rise of distortion amount shows that wavelet compression performance does not change proportionally to the distortion content. This shows that adapting DWT parameters cannot be done in a simple way like decision tree.

12.3 Time of parametrization

The validation of time overhead is crucial for this solution, as it is intended to be used in large-scale IoT systems with limited resources. The low timing overhead is crucial for this solution to be effective and useful. Compared to doing Bayesian Optimization directly on the provided data, this solution consumes significantly less resources. The average execution time of 0.0011s permits for a 78 545 454 calibrating operations per day while running the script on a low-end PC. By 2028, Europe is projected to use 326 million SMs [164], which means that it would take less than 5 days to parameterize every SM in Europe using a single PC.

Computation time	**NN (proposed)**	**Bayesian Optimization**	**Difference**
Minimum	0.0006s	2.1888s	3648 times faster
Maximum	0.0049s	3.8016s	775 times faster
Average	0.0011s	2.8743s	2613 times faster

Table 12.4: Comparison of computation times for DWT parameters. Measurements were done on Dell Inc. Latitude 5400 PC, 16 GB RAM, Intel® Core™ i5-8365U CPU @ 1.60GHz × 8, Ubuntu 22.04.4 LTS 64-bit, Python 3.10.12

In conclusion, the proposed system demonstrates superior performance in terms of CR, decompressed signal quality, and time overhead compared to existing methodologies. The empirical evidence supports its potential for practical applications in dynamic and resource-constrained environments.

Chapter 13

Contributions

The main contributions of this research project are as follows:

1. **A novel way of DWT parameterization**, that provides a better CR than state-of-the-art solutions, while reducing compression loss. The parameterization method targets DC of PG signals with transient and harmonic distortion to improve performance of PQ analysis and protection systems.

2. Method to label signals with wavelet transform parameters that offer the best CR within declared threshold of MSE for the defined search space.

3. The CR and MSE were examined across the data sets in a simulation to verify the influence of signal characteristics (THD and transients) on compression performance.

4. An exhaustive **review of data compression in IoT** sensors network.

5. **A system architecture** that can work with real signals provided by the user or generate synthetic signals was prepared and presented.

6. **Sensors software in C++** was prepared to be implemented directly on end-point devices.

7. **NN** that is able to greatly reduce the time of parameterization of the DWT based compression system was introduced.

8. **Theoretical study** on DC with a focus on parameterization of usage of DWT for compression.

The goal of this project was to develop a solution that will improve DC algorithms using DWT. The method is beneficial in particulat for compression of singals with significant distortion, especially transients. Such signals are important in applications like PS protection or PQ analysis.

Chapter 14

Summary

The goal of this research was to improve a way DWT is utilized to compress the data in the domain of SG. Initially, a comprehensive review of the literature was conducted to target efforts and define areas that needed improvement. Following trends was an important aspect of the state-of-the-art review, since IoT systems often bear a great responsibility, and using algorithms that are not verified may not be welcomed. The literature review also presents the need for reduction in data stored and transmitted by IoT systems.

The primary achievement is the creation of an adaptive DWT parameterization system that offers a better CR than existing solutions. This system is not only theoretical but is also implementation-ready, meaning that all of its core components have been prepared and tested. These components include data preparation software, a NN, embedded software, and a validation framework. The development and testing of these components ensure that the system can be deployed in real-world scenarios without significant modifications. The creation of the adaptive DC system involved several key steps. Initially, a thorough analysis of current DC methods was performed, focusing on those applicable to IoT sensor networks in SG. This exhaustive review provided insights into the strengths and weaknesses of existing technologies and helped to identify the specific requirements for an improved compression system. The need for efficient DC in IoT networks is paramount due to the large volumes of data generated by sensors, which must be stored and transmitted efficiently to avoid overwhelming network resources.

During the literature review, various compression algorithms were examined, focusing on those that were highly efficient and suitable for implementation in resource-constrained environments like IoT sensor networks. The review covered a wide range of techniques, including lossless and lossy compression methods, each with their own advantages and limitations. The analysis identified that many existing solutions, while effective in some scenarios, did not fully meet the specific needs of IoT systems, particularly in terms of RT data processing and energy efficiency.

The architecture of the proposed system was designed and presented as part of the project. This architecture outlines the flow of data through the system, from the initial capture of data by IoT sensors to the final compressed output. Key components of the system include a NN designed to parametrize the compression process, embedded software developed in C++ for deployment on user devices, and a validation framework to ensure the reliability and efficiency of the compression. One of the significant innovations in this project is the development of a NN capable of significantly reducing the time required for the parametrization of a

DWT-based compression system. This NN selects the parameters needed for the DWT, thereby enhancing the overall compression efficiency. By reducing the time and computational resources needed for parameter selection, the NN makes the compression system more suitable for RT applications in IoT networks. The NN was trained using a comprehensive dataset that represented a variety of data types and conditions in IoT sensor networks. This training process involved fine-tuning the NN's parameters to ensure it could generalize well to new, unseen data, thereby maintaining high compression efficiency across different scenarios. Insights about features of the signals and compression efficiency were also discussed. The resulting NN not only improves the CR but also reduces the latency associated with data processing, making it an ideal solution for time-sensitive applications. Presented design lets the system do the parameterization in parallel to the normal operation, which will not any runtime overhead.

The embedded software component of the system was developed in C++, chosen for its performance and compatibility with a wide range of IoT devices. This software is designed to be directly implemented on users' devices, allowing for seamless integration with existing IoT infrastructure. The development process included rigorous testing to ensure that the software performs reliably under various conditions and scenarios. The C++ software was developed with a focus on modularity and flexibility, allowing it to be easily adapted to different hardware platforms and sensor types. A variety of testing methodologies were employed, including unit tests, integration tests, and field tests, to validate the software performance. These tests ensured that the software could handle the typical data rates and volumes of IoT sensor networks without introducing significant overhead or latency.

In addition to the software components, a comprehensive validation framework was developed to test the efficacy of the DC system. This framework includes tools and methodologies for assessing the performance of the compression algorithms, ensuring that they meet the required standards for accuracy and efficiency. The validation framework is a crucial part of the project, as it provides the means to verify that the system works as intended and can be trusted for use in critical IoT applications.

The validation framework was designed to simulate a wide range of operating conditions and data scenarios, allowing a thorough evaluation of the compression system performance. This included testing the system with different types of sensor data, varying levels of data noise and redundancy, and different network conditions. The results of these tests demonstrated that the compression system consistently outperformed existing solutions in terms of both CR and MSE.

The theoretical study conducted as part of this project focused on the parametrization of wavelet compression. This study provided the foundational knowledge necessary to develop the NN and other components of the compression system. By exploring the theoretical aspects of DC, the most effective techniques for selection of the compression parameters and improving overall system performance were identified.

The theoretical research included an in-depth analysis of WT techniques, examining various wavelet functions and their suitability for different types of IoT sensor data. This analysis helped to identify the most promising wavelet functions for the compression system, leading to further refinements in the NN's design process. The study also explored the mathematical underpinnings of wavelet compression, providing valuable insights into the trade-offs between compression efficiency and computational complexity.

Overall, this research project has made significant contributions to the field of DC for IoT networks. Especially for compression signals with transient distortion. The development of an adaptive compression system, backed by a robust theoretical foundation and practical validation, provides a valuable tool to improve the efficiency of IoT systems. The comprehensive approach taken in this project, from literature review to system validation, ensures that the developed solution is both innovative and practical, ready to meet the demands of modern IoT applications.

The project's outcomes not only address current challenges in IoT DC but also pave the way for future advancements. The adaptive nature of the developed system allows it to evolve with technological advancements, ensuring its relevance and applicability in the ever-changing landscape of SG technology. This research has laid the groundwork for further exploration and development in this critical area, with the potential to significantly impact the efficiency and effectiveness of SG systems.

Future work in this area could explore several promising directions. One potential avenue is the integration of additional ML techniques to further enhance the adaptive capabilities of the compression system. For example, reinforcement learning could be used to dynamically adjust compression parameters in RT based on the specific characteristics of the data being processed. Additionally, the system could be expanded to support a broader range of data types and compression algorithms, providing even greater flexibility and efficiency.

Another important area for future research is the parameterization of the compression system for energy efficiency. IoT devices often operate on limited power sources, such as batteries or energy harvesting systems, making energy efficiency a critical consideration. By incorporating energy-aware algorithms and optimization techniques, the compression system could further reduce its power consumption, extending the operational lifetime of IoT devices and improving overall system sustainability.

Finally, the deployment of the compression system in real-world IoT applications will provide valuable feedback and insights that can be used to refine and improve the system. Collaborations with industry partners and field tests in diverse IoT environments will help identify practical challenges and opportunities for enhancement, ensuring that the compression system continues to evolve and meet the needs of future IoT applications.

In conclusion, the research project has successfully developed an adaptive DC system that offers significant improvements over existing solutions. Through a combination of theoretical research, practical development, and testing of real use cases, the project has created a fully implemented solution that addresses the unique challenges of DC in IoT sensor networks. This work represents a step forward in the field, providing a foundation for othed adaptive compression techniques and real-world applications in IoT technology.

Chapter 15

Directions for further research

Even though proposed solution provides significant benefits, by no means it exhausts possibilities for the development in the field of DC, DWT based methods, and Artificial Intelligence-supported parametrization. Possible areas of improvement are:

- **Improvement of signal reconstruction quality** - One potential area for further research is the reduction of MSE introduced during the compression process. While current algorithms achieve a balance between CR and data fidelity, there is always room for improvement. Researchers could explore advanced error minimization techniques that leverage ML models to predict and correct compression-induced errors. Additionally, the development of more sophisticated wavelet functions specifically tailored to minimize MSE for different types of sensor data could lead to significant enhancements in data accuracy post-compression. Improved quantization techniques can significantly reduce MSE by ensuring that the mapping process introduces minimal distortion. Researchers can explore adaptive quantization methods that adjust the quantization levels based on the signal characteristics, leading to more accurate representations of the original data. Adopting cross-disciplinary approaches that combine insights from signal processing, ML, information theory, and domain-specific knowledge can lead to innovative solutions for reducing MSE. Collaboration between researchers from different fields can result in the development of novel compression techniques that leverage the latest advancements in each discipline. For instance, combining the principles of information theory with ML models can help design compression algorithms that are both efficient and robust.
- **Algorithm determinism** - It is very difficult to create a deterministic algorithm based on NN, however it is possible to reduce the level of uncertanity. That approach might be needed to popularize this framework in IoT systems that usually bear huge responsibility in domains such as aviation, space, healthcare, automotive or industry.
- **Further increase in compression ratio** - Achieving even higher CRs without sacrificing data integrity remains a key goal. Future research could focus on developing innovative compression algorithms that push the boundaries of current methodologies. This might include exploring new mathematical models and transformations that offer superior compression capabilities. Additionally, integrating more advanced Artificial Intelligence techniques, such as deep reinforcement learning, could

enable the dynamic adjustment of compression parameters in RT, improving the CR based on the specific characteristics of the data being processed.

- **Reduction of used memory** - Reducing the memory footprint of compression algorithms is essential for their application in memory-constrained IoT devices. Research could be directed towards the development of more memory-efficient algorithms that maintain high compression performance. This might involve adapting the data structures and algorithms used in the compression process or developing new methods for efficiently managing memory usage. Additionally, exploring hardware-software co-design approaches, where both the hardware and software are designed together, could lead to significant reductions in memory requirements.

- **Application in different domains, such as audio, image, video encoding** - The application of the proposed compression techniques in different domains, such as audio, image, and video encoding, represents a promising area for further research. Each of these domains has unique characteristics and requirements, and adapting the compression algorithms to meet these needs could lead to significant advancements. For instance, the development of specialized wavelet functions and NN architectures tailored for image or video data could enhance compression efficiency and quality. Additionally, cross-domain research could lead to the discovery of new techniques and approaches that benefit multiple application areas. Exploring the potential for these compression techniques in the audio domain could involve fine-tuning algorithms to handle the nuances of sound waves and auditory data, thereby improving audio quality while reducing file size. Similarly, investigating how these techniques can be applied to streaming video content could have substantial implications for bandwidth usage and streaming quality, making high-definition video more accessible in various network conditions. The integration of ML models trained on domain-specific datasets can further improve compression algorithms, leading to more efficient storage and transmission of multimedia data. By continuing to explore and refine these techniques across different domains, researchers can unlock new levels of performance and utility in digital media compression, paving the way for more advanced and adaptable compression solutions. This holistic approach not only improves individual domain applications but also fosters innovation that transcends traditional boundaries, ultimately contributing to the evolution of compression technology as a whole.

- **Increased generalization of neural network** - Enhancing the generalization capabilities of the NN used in the compression system is another important research direction. A NN that generalizes well can effectively compress a wide variety of data types without requiring extensive retraining. Future research could focus on developing more robust NN architectures and training methodologies that improve generalization. This might involve the use of advanced regularization techniques, data augmentation strategies, and transfer learning approaches to enable the NN to learn from diverse datasets and perform well in different scenarios.

- **Further improvement of embedded sub-system** - Further improving the embedded sub-system for improved performance and efficiency is a key area for ongoing research. This could involve refining the software algorithms to reduce computational overhead and enhance processing speed. Additionally, exploring new hardware designs and architectures that are specifically designed for the compression tasks could lead to significant performance gains. This might include the development of custom processing units or the integration of specialized co-processors that accelerate specific parts of the compression process.

- **Compliance with communication protocols** - Ensuring compliance with various communication protocols used in IoT networks is crucial for the widespread adoption of the compression system. Future research could focus on developing compression algorithms that are compatible with existing and emerging communication standards. This might involve adapting the algorithms to meet the specific requirements and constraints of different protocols, such as low latency, low bandwidth, and high reliability. Additionally, researchers could explore ways to integrate the compression system seamlessly with different types of network infrastructure, ensuring interoperability and ease of deployment.

- **Adding methods that will increase the security of transmitted and stored data** - Enhancing the security of transmitted and stored data is an essential aspect of future research. This could involve developing new methods for securing compressed data, such as integrating encryption algorithms with the compression process to provide end-to-end data protection. Additionally, researchers could explore techniques for detecting and mitigating potential security threats, such as data tampering or unauthorized access. Ensuring that the compression system meets stringent security standards will be crucial for its adoption in sensitive applications, such as SG systems and other critical infrastructure. One promising approach could be the development of hybrid algorithms that combine the strengths of both compression and encryption techniques, ensuring that data remains secure without compromising on compression efficiency. These algorithms could be designed to dynamically adjust their parameters based on the sensitivity of the data and the potential threat landscape. Moreover, advancements in quantum computing could be leveraged to create next-generation encryption methods that are resilient against future threats posed by quantum attacks, thus ensuring long-term security for stored and transmitted data. Another area of research could focus on the implementation of robust authentication mechanisms that work in tandem with compression algorithms. These mechanisms could include digital signatures, blockchain-based verification processes, and multi-factor authentication to ensure that only authorized users have access to the compressed data. Additionally, the use of ML and artificial intelligence to monitor and analyze data transmission patterns could help in early detection of anomalies and potential security breaches, allowing for proactive measures to be taken. Furthermore, it would be beneficial to explore the development of secure data storage solutions that integrate compression algorithms. This could involve creating specialized hardware that supports encrypted compression, providing an additional layer of security. For instance, secure enclaves within processors could be used to handle sensitive DC tasks, ensuring that data remains protected even at the hardware level. Implementing comprehensive security protocols that encompass the entire lifecycle of data – from its initial compression to storage and eventual transmission – will be crucial. This holistic approach ensures that data is protected at every stage, reducing the risk of breaches and unauthorized access. Regular security audits and updates to the compression algorithms and encryption methods would be essential to address emerging threats and vulnerabilities. Collaborative efforts between academia, industry, and government bodies could also play a pivotal role in advancing the security of compressed data. By establishing standardized security frameworks and guidelines, stakeholders can ensure that compression technologies are developed and deployed with the highest levels of security in mind.

- **Usage in different domains** Despite not being tested with data other than PG signals, this framework might be used in domains different than SG, since it is suitable to compress other types of signals. This would require retraining the NN with data typical to domains like audio or image compression, but, by the design, the method is suitable to compress other types of signals as well.

Although the proposed solution offers significant advancements in DC for IoT sensor networks, there remain numerous opportunities for further research and development. By exploring these areas, researchers can continue to push the boundaries of what is possible, leading to even more efficient, reliable, and secure DC solutions.

Bibliography

[1] K. Prokop, A. Bień, and S. Barczentewicz, "Compression techniques for real-time control and non-time-critical big data in smart grids: A review," *Energies*, vol. 16, no. 24, 2023. [Online]. Available: https://www.mdpi.com/1996-1073/16/24/8077

[2] ——, "Downtime measurements of generator-powered microgrid during planned and unplanned transfer to island mode," *International Journal of Electronics and Telecommunications*, vol. vol. 69, no. No 3, pp. 493–498, 2023. [Online]. Available: http://journals.pan.pl/Content/128287/PDF/11_4163_Prokop_sk.pdf

[3] R. Jumar, H. Maaß, and V. Hagenmeyer, "Comparison of lossless compression schemes for high rate electrical grid time series for smart grid monitoring and analysis," *Computers Electrical Engineering*, vol. 71, pp. 465–476, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790617334791

[4] J. Ning, J. Wang, W. Gao, and C. Liu, "A wavelet-based data compression technique for smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 1, pp. 212–218, 2011.

[5] N. C. F. Tse, J. Y. C. Chan, W.-H. Lau, J. T. Y. Poon, and L. L. Lai, "Real-time power-quality monitoring with hybrid sinusoidal and lifting wavelet compression algorithm," *IEEE Transactions on Power Delivery*, vol. 27, no. 4, pp. 1718–1726, 2012.

[6] S. M. Blair, A. J. Roscoe, and J. Irvine, "Real-time compression of iec 61869-9 sampled value data," in *2016 IEEE International Workshop on Applied Measurements for Power Systems (AMPS)*, 2016, pp. 1–6.

[7] S. Z. Marshoodulla and G. Saha, "A survey of data mining methodologies in the environment of iot and its variants," *Journal of Network and Computer Applications*, vol. 228, p. 103907, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804524000845

[8] P. Pandiyan, S. Saravanan, R. Kannadasan, S. Krishnaveni, M. H. Alsharif, and M.-K. Kim, "A comprehensive review of advancements in green iot for smart grids: Paving the path to sustainability," *Energy Reports*, vol. 11, pp. 5504–5531, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352484724002981

[9] Y. Fan, L. Zhang, D. Li, and Z. Wang, “Progress in self-powered, multi-parameter, micro sensor technologies for power metaverse and smart grids,” *Nano Energy*, vol. 118, p. 108959, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2211285523007966

[10] J. Powell, A. McCafferty-Leroux, W. Hilal, and S. A. Gadsden, “Smart grids: A comprehensive survey of challenges, industry applications, and future trends,” *Energy Reports*, vol. 11, pp. 5760–5785, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352484724003299

[11] K. Islam, D. Kim, and A. Abu-Siada, “A review on adaptive power system protection schemes for future smart and micro grids, challenges and opportunities,” *Electric Power Systems Research*, vol. 230, p. 110241, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779624001299

[12] M. Khalid, “Smart grids and renewable energy systems: Perspectives and grid integration challenges,” *Energy Strategy Reviews*, vol. 51, p. 101299, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2211467X24000063

[13] R. Pandya and F. Bhavsar, “Study on compensation of voltage sag and voltage swell by using dvr (dynamic voltage restorer),” in *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, 2018, pp. 1–4.

[14] O. Mohamed Salim and A. El-Sayed Salem Aboraya, “Power quality enhancement of unbalanced distributed-resources based on triple-loop compensation,” *Ain Shams Engineering Journal*, vol. 15, no. 2, p. 102454, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S209044792300343X

[15] K. Sobolewski and J. Sroka, “Voltage thd suppression by autonomous power source - case study,” in *2020 IEEE 21st International Conference on Computational Problems of Electrical Engineering (CPEE)*, 2020, pp. 1–4.

[16] R. Kljajić, P. Marić, H. Glavaš, and M. Žnidarec, “Microgrid stability: A review on voltage and frequency stability,” in *2020 IEEE 3rd International Conference and Workshop in Óbuda on Electrical and Power Engineering (CANDO-EPE)*, 2020, pp. 000 047–000 052.

[17] A. K. Panigahi and R. K. Sahu, “Automatic generation control and its impact on long-term voltage and frequency stability,” in *2024 IEEE International Students’ Conference on Electrical, Electronics and Computer Science (SCEECS)*, 2024, pp. 1–6.

[18] “Standard for measurement and limits of voltage fluctuations and associated light flicker on ac power systems,” *IEEE Std 1453-2022 (Revision of IEEE Std 1453-2015)*, pp. 1–83, 2022.

[19] P. G. V. Axelberg, M. H. J. Bollen, and I. Y.-H. Gu, “Trace of flicker sources by using the quantity of flicker power,” *IEEE Transactions on Power Delivery*, vol. 23, no. 1, pp. 465–471, 2008.

[20] L. Yunos, M. L. Jane, P. J. Murphy, and K. Zuber, “Frequency selective surface on low emissivity windows as a means of improving telecommunication signal transmission: A review,” *Journal of Building Engineering*, vol. 70, p. 106416, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352710223005958

[21] J. Tverdohleb, I. Limarev, V. Dubrovin, and V. Logominov, “Wavelet analysis of complex nonstationary oscillatory signals,” in *2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC ST)*, 2017, pp. 12–20.

[22] N. S. Hasan, N. Rosmin, D. A. A. Osman, and A. H. Musta’amal@Jamal, “Reviews on multilevel converter and modulation techniques,” *Renewable and Sustainable Energy Reviews*, vol. 80, pp. 163–174, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032117307955

[23] A. António-Ferreira, C. Collados-Rodríguez, and O. Gomis-Bellmunt, “Modulation techniques applied to medium voltage modular multilevel converters for renewable energy integration: A review,” *Electric Power Systems Research*, vol. 155, pp. 21–39, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779617303334

[24] Q. Zhang, R. Shi, H. Geng, J. Liu, X. Jin, H. Xing, and D. Bai, “Power grid inspection image denoising for mixed poisson-gaussian noise,” in *2022 IEEE 5th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, 2022, pp. 376–380.

[25] C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[26] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 2006.

[27] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*. Pearson, 2011.

[28] Z. Fan, S. Kulkarni, S. Gormus, C. Efthymiou, G. Kalogridis, M. Sooriyabandara, Z. F. Zhu, S. Lambotharan, and W. Chin, “Smart grid communications: Overview of research challenges, solutions, and standardization activities,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 21–38, 2013.

[29] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[30] G. Keiser, *Optical Fiber Communications*. McGraw-Hill, 2003.

[31] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, “Smart grid technologies: Communication technologies and standards,” *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 529–539, 2011.

[32] A.-H. Mohsenian-Rad, V. W. S. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, “Optimal residential load control with price prediction in real-time electricity pricing environments,” *IEEE Transactions on Smart Grid*, vol. 1, no. 2, pp. 120–133, 2010.

[33] W. H. Kersting, *Distribution system modeling and analysis*. CRC press, 2001.

[34] S. Ghosh and D. L. Lubkeman, "Intelligent fault analysis in power systems using artificial neural networks," *IEEE Transactions on Power Delivery*, vol. 9, no. 2, pp. 803–810, 2010.

[35] G. Găşpăresc, "Data compression of power quality disturbances using wavelet transform and spline interpolation method," in *2010 9th International Conference on Environment and Electrical Engineering*, 2010, pp. 285–288.

[36] J. Yin, I. Gorton, and S. Poorva, "Toward real time data analysis for smart grids," 2012, Conference paper, p. 827 – 832. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84876517193&doi=10.1109%2fSC.Companion.2012.116&partnerID=40&md5=52a572db08a977ced12ee73ed1546641

[37] M. P. Tcheou, L. Lovisolo, M. V. Ribeiro, E. A. B. da Silva, M. A. M. Rodrigues, J. M. T. Romano, and P. S. R. Diniz, "The compression of electric signal waveforms for smart grids: State of the art and future trends," *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 291–302, 2014.

[38] S. M. S. Alam, B. Natarajan, and A. Pahwa, "Distribution grid state estimation from compressed measurements," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1631–1642, 2014.

[39] A. Unterweger and D. Engel, "Resumable load data compression in smart grids," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 919–929, 2015.

[40] J. Khan, S. M. A. Bhuiyan, G. Murphy, and M. Arline, "Embedded-zerotree-wavelet-based data denoising and compression for smart grid," *IEEE Transactions on Industry Applications*, vol. 51, no. 5, pp. 4190–4200, 2015.

[41] J. Khan, S. Bhuiyan, G. Murphy, and M. Arline, "Embedded zerotree wavelet based data compression for smart grid," in *2013 IEEE Industry Applications Society Annual Meeting*, 2013, pp. 1–8.

[42] S.-Y. Chiu, H. H. Nguyen, R. Tan, D. K. Y. Yau, and D. Jung, "Jice: Joint data compression and encryption for wireless energy auditing networks," in *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2015, pp. 453–461.

[43] F. Zhang, L. Cheng, X. Li, Y. Sun, W. Gao, and W. Zhao, "Application of a real-time data compression and adapted protocol technique for wams," *IEEE Transactions on Power Systems*, vol. 30, no. 2, pp. 653–662, 2015.

[44] F. Eichinger, P. Efros, S. Karnouskos, and K. Böhm, "A time-series compression technique and its application to the smart grid," *VLDB Journal*, vol. 24, no. 2, p. 193 – 218, 2015. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84925291103&doi=10.1007%2fs00778-014-0368-8&partnerID=40&md5=ba899b4a72006495ae5a826a163d3942

[45] G. Luo, K. Yi, S.-W. Cheng, Z. Li, W. Fan, C. He, and Y. Mu, "Piecewise linear approximation of streaming time series data with max-error guarantees," in *2015 IEEE 31st International Conference on Data Engineering*, 2015, pp. 173–184.

[46] P. H. Gadde, M. Biswal, S. Brahma, and H. Cao, "Efficient compression of pmu data in wams," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2406–2413, 2016.

[47] Y. Wang, Q. Chen, C. Kang, Q. Xia, Y. Tan, Z. Zeng, and M. Luo, "Residential smart meter data compression and pattern extraction via non-negative k-svd," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, 2016, pp. 1–5.

[48] E. Prathibha, A. Manjunatha, and S. Basavaraj, "Dual tree complex wavelet transform based approach for power quality monitoring and data compression," in *2016 Biennial International Conference on Power and Energy Systems: Towards Sustainable Energy (PESTSE)*, 2016, pp. 1–5.

[49] J. C. S. de Souza, T. M. L. Assis, and B. C. Pal, "Data compression in smart distribution systems via singular value decomposition," *IEEE Transactions on Smart Grid*, vol. 8, no. 1, pp. 275–284, 2017.

[50] A. Unterweger and D. Engel, "Lossless compression of high-frequency voltage and current data in smart grids," in *2016 IEEE International Conference on Big Data (Big Data)*, 2016, pp. 3131–3139.

[51] W. Ren, T. Yardley, and K. Nahrstedt, "Isaac: Intelligent synchrophasor data real-time compression framework for wams," in *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2017, pp. 430–436.

[52] Y. Wang, Q. Chen, C. Kang, Q. Xia, and M. Luo, "Sparse and redundant representation-based smart meter data compression and pattern extraction," *IEEE Transactions on Power Systems*, vol. 32, no. 3, pp. 2142–2151, 2017.

[53] A. Chatterjee, R. J. Shah, and K. S. Hasan, "Efficient data compression for iot devices using huffman coding based techniques," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 5137–5141.

[54] Y. Lee, E. Hwang, and J. Choi, "A unified approach for compression and authentication of smart meter reading in ami," *IEEE Access*, vol. 7, pp. 34 383–34 394, 2019.

[55] N. Kayedpour, A. E. Samani, J. D. M. De Kooning, L. Vandevelde, and G. Crevecoeur, "A data-driven approach using deep learning time series prediction for forecasting power system variables," in *2019 IEEE 2nd International Conference on Renewable Energy and Power Engineering (REPE)*, 2019, pp. 43–47.

[56] D. L. S. Mendes, A. F. da S. Veloso, R. A. L. Rabêlo, J. J. P. C. Rodrigues, and J. V. dos Reis Junior, "An adaptive data compression mechanism for wireless sensor networks in the smart grid scenarios," in *2020 5th International Conference on Smart and Sustainable Technologies (SpliTech)*, 2020, pp. 1–6.

[57] G. Kuldeep and Q. Zhang, "Revisiting compressive sensing based encryption schemes for iot," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6.

[58] M. R. Chowdhury, S. Tripathi, and S. De, "Adaptive multivariate data compression in smart metering internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1287–1297, 2021.

[59] R. Vestergaard, D. E. Lucani, and Q. Zhang, "A randomly accessible lossless compression scheme for time-series data," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 2145–2154.

[60] X. Yu, Y. Peng, F. Li, S. Wang, X. Shen, H. Mai, and Y. Xie, "Two-level data compression using machine learning in time series database," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, 2020, pp. 1333–1344.

[61] Q.-T. Doan, A. S. M. Kayes, W. Rahayu, and K. Nguyen, "Integration of iot streaming data with efficient indexing and storage optimization," *IEEE Access*, vol. 8, pp. 47 456–47 467, 2020.

[62] L. Das, D. Garg, and B. Srinivasan, "Neuralcompression: A machine learning approach to compress high frequency measurements in smart grid," *Applied Energy*, vol. 257, p. 113966, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261919316538

[63] L. Yan, J. Han, R. Xu, and Z. Li, "Model-free lossless data compression for real-time low-latency transmission in smart grids," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2601–2610, 2021.

[64] J.-F. Huang, G.-H. Zhang, and S.-Y. Hsieh, "Real-time energy data compression strategy for reducing data traffic based on smart grid ami networks," *Journal of Supercomputing*, vol. 77, no. 9, p. 10097 – 10116, 2021. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85101740427&doi=10.1007%2fs11227-020-03557-8&partnerID=40&md5=6ac9f8375c23f90a0cc9ac4c8541ffbd

[65] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, and M. S. Hossain, "Deep anomaly detection for time-series data in industrial iot: A communication-efficient on-device federated learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6348–6358, 2021.

[66] R. Ayub and W. G. Morsi, "Real-time wavelet-based data compression in light of iec61850 communication protocol," in *2022 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2022, pp. 151–158.

[67] K. M. Jose and W. G. Morsi, "Smart grid data compression of power quality events using wavelet transform," in *2022 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2022, pp. 159–164.

[68] H. Wang, H. Rao, X. Gong, Z. Chen, D. Mao, and J. Zhang, "Electric power asynchronous heterogeneous data accelerated compression for edge computing," in *2022 IEEE 7th International Conference on Smart Cloud (SmartCloud)*, 2022, pp. 121–126.

[69] J. Khan, "Fuzzy transform for smart grid data compression," in *2022 IEEE Industry Applications Society Annual Meeting (IAS)*, 2022, pp. 1–5.

[70] M. Zhou, B. Deng, M. Kong, and Y. Yuan, "A real-time spectrum data compression method based on character encoding," 2022, Conference paper, p. 410 – 414. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85131838484&doi=10.1109%2fICSP54964.2022.9778574&partnerID=40&md5=ac53395d135bdaff14e77f8ac2335147

[71] P. Kumari, R. Mishra, H. P. Gupta, T. Dutta, and S. K. Das, "An energy efficient smart metering system using edge computing in lora network," *IEEE Transactions on Sustainable Computing*, vol. 7, no. 4, pp. 786–798, 2022.

[72] D. Subbarao, A. S. Kumar, S. Bikshapthi, N. Malathi, and M. Ashraf, "Analysis of data compression techniques in smart grids for optimising mean-square-error," *Applied Nanoscience (Switzerland)*, vol. 13, no. 2, p. 1591 – 1599, 2023. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85115339102&doi=10.1007%2fs13204-021-02028-7&partnerID=40&md5=a64e348eedaaaa2b70d16cbaffe881fe

[73] B. Liu, Y. Hou, W. Luan, Z. Liu, S. Chen, and Y. Yu, "A divide-and-conquer method for compression and reconstruction of smart meter data," *Applied Energy*, vol. 336, p. 120851, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261923002155

[74] A. Giani, E. Bitar, M. Garcia, and M. McQueen, "Smart grid data integrity attacks," *IEEE Transactions on Smart Grid*, 2013.

[75] S. Tan, W. Song, M. Stewart, and J. Yang, "Online data integrity attacks against real-time electrical market in smart grid," *IEEE Transactions on Smart Grid*, 2016.

[76] R. Shuvro, P. Das, J. Jyoti, and J. Abreu, "Data-integrity aware stochastic model for cascading failures in power grids," *IEEE Transactions on Power Systems*, 2022.

[77] D. An, Q. Yang, W. Liu, and Y. Zhang, "Defending against data integrity attacks in smart grid: A deep reinforcement learning-based approach," *IEEE Access*, 2019.

[78] A. Giani, E. Bitar, M. Garcia, and M. McQueen, "Smart grid data integrity attacks: characterizations and countermeasures," *IEEE Transactions on Smart Grid*, 2011.

[79] D. An, F. Zhang, F. Cui, and Q. Yang, "Toward data integrity attacks against distributed dynamic state estimation in smart grid," *IEEE Transactions on Power Systems*, 2023.

[80] A. Farraj, E. Hammad, and D. Kundur, "A distributed control paradigm for smart grid to address attacks on data integrity and availability," *IEEE Transactions on Signal Processing*, 2017.

[81] Q. Yang, D. Li, W. Yu, Y. Liu, and D. An, "Toward data integrity attacks against optimal power flow in smart grid," *IEEE Internet of Things Journal*, 2017.

[82] S. Sridhar and G. Manimaran, "Data integrity attack and its impacts on voltage control loop in power grid," *2011 IEEE power and energy society general meeting*, 2011.

[83] K. Sayood, *Introduction to Data Compression*. Morgan Kaufmann, 2017.

[84] D. Korn and K.-P. Vo, "Algorithms for lossy compression," *IEEE Data Compression Conference*, pp. 477–486, 2001.

[85] R. H. Fowler and H. M. Gladney, *Data Compression in Digital Systems*. CRC Press, 1987.

[86] T. C. Bell, I. H. Witten, and J. G. Cleary, "Modeling for text compression," *ACM Computing Surveys*, vol. 21, no. 4, pp. 557–591, 1989.

[87] I. H. Witten, A. Moffat, and T. C. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, 1999.

[88] S. Katzenbeisser and F. A. P. Petitcolas, "Information hiding techniques for steganography and digital watermarking," *IEEE Security & Privacy*, vol. 3, no. 5, pp. 72–74, 2000.

[89] D. Salomon and G. Motta, *Data Compression: The Complete Reference*. Springer, 2004.

[90] J. Rizzo and T. Duong, "The crime attack," *OWASP AppSec USA*, 2012.

[91] P. Zhang, Y. Ni, and J. Wang, "Research on signal denoising method based on adaptive lifting wavelet transform," in *2018 International Conference on Sensor Networks and Signal Processing (SNSP)*, 2018, pp. 306–310.

[92] I. Daubechies, *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611970104

[93] S. Singh, B. Jain, and S. Jain, "Wavelet based real-time power quality monitoring," in *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2016, pp. 1–6.

[94] X. Cheng and H. Zhang, "Research on real-time data transmission and signal processing system of cim practical training teaching platform based on 5g network," in *2024 IEEE 2nd International Conference on Image Processing and Computer Applications (ICIPCA)*, 2024, pp. 845–849.

[95] P. Lertwanitrot, A. Ngaopitakkul, and S. Ananwattanaporn, "Comparative study for discrete wavelet transform between single and double detection approach to fault classification on transmission line," in *2024 16th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, 2024, pp. 493–498.

[96] S. Mallat, *A Wavelet Tour of Signal Processing*, 01 1999.

[97] M. Vetterli and J. Kovačevic, *Wavelets and subband coding*. USA: Prentice-Hall, Inc., 1995.

[98] Y. Ding and I. W. Selesnick, "Artifact-free wavelet denoising: Non-convex sparse regularization, convex optimization," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1364–1368, 2015.

[99] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Philadelphia, PA: Wellesley-Cambridge Press, 1996. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9780961408879

[100] R. Jadhav and A. Mahajan, "Smart grid data compression and reconstruction by wavelet packet transform," *MethodsX*, p. 102872, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2215016124003248

[101] H. Yang, D. Tan, N. Ramalingam, K.-S. Lim, C.-G. Tan, and H. Ahmad, "Enhancing data sparsity in spectral signals using wavelet decomposition for improved compression and storage efficiency," *Optical Fiber Technology*, vol. 86, p. 103848, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1068520024001937

[102] R. Coifman and M. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 713–718, 1992.

[103] D. Zhigang, Z. Jingxuan, and J. Chunrong, "An improved wavelet threshold denoising algorithm," in *2013 Third International Conference on Intelligent System Design and Engineering Applications*, 2013, pp. 297–299.

[104] H. Kekre, P. Natu, and T. Sarode, "Color image compression using vector quantization and hybrid wavelet transform," *Procedia Computer Science*, vol. 89, pp. 778–784, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050916311243

[105] H. F. Ates and E. Tamer, "Hierarchical quantization indexing for wavelet and wavelet packet image coding," *Signal Processing: Image Communication*, vol. 25, no. 2, pp. 111–120, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0923596509001234

[106] T. Koya, S. Chandran, and K. Vijayalakshmi, "Analysis of application of arithmetic coding on dct and dct-dwt hybrid transforms of images for compression," in *2017 International Conference on Networks Advances in Computational Technologies (NetACT)*, 2017, pp. 288–293.

[107] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, 1996.

[108] D. Taubman, "High performance scalable image compression with ebcot," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, 2000.

[109] A. Islam and W. A. Pearlman, "An embedded and efficient low-complexity hierarchical image coder," in *Proceedings DCC'99. Data Compression Conference*. IEEE, 1999, pp. 4–13.

[110] R. Praisline Jasmi, B. Perumal, and M. Pallikonda Rajasekaran, "Comparison of image compression techniques using huffman coding, dwt and fractal algorithm," in *2015 International Conference on Computer Communication and Informatics (ICCCI)*, 2015, pp. 1–5.

[111] R. Sharma, A. Kesarwani, and P. M. Mathur, "A novel compression algorithm using dwt," in *2014 Annual IEEE India Conference (INDICON)*, 2014, pp. 1–4.

[112] C. Vaz and D. G. Daut, "Effects of wavelet choice in fast-fading channel estimation using wavelet transform-based deconvolution," in *2011 11th International Symposium on Communications Information Technologies (ISCIT)*, 2011, pp. 93–96.

[113] H. N. Murthy and M. Meenakshi, "Optimum choice of wavelet function and thresholding rule for ecg signal denoising," in *2015 International Conference on Smart Sensors and Systems (IC-SSS)*, 2015, pp. 1–5.

[114] J. Kim, H. M. Hasanien, and R. K. Tagayi, "Investigation of noise suppression in experimental multi-cell battery string voltage applying various mother wavelets and decomposition levels in discrete wavelet transform for precise state-of-charge estimation," *Journal of Energy Storage*, vol. 73, p. 109196, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X2302594X

[115] S. A. Lashari, R. Ibrahim, and N. Senan, "De-noising analysis of mammogram images in the wavelet domain using hard and soft thresholding," in *2014 4th World Congress on Information and Communication Technologies (WICT 2014)*, 2014, pp. 353–357.

[116] N. A. Golilarz and H. Demirel, "Image de-noising using un-decimated wavelet transform (uwt) with soft thresholding technique," in *2017 9th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2017, pp. 16–19.

[117] N. A. Golilarz, H. Gao, W. Ali, and M. Shahid, "Hyper-spectral remote sensing image de-noising with three dimensional wavelet transform utilizing smooth nonlinear soft thresholding function," in *2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 2018, pp. 142–146.

[118] M. Vijay, L. Saranya Devi, M. Shankaravadivu, and M. Santhanamari, "Image denoising based on adaptive spatial and wavelet thresholding methods," in *IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012)*, 2012, pp. 161–166.

[119] L. Shuangquan, W. Chao, Z. Jiahui, and J. Binling, "The research of wavelet adaptive threshold gray scale image denoising," in *Proceedings of 2013 2nd International Conference on Measurement, Information and Control*, vol. 02, 2013, pp. 968–971.

[120] R. Fu, J. Zhang, R. Wang, and T. Xu, "Improved wavelet thresholding function and adaptive thresholding for noise reduction," in *2023 11th International Conference on Intelligent Computing and Wireless Optical Communications (ICWOC)*, 2023, pp. 25–30.

[121] S. Aouay, S. Jamoussi, and Y. B. Ayed, "Particle swarm optimization based method for bayesian network structure learning," in *2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO)*, 2013, pp. 1–6.

[122] N. Dikhaminjia, G. Tsintsadze, Z. Kiguradze, J. He, M. Tsiklauri, J. Drewniak, A. Chada, and B. Mutnury, "Optimization of joint equalization of high-speed signals using bayesian machine learning," in *2021 IEEE International Joint EMC/SI/PI and EMC Europe Symposium*, 2021, pp. 48–52.

[123] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning.*, ser. Adaptive computation and machine learning. MIT Press, 2006.

[124] K. Tamura, "A heuristic strategy of a bayesian optimization algorithm for evaluation-times constrained optimization (eco) problems," in *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*, 2018, pp. 764–769.

[125] T. Nguyen, B. Shi, H. Ma, E.-P. Li, X. Chen, A. C. Cangellaris, and J. Schutt-Ainé, "Comparative study of surrogate modeling methods for signal integrity and microwave circuit applications," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 11, no. 9, pp. 1369–1379, 2021.

[126] S. Koziel, S. Ogurtsov, and J. P. Jacobs, "Low-cost design optimization of slot antennas using bayesian support vector regression and space mapping," in *2012 Loughborough Antennas Propagation Conference (LAPC)*, 2012, pp. 1–4.

[127] H. Wang, B. van Stein, M. Emmerich, and T. Back, "A new acquisition function for bayesian optimization based on the moment-generating function," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 507–512.

[128] F. Passos, R. González-Echevarría, E. Roca, R. Castro-López, and F. Fernández, "Surrogate modeling and optimization of inductor performances using kriging functions," in *2015 International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, 2015, pp. 1–4.

[129] W. Gan, Z. Ji, and Y. Liang, "Acquisition functions in bayesian optimization," in *2021 2nd International Conference on Big Data Artificial Intelligence Software Engineering (ICBASE)*, 2021, pp. 129–135.

[130] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf

[131] E. Brochu, V. Cora, and N. Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *CoRR*, vol. abs/1012.2599, 12 2010.

[132] M. S. Neyestanak, E. Azizi, and E. Salarpour, "Iot performance improving for indoor and outdoor environments," in *2021 5th International Conference on Internet of Things and Applications (IoT)*, 2021, pp. 1–6.

[133] A. Vakaloudis and C. O'Leary, "A framework for rapid integration of iot systems with industrial environments," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 601–605.

[134] K. E. S. Desikan, V. J. Kotagi, and C. S. R. Murthy, "Decoding the interplay between latency, reliability, cost, and energy while provisioning resources in fog-computing-enabled iot networks," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2404–2416, 2023.

[135] M. Goyal, K. Tatwawadi, S. Chandak, and I. Ochoa, "Deepzip: Lossless data compression using recurrent neural networks," in *2019 Data Compression Conference (DCC)*, 2019, pp. 575–575.

[136] C. Li and R. Zheng, "Load data compression based on integrated neural network model," in *2019 Chinese Control And Decision Conference (CCDC)*, 2019, pp. 6203–6209.

[137] S. Coulibaly, B. Kamsu-Foguem, D. Kamissoko, and D. Traore, "Deep convolution neural network sharing for the multi-label images classification," *Machine Learning with Applications*, vol. 10, p. 100422, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666827022000974

[138] E. Stevens, L. P. G. Antiga, and T. Viehmann, 2020.

[139] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[140] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[141] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[142] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[143] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 448–456.

[144] Y. Shao, G. N. Taff, and S. J. Walsh, "Comparison of early stopping criteria for neural-network-based subpixel classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 1, pp. 113–117, 2011.

[145] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[146] T. Takase, S. Oyama, and M. Kurihara, "Evaluation of stratified validation in neural network training with imbalanced data," in *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2019, pp. 1–4.

[147] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *Communications of the ACM*, vol. 64, 11 2016.

[148] A. Suliman and N. Nazri, "A new hybrid model of software engineering and systems engineering for embedded system development methodology," in *Proceedings of the 6th International Conference on Information Technology and Multimedia*, 2014, pp. 346–350.

[149] M. Kolekar, C. Jha, and P. Kumar, "Ecg data compression using modified run length encoding of wavelet coefficients for holter monitoring," *IRBM*, vol. 43, no. 5, pp. 325–332, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1959031821001159

[150] J. Zhang and D. Sun, "Improvement of data compression technology for power dispatching based on run length encoding," *Procedia Computer Science*, vol. 183, pp. 526–532, 2021, proceedings of the 10th International Conference of Information and Communication Technology. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187705092100569X

[151] M. Stabno and R. Wrembel, "Rlh: Bitmap compression technique based on run-length and huffman encoding," *Information Systems*, vol. 34, no. 4, pp. 400–414, 2009, data Warehousing and On-Line Analytical Processing. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306437908000902

[152] S. T. Klein, E. Opalinsky, and D. Shapira, "Synchronizing dynamic huffman codes," *Discrete Applied Mathematics*, vol. 358, pp. 23–32, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166218X24002853

[153] "https://wavelet2d.sourceforge.net/."

[154] "https://www.gnu.org/software/gsl/."

[155] "https://opencv.org/."

[156] "https://github.com/rafat/wavelib."

[157] X. Li, P. Zhang, E. Hui, and G. Chen, "Based on embedded zerotree wavelet coding data compression algorithm research of nc machine tool control system," in *2019 4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, 2019, pp. 936–9364.

[158] C. S. Lai, "Compression of power system signals with wavelets," in *2014 International Conference on Wavelet Analysis and Pattern Recognition*, 2014, pp. 109–115.

[159] C. J. Ramakrishna, D. B. K. Reddy, B. Priya, P. Amritha, and K. Lakshmy, "Analysis of lightweight cryptographic algorithms for iot gateways," *Procedia Computer Science*, vol. 233, pp. 235–242, 2024, 5th International Conference on Innovative Data Communication Technologies and Application (ICIDCA 2024). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050924005726

[160] A. Shen, M. Kuzlu, M. Pipattanasomporn, S. Rahman, and L. Chen, "A performance testing method for embedded software platforms," in *2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, 2016, pp. 135–140.

[161] M. Zeller, C. Prehofer, G. Weiss, D. Eilers, and R. Knorr, “Towards self-adaptation in real-time, networked systems: Efficient solving of system constraints for automotive embedded systems,” in *2011 IEEE Fifth International Conference on Self-Adaptive and Self-Organizing Systems*, 2011, pp. 79–88.

[162] M. Garg and R. Lai, “Measuring the constraint complexity of automotive embedded software systems,” in *2014 International Conference on Data and Software Engineering (ICODSE)*, 2014, pp. 1–6.

[163] B. A. Bhuiyan, M. W. Absar, and A. Roy, “Performance comparison of various wavelets in compression of pmu generated data in smart grid,” in *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)*, 2017, pp. 1–6.

[164] J. S. Jones, “326 million smart meters across europe by 2028 – report,” *https://www.smart-energy.com/industry-sectors/smart-meters/326-million-smart-meters-across-europe-by-2028-report/*, 2024.