Robust Bootstrap Methods and Applications

Zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.)
Genehmigte Dissertation im Fachbereich Elektrotechnik und Informationstechnik von Pertami Junia Kunz
Tag der Einreichung: Dez 2025, Tag der Prüfung: 23. März 2026
Year of publication of the dissertation on TUprints: 2026

1. Gutachten: Prof. Dr. -Ing. Abdelhak Zoubir
2. Gutachten: Prof. Dr. Frank Aurzada
Darmstadt, Technische Universität Darmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

Electrical Engineering and Information Technology Department

Institut für Nachrichtentechnik

SIGNAL PROCESSING GROUP

Robust Bootstrap Methods and Applications

Accepted doctoral thesis in the department of Electrical Engineering and Information Technology by Pertami Junia Kunz

Date of submission: Dez 2025
Date of thesis defense: 23. März 2026
Year of publication of the dissertation on TUprints: 2026

Darmstadt, Technische Universität Darmstadt

Bitte zitieren Sie dieses Dokument als:
URN: https://nbn-resolving.de/urn:nbn:de:tuda-tuda-155300
URL: https://tuprints.ulb.tu-darmstadt.de/handle/tuda/15530

Dieses Dokument wird bereitgestellt von tuprints,
E-Publishing-Service der TU Darmstadt
https://tuprints.ulb.tu-darmstadt.de
tuprints@ulb.tu-darmstadt.de

Erklärungen laut Promotionsordnung

§ 8 Abs. 1 lit. d PromO

Ich versichere hiermit, dass zu einem vorherigen Zeitpunkt noch keine Promotion versucht wurde. In diesem Fall sind nähere Angaben über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuchs mitzuteilen.

§ 9 Abs. 1 PromO

Ich versichere hiermit, dass die vorliegende Dissertation – abgesehen von den in ihr ausdrücklich genannten Hilfen – selbstständig verfasst wurde und dass die „Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Technischen Universität Darmstadt“ und die „Leitlinien zum Umgang mit digitalen Forschungsdaten an der TU Darmstadt“ in den jeweils aktuellen Versionen bei der Verfassung der Dissertation beachtet wurden.

§ 9 Abs. 2 PromO

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, Dez 2025

Pertami J. Kunz

Preface

The research presented in this dissertation emerged from a sustained effort to understand how resampling methods can be made more responsive to the complexities of modern data. While the bootstrap has become a foundational tool in statistics, my work gradually revealed the extent to which real-world datasets, often imbalanced, highly structured, or driven by intricate temporal patterns, push beyond the limits of traditional methodology. This dissertation represents my attempt to respond to those challenges and to contribute methods that better reflect the data environments in which contemporary scientists and analysts now operate.

Throughout this work, my focus has been shaped by interdisciplinary collaboration and practical problem-solving. Engaging with diverse applications, including biomedical signals, sensor-based food monitoring, and financial time series, continually highlighted the need for resampling techniques that respect heterogeneity, adapt to structural dependencies, and remain computationally feasible. The ideas developed here evolved through many iterations, from early explorations of stratification to more mature considerations of influence functions, robustness, and complex seasonality. Although the methods introduced in the chapters that follow vary in purpose and structure, they are united by a commitment to enhancing the flexibility and reliability of bootstrap-based inference.

The process of writing this dissertation has been both challenging and deeply rewarding. I owe my deepest gratitude to Prof. Dr.-Ing. Abdelhak M. Zoubir, whose guidance, encouragement, and high standards have shaped not only the direction of this work but also my growth as a researcher. I am equally grateful to my colleagues and collaborators for the many conversations, critiques, and insights that enriched this work and broadened my perspective. I wish to express my heartfelt thanks to my family. Their unwavering support, patience, and belief in me provided the foundation that allowed me to pursue this research with dedication and resilience. Finally, I gratefully acknowledge the support of the Graduate School CE within the Centre for Computational Engineering. This support made it possible for me to pursue and complete this research.

Zusammenfassung

Bootstrap-Resamplingverfahren sind grundlegende Werkzeuge der modernen Statistik, doch klassische Ansätze stoßen an ihre Grenzen, wenn sie auf heterogene, unausgewogene oder strukturell komplexe Daten angewendet werden. Diese Dissertation stellt methodische Weiterentwicklungen vor, die darauf abzielen, die Anwendbarkeit und Zuverlässigkeit von Bootstrap-Verfahren in solchen anspruchsvollen Datensituationen zu erweitern. Die Beiträge lassen sich in drei thematische Schwerpunkte einordnen: Oversampling-Stratifizierte Bootstrap-Methoden, Komplex-Saisonale Bootstrap-Methoden und spezialisierte bootstrapbasierte Verfahren auf Grundlage von Einflussfunktionen oder effizienter Berechnung.

Der erste Schwerpunkt umfasst Oversampling-Stratifizierte Bootstrap-Methoden, die starke Klassenungleichgewichte und strukturelle Heterogenität adressieren. Dazu gehören der Mahalanobis-Distanz-Stratifizierte Bootstrap, der Heterogenitäts-Stratifizierte Bootstrap sowie zwei Erweiterungen, der Heterogenitäts-Intensivierte Bootstrap und der Heterogenitäts-Ratio-Stratifizierte Bootstrap. Diese Verfahren nutzen distanz- oder heterogenitätsbasierte Stratifizierung, um die Repräsentation der Minderheitsklasse zu verbessern, die Varianz von Schätzern zu stabilisieren und lokale Strukturen zu erhalten. Die Leistungsfähigkeit der Methoden wird anhand von Simulationen sowie einer Fallstudie zur Klassifikation intrakranieller EEG-Daten untersucht.

Der zweite thematische Schwerpunkt widmet sich Komplex-Saisonalen Bootstrap-Methoden, die für Datensätze entwickelt wurden, die mehrere überlagerte oder nicht ganzzahlige Saisonalitäten aufweisen. Der Komplex-Saisonale Zirkulare Block-Bootstrap erweitert klassische Block-Resampling-Ansätze auf multi-saisonale Strukturen, während robuste Varianten auf Grundlage von Einflussfunktionen und Blockmittelungen eine bessere Performance bei schweren Verteilungen, Volatilitätswechseln und strukturellen Unregelmäßigkeiten ermöglichen. Diese Methoden liefern verlässlichere Unsicherheitsabschätzungen für Zeitreihen, in denen herkömmliche Block-Bootstrap-Verfahren unzureichend sind.

Der dritte Schwerpunkt stellt zwei spezialisierte Bootstrap-Verfahren vor. Der Blitz-Bootstrap bietet eine schnelle approximative Resampling-Methode, die insbesondere in Situationen mit begrenzten Rechenressourcen geeignet ist. Der Bootstrap mit Zyklischer Volatilitäts-Einflussfunktion und seine Varianten konzentrieren sich auf Zeitreihen mit zyklischer oder regimespezifischer Volatilität, wie sie in finanziellen Anwendungen häufig

auftreten.

Insgesamt erweitern die in dieser Dissertation vorgestellten Beiträge die Bootstrap-Methodik, indem sie Heterogenität, Klassenungleichgewicht, komplexe Saisonalität, Volatilitätsstrukturen und rechnerische Einschränkungen berücksichtigen. Die entwickelten Verfahren vergrößern die Einsatzmöglichkeiten bootstrapbasierter Inferenz in einer Vielzahl realer Anwendungsfelder und ermöglichen verlässlichere und besser interpretierbare Unsicherheitsquantifizierungen für komplexe moderne Datensätze.

Abstract

Bootstrap resampling techniques are essential tools in modern statistics, yet classical approaches often fail when confronted with heterogeneous, imbalanced, or structurally complex data. This dissertation introduces methodological advances designed to extend the applicability and reliability of bootstrap procedures in such challenging settings. The contributions span three major themes: oversampling stratified bootstrap methods, complex seasonal bootstrap methods, and specialized influence-function-based or computationally efficient bootstrap variants.

The first set of contributions develops oversampling stratified bootstrap methods that address severe class imbalance and structural heterogeneity. These include the Mahalanobis-distance Stratified Bootstrap, the Heterogeneity-Stratified Bootstrap, and two extensions known as the Heterogeneity-Intensified Bootstrap and the Heterogeneity Ratio-Stratified Bootstrap. These methods incorporate distance-based or heterogeneity-based stratification to enhance minority-class representation, stabilize estimator variance, and preserve local structure. Their performance is evaluated through simulations and through a detailed case study involving intracranial EEG classification.

The second theme introduces complex seasonal bootstrap methods designed for datasets that exhibit multiple overlapping or non-integer seasonal patterns. The Complex Seasonal Circular Block Bootstrap generalizes block resampling to multi-seasonal environments, while its robust variants based on influence functions and block averaging improve performance in the presence of heavy tails, volatility shifts, and structural irregularities. These methods provide more reliable uncertainty quantification for time series in which classical block bootstrap procedures are inadequate.

The final set of contributions presents two specialized bootstrap procedures. The Blitz-Boot offers a fast approximate resampling approach suitable for situations where computational resources are limited. The Cyclic Volatility Influence Function Bootstrap and its variants focus on time series with cyclical or regime-based volatility, such as those commonly encountered in financial applications.

Taken together, these contributions advance bootstrap methodology by providing tools that incorporate heterogeneity, class imbalance, complex seasonality, volatility structure, and computational constraints. The methods introduced in this dissertation broaden the

applicability of bootstrap inference to a wide range of real-world scenarios and offer more reliable and interpretable uncertainty quantification for complex modern datasets.

Contents

Preface **v**

1 Introduction **1**
1.1 Background . . . 1
1.2 Motivation . . . 3

2 Literature Review **5**
2.1 Stratified Sampling Methods . . . 7
2.2 Special Block Bootstrap Methods . . . 11
2.3 Robust Bootstrap Methods . . . 17

3 Summary of Contributions **23**

4 Oversampling Stratified Bootstrap Methods **27**
4.1 Mahalanobis-distance Stratified Bootstrap (MS-Boot) . . . 28
4.1.1 Introduction . . . 28
4.1.2 Problem Formulation . . . 30
4.1.3 Methods . . . 32
4.1.4 Case Study: Intracranial EEG . . . 47
4.1.5 Results . . . 51
4.1.6 Conclusion and Future Work . . . 51
4.2 Heterogeneity-Stratified Bootstrap (HS-Boot) . . . 59
4.2.1 Introduction . . . 59
4.2.2 Methodology . . . 60
4.2.3 Experiment and Results . . . 64
4.2.4 Conclusion and Future Work . . . 65
4.3 The Heterogeneity-Intensified and Heterogeneity Ratio-Stratified Bootstrap (HiS- and HeRS-Boot) . . . 69
4.3.1 Introduction . . . 69
4.3.2 Methodology . . . 70
4.3.3 Experiment and Results . . . 70

4.3.4 Conclusion and Future Work 71

5 Complex Seasonal Bootstrap Methods 79
5.1 The Complex Seasonal Circular Block Bootstrap (XSCBB) 81
5.1.1 Introduction 81
5.1.2 Methodology 84
5.1.3 Experiment and Results 86
5.1.4 Conclusion and Future Work 86
5.2 The Block Robust Empirical Standardised Influence Function (BRESIF)-XSCBB and the Block Average (BA)-XSCBB 92
5.2.1 Introduction 92
5.2.2 Problem Formulation 93
5.2.3 Methodology 94
5.2.4 Experiment and Results 98
5.2.5 Conclusion and Future Work 103

6 Other Special Bootstrap Methods 105
6.1 Blitz-Boot 107
6.1.1 Introduction 107
6.1.2 Problem Formulation 108
6.1.3 Methodology 109
6.1.4 Experiments and Results 110
6.1.5 Conclusion 114
6.2 The Cyclic Volatility Influence Function (CV-IF) Bootstrap 116
6.2.1 Introduction 116
6.2.2 Problem Formulation 117
6.2.3 Methodology 118
6.2.4 Experiment and Results 121
6.2.5 Conclusion and Future Work 124

7 Conclusion 127

List of Figures 137

List of Tables 139

List of Algorithms 141

List of Acronyms 143

Glossary **147**

1 Introduction

Statistical inference increasingly relies on resampling methods, particularly the bootstrap, to quantify uncertainty in complex data settings where analytical variance expressions are unavailable or unreliable. Although the bootstrap is flexible and widely applicable, its classical formulations often break down when data exhibit temporal dependence, multi-seasonal structure, heterogeneity, class imbalance or outliers. Such features are common in modern signal processing, biomedical and environmental applications, where real-world data rarely satisfy idealised assumptions. This dissertation develops bootstrap methodologies that address these limitations by incorporating structure, robustness and adaptive sampling principles into the resampling process. The following sections introduce the theoretical background and practical motivations that guide these methodological contributions.

1.1 Background

The bootstrap method, first introduced by Efron in 1979 [1], has become a central tool in modern statistics. It provides a powerful and flexible framework for estimating sampling distributions, confidence intervals, and hypothesis tests without relying on strict parametric assumptions. The core idea is to resample with replacement from the observed data in order to approximate the variability of a statistic under repeated sampling. Because of its simplicity and broad applicability, the bootstrap has been widely adopted in fields such as economics, biomedical research, and increasingly, signal processing.

In signal processing, resampling methods offer important advantages for assessing estimator performance, generating surrogate data, and quantifying uncertainty, especially when analytical variance formulas are difficult or impossible to obtain. However, most classical bootstrap techniques rely on the assumption that observations are independent and identically distributed. This assumption is rarely satisfied in practical settings, where signals typically exhibit time dependence, periodic structures, nonstationarity, or contamination from noise and outliers.

To accommodate temporal dependence in time-series data, extensions such as the

block bootstrap [2], [3] and the circular block bootstrap [4] were introduced. These methods preserve short-range temporal correlations by resampling blocks of consecutive observations. Although effective for certain types of dependence, they struggle to capture more complex structures such as multiple or interacting seasonalities, cyclic changes in variance, or irregular repetition patterns that frequently occur in real-world signals. Examples include electricity consumption data with overlapping cycles or sensor streams influenced by environmental rhythms.

In classification and detection tasks, particularly those involving biomedical or environmental signals, severe class imbalance poses another major challenge. Rare but critical events, such as epileptic seizures or hazardous emissions, are often underrepresented, leading standard learning algorithms to perform poorly. Oversampling methods such as synthetic minority oversampling technique (SMOTE) [5] attempt to alleviate imbalance by generating synthetic minority samples, but they may introduce artificial noise or fail to reflect the true underlying data geometry. Bootstrap-based approaches offer a statistically principled alternative, yet they require modifications to account for heterogeneity, high dimensionality, and the structure of signal domains.

Recent research has also explored robust bootstrap methods [6], which aim to reduce the influence of outliers and non-Gaussian noise. Approaches based on robust estimators, influence functions, and stratified sampling have shown potential in settings where signal quality varies over time or across sensors, but these ideas remain underdeveloped for many signal-processing tasks.

Despite these advances, significant gaps remain in the design of bootstrap procedures tailored to the needs of modern signal processing. Existing methods often address dependence or robustness in isolation, and few approaches integrate multiple challenges simultaneously, such as nonstationarity, imbalance, volatility, and uncertainty in parameter estimation. Moreover, many current frameworks are specialized to a single task, whereas practical applications demand approaches that generalize across forecasting, detection, and classification problems.

This dissertation aims to bridge these gaps by developing adaptive and task-specific bootstrap methods that operate reliably in structured, imbalanced, and volatile signal environments. The methods introduced here build on and extend the existing literature, offering new tools designed to meet the demands of contemporary signal-processing applications.

1.2 Motivation

Although bootstrap methods provide a powerful framework for estimating uncertainty, their direct application to real-world signal processing remains limited. Many practical scenarios involve characteristics that violate the assumptions underlying classical bootstrap procedures, which motivates the development of more adaptive and data-aware resampling strategies.

A large portion of modern signal data exhibits strong structural dependencies. Time series in energy systems, transportation networks, or physiological monitoring often contain multiple interacting cycles, long-range correlations, and fluctuations that evolve over time. For example, electricity demand follows daily and weekly rhythms, but it is also influenced by temperature, holidays, and unpredictable consumer patterns. Traditional block bootstrap techniques preserve short temporal correlations but do not align with such multi-layered structures. When these structures are ignored during resampling, the resulting uncertainty estimates may become unreliable or overly conservative.

Another challenge arises from changes in variance. In domains such as finance, industrial monitoring, or power grid management, the variability of a signal is as informative as its average behavior. Periods of elevated volatility often carry important decision-relevant information. However, standard bootstrap procedures tend to smooth out these dynamics, which prevents accurate representation of volatility regimes in the resampled data.

Classification and detection tasks introduce additional difficulties. Many important signal processing applications involve rare but critical events. Seizures in Electroencephalogram (EEG) recordings, early indications of food spoilage in gas sensor arrays, and anomalous emissions in environmental monitoring are all examples of events that occur infrequently yet carry substantial weight in decision-making. Learning algorithms trained on such data are frequently dominated by the majority class and struggle to detect minority events reliably. Oversampling approaches such as SMOTE or Adaptive Synthetic Sampling Approach for Imbalanced Learning (ADASYN) attempt to correct this imbalance by generating synthetic data points, but they may distort the underlying data geometry, especially in high-dimensional or highly correlated signal domains. A bootstrap-based approach has the potential to preserve distributional characteristics more faithfully, provided the resampling scheme accounts for heterogeneity and local structure.

Parameter estimation problems in signal processing further highlight the need for improved resampling tools. Tasks such as determining the direction of arrival of signals, estimating frequencies in sensor arrays, or extracting latent dynamics from noisy measurements require high precision. Small deviations in estimated parameters can accumulate and degrade subsequent processing steps. Bootstrap methods could enhance the reliability of these estimators by quantifying their uncertainty, but existing approaches are

not designed to accommodate the interaction of noise, structure, and temporal dynamics present in these applications.

Practical signal datasets also tend to include outliers, abrupt changes, and sensor faults. Robust statistics provide mechanisms for identifying and mitigating the influence of such irregularities, often through influence functions or down-weighting strategies. Yet these ideas have rarely been incorporated into bootstrap frameworks tailored for signal data, despite their potential to greatly improve stability.

Taken together, these challenges reveal a clear need for bootstrap methods that are aware of the structural, statistical, and operational characteristics of modern signal processing tasks. There is a demand for resampling frameworks that adapt to complex temporal patterns, reflect distributional heterogeneity, handle class imbalance effectively, respect volatility dynamics, and remain computationally feasible in large-scale settings.

This dissertation responds to these demands by developing a set of resampling techniques specifically designed for structured, heterogeneous, and imbalanced signal environments. These methods include seasonal and volatility-aware bootstrap procedures for forecasting, heterogeneity-oriented resampling techniques for classification, and bootstrap-based tools for robust parameter estimation. Each method is motivated by practical challenges encountered in real-world signal processing, and together they form a unified contribution toward modern, application-driven bootstrap methodology.

2 Literature Review

The bootstrap, introduced by Efron in 1979 [1], [7], is a general resampling technique for estimating the distribution of a statistic when analytical calculations are unavailable or unreliable. Bootstrap methods are particularly useful in modern statistical and signal processing applications, where data may be high dimensional, heterogeneous or dependent, and where classical asymptotic approximations may fail or converge slowly. The fundamental idea is to approximate the sampling distribution of a statistic by repeatedly resampling from the empirical distribution of the observed data.

Let $X_1, \ldots, X_n$ be independent and identically distributed observations from an unknown distribution P. Let $T = T(P)$ denote a parameter of interest, such as a mean, variance, regression coefficient or any smooth functional, and let $\hat{T}_n = T(\hat{P}_n)$ be the estimator obtained by applying T to the empirical distribution

$$\hat{P}_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{X_i}. \tag{2.1}$$

The sampling distribution of $\hat{T}_n$ under P is generally unknown. The bootstrap estimates this distribution by resampling with replacement from $\hat{P}_n$. A bootstrap sample is defined as

$$X_1^*, \ldots, X_n^* \sim \hat{P}_n^{\otimes n}, \tag{2.2}$$

where every X_i^* is drawn independently and uniformly from $\{X_1, \ldots, X_n\}$ and $\hat{P}_n^{\otimes n}$ denotes the product measure corresponding to n independent draws from $\hat{P}_n$. The bootstrap estimator is

$$\hat{T}_n^* = T(\hat{P}_n^*), \tag{2.3}$$

and its conditional distribution given the data,

$$\mathcal{L}(\hat{T}_n^* \mid X_1, \ldots, X_n), \tag{2.4}$$

serves as an approximation to the distribution of $\hat{T}_n$.

Bootstrap Approximation and Consistency

The bootstrap aims to approximate the distribution of the centered and scaled statistic

$$\sqrt{n}(\hat{T}_n - T), \tag{2.5}$$

by the conditional distribution of

$$\sqrt{n}(\hat{T}_n^* - \hat{T}_n). \tag{2.6}$$

Consistency of the bootstrap requires that

$$\sup_{x\in\mathbb{R}} \left| \mathbb{P}\Big(\sqrt{n}(\hat{T}_n - T) \le x\Big) - \mathbb{P}^*\Big(\sqrt{n}(\hat{T}_n^* - \hat{T}_n) \le x\Big) \right| \to 0, \tag{2.7}$$

in probability, where $\mathbb{P}^*$ denotes the bootstrap distribution. For smooth functionals of the distribution P, this holds under mild regularity conditions [8]. A key theoretical tool is the von Mises expansion of a functional $T(P)$,

$$T(\hat{P}_n) - T(P) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{IF}(X_i; T, P) + R_n, \tag{2.8}$$

where $\mathrm{IF}(x; T, P)$ is the influence function of T and R_n is a second order remainder. If $R_n = o_p(n^{-1/2})$, the bootstrap consistently approximates the sampling distribution of $\hat{T}_n$.

Bootstrap Confidence Intervals

Bootstrap distributions are used to construct confidence intervals for T. For instance, the percentile interval is defined as

$$[T^*_{\alpha/2}, T^*_{1-\alpha/2}], \tag{2.9}$$

where T^*_q is the q-quantile of the bootstrap distribution. Other variants include the bias-corrected and accelerated (BCa) interval and the studentized bootstrap interval [7]. The BCa estimator adjusts for skewness and estimator bias, while the studentized method uses bootstrap estimates of the variance to achieve improved accuracy.

Bootstrap for Dependent Data

For dependent observations, such as those encountered in time series or signal processing, resampling individual data points destroys serial correlation and invalidates inference. In such settings the block bootstrap and its variants were developed [2], [3], [4]. Instead of resampling data points, these methods resample blocks of consecutive observations to preserve local dependence. Block bootstrap methods form the basis of the seasonal and complex seasonal block bootstrap procedures discussed in later chapters.

Bootstrap for Imbalanced and Heterogeneous Data

When data arise from heterogeneous populations or exhibit class imbalance, the standard bootstrap reproduces the empirical distribution without adjusting for minority classes or structurally distinct subregions. This may lead to poor variance estimates or unstable classifiers. Stratified bootstrap methods provide one remedy by enforcing resampling within predefined strata. More sophisticated approaches, such as distance-weighted or heterogeneity-based bootstrap strategies, further adapt the bootstrap mechanism to local data structure. These ideas motivate the oversampling stratified bootstrap methods presented in Chapter 4.

Robust Bootstrap Methods

In the presence of outliers or heavy-tailed noise, standard bootstrap estimators may fail to reflect the true sampling behavior. Robust bootstrap procedures incorporate influence functions, downweighting schemes or robust estimators to mitigate the effect of contamination [6]. Robust ideas also appear in block bootstrap frameworks, particularly in seasonal or volatility-rich time series settings, where outliers or variance shifts significantly distort resampled blocks. Such methods form the basis of the robust complex seasonal bootstrap procedures developed in Chapter 5.

Summary

The bootstrap provides a flexible and widely applicable framework for statistical inference. Its effectiveness derives from the ability to approximate sampling distributions without explicit parametric assumptions. The method generalizes naturally to dependent data through block resampling and can be adapted to heterogeneous, imbalanced or contaminated datasets through stratification, weighting or robustness principles. In the remainder of this chapter, we provide a detailed discussion of the foundations and existing methods for stratified sampling, special block bootstrap procedures and robust bootstrap techniques.

2.1 Stratified Sampling Methods

Stratified sampling is one of the foundational concepts in sampling theory and has played a central role in survey design, Monte Carlo methods, and resampling procedures. The essential idea is to partition a population or dataset into internally homogeneous subgroups, known as strata, so that sampling within each stratum produces more efficient estimators

than simple random sampling applied to the entire population. Classical work such as [9] provides the theoretical basis for stratified inference, while later developments extended these ideas to the bootstrap and related resampling techniques [7], [8], [10]. Stratification is particularly relevant in modern applications where data are heterogeneous, imbalanced, or structurally complex, as is often the case in signal processing, biomedical monitoring, and environmental sensing.

Simple Random Sampling

Under simple random sampling without replacement, a sample of size n is drawn uniformly from a population of size N. The population mean and variance are

$$\bar{Y} = \frac{1}{N}\sum_{i=1}^{N} Y_i, \qquad S^2 = \frac{1}{N-1}\sum_{i=1}^{N}(Y_i - \bar{Y})^2. \tag{2.10}$$

The unbiased estimator of S^2 is given in Theorem 2.4 of [9]:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{y})^2, \tag{2.11}$$

where $\bar{y}$ is the sample mean. This result forms the baseline against which variance reduction through stratification is assessed.

Stratified Random Sampling

Consider a partition of the population into L strata of known sizes

$$N_1 + N_2 + \cdots + N_L = N. \tag{2.12}$$

From each stratum h, a sample of size n_h is drawn, with sampling fraction

$$f_h = \frac{n_h}{N_h}, \qquad W_h = \frac{N_h}{N}. \tag{2.13}$$

Within each stratum, the true and sample means are

$$\bar{Y}_h = \frac{1}{N_h}\sum_{i=1}^{N_h} Y_{hi}, \qquad \bar{y}_h = \frac{1}{n_h}\sum_{i=1}^{n_h} Y_{hi}, \tag{2.14}$$

and the stratum variance is

$$S_h^2 = \frac{1}{N_h - 1} \sum_{i=1}^{N_h} (Y_{hi} - \bar{Y}_h)^2. \tag{2.15}$$

The stratified estimator of the population mean is

$$\bar{y}_{st} = \sum_{h=1}^{L} W_h \, \bar{y}_h. \tag{2.16}$$

This estimator equals the ordinary sample mean $\bar{y}$ whenever all strata are sampled with proportional allocation:

$$\frac{n_h}{N_h} = \frac{n}{N}. \tag{2.17}$$

Unbiasedness and Variance Properties

Theorem 5.1 of [9] states that if $\bar{y}_h$ is unbiased for each stratum mean $\bar{Y}_h$, then the stratified estimator $\bar{y}_{st}$ is unbiased for $\bar{Y}$.

Independence of sampling across strata gives the variance formula of Theorem 5.2:

$$V(\bar{y}_{st}) = \sum_{h=1}^{L} W_h^2 \, V(\bar{y}_h). \tag{2.18}$$

Since

$$V(\bar{y}_h) = (1 - f_h) \, \frac{S_h^2}{n_h}, \tag{2.19}$$

Theorem 5.3 yields the finite-population variance expression:

$$V(\bar{y}_{st}) = \frac{1}{N^2} \sum_{h=1}^{L} N_h (N_h - n_h) \frac{S_h^2}{n_h} = \sum_{h=1}^{L} W_h^2 \frac{S_h^2}{n_h} (1 - f_h). \tag{2.20}$$

Variance Estimation

An unbiased estimator of the variance of the stratified mean is given by Theorem 5.5:

$$v(\bar{y}_{st}) = \frac{1}{N^2} \sum_{h=1}^{L} N_h (N_h - n_h) \frac{s_h^2}{n_h} = \sum_{h=1}^{L} \frac{W_h^2 s_h^2}{n_h} - \sum_{h=1}^{L} \frac{W_h s_h^2}{N}, \tag{2.21}$$

where s_h^2 denotes the sample variance within each stratum.

Optimal Allocation

If sampling costs differ across strata, with total cost

$$C = c_0 + \sum_{h=1}^{L} c_h n_h, \tag{2.22}$$

then Theorem 5.6 shows that the variance of $\bar{y}_{st}$ is minimized for a fixed total cost when

$$n_h \propto \frac{W_h S_h}{\sqrt{c_h}}. \tag{2.23}$$

Equivalently,

$$\frac{n_h}{n} = \frac{W_h S_h / \sqrt{c_h}}{\sum_{j=1}^{L} W_j S_j / \sqrt{c_j}}. \tag{2.24}$$

If all costs are equal, the optimal allocation becomes the classical Neyman allocation [11] described in Theorem 5.7:

$$n_h = n \frac{W_h S_h}{\sum_{j=1}^{L} W_j S_j} = n \frac{N_h S_h}{\sum_{j=1}^{L} N_j S_j}. \tag{2.25}$$

Mean Squared Error

Stratified estimators may become biased if incorrect weights w_h are used in place of W_h. Equation 5A.9 in [9] provides the mean squared error:

$$\text{MSE}(\bar{y}_{st}) = \sum_{h=1}^{L} \frac{w_h^2 s S_h^2}{n_h}(1 - f_h) + \left[\sum_{h=1}^{L} (w_h - W_h)\bar{Y}_h\right]^2. \tag{2.26}$$

This decomposition parallels the bias–variance trade-offs that appear later in bootstrap-based frameworks.

Stratified Bootstrap Methods

The classical theory extends naturally to the bootstrap. Instead of drawing a single sample from each population stratum, the bootstrap resamples with replacement from the empirical stratum distributions:

$$D_h^* \sim \text{Multinomial}(n_h; 1/N_h, \ldots, 1/N_h), \tag{2.27}$$

leading to the bootstrap version of the stratified estimator,

$$\bar{y}^*_{st} = \sum_{h=1}^{L} W_h \, \bar{y}^*_h. \tag{2.28}$$

Stratified bootstrap techniques reduce Monte Carlo variance and improve approximation accuracy in heterogeneous or imbalanced datasets [8], [10]. Variants use weighted resampling, strata defined by heterogeneity measures, or adaptive allocation rules analogous to Neyman allocation.

These ideas form a precursor to the oversampling stratified bootstrap methods developed in Chapter 4, where strata are defined through Mahalanobis distances, heterogeneity scores, or other structural features relevant to high-dimensional signal data.

Summary

Stratified sampling provides the theoretical basis for many advanced resampling procedures. The variance decompositions, allocation strategies, and bias–variance relationships established in classical sampling theory underpin modern stratified bootstrap methods. These foundations justify the development of adaptive, heterogeneity-aware, and imbalance-correcting bootstrap procedures that appear in later chapters.

2.2 Special Block Bootstrap Methods

Bootstrap methods for independent data are well established, but many applications in signal processing and time series analysis involve dependent observations. In such cases, resampling individual data points destroys the temporal structure and leads to unreliable inference. Block bootstrap methods overcome this limitation by resampling contiguous blocks of observations and are therefore central to bootstrap methodology for stationary and weakly dependent processes [12], [13], [14]. This chapter reviews the main block bootstrap ideas and then focuses on special block bootstrap procedures that are particularly relevant for stationary time series, seasonal data, and complex dependence structures.

Throughout, let $\{X_t, t \in \mathbb{Z}\}$ be a strictly stationary time series with mean $\mu = \mathbb{E}[X_t]$ and autocovariance function

$$\gamma(k) = \mathrm{Cov}(X_t, X_{t+k}), \qquad k \in \mathbb{Z}. \tag{2.29}$$

Given a sample $X_1, \ldots, X_n$ from this process, many inferential problems concern statistics of the form

$$T_n = T(X_1, \ldots, X_n), \tag{2.30}$$

such as the sample mean, regression coefficients, or parameter estimators from time series models. For dependent data, the limiting distribution of T_n often involves long-run variances and is difficult to obtain explicitly. Block bootstrap methods approximate the sampling distribution of T_n by resampling blocks that preserve local dependence.

Classical Block Bootstrap for Stationary Observations

Künsch [2] introduced the block bootstrap for general stationary observations. The key idea is to treat contiguous segments of the series as basic resampling units instead of individual observations.

Let b denote a chosen block length. Define overlapping blocks of length b,

$$B_1 = (X_1, \ldots, X_b), \quad B_2 = (X_2, \ldots, X_{b+1}), \quad \ldots, \quad B_N = (X_N, \ldots, X_n), \tag{2.31}$$

where $N = n - b + 1$. The empirical distribution of blocks is

$$\hat{P}_B = \frac{1}{N} \sum_{i=1}^{N} \delta_{B_i}, \tag{2.32}$$

where δ_{B_i} denotes the Dirac measure at block B_i, assigning probability one to B_i.

A block bootstrap sample of length n is obtained by drawing K blocks with replacement from $\hat{P}_B$ and concatenating them,

$$B_{I_1}, B_{I_2}, \ldots, B_{I_K}, \qquad I_k \sim \text{i.i.d. Uniform}\{1, \ldots, N\}, \tag{2.33}$$

then truncating to the first n observations. The block length b increases with n such that

$$b \to \infty, \qquad \frac{b}{n} \to 0, \tag{2.34}$$

which balances bias and variance contributions from the resampling of dependent data. Under appropriate mixing conditions on $\{X_t\}$, the block bootstrap consistently approximates the distribution of suitably normalized functionals T_n [2].

For example, consider the sample mean

$$\bar{X}_n = \frac{1}{n} \sum_{t=1}^{n} X_t. \tag{2.35}$$

Under standard assumptions, $\sqrt{n}(\bar{X}_n - \mu)$ converges in distribution to $\mathcal{N}(0, \sigma^2)$ with long-run variance

$$\sigma^2 = \gamma(0) + 2\sum_{k=1}^{\infty} \gamma(k). \tag{2.36}$$

The block bootstrap approximates the distribution of $\sqrt{n}(\bar{X}_n - \mu)$ by the conditional distribution of

$$\sqrt{n}(\bar{X}_n^* - \bar{X}_n), \tag{2.37}$$

where $\bar{X}_n^*$ is the mean of a block bootstrap sample.

Moving Blocks Bootstrap

Liu and Singh [3] proposed the moving blocks bootstrap as a specific implementation of the block bootstrap that emphasizes overlapping blocks. The construction is as described above: all possible overlapping blocks of length b are formed, and resampling is performed from this collection.

More formally, the Moving Blocks Bootstrap (MBB) algorithm can be summarised as:

1. Choose a block length b and let $N = n - b + 1$.
2. Form overlapping blocks $B_1, \ldots, B_N$ of length b.
3. Draw indices $I_1, \ldots, I_K$ independently and uniformly from $\{1, \ldots, N\}$, where $K = \lceil n/b \rceil$.
4. Concatenate the selected blocks and truncate to length n.

The resulting bootstrap series $(X_1^*, \ldots, X_n^*)$ preserves local dependence up to lag $b-1$. As in the general block bootstrap, consistency requires $b \to \infty$ and $b/n \to 0$ under suitable mixing assumptions.

The MBB is widely used for time series bootstrap due to its conceptual simplicity and its ability to capture short-range dependence. In signal processing, it has been used to assess the variability of spectral estimators, time-varying filters, and parametric time series models [12], [13], [14].

Circular Block Bootstrap

One drawback of MBB is the treatment of end effects. The first and last observations of the series appear in fewer blocks than interior points. Politis and Romano proposed the

Circular Block Bootstrap (CBB) to mitigate this issue by wrapping the time series around a circle [4].

Define the circularly extended series

$$\tilde{X}_t = X_{((t-1) \bmod n)+1}, \qquad t \in \mathbb{Z}, \tag{2.38}$$

and form blocks

$$\tilde{B}_i = (\tilde{X}_i, \ldots, \tilde{X}_{i+b-1}), \qquad i = 1, \ldots, n. \tag{2.39}$$

The empirical distribution of these blocks is

$$\hat{P}_{\text{CBB}} = \frac{1}{n} \sum_{i=1}^{n} \delta_{\tilde{B}_i}. \tag{2.40}$$

A circular block bootstrap series is obtained by resampling with replacement from $\hat{P}_{\text{CBB}}$ and concatenating the blocks until at least n observations are obtained, then truncating.

In contrast to MBB, each observation appears in exactly b blocks, which improves the homogeneity of the resampling scheme. For many statistics, CBB and MBB yield similar asymptotic behaviour, but CBB tends to perform better in finite samples when boundary effects are non-negligible [4].

Seasonal Block Bootstrap

Many time series in energy, climatology, transportation, and communication exhibit deterministic seasonal components. A typical model decomposes the process as

$$X_t = \mu_t + Y_t, \qquad \mu_t = \mu_{t-d}, \quad t \in \mathbb{Z}, \tag{2.41}$$

where d is an integer period and $\{Y_t\}$ is a strictly stationary zero-mean sequence [15]. If μ_t is not constant, the standard block bootstrap that treats X_t as stationary is not applicable. Seasonal block bootstrap methods adapt block resampling to this periodic mean structure.

Politis [15] introduced the Seasonal Block Bootstrap (SBB) for time series with a single known seasonality d. Suppose $N = nd$ for some integer n. The SBB procedure can be summarised as follows [15], [16]:

1. Partition the series into n seasonal cycles of length d.
2. Choose a block length $b < n$ and set $k = \lfloor n/b \rfloor$.
3. Draw indices $i_0, \ldots, i_{k-1}$ independently and uniformly from $\{1, \ldots, n - b + 1\}$.

4. Form blocks of length bd by concatenating b consecutive cycles,

$$X_{i_m d+1}, \ldots, X_{(i_m+b)d}, \qquad m = 0, \ldots, k-1. \tag{2.42}$$

5. Concatenate the selected blocks to form a bootstrap series of length $l = kbd$ and truncate or extend to the desired length.

In this construction, block starting points and block lengths are constrained to be integer multiples of d. The SBB is thus a version of the block bootstrap that respects the seasonal structure by resampling whole seasonal cycles and their neighbouring cycles together.

The SBB captures the joint behaviour of the deterministic seasonal component and the stationary residual. For statistics that are functionals of the deseasonalised process or for forecasting problems, this alignment is crucial.

Generalised Seasonal Block Bootstrap

The restriction that block size and starting points must be multiples of the period d can be limiting, for example when d is large relative to the sample size or when one wishes to use smaller blocks to capture short-range dependence. Dudek, Leśkow, Paparoditis and Politis [16] proposed the Generalised Seasonal Block Bootstrap (GSBB) and its circular version Generalised Seasonal Circular Block Bootstrap (GSCBB) to relax these constraints.

The GSBB and GSCBB operate on the same model

$$X_t = \mu_t + Y_t, \qquad \mu_t = \mu_{t-d}, \quad \{Y_t\} \text{ stationary}, \tag{2.43}$$

but allow block starting positions to shift by integer multiples of d around a given index. The GSCBB may be summarised as follows [16].

1. Choose a positive integer block size $b < N$ and set $l = \lfloor N/b \rfloor$.

2. For $t = 1, b+1, 2b+1, \ldots, lb+1$, define a set of admissible starting points

$$S_{t,N} = \{t - dR_{1,N}, t - d(R_{1,N}-1), \ldots, t, t+d, \ldots, t+d(R_{2,N}-1), t+dR_{2,N}\}, \tag{2.44}$$

where

$$R_{1,N} = \left\lfloor \frac{t-1}{d} \right\rfloor, \qquad R_{2,N} = \left\lfloor \frac{N-b-t}{d} \right\rfloor. \tag{2.45}$$

3. Draw k_t uniformly from $S_{t,N}$ and set

$$(X_t^*, \ldots, X_{t+b-1}^*) = (X_{k_t}, \ldots, X_{k_t+b-1}). \tag{2.46}$$

4. Join the resulting blocks to obtain $X_1^*, \ldots, X_{(l+1)b}^*$. Retain the first N observations as the bootstrap series.

This construction allows block starting points to fluctuate along the seasonal cycle while maintaining seasonal alignment. The circular design improves the treatment of boundary effects, similarly to the ordinary circular block bootstrap. Dudek et al. [16] established theoretical properties of GSBB and GSCBB and demonstrated their usefulness for time series with strong seasonal behaviour.

Tapered and Robust Seasonal Block Bootstrap

Tapered block bootstrap methods use tapering functions to smooth transitions at block boundaries, which reduces edge effects and variance inflation. Dudek, Paparoditis and Politis [17] introduced the Generalised Seasonal Tapered Circular Block Bootstrap (GSTBB) as a seasonal counterpart of tapered block methods. In GSTBB, each block is multiplied by a taper

$$w(j), \qquad j = 1, \ldots, b, \tag{2.47}$$

with $w(1)$ and $w(b)$ close to zero and $w(j)$ near one in the interior. The resulting pseudo-series is a weighted sum of overlapping tapered blocks. For seasonal series, the taper is applied in a way that respects seasonal alignment.

Tapering has two main effects. First, it reduces the influence of block boundaries on resampled series, which leads to more accurate approximation of covariance structures. Second, it can improve the numerical stability of bootstrap-based estimators of long-run variances and other functionals that are sensitive to high-frequency artefacts.

Robust extensions of block bootstrap procedures incorporate influence functions or non-random weights to downweight outliers and heavy-tailed observations [18], [19], [20], [21]. These ideas have recently been combined with seasonal block bootstrap structures to obtain robust seasonal block methods for heavy-tailed or contaminated time series. Building on GSBB, GSTBB and circular seasonal methods, block-robust approaches use standardized influence functions or block averages to produce bootstrap series that are less sensitive to extreme values.

Special Block Bootstrap Methods in Signal Processing

In signal processing applications, block bootstrap methods are often combined with parametric models such as Autoregressive Moving Average (ARMA) or Generalised Autoregressive Conditionally Heteroscedastic (GARCH). The bootstrap can operate on residuals from fitted models, on blocks of deseasonalised series, or on transformed quantities such as

innovations. For processes with time-varying volatility, several works have developed bootstrap techniques for Autoregressive Conditionally Heteroscedastic (ARCH) and GARCH models [22], [23], [24], [25], while other studies have considered bootstrap quantile estimation and model specification tests in the presence of conditional heteroskedasticity [26], [27]. These approaches complement block bootstrap methods by providing parametric or semi-parametric alternatives for volatility-rich environments.

Seasonal block bootstrap methods such as SBB, GSBB, GSCBB and GSTBB provide nonparametric resampling tools that respect seasonal dependence structures in the observed data. They are particularly attractive for forecasting problems in energy and environmental applications, where multi-periodic behaviour, time-varying volatility, and outliers often coexist.

Recent work has extended circular seasonal block ideas to more complex multi-seasonal structures, where several seasonalities (for example daily, weekly and yearly cycles) interact [16], [17]. In such settings, the design of blocks and their resampling rules must reflect multiple periodicities simultaneously. This motivates more flexible circular block constructions and robustifications, as developed later in Chapter 5.

Summary

Block bootstrap methods are essential for resampling dependent data. The moving blocks and circular block bootstrap provide general tools for stationary time series, while seasonal block bootstrap methods adapt the block concept to processes with deterministic seasonal patterns. Generalized and tapered seasonal block bootstraps relax constraints on block starting points and introduce tapering to reduce edge effects. Robust variants incorporate influence-function-based weights or block averages to handle outliers and heavy tails.

These special block bootstrap procedures form the methodological foundation for the complex seasonal bootstrap methods and volatility-aware block bootstrap variants developed in later chapters. They demonstrate how structural features such as dependence, seasonality and volatility can be integrated into the resampling mechanism, which is a central theme of this dissertation.

2.3 Robust Bootstrap Methods

Many statistical procedures are sensitive to outliers, heavy-tailed noise or structural contamination in the data. In such settings, the classical bootstrap may fail to provide reliable approximations of sampling distributions because it reproduces influential observations with the same probability as non-contaminated ones. This can distort variance estimates,

widen or shrink confidence intervals or produce unstable estimators. Robust bootstrap methods aim to mitigate these issues by combining tools from robust statistics [6], [19], [20] with resampling principles.

Let $T(\hat{P}_n)$ denote an estimator of a parameter $T(P)$ computed from the empirical distribution $\hat{P}_n$. If the estimator is sensitive to outliers, the classical bootstrap distribution of $T(\hat{P}_n^*)$ may reflect these distortions. Robust methods typically modify either the estimator, the resampling weights, or the empirical distribution to control the influence of extreme observations.

Influence Functions and Robust Estimation

The influence function of a functional T at point x under distribution P is defined as [19]

$$\mathrm{IF}(x; T, P) = \lim_{\varepsilon \to 0} \frac{T\big((1-\varepsilon)P + \varepsilon \delta_x\big) - T(P)}{\varepsilon}. \tag{2.48}$$

If $\mathrm{IF}(x; T, P)$ is unbounded, the estimator is sensitive to contamination at x. Robust estimators replace T with a functional whose influence function is bounded or redescending. When a robust estimator $\tilde{T}(\hat{P}_n)$ is used in place of $T(\hat{P}_n)$, a bootstrap distribution based on $\tilde{T}$ inherits these robustness properties.

For example, if $\tilde{T}$ satisfies the von Mises expansion,

$$\tilde{T}(\hat{P}_n) = \tilde{T}(P) + \frac{1}{n}\sum_{i=1}^{n} \mathrm{IF}(X_i; \tilde{T}, P) + o_p(n^{-1/2}), \tag{2.49}$$

and the influence function is bounded, the bootstrap approximation of

$$\sqrt{n}\big(\tilde{T}(\hat{P}_n^*) - \tilde{T}(\hat{P}_n)\big) \tag{2.50}$$

is significantly more stable compared to classical estimators.

Weighted and Winsorized Bootstrap

A widely used robustification is to modify the empirical distribution by applying weights to downweight outlying observations. Define weights $w_1, \ldots, w_n$ and the weighted empirical measure

$$\hat{P}_n^{(w)} = \sum_{i=1}^{n} w_i \delta_{X_i}, \qquad \sum_{i=1}^{n} w_i = 1. \tag{2.51}$$

A weighted bootstrap sample is drawn as

$$X_i^* \sim \hat{P}_n^{(w)}. \tag{2.52}$$

Common choices include:

$$w_i \propto \psi\left(\frac{X_i - \hat{\mu}}{\hat{\sigma}}\right), \tag{2.53}$$

where ψ is a bounded score function from M-Estimation. Alternatively, observations beyond a threshold may be Winsorized or trimmed before resampling.

These approaches reduce the probability that extreme data points dominate the bootstrap distribution. Weighted bootstrap techniques are particularly useful in heavy-tailed settings or when contamination is asymmetric.

Robust Block Bootstrap

In time series and signal processing, outliers often propagate through the resampling mechanism if entire blocks containing anomalous observations are repeatedly selected. This motivates robust block bootstrap constructions that incorporate influence functions or blockwise transformations.

One approach replaces each block with a robust transformation such as a block average or a blockwise M-Estimation. If B_i denotes the i-th block,

$$\tilde{B}_i = g(B_i), \tag{2.54}$$

where g is a robust functional applied to the block. For example,

$$g(B_i) = \frac{1}{|B_i|} \sum_{t \in B_i} \psi\big((X_t - \hat{\mu})/\hat{\sigma}\big), \tag{2.55}$$

with ψ bounded. The bootstrap then resamples $\tilde{B}_i$ rather than the raw blocks B_i.

A second approach employs standardized influence functions evaluated at the block level. If $\mathrm{IF}(B_i)$ denotes the influence contribution of the block to a statistic of interest, block weights can be modified as

$$w_i \propto \frac{1}{1 + |\mathrm{IF}(B_i)|}. \tag{2.56}$$

Such blockwise weighting improves robustness under heavy-tailed noise or structural changes in variance. Robust block methods are essential for complex seasonal or high-volatility data, where extreme observations tend to cluster.

Robust Bootstrap for Heavy-Tailed Distributions

When the underlying data distribution has heavy tails, classical bootstrap variance estimates may diverge or converge slowly. A common remedy is the m-out-of-n bootstrap, where bootstrap samples of size $m < n$ are drawn [28]. The asymptotic regime requires

$$m \to \infty, \qquad \frac{m}{n} \to 0. \tag{2.57}$$

Under heavy tails, the m-out-of-n bootstrap better approximates the limiting distribution of many estimators, including sample means and regression coefficients.

Another strategy is to bootstrap from a smoothed and truncated empirical distribution,

$$\hat{P}_n^{\text{trunc}} = \frac{1}{n} \sum_{i=1}^{n} \delta_{\tau(X_i)}, \tag{2.58}$$

where τ is a truncation operator such as

$$\tau(x) = \text{sign}(x) \min(|x|, c). \tag{2.59}$$

Such truncation stabilizes variance estimates in settings with extreme values or non-Gaussian contamination.

Robust Methods in Signal Processing

In signal processing applications, robust bootstrap techniques are particularly important because measurement noise, interference, multipath effects or sensor faults may introduce structural contamination. Recent signal processing literature has incorporated robust bootstrap components into spectral estimation, adaptive filtering and autoregressive modeling [14], [18], [21].

Typical strategies include:

- resampling robust residuals rather than raw data,
- applying robust score functions to blocks,
- downweighting high-energy transients,
- combining bootstrap with robust model fitting for Autoregressive (AR), ARMA or GARCH models.

Summary

Robust bootstrap methods modify the resampling mechanism or the underlying estimator to reduce sensitivity to outliers, heavy tails or structural contamination. Tools from robust statistics, including influence functions, M-Estimations, weighted empirical distributions and truncation schemes, play key roles in constructing robust bootstrap distributions. Robust block bootstrap variants further extend these ideas to dependent data. These methods provide essential theoretical and practical foundations for the robust seasonal bootstrap and volatility-aware resampling procedures developed in later chapters.

3 Summary of Contributions

Chapters 4 to 6 of this dissertation present the main methodological and applied contributions developed to advance bootstrap resampling for structured, heterogeneous, imbalanced, and volatile data environments. The contributions span three major families of methods: oversampling stratified bootstrap techniques, complex seasonal bootstrap procedures, and specialized influence-function or computationally efficient bootstrap variants. The following overview summarises the key ideas and innovations introduced in each chapter.

- **Oversampling Stratified Bootstrap Methods (Chapter 4)**
  - Development of bootstrap methods for classification tasks with severe class imbalance, high dimensionality, and heterogeneous structure.
  - **Mahalanobis-distance Stratified Bootstrap (MSBoot)** : A distance-based oversampling strategy that allocates sampling weights using Mahalanobis geometry. A journal article based on this method is currently in preparation.
  - **Heterogeneity-Stratified Bootstrap (HS-Boot)** : A resampling framework that uses heterogeneity measures to create adaptive strata, enabling the bootstrap to reflect varying complexity across the feature space. The corresponding study was published as “Heterogeneity-Stratified Bootstrap Oversampling for Training a Spoiled Food Detector” at DSP 2023 [29].
  - **The Heterogeneity-Intensified and Heterogeneity Ratio-Stratified Bootstrap (HiS- and HeRS-Boot)** : Two extensions that intensify heterogeneity effects or re-weight sampling ratios, enhancing performance in domains with extreme imbalance or strong structural variability. These methods were published in IEEE SENSORS 2023 under the title “The Heterogeneity-Intensified and Heterogeneity Ratio-Stratified Bootstrap (HiS- and HeRS-Boot) Oversampling to Boost a Detector Performance” [30].
  - Demonstration of these methods through case studies including intracranial EEG classification and detection of spoiled food and allergens.

- **Complex Seasonal Bootstrap Methods(Chapter 5)**
  - Introduction of bootstrap procedures capable of handling multi-seasonal periodicity in time series.
  - **Complex Seasonal Circular Block Bootstrap (XSCBB)** : A generalisation of the circular block bootstrap that preserves multiple overlapping seasonal patterns simultaneously. The method was published at EUSIPCO 2023 as "Complex Seasonal Circular Block Bootstrap for Electricity Load Forecasting" [31].
  - **The Block Robust Empirical Standardised Influence Function (BRESIF)-XSCBB and the Block Average (BA)-XSCBB** : Robust extensions that incorporate influence functions or block-averaging strategies, improving resistance to heavy tails, volatility changes, and structural irregularities. These methods were published in EUSIPCO 2024 as "Integrating Block Robust Empirical Standardised Influence Function and Block Average to Robustify Seasonal Block Bootstrap Methods" [32].
  - Applications to forecasting and uncertainty quantification in energy time series.
- **Other Special Bootstrap Methods (Chapter 6)**
  - **Block-Toeplitz Bootstrap (Blitz-Boot)** : An efficient bootstrap designed for settings where rapid inference is required, offering approximate resampling under tight computational constraints. This work appears in the 2024 CoSeRa Workshop under the title "The Blitz-Boot Approach: Improving the Accuracy of Joint Direction-of-Arrival and Frequency Estimation" [33].
  - **Cyclic Volatility Influence Function (CV-IF) Bootstrap** : A framework for modelling and resampling data with cyclical or regime-dependent volatility, especially relevant to financial and industrial signals. This work appears in the 2025 International Conference on Digital Signal Processing (DSP) under the title "The Cyclic-Volatility Influence Function Bootstrap – A Resampling Framework for Periodic and Volatility-Driven Signals" [34].
  - Under the framework of the CV-IF Bootstrap, the introduction of **GARCH-IF** and **Gated Sin-IF** variants, enabling improved uncertainty quantification in complex volatility environments [34].
  - Empirical studies demonstrating how these specialised methods outperform classical bootstraps when volatility structure plays a central role.

Across these chapters, the contributions collectively broaden the scope of bootstrap methodology by addressing dependence, heterogeneity, imbalance, multi-seasonality, volatility patterns, and computational efficiency. Chapters 4-6 demonstrate that thoughtfully designed bootstrap procedures can improve inference quality in modern signal-processing and time-series applications.

4 Oversampling Stratified Bootstrap Methods

This chapter introduces a family of oversampling stratified bootstrap methods designed to address one of the most persistent challenges in modern classification and detection tasks: severe class imbalance combined with heterogeneous data structure. Many real-world signal processing applications, such as intracranial EEG analysis, environmental monitoring, or sensor-based quality assessment, involve rare but critical events that are underrepresented in the data. Standard learning algorithms often fail to recognise these minority patterns, while traditional oversampling approaches may distort the underlying geometry or introduce synthetic noise. Bootstrap-based resampling provides a principled alternative, but existing methods do not adequately account for heterogeneity or the complex statistical structure of high-dimensional signals.

The chapter develops three major methodological contributions that extend the bootstrap framework to better capture these characteristics. The first is the *Mahalanobis-distance Stratified Bootstrap (MS-Boot)*, which uses distance-based information to guide the allocation of sampling weights. The second contribution is the *Heterogeneity-Stratified Bootstrap (HS-Boot)*, a versatile framework that partitions the feature space into strata defined by local heterogeneity measures, enabling the bootstrap to adapt its sampling behaviour to regions of varying complexity. Building on this foundation, the chapter introduces two further extensions, the *Heterogeneity-Intensified Bootstrap (HiS-Boot)* and the *Heterogeneity Ratio-Stratified Bootstrap (HeRS-Boot)*, which provide additional control over sampling density and class representation in highly imbalanced or structurally diverse environments.

Throughout the chapter, these methods are evaluated through real life applications. The results demonstrate that incorporating heterogeneity and distance information into the bootstrap process leads to more reliable minority-class representation, improved classifier performance, and more stable uncertainty estimates. This chapter thus establishes a comprehensive and adaptable set of resampling tools for addressing imbalance and structure in complex signal-processing datasets.

4.1 Mahalanobis-distance Stratified Bootstrap (MS-Boot)

Severe class imbalance remains a prime obstacle to reliable pattern recognition in domains ranging from fraud analytics to rare-event medicine. We introduce the Mahalanobis distance Stratified Bootstrap (MS-Boot), a hyperparameter-free resampling scheme that assigns each minority instance a sampling weight proportional to its Mahalanobis distance from the minority centroid in a feature space and enforces a specified class ratio per training split. By concentrating the learner on under-represented yet informative regions of the minority set, MS-Boot avoids the variance inflation of uniform or synthetic oversampling. We provide a formal guarantee that, for a broad family of margin-based classifiers, distance-weighted resampling strictly lowers the expected empirical risk relative to uniform bootstrapping and to no resampling under mild regularity assumptions. Empirically, MS-Boot is classifier-agnostic: across a diverse set of standard models, it consistently improves detection performance on public EEG-seizure datasets. The results confirm that MS-Boot offers a principled, plug-and-play alternative to heuristic remedies such as the SMOTE, thus delivering consistent sensitivity gains without sacrificing precision or adding model complexity while keeping training priors balanced.

4.1.1 Introduction

Learning from severely imbalanced class distributions is a long-standing bottleneck in pattern recognition, with applications as diverse as fraud detection, predictive maintenance, and rare-disease diagnostics [35], [36]. When the minority class represents less than 5% of the available sample, which is typical in EEG seizure datasets or financial anomalies, the empirical risk is dominated by majority-class behavior. Consequently, conventional learners gravitate toward trivial, majority-biased decision rules with deceptively high overall accuracy yet poor sensitivity to the rare events that matter most in practice. In this work we address this regime by re-balancing each training split to a desired class prior via a distance-aware bootstrap, while preserving the decision-relevant structure in the minority class.

Cost-sensitive learning re-weights the loss but requires manual or heuristic cost matrices and reliable class-prior estimates, which are rarely known a priori [37]. Synthetic oversampling e.g., SMOTE [5], Borderline-SMOTE [38], and ADASYN [39], fabricates minority instances in feature space; however, such heuristics can inject unrealistic points, blur class boundaries, and inflate variance, especially in high-dimensional regimes.

Uniform bootstrap oversampling preserves empirical instances but allocates sampling budget evenly across the minority set. When re-balancing to 1:1, uniform oversampling indiscriminately replicates minority points, including redundant or uninformative ones,

so improvements over the naïve baseline are often marginal [7].

Stratified variants refine the plain bootstrap by altering the resampling probabilities within predefined strata to better reflect the structure in the data. A range of stratified-bootstrap schemes has been proposed and shown to enhance established detection and estimation procedures [29], [30], [40]. These results motivate distance-aware stratification for class-imbalance settings, where the resampling distribution can be shaped to emphasize informative minority regions while still drawing only from observed examples.

Intuitively, not all minority instances are equally valuable. Those geometrically remote from the minority centroid often lie near class boundaries, carrying disproportionate information about the decision surface. Recent work has therefore explored distance-weighted sampling rules [41], yet existing methods either (i) rely on ad-hoc distance metrics in the raw feature space, (ii) introduce hyper-parameters that must be tuned for every new dataset, or (iii) lack formal performance guarantees.

We propose the MS-Boot, a hyperparameter-free resampling scheme that remedies the foregoing shortcomings in one unified procedure. MS-Boot operates in the available feature space, computes each minority instance's Mahalanobis distance from the minority centroid using the sample covariance, and assigns a sampling probability proportional to that distance. This analytically tractable rule concentrates the learner on informative minority regions while enforcing a certain class ratio in every bootstrap replicate.

The key contributions of this work are:

1 We propose and formalise MS-Boot, a hyperparameter-free, distance-aware bootstrap that is drop-in compatible with any learner and feature modality, and that re-balances each training split to a given prior.

2 We derive tight first-order bounds via an influence-function analysis of sample-weighted empirical risk, showing that MS-Boot reduces the expected empirical risk for margin-based classifiers under mild regularity conditions. We also provide a geometric interpretation based on the monotone relation between a signed Mahalanobis coordinate and the latent margin.

3 We demonstrate consistent, statistically significant gains across diverse classifiers on a public intracranial EEG seizure dataset.

The remainder of the section is organized as follows. Subsection 4.1.2 formulates the problem. Subsection 4.1.3 introduces the MS-Boot and develops the theoretical analysis. Subsection 4.1.4 describes the details of the practical case study: the datasets, the feature extraction method, and the evaluation metrics. Subsection 4.1.5 reports results across seven classifiers and three resampling strategies including the proposed MS-Boot. Subsection 4.1.6 concludes and discusses limitations.

4.1.2 Problem Formulation

Given the labelled dataset

$$\mathcal{D} = \big\{(\mathbf{z}_i,\, y_i)\big\}_{i=1}^{N}, \qquad y_i \in \{0,1\},\;\; \mathbf{z}_i \in \mathbb{R}^m. \tag{4.1}$$

Let the set of indices $\mathcal{I}_1 = \{i : y_i = 1\}$ and $\mathcal{I}_0 = \{i : y_i = 0\}$ with sample sizes $N_1 = |\mathcal{I}_1| \ll N_0 = |\mathcal{I}_0|$, and total observations $N = N_0 + N_1$. The class priors are $p_1 = N_1/N$ and $p_0 = 1 - p_1$, and the sample mean and sample covariance matrix of class-1, respectively, are

$$\bar{\mathbf{z}} = \tfrac{1}{N_1} \sum_{i \in \mathcal{I}_1} \mathbf{z}_i, \qquad \mathbf{S} = \tfrac{1}{N_1 - 1} \sum_{i \in \mathcal{I}_1} (\mathbf{z}_i - \bar{\mathbf{z}})(\mathbf{z}_i - \bar{\mathbf{z}})^\intercal. \tag{4.2}$$

The squared Mahalanobis distance of a minority point is

$$d_i^2 = (\mathbf{z}_i - \bar{\mathbf{z}})^\intercal \mathbf{S}^{-1} (\mathbf{z}_i - \bar{\mathbf{z}}), \qquad i \in \mathcal{I}_1. \tag{4.3}$$

We compare three resampling schemes:

(a) No Oversampling (NO): train on the original $\mathcal{D}$.

(b) Uniform bootstrap oversampling (BO) draw N_1^* minority indices with replacement, each with probability $1/N_1$, and combine these with all N_0 majority instances.

(c) MS-Boot (MB): draw N_1^* minority indices with replacement, where $i \in \mathcal{I}_1$ is selected with probability

$$w_i = \frac{d_i}{\sum_{j \in \mathcal{I}_1} d_j}, \tag{4.4}$$

and combine these with all N_0 majority instances.

Define the uniform probability vector over N_1 elements $\mathbf{u} = (1/N_1, \ldots, 1/N_1)$. Let the bootstrap multinomial count vector over indices in $\mathcal{I}_1$ be $\boldsymbol{\delta}^{(b)} \sim \text{Multinomial}(N_1^*, \mathbf{p})$, $\boldsymbol{\delta}^{(b)} = \left(\delta_i^{(b)} : i \in \mathcal{I}_1\right) \in \mathbb{N}^{|\mathcal{I}_1|}$, $\sum_{i \in \mathcal{I}_1} \delta_i^{(b)} = N_1^*$. For BO, set $\mathbf{p} = \mathbf{u}$; for MB set $\mathbf{p} = \mathbf{w}$ from (4.4).

The b-th bootstrap sample with size $N_0 + N_1^*$ is

$$\widehat{\mathcal{D}}^{(b)} = \big\{(\mathbf{z}_i, y_i)\big\}_{i \in \mathcal{I}_0} \;\cup\; \Big\{(\mathbf{z}_i, y_i) \text{ repeated } \boldsymbol{\delta}_i^{(b)} \text{ times}\Big\}_{i \in \mathcal{I}_1}. \tag{4.5}$$

Any classifier with parameters $\boldsymbol{\theta}$ is trained by empirical–risk minimization (ERM) on $\widehat{\mathcal{D}}^{(b)}$:

$$\widehat{\boldsymbol{\theta}}^{(b)} = \arg\min_{\boldsymbol{\theta}} \frac{1}{\big|\widehat{\mathcal{D}}^{(b)}\big|} \sum_{(\mathbf{z},y) \in \widehat{\mathcal{D}}^{(b)}} \ell\big(y,\, s_{\boldsymbol{\theta}}(\mathbf{z})\big) + \lambda \Omega(\boldsymbol{\theta}), \tag{4.6}$$

where $\lambda > 0$ is the regularisation strength, and $\Omega(\boldsymbol{\theta})$ a penalty function on the parameters $\boldsymbol{\theta}$. The score $s_{\boldsymbol{\theta}}(\mathbf{z})$ is an algorithm–specific score and ℓ is convex in that score.

For example, consider a linear classifier with the parameters $\boldsymbol{\theta} = (\mathbf{w}, c)$ and score function $s_{\boldsymbol{\theta}}(\mathbf{z}) = \mathbf{w}^{\intercal}\mathbf{z} + c$. Using the logistic loss with labels $y \in \{0, 1\}$, $\ell(y, t) = \log\big(1 + e^{t}\big) - y\,t$, the regularised ERM objective on a training set $\widehat{\mathcal{D}}^{(b)}$ takes the form and a standard ℓ_2 (ridge) regulariser is $\Omega(\boldsymbol{\theta}) = \|\mathbf{w}\|_2^2 = \sum_{j=1}^{m} w_j^2$.

On a held-out test set, with true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), we report the empirical rates

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{4.7}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}, \tag{4.8}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{4.9}$$

$$F_1 = \frac{2\,\text{Precision} \times \text{TPR}}{\text{Precision} + \text{TPR}}. \tag{4.10}$$

These are empirical estimates of the population quantities, i.e. test power $1 - \beta$ and Type-I error α, at the chosen threshold.

Our goal is to design a bootstrap rule that, on average over repeated training runs, yields classifiers with high statistical power and low Type-I error. Given a realised classifier $\widehat{f}$, its population Type-I error and power under P are, respectively, $\alpha(\widehat{f}) = \Pr\{\widehat{f}(\mathbf{Z}) = 1 \mid \mathbf{y} = 0\}$ and $\text{Power}(\widehat{f}) = \Pr\{\widehat{f}(\mathbf{Z}) = 1 \mid \mathbf{y} = 1\}$, which are deterministic quantities once $\widehat{f}$ is fixed. The randomness in our formulation comes entirely from the bootstrap mechanism that produces $\widehat{f}$. We write

$$J(\widehat{\boldsymbol{\theta}}) \;\equiv\; J(\widehat{f}) \;=\; \varphi\big(\text{Power}(\widehat{f}),\, \alpha(\widehat{f})\big), \tag{4.11}$$

for some deterministic utility function φ that is nondecreasing in its first argument (higher power is better) and non-increasing in its second argument (lower Type-I error is better). Our target weight vector is then

$$\mathbf{p}^{*} \;=\; \arg\max_{\mathbf{p} \in \Delta_{N_1 - 1}} \; \mathbb{E}_{\text{boot}(\mathbf{p})}\big[\, J(\widehat{\theta})\,\big], \tag{4.12}$$

where the expectation $\mathbb{E}_{\text{boot}(\mathbf{p})}[\cdot]$ is taken with respect to the randomness of the bootstrap resampling.

Algorithm 1: MS-Boot (Mahalanobis–distance Stratified Bootstrap)

Step 1 Collect labeled dataset $\mathcal{D} = \{(\mathbf{z}_i, y_i)\}_{i=1}^{N}$ with $y_i \in \{0, 1\}$; number of bootstrap rounds B; target minority draw N_1^* (for e.g. $N_1^* = N_0$ for 1:1).

Step 2 Draw bootstrap samples $\{\widehat{\mathcal{D}}^{(b)}\}_{b=1}^{B}$

Step 3 Split indices $\mathcal{I}_1 = \{i : y_i = 1\}$, $\mathcal{I}_0 = \{i : y_i = 0\}$;

Step 4 Compute class-1 sample mean and sample covariance matrix according to Eq. (4.2), the Mahalanobis distances according to Eq. (4.3), and the sampling weights with fallback to uniform if $\sum_{j \in \mathcal{I}_1} d_j = 0$ according to Eq. (4.4).

Step 5 For $b = 1$ **to** B:

- Draw minority bootstrap counts:

$$\big(\delta_i^{(b)}\big)_{i \in \mathcal{I}_1} \sim \text{Multinomial}\big(N_1^*,\ (w_i)_{i \in \mathcal{I}_1}\big).$$

- Form the bootstrapped sample:

$$\widehat{\mathcal{D}}^{(b)} \leftarrow \big\{(\mathbf{z}_i, y_i)\big\}_{i \in \mathcal{I}_0} \cup \Big\{(\mathbf{z}_i, y_i) \text{ replicated } \delta_i^{(b)} \text{ times}\Big\}_{i \in \mathcal{I}_1}.$$

- Train a classifier on $\widehat{\mathcal{D}}^{(b)}$ via ERM.

4.1.3 Methods

To preferentially resample observations that are "extreme" in the feature space, we compute a Mahalanobis-distance–based weight for each training instance and use these weights in a non-uniform bootstrap.

Weighted Bootstrap via Mahalanobis Distance

The squared Mahalanobis distance of each minority observation $i \in \mathcal{I}_1$ and the sampling probabilities are as in (4.3) and (4.4). Observations with larger Mahalanobis distance, i.e., more "outlying", receive higher resampling probability.

Algorithm 1 summarises the proposed resampling method. Subsequent analysis or model fitting is performed on each $\widehat{\mathcal{D}}^{(b)}$, and results are aggregated over b [7], [42].

Distance–Weighted Oversampling on Class–Imbalance Metrics

For the quadratic discriminant analysis (QDA), decision trees/Classification and Regression Trees (CART), Naïve Bayes (NB), Logistic Regression (LR), Multi Layer Perceptrons (MLPs), Support Vector Machine (SVM), the fitted score can be written in affine form after a (possibly nonlinear) feature map ψ:

$$s_{\boldsymbol{\theta}}(\mathbf{z}) = \big\langle \widetilde{\psi}(\mathbf{z}),\, \widetilde{\boldsymbol{\theta}} \big\rangle, \qquad \widetilde{\psi}(\mathbf{z}) \coloneqq \begin{bmatrix} \psi(\mathbf{z}) \\ 1 \end{bmatrix}, \tag{4.13}$$

so the intercept is absorbed into $\widetilde{\boldsymbol{\theta}}$. Here $\psi(\mathbf{z})$ is algorithm-specific (e.g., quadratic features for QDA, kernel features for SVM, last hidden-layer activations for MLPs). For tree-based classifiers, a scalar score can be defined by leaf-wise log-odds: $s_{\boldsymbol{\theta}}(\mathbf{z}) = \sum_l \boldsymbol{\theta}_l \, \mathbb{I}\{\mathbf{z} \in \text{leaf } l\}$, proportional to the difference in log posteriors in the terminal node [43].

Define the population margin

$$\gamma(\mathbf{z}) = \Pr\{y = 1 \mid \mathbf{z}\} - \tfrac{1}{2}. \tag{4.14}$$

Let P_1 denote the class-1 conditional distribution and write

$$m_1(\boldsymbol{\theta}) = \mathbb{E}_{P_1}\big[s_{\boldsymbol{\theta}}(\mathbf{Z})\big], \qquad v_1^2(\boldsymbol{\theta}) = \mathrm{Var}_{P_1}\big[s_{\boldsymbol{\theta}}(\mathbf{Z})\big].$$

Set $\widetilde{\psi}_1 \coloneqq \mathbb{E}_{P_1}[\widetilde{\psi}(\mathbf{Z})]$, so that $m_1(\boldsymbol{\theta}) = \langle \widetilde{\psi}_1, \widetilde{\boldsymbol{\theta}} \rangle$. Assume a location–scale form for the class-1 score:

$$s_{\widehat{\boldsymbol{\theta}}}(\mathbf{Z}) \,\big|\, (y = 1) \;\overset{d}{=}\; m_1(\widehat{\boldsymbol{\theta}}) + v_1(\widehat{\boldsymbol{\theta}})\, U,$$

for a fixed, parameter-free random variable U. Then the power of the fitted rule $\widehat{f}(\mathbf{z}) = \mathbb{I}\{s_{\widehat{\boldsymbol{\theta}}}(\mathbf{z}) \geq 0\}$ is

$$\begin{aligned} \mathrm{Power}\big(\widehat{f}\big) &= \Pr\{s_{\widehat{\boldsymbol{\theta}}}(\mathbf{Z}) \geq 0 \mid y = 1\} \\ &= 1 - F_U\left(-\frac{m_1(\widehat{\boldsymbol{\theta}})}{v_1(\widehat{\boldsymbol{\theta}})} \right) \;=:\; \Phi\left(\frac{m_1(\widehat{\boldsymbol{\theta}})}{v_1(\widehat{\boldsymbol{\theta}})} \right), \end{aligned} \tag{4.15}$$

where Φ is strictly increasing, determined by U.

Without the location–scale assumption, the one-sided Chebyshev's inequality (Cantelli's inequality) bound [44] gives the distribution-free inequality

$$\begin{aligned} \mathrm{Power}(\widehat{f}) &= \Pr\big\{ s_{\widehat{\boldsymbol{\theta}}}(\mathbf{Z}) \geq 0 \mid y = 1 \big\} \\ &\geq 1 - \frac{v_1^2(\widehat{\boldsymbol{\theta}})}{v_1^2(\widehat{\boldsymbol{\theta}}) + m_1^2(\widehat{\boldsymbol{\theta}})}, \end{aligned} \tag{4.16}$$

which is increasing in $m_1(\widehat{\boldsymbol{\theta}})/v_1(\widehat{\boldsymbol{\theta}})$.

Set $w_i \propto \delta_i$ to be the resampling weights induced by bootstrap counts. Sort indices so that the Mahalanobis distances $d_1 \leq \cdots \leq d_{N_1}$.

Theorem 4.1.1 (Classifier-free power improvement, first order)**.** *Let $\widehat{\mathbf{z}} = \mathbb{I}\{\, s_{\boldsymbol{\theta}}(\mathbf{z}) \geq 0 \,\}$ with $s_{\boldsymbol{\theta}}(\mathbf{z}) = \langle \psi(\mathbf{z}), \boldsymbol{\theta} \rangle$ a scoring rule that is monotone in the posterior margin $\gamma(\mathbf{z}) = \Pr\{y = 1 \mid \mathbf{Z} = \mathbf{z}\} - \frac{1}{2}$. Assume:*

i. $s_{\boldsymbol{\theta}}$ is Fisher-consistent for the 0–1 risk [45];

ii. $\widehat{\boldsymbol{\theta}}$ is obtained by ERM with a convex, differentiable loss and is a Hadamard-differentiable functional of the empirical measure such that small re-weightings admit a first-order von Mises expansion

$$\widehat{\boldsymbol{\theta}} = \boldsymbol{\theta}_{\mathrm{NO}} + \sum_{i=1}^{N} \chi_i \, (w_i - 1) + o(\|\mathbf{w} - 1\|) \tag{4.17}$$

with χ_i being the influence coefficients;

iii. On class $y = 1$, the Mahalanobis distance d_i is positively associated with the latent margin: larger d_i correspond (in distribution) to larger scores $s_{\boldsymbol{\theta}}(\mathbf{z}_i)$.

Then, taking expectation over the resampling mechanism,

$$\mathbb{E}_{\mathrm{MB}}\Big[\mathrm{Power}(\widehat{f})\Big] \;\geq\; \mathbb{E}_{\mathrm{BO}}\Big[\mathrm{Power}(\widehat{f})\Big] \;\geq\; \mathbb{E}_{\mathrm{NO}}\Big[\mathrm{Power}(\widehat{f})\Big]. \tag{4.18}$$

Define the first-order coefficients $\kappa_i := \langle \widetilde{\psi}_1, \chi_i \rangle$, where χ_i are the influence coefficients from the von Mises expansion in (4.17). By (iii), we have $\kappa_1 \leq \cdots \leq \kappa_{N_1}$.

Lemma 4.1.2 (Monotone shifts increase the linear margin functional)**.** *Fix $\kappa_1 \leq \cdots \leq \kappa_{N_1}$ and define $L(w) = \sum_{i \in \mathcal{I}_1} \kappa_i w_i$ for $w \in \mathbb{R}^{N_1}_{\geq 0}$ with $\sum_i w_i$ fixed. If $w' = w + \varepsilon(e_s - e_r)$ with $s > r$ and $0 < \varepsilon \leq w_r$, then*

$$L(w') - L(w) = \varepsilon(\kappa_s - \kappa_r) \geq 0. \tag{4.19}$$

Hence any finite sequence of such upward transfers weakly increases $L(w)$.

Corollary 4.1.2.1 (Ordering of expected weights)**.** *Let $\bar{\mathbf{w}}^{(\cdot)} := \mathbb{E}[\mathbf{W}]$ denote the expected weight vector under NO, BO, and MB. With indices ordered by d_i, the vector $\bar{\mathbf{w}}^{\mathrm{MB}}$ is obtained from $\bar{\mathbf{w}}^{\mathrm{BO}}$ by a finite sequence of upward transfers, and $\bar{\mathbf{w}}^{\mathrm{BO}}$ from $\bar{\mathbf{w}}^{\mathrm{NO}}$ likewise. Consequently,*

$$L\big(\bar{\mathbf{w}}^{\mathrm{MB}}\big) \;\geq\; L\big(\bar{\mathbf{w}}^{\mathrm{BO}}\big) \;\geq\; L\big(\bar{\mathbf{w}}^{\mathrm{NO}}\big). \tag{4.20}$$

Proof of Theorem 4.1.1. Weighted ERM with bootstrap counts is ERM with weights $w_i \propto \delta_i$. By Assumption (ii), $\widehat{\boldsymbol{\theta}}$ admits a von Mises expansion (4.17). Therefore the class-1 sample mean score satisfies, to first order,

$$m_1(\widehat{\boldsymbol{\theta}}) = m_1(\boldsymbol{\theta}_{\text{NO}}) + \sum_{i \in \mathcal{I}_1} \underbrace{\langle \widetilde{\psi}_1, \chi_i \rangle}_{\kappa_i} (w_i - 1) + o(\cdot), \tag{4.21}$$

so that $\mathbb{E}[m_1(\widehat{\boldsymbol{\theta}})] = m_1(\boldsymbol{\theta}_{\text{NO}}) + L(\bar{\mathbf{w}}) - L(\mathbf{1}) + o(1)$. By the coefficient ordering and Corollary 4.1.2.1, $L(\bar{\mathbf{w}})$ increases from NO $\to$ BO $\to$ MB; hence

$$\mathbb{E}_{\text{MB}}[m_1(\widehat{\boldsymbol{\theta}})] \;\geq\; \mathbb{E}_{\text{BO}}[m_1(\widehat{\boldsymbol{\theta}})] \;\geq\; \mathbb{E}_{\text{NO}}[m_1(\widehat{\boldsymbol{\theta}})]. \tag{4.22}$$

Finally, pass from mean score to power using any monotone link: if $s_{\widehat{\boldsymbol{\theta}}}(\mathbf{Z}) \mid y = 1$ is location–scale with fixed shape, then Power $= \Phi(m_1/v_1)$ with Φ strictly increasing;otherwise the Cantelli's inequality bound (4.16) is increasing in m_1/v_1. Monotonicity yields the stated power ordering. □

Theorem 4.1.1 shows that reallocating minority sampling mass from "easy" points with small d_i to "hard/informative" points with large d_i, larger influence strictly increases the class-1 mean score. Via any monotone margin-to-power link, this raises Power at the same operating threshold and thus for the same nominal Type-I error α, compared with uniform oversampling (BO) or no oversampling (NO).

In the sequel, we detail how Theorem 4.1.1 applies to

- quadratic discriminant analysis,
- classification trees,
- naïve Bayes,
- logistic regression,
- fully connected multilayer perceptrons, and
- SVMs with polynomial or Gaussian kernels.

Throughout, we use $y \in \{0, 1\}$, except in hinge-loss models where we map $\tilde{y} = 2y - 1$ when needed.

Quadratic Discriminant Analysis (QDA)

Assume $\mathbf{Z} \mid (y = k) \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ for $k \in \{0, 1\}$ with priors π_k. The posterior log-likelihood ratio (LLR) is

$$\begin{aligned} g(\mathbf{z}; \theta) &:= \log \frac{p(\mathbf{z} \mid y = 1)}{p(\mathbf{z} \mid y = 0)} + \log \frac{\pi_1}{\pi_0} \\ &= -\tfrac{1}{2}(\mathbf{z} - \boldsymbol{\mu}_1)^\intercal \boldsymbol{\Sigma}_1^{-1}(\mathbf{z} - \boldsymbol{\mu}_1) + \tfrac{1}{2}(\mathbf{z} - \boldsymbol{\mu}_0)^\intercal \boldsymbol{\Sigma}_0^{-1}(\mathbf{z} - \boldsymbol{\mu}_0) - \tfrac{1}{2} \log \frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|} + \log \frac{\pi_1}{\pi_0}. \end{aligned} \tag{4.23}$$

The QDA classifier is $\widehat{f}(\mathbf{z}) = \mathbb{I}\{g(\mathbf{z}; \widehat{\theta}) \geq 0\}$ with $\widehat{\theta} = (\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}_k, \widehat{\pi}_k)$ estimated by ERM on the (re)sampled data.

A standard separation index is the Bhattacharyya distance [46]

$$B(\boldsymbol{\theta}) = \tfrac{1}{8}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^\intercal \left(\tfrac{\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1}{2}\right)^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) + \tfrac{1}{2} \log \frac{\left|(\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1)/2\right|}{\sqrt{|\boldsymbol{\Sigma}_0||\boldsymbol{\Sigma}_1|}}, \tag{4.24}$$

where larger $B(\boldsymbol{\theta})$ implies smaller Bayes error Chernoff/Bhattacharyya); if $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$, the first term reduces to $\frac{1}{8}\Delta$ with $\Delta = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^\intercal \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$ [47].

Theorem 4.1.3 (Power ordering for Gaussian QDA via Theorem 4.1.1)**.** *Let $s_{\boldsymbol{\theta}} s(\mathbf{z}) \equiv g(\mathbf{z}; \boldsymbol{\theta})$ from* (4.23) *and fix the decision threshold at 0. Under Assumptions (i)–(iii) of Theorem 4.1.1, the expected power satisfies*

$$\mathbb{E}_{\text{MB}}\left[\text{Power}(\widehat{f})\right] \;\geq\; \mathbb{E}_{\text{BO}}\left[\text{Power}(\widehat{f})\right] \;\geq\; \mathbb{E}_{\text{NO}}\left[\text{Power}(\widehat{f})\right]. \tag{4.25}$$

Proof. Direct from Theorem 4.1.1 with g as the score: (i) the likelihood–ratio test is Fisher consistent and monotone in the posterior margin; (ii) ERM grants a first–order expansion under re-weighting; (iii) MB concentrates more weight on higher–margin minority points than BO/NO. Hence the class-1 sample mean score increases in expectation from NO $\to$ BO $\to$ MB, and any monotone link, such as location–scale or Cantelli's inequality, yields the stated power ordering. □

Corollary 4.1.3.1 (Type I error and population precision/F_1 under mild anti-drift)**.** *Denote the population precision by* $\text{Precision}^{\text{pop}}$ *and the population F_1 by F_1^{pop}. If the class-0 mean score $m_0(\widehat{\boldsymbol{\theta}}) = \mathbb{E}\left[g(\mathbf{Z}; \widehat{\boldsymbol{\theta}}) \mid y = 0\right]$ does not increase from NO $\to$ BO $\to$ MB (no rise in Type I error), then*

$$\mathbb{E}_{\text{MB}}[\alpha] \;\leq\; \mathbb{E}_{\text{BO}}[\alpha] \;\leq\; \mathbb{E}_{\text{NO}}[\alpha]. \tag{4.26}$$

For a fixed prior π_1, the population precision and F_1 satisfy

$$\text{Precision}^{\text{pop}} = \frac{\pi_1 \text{Power}}{\pi_1 \text{Power} + (1-\pi_1)\alpha}, \tag{4.27}$$

$$F_1^{\text{pop}} = \frac{2\text{Precision}^{\text{pop}} \cdot \text{Power}}{\text{Precision}^{\text{pop}} + \text{Power}}. \tag{4.28}$$

Since $\text{Precision}^{\text{pop}}$ *and* F_1^{pop} *are increasing in* Power *and decreasing in* α*, they both increase in expectation from NO* $\to$ *BO* $\to$ *MB.*

With $\delta = (\delta_i)_{i \in \mathcal{I}_1}$ the minority bootstrap counts (NO: $\delta_i \equiv 1$; BO/MB: $\sum_i \delta_i = N_1^*$), define centered counts $\tilde{\delta}_i = \delta_i - 1$. For BO/MB, $\mathbb{E}[\delta_i] = 1$, hence

$$\mathbb{E}[\widehat{\boldsymbol{\mu}}_1] = \frac{1}{N_1} \sum_{i \in \mathcal{I}_1} \mathbf{z}_i =: \boldsymbol{\mu}_1. \tag{4.29}$$

Moreover,

$$\widehat{\boldsymbol{\mu}} = \frac{1}{N_1^*} \sum_{i \in \mathcal{I}_1} \delta_i \mathbf{z}_i = \bar{\boldsymbol{\mu}} + \frac{N_1}{N_1^*} \mathbf{r}, \tag{4.30}$$

$$\bar{\boldsymbol{\mu}} := \frac{1}{N_1} \sum_{i \in \mathcal{I}_1} \mathbf{z}_i,$$

$$\tilde{\delta}_i := \delta_i - 1,$$

$$\mathbf{r} := \frac{1}{N_1} \sum_{i \in \mathcal{I}_1} \tilde{\delta}_i \mathbf{z}_i. \tag{QDA-mean}$$

Lemma 4.1.4 (Convexity leading to BO/MB $\geq$ NO for QDA separation)**.** *Condition on* $\widehat{\boldsymbol{\Sigma}}_0, \widehat{\boldsymbol{\Sigma}}_1$ *and set* $\mathbf{A} = \frac{1}{2}(\widehat{\boldsymbol{\Sigma}}_0 + \widehat{\boldsymbol{\Sigma}}_1) \succ \mathbf{0}$*,* $\boldsymbol{\eta} = \widehat{\boldsymbol{\mu}}_0 - \bar{\boldsymbol{\mu}}$*, and write* $\widehat{\boldsymbol{\mu}}_1 = \bar{\boldsymbol{\mu}} + (N_1/N_1^*)\mathbf{r}$ *as in* (QDA-mean)*. Then*

$$\Delta(\widehat{\boldsymbol{\theta}}) = \left\| \boldsymbol{\eta} + \kappa \mathbf{r} \right\|_{\mathbf{A}^{-1}}^2, \qquad \kappa := -\tfrac{N_1}{N_1^*} < 0, \tag{4.31}$$

is convex in $\mathbf{r}$*. Hence, since* $\mathbb{E}_{\text{BO/MB}}[\mathbf{r}] = \mathbf{0}$ *and under NO, we have* $\mathbf{r} \equiv \mathbf{0}$*,*

$$\mathbb{E}_{\text{MB}}[\Delta] \geq \|\boldsymbol{\eta}\|_{\mathbf{A}^{-1}}^2 = \mathbb{E}_{\text{NO}}[\Delta], \qquad \mathbb{E}_{\text{BO}}[\Delta] \geq \mathbb{E}_{\text{NO}}[\Delta]. \tag{4.32}$$

Proof. $\Delta(\widehat{\boldsymbol{\theta}}) = \|\boldsymbol{\eta} + \kappa \mathbf{r}\|_{\mathbf{A}^{-1}}^2$ has Hessian $2|\kappa|^2 \mathbf{A}^{-1} \succeq 0$, so by Jensen inequality [48, Theorem 3.3.6] with $\mathbb{E}[\mathbf{r}] = \mathbf{0}$ (BO/MB), we get the inequalities; NO uses $\mathbf{r} \equiv \mathbf{0}$. □

With equal costs/priors, the monotonicity of the LLR implies

$$\text{Power} = \Phi\big(\tfrac{1}{2}\sqrt{\Delta}\big), \qquad \alpha = 1 - \Phi\big(\tfrac{1}{2}\sqrt{\Delta}\big), \tag{4.33}$$

for a strictly increasing Φ. Hence the lemma implies $\mathbb{E}[\text{Power}]$ increases and $\mathbb{E}[\alpha]$ decreases from NO to BO/MB. To additionally order MB $\geq$ BO in power, invoke Theorem 4.1.1: MB places more expected weight on higher-margin minority points (via d_i alignment), which increases the class-1 sample mean score to first order by the monotone-transfer argument, yielding

$$\mathbb{E}_{\text{MB}}[\text{Power}] \;\geq\; \mathbb{E}_{\text{BO}}[\text{Power}], \qquad \mathbb{E}_{\text{MB}}[\alpha] \;\leq\; \mathbb{E}_{\text{BO}}[\alpha], \tag{4.34}$$

and, with fixed priors, the same ordering for Precision and F_1.

Classification Trees (CART)

A CART classifier recursively partitions the input space and assigns to every terminal node (leaf) τ an empirical class probability

$$\widehat{\pi}_\tau = \frac{1}{|\mathcal{D}_\tau|} \sum_{(\mathbf{z}_i, y_i) \in \mathcal{D}_\tau} \mathbb{I}\{y_i = 1\}, \qquad \mathcal{D}_\tau = \{(\mathbf{z}_i, y_i) : \mathbf{z}_i \in \tau\}. \tag{4.35}$$

The tree predicts class 1 if $\widehat{\pi}_\tau > \frac{1}{2}$. The leafwise log–odds score

$$s(\mathbf{z}) = \log \frac{\widehat{\pi}_{\tau(\mathbf{z})}}{1 - \widehat{\pi}_{\tau(\mathbf{z})}} = \log \frac{n_{\tau(\mathbf{z}),1}}{n_{\tau(\mathbf{z}),0}}, \tag{4.36}$$

is compared to 0 [43], where $n_{\tau 1}$ and $n_{\tau 0}$ are the class counts in leaf τ.

For a fixed partition $\{\tau\}$ (grown independently of resampling), only the minority counts change:

$$n^*_{\tau 1} = \sum_{i \in \mathcal{I}_1 \cap \mathcal{D}_\tau} \delta_i, \qquad\qquad n_{\tau 0} = |\mathcal{I}_0 \cap \mathcal{D}_\tau| \text{ (fixed)}, \tag{4.37}$$

$$\widehat{\pi}_\tau = \frac{n^*_{\tau 1}}{n^*_{\tau 1} + n_{\tau 0}}. \tag{4.38}$$

Thus, $s(\mathbf{z})$ in (4.36) depends on resampling only through $n^*_{\tau 1}$.

Theorem 4.1.5 (Power improvement for fixed-partition trees)**.** *Fix a fully grown CART whose partition $\{\tau\}$ is independent of the resampling scheme. Assume:*

i. $s(\mathbf{z})$ *in* (4.36) *is monotone in the posterior margin and induces* $\widehat{f}(\mathbf{z}) = \mathbb{I}\{s(\mathbf{z}) \geq 0\}$.

ii. If $\bar{d}_{\tau,1}$ is the mean Mahalanobis distance of minority samples in leaf τ, then $\bar{d}_{\tau,1}$ is nondecreasing in $P_1(\tau) = \Pr\{\mathbf{Z} \in \tau \mid y = 1\}$.

iii. Within leaf τ, a small increase $\Delta n^*_{\tau 1} > 0$ changes $\boldsymbol{\theta}_\tau = \log(n^*_{\tau 1}/n_{\tau 0})$ by $\Delta n^*_{\tau 1}/n^*_{\tau 1}$.

Then, taking expectation over the bootstrap mechanism,

$$\mathbb{E}_{\text{MB}}\left[\text{Power}(\widehat{f})\right] \;\geq\; \mathbb{E}_{\text{BO}}\left[\text{Power}(\widehat{f})\right] \;\geq\; \mathbb{E}_{\text{NO}}\left[\text{Power}(\widehat{f})\right]. \tag{4.39}$$

Proof. Represent the score as $s_{\boldsymbol{\theta}}(\mathbf{z}) = \sum_\tau \boldsymbol{\theta}_\tau \mathbb{I}\{\mathbf{z} \in \tau\}$, with $\boldsymbol{\theta}_\tau = \log(n^*_{\tau 1}/n_{\tau 0})$. The class-1 mean score is $m_1(\boldsymbol{\theta}) = \sum_\tau \boldsymbol{\theta}_\tau P_1(\tau)$. A unit increase of normalized weight for a minority point i in leaf τ raises $\boldsymbol{\theta}_\tau$ by $1/n^*_{\tau 1}$; hence the first-order influence on m_1 equals $\kappa_i \propto P_1(\tau)/n^*_{\tau 1}$, constant for all i in the same leaf. By (ii), ordering minority points by d_i aligns (in expectation) with the ordering of the leaf-level coefficients κ_i. Therefore the linear functional $L(w) = \sum_i \kappa_i w_i$ (first-order change of m_1 under weights w) increases under finite sequences of upward transfers toward larger d_i. In expectation, MB is obtained from BO (and BO from NO) by such transfers, so $\mathbb{E}[m_1]$ increases from NO $\to$ BO $\to$ MB. □

If the partition is learned on the resampled data, the leaf structure may change across schemes and additional conditions are needed (e.g., stability of splits or margin-aligned split selection). Theorem 4.1.5 isolates the effect of re-weighting within leaves, which is what MB directly targets.

Let the partition be $\{\tau_1, \ldots, \tau_L\}$, with majority counts $n_{\tau_\ell 0}$ fixed across schemes. For minority resampling with counts δ_i and selection probabilities p_i on $i \in \mathcal{I}_1$, define the leaf-level probabilities

$$p_\ell^{(\text{scheme})} = \sum_{i \in \mathcal{I}_1 \cap \mathcal{D}_{\tau_\ell}} p_i, \tag{4.40}$$

so $(n^*_{\tau_\ell 1})_{\ell=1}^L \sim \text{Multinomial}\big(N_1^*, (p_\ell^{(\text{scheme})})_{\ell=1}^L\big)$ and, marginally,

$$n^*_{\tau_\ell 1} \sim \text{Binomial}\big(N_1^*, p_\ell^{(\text{scheme})}\big). \tag{4.41}$$

Let $h_\ell(p) := \Pr\{\text{Binomial}(N_1^*, p) \geq n_{0,\ell}\}$ be the probability that $\widehat{\pi}_{\tau_\ell} > \frac{1}{2}$ in leaf τ_ℓ with threshold $n_{0,\ell} = \lceil \frac{1}{2}(n_{\tau_\ell 0} + n^*_{\tau_\ell 1}) \rceil$. Each h_ℓ denotes the right derivative with respect to p. It exists and is nonnegative because h_ℓ is a binomial tail cumulative distribution function (CDF), hence nondecreasing. Consequently,

$$\mathbb{E}[\text{Power}(\widehat{f})] = \sum_{\ell=1}^{L} P_1(\tau_\ell) h_\ell\left(p_\ell^{(\text{scheme})}\right). \tag{4.42}$$

Sort leaves so that $\kappa_\ell := P_1(\tau_\ell)$ is nondecreasing. Under BO, $p_\ell^{(\mathrm{BO})}$ is the minority share in leaf τ_ℓ; under MB, mass shifts in expectation toward leaves whose minority points have larger d_i, which, by the alignment assumption, are those with larger $P_1(\tau_\ell)$. Thus $p^{(\mathrm{MB})}$ is obtained from $p^{(\mathrm{BO})}$ by a finite sequence of upward transfers (T-transforms) in the sorted order.

Linearizing (4.42) gives

$$L(p) = \sum_{\ell=1}^{L} \kappa_\ell h'_\ell\,(p_\ell)\, p_\ell, \quad h'_\ell \geq 0,$$

so an upward transfer $\varepsilon(e_s - e_r)$ with $s > r$ increases $L(p)$. Iterating transfers yields, to first order,

$$\mathbb{E}_{\mathrm{MB}}[\mathrm{Power}] \;\geq\; \mathbb{E}_{\mathrm{BO}}[\mathrm{Power}] \;\geq\; \mathbb{E}_{\mathrm{NO}}[\mathrm{Power}],$$

consistent with Theorem 4.1.5.

If, in addition, the class-0 mean score does not increase from NO $\to$ BO $\to$ MB or no drift in false rejections, the same reasoning gives the reversed ordering for Type I error α, and thus the corresponding improvements for precision and F_1 at fixed prior.

Gaussian Naïve Bayes

Assume conditional independence of features given the class:

$$p(\mathbf{z} \mid y = k) = \prod_{j=1}^{m} \mathcal{N}\big(z_j \mid \mu_{kj}, \sigma_{kj}^2\big), \qquad k \in \{0, 1\}. \tag{4.43}$$

Let $\widehat{\mu}_{kj}, \widehat{\sigma}_{kj}^2$ be the class–conditional maximum likelihood estimates (MLEs) on a (re)sampled training set. The posterior log–likelihood ratio is

$$\begin{aligned} s_{\widehat{\theta}}(\mathbf{z}) &= \log \frac{p(\mathbf{z} \mid y = 1)}{p(\mathbf{z} \mid y = 0)} + \log \frac{\pi_1}{\pi_0} \\ &= \sum_{j=1}^{m} \left\{ -\tfrac{(z_j - \widehat{\mu}_{1j})^2}{2\widehat{\sigma}_{1j}^2} + \tfrac{(z_j - \widehat{\mu}_{0j})^2}{2\widehat{\sigma}_{0j}^2} \right\} - \tfrac{1}{2} \sum_{j=1}^{m} \log \frac{\widehat{\sigma}_{1j}^2}{\widehat{\sigma}_{0j}^2} + \log \frac{\pi_1}{\pi_0}, \end{aligned} \tag{4.44}$$

with decision rule $\widehat{f}(\mathbf{z}) = \mathbb{I}\{\, s_{\widehat{\theta}}(\mathbf{z}) \geq 0 \,\}$.

If $\sigma_{1j}^2 = \sigma_{0j}^2 = \sigma_j^2$ (after standardising features) and $\pi_1 = \pi_0$, then

$$\Delta_{\text{NB}}(\widehat{\theta}) = \sum_{j=1}^{m} \frac{\left(\widehat{\mu}_{1j} - \widehat{\mu}_{0j}\right)^2}{\sigma_j^2}, \tag{4.45}$$

$$\text{Power} = \Phi\left(\tfrac{1}{2}\sqrt{\Delta_{\text{NB}}}\right), \tag{4.46}$$

$$\alpha = 1 - \Phi\left(\tfrac{1}{2}\sqrt{\Delta_{\text{NB}}}\right), \tag{4.47}$$

with Φ strictly increasing (cf. [49, Sec. 3.5]).

For any counts,

$$\widehat{\mu}_{1j}(\delta) = \frac{1}{N_1^*} \sum_{i \in \mathcal{I}_1} \delta_i \, z_{ij}, \tag{4.48}$$

$$\widehat{\sigma}_{1j}^2(\delta) = \frac{1}{N_1^*} \sum_{i \in \mathcal{I}_1} \delta_i \big(z_{ij} - \widehat{\mu}_{1j}(\delta)\big)^2. \tag{4.49}$$

Theorem 4.1.6 (Ordering of Gaussian Naïve Bayes (GNB) power and Type-I error)**.** *Assume (i) homoscedastic feature variances $\sigma_{1j}^2 = \sigma_{0j}^2 = \sigma_j^2$ and equal priors (so that* (4.47) *holds); (ii) the generic assumptions of Theorem 4.1.1 (regular ERM/von Mises expansion; alignment of d_i with the class-1 margin). Then, taking expectation over the resampling,*

$$\mathbb{E}_{\text{s}}\left[\text{Power}\right] \;\geq\; \mathbb{E}_{\text{BO}}\left[\text{Power}\right] \;\geq\; \mathbb{E}_{\text{NO}}\left[\text{Power}\right], \tag{4.50}$$

and consequently the expected Type-I error satisfies

$$\mathbb{E}_{\text{MB}}[\alpha] \;\leq\; \mathbb{E}_{\text{BO}}[\alpha] \;\leq\; \mathbb{E}_{\text{NO}}[\alpha]. \tag{4.51}$$

Proof. With homoscedastic variances, (4.45) is a quadratic form in the linear statistic $\widehat{\mu}_1(\delta)$, hence convex in δ. Under BO, $\mathbb{E}[\widehat{\mu}_1(\delta)] = \mu_1$, and under NO, $\widehat{\mu}_1(\delta) \equiv \mu_1$. By Jensen [48],

$$\mathbb{E}_{\text{BO}}\left[\Delta_{\text{NB}}\right] \;\geq\; \mathbb{E}_{\text{NO}}\left[\Delta_{\text{NB}}\right],$$

and the monotone link (4.47) gives higher power and lower α from NO to BO. For MB vs. BO, apply Theorem 4.1.1 with the score (4.44): distance weighting concentrates expected minority mass on larger d_i, increasing the class-1 mean margin to first order more than BO; thus $\mathbb{E}_{\text{MB}}[\text{Power}] \geq \mathbb{E}_{\text{BO}}[\text{Power}]$, and the α ordering follows from (4.47). □

For GNB with equal variances, the separation (Δ_{NB}) is a convex function of the resampled minority mean. Hence even uniform oversampling (BO) improves the expected separation, and thus power, over using the raw sample (NO) by a Jensen's step. MS-Boot (MB) then goes further.

Logistic Regression

Logistic regression models the minority posterior via

$$\Pr\{y = 1 \mid \mathbf{z}; \boldsymbol{\theta}\} = \sigma\big(\boldsymbol{\theta}^\intercal \mathbf{z}\big), \qquad \sigma(t) = (1 + e^{-t})^{-1}, \tag{4.52}$$

and classifies by $\widehat{f}(\mathbf{z}) = \mathbb{I}\{\, \boldsymbol{\theta}^\intercal \mathbf{z} \geq 0 \,\}$, with $\widehat{\boldsymbol{\theta}}$ the von Mises expansion/maximum likelihood estimation (MLE) on the resampled data. Write $w_i \propto \delta_i$ for normalised minority weights and $\mathbf{m} := \mathbb{E}_{P_1}[\mathbf{Z}]$ for the class-1 mean.

Theorem 4.1.7 (Ordering of LR power)**.** *Assume:*

i. $\widehat{\boldsymbol{\theta}}$ *is Hadamard-differentiable in the empirical measure and thus admits a first-order von Mises expansion expansion under re-weighting.*

ii. On $y = 1$*, the Mahalanobis distance* d_i *is positively associated with the margin* $s(\mathbf{z}_i) = \boldsymbol{\theta}^\intercal \mathbf{z}_i$*.*

iii. The decision threshold is 0*.*

Then, in expectation over the resampling mechanism,

$$\mathbb{E}_{\text{MB}}[\text{Power}] \;\geq\; \mathbb{E}_{\text{BO}}[\text{Power}] \;\geq\; \mathbb{E}_{\text{NO}}[\text{Power}].$$

Moreover, if the class-0 mean score does not increase from NO $\to$ *BO* $\to$ *MB (no Type-I drift), then*

$$\mathbb{E}_{\text{MB}}[\alpha] \;\leq\; \mathbb{E}_{\text{BO}}[\alpha] \;\leq\; \mathbb{E}_{\text{NO}}[\alpha].$$

Proof. By (i), a small change of minority weights w_i yields

$$\widehat{\boldsymbol{\theta}} = \boldsymbol{\theta}_{\text{NO}} + \sum_{i \in \mathcal{I}_1} \mathbf{A}^{-1}\, \psi(\mathbf{z}_i, 1; \boldsymbol{\theta}_{\text{NO}})\, (w_i - 1) + o(\|w - 1\|),$$

where ψ is the logistic score and $\mathbf{A}$ the expected Hessian at $\boldsymbol{\theta}_{\text{NO}}$. Thus the class-1 mean margin is given by

$$\begin{aligned} \mu_1 :=& \mathbb{E}_{P_1}\left[s_{\widehat{\boldsymbol{\theta}}}(\mathbf{Z})\right] = \mathbf{m}^\intercal \widehat{\boldsymbol{\theta}} \\ =& \mu_1^{\text{NO}} + \sum_{i \in \mathcal{I}_1} c_i\, (w_i - 1) + o(\cdot), \end{aligned} \tag{4.53}$$

$$c_i := \mathbf{m}^\intercal \mathbf{A}^{-1}\, \psi\left(\mathbf{z}_i, 1; \boldsymbol{\theta}_{\text{NO}}\right). \tag{4.54}$$

By (ii), sorting minority points by d_i sorts (in expectation) the coefficients so that $c_1 \leq \cdots \leq c_{N_1}$. The first-order change is the linear form $L(w) = \sum_i c_i w_i$. Moving expected

weight from smaller to larger d_i indices (NO $\rightarrow$ BO $\rightarrow$ MB) is a finite sequence of upward transfers, which increases L by the monotone-transfer lemma; hence $\mathbb{E}[\mu_1]$ increases in that order. Finally, the test power Power $= \Pr\{s_{\widehat{\boldsymbol{\theta}}}(\mathbf{Z}) \geq 0 \mid y = 1\}$ is monotone in μ_1 (via a location–scale link or the Cantelli bound), giving the stated power ordering. The Type-I error ordering follows under the no–drift condition by the same monotonicity applied to the class-0 score distribution. □

Two–Hidden–Layer MLP (8–4 Architecture)

Consider a two–hidden–layer network with $h{=}8$ Rectified Linear Unit (ReLU), then $r{=}4$ logistic units, and a linear score. For $\mathbf{x} \in \mathbb{R}^n$,

$$\mathbf{h}(\mathbf{x}) = \mathrm{ReLU}\left(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1\right) \in \mathbb{R}^8, \tag{4.55}$$

$$\mathbf{u}(\mathbf{x}) = \varphi\left(\mathbf{W}_2\mathbf{h}(\mathbf{x}) + \mathbf{b}_2\right) \in \mathbb{R}^4, \tag{4.56}$$

$$s_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{w}_3^{\top}\mathbf{u}(\mathbf{x}) + b_3, \qquad \widehat{f}(\mathbf{x}) = \mathbb{I}\{\, s_{\boldsymbol{\theta}}(\mathbf{x}) \geq 0 \,\}, \tag{4.57}$$

with $\varphi(t) = (1 + e^{-t})^{-1}$ and parameters $\boldsymbol{\theta} = \{\mathbf{W}_1, \mathbf{b}_1, \mathbf{W}_2, \mathbf{b}_2, \mathbf{w}_3, b_3\}$. Parameters are estimated by ERM with logistic loss on a (re)sampled set:

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \; \frac{1}{N}\sum_{i=1}^{N} \ell\left(y_i, s_{\boldsymbol{\theta}}(\mathbf{x}_i)\right), \tag{4.58}$$

$$\ell(y, t) = \log\bigl(1 + e^{t}\bigr) - y\,t, \qquad y \in \{0, 1\}. \tag{4.59}$$

Let $R(\boldsymbol{\theta}) = \mathbb{E}[\ell(y, s_{\boldsymbol{\theta}}(\mathbf{Z}))]$ and suppose R is twice differentiable with positive–definite Hessian at $\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} R(\boldsymbol{\theta})$. For a small minority–only reweighting $w_i \propto \delta_i$, the influence–function approximation (e.g., [50]) gives

$$\widehat{\boldsymbol{\theta}} = \boldsymbol{\theta}^\star - H_{\boldsymbol{\theta}^\star}^{-1}\left[\frac{1}{N}\sum_{i \in \mathcal{I}_1}(w_i - 1)\,\nabla_{\boldsymbol{\theta}}\ell\left(1, s_{\boldsymbol{\theta}^\star}(\mathbf{x}_i)\right)\right] + o(\|w - 1\|),$$

with $H_{\boldsymbol{\theta}^\star} = \nabla^2_{\boldsymbol{\theta}} R(\boldsymbol{\theta}^\star) \succ 0$ and $\nabla_{\boldsymbol{\theta}}\ell(y, s) = (\varphi(s) - y)\,\nabla_{\boldsymbol{\theta}} s$.

Define the class-1 mean margin $m_1(\boldsymbol{\theta}) = \mathbb{E}_{P_1}[s_{\boldsymbol{\theta}}(\mathbf{X})]$. A first-order expansion around $\boldsymbol{\theta}^\star$ yields

$$m_1(\widehat{\boldsymbol{\theta}}) = m_1(\boldsymbol{\theta}^\star) + \left\langle \mathbf{g}_1,\, \widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star \right\rangle + o\left(\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\|\right), \tag{4.60}$$

$$\mathbf{g}_1 := \mathbb{E}_{P_1}\left[\nabla_{\boldsymbol{\theta}} s_{\boldsymbol{\theta}^\star}(\mathbf{X})\right]. \tag{4.61}$$

Substituting (4.60),

$$m_1(\widehat{\boldsymbol{\theta}}) = m_1(\boldsymbol{\theta}^\star) - \sum_{i \in \mathcal{I}_1} \beta_i \, (w_i - 1) + o(\|w - 1\|), \tag{4.62}$$

$$\beta_i := \Big\langle \mathbf{g}_1, \, H_{\boldsymbol{\theta}^\star}^{-1} \nabla_{\boldsymbol{\theta}} \ell \, (1, s_{\boldsymbol{\theta}^\star}(\mathbf{x}_i)) \Big\rangle. \tag{4.63}$$

Since for $y{=}1$ one has $\nabla_{\boldsymbol{\theta}} \ell(1, s_{\boldsymbol{\theta}^\star}) = -(1 - \varphi(s_{\boldsymbol{\theta}^\star})) \, \nabla_{\boldsymbol{\theta}} s_{\boldsymbol{\theta}^\star}$ and $H_{\boldsymbol{\theta}^\star}^{-1} \succ 0$, the coefficients $\beta_i \geq 0$ and increase with the deficit $-s_{\boldsymbol{\theta}^\star}(\mathbf{x}_i)$.

Theorem 4.1.8 (Ordering of $\text{MLP}_{8\to4}$ power and Type-I error)**.** *Assume:*

i. R *is twice differentiable and* $H_{\boldsymbol{\theta}^\star} \succ 0$*;*

ii. on $y{=}1$*, the Mahalanobis distance* d_i *is positively associated with the deficit* $-s_{\boldsymbol{\theta}^\star}(\mathbf{x}_i)$*;*

iii. the decision threshold is 0*.*

Then, in the first-order (influence-function) approximation and in expectation over NO, BO, MB (minority-only):

$$\mathbb{E}_{\text{MB}}[\text{Power}] \;\geq\; \mathbb{E}_{\text{BO}}[\text{Power}] \;\geq\; \mathbb{E}_{\text{NO}}[\text{Power}],$$

and

$$\mathbb{E}_{\text{MB}}[\alpha] \;\leq\; \mathbb{E}_{\text{BO}}[\alpha] \;\leq\; \mathbb{E}_{\text{NO}}[\alpha].$$

Proof. Ordering minority points by d_i aligns (by (ii)) with increasing deficit and hence with larger β_i in (4.63). The first-order change of m_1 is $L(w) = -\sum_i \beta_i (w_i - 1)$. Upward transfers (NO $\to$ BO $\to$ MB) decrease $L(w)$ and increase m_1. A monotone link from mean margin to the rejection probability under $y{=}1$ gives the stated power ordering; under the same link, the Type-I error $\alpha = \Pr\{s_{\widehat{\boldsymbol{\theta}}}(\mathbf{Z}) \geq 0 \mid y = 0\}$ is non-increasing in the sequence (or assume no drift in the class-0 mean score), yielding the α ordering above. □

The argument does not require exact linearity of $\widehat{\boldsymbol{\theta}}$ in the counts; the influence-function linearization is the standard first-order tool for deep ERM [50]. Mini-batch Stochastic Gradient Descent (SGD) or Adam produces iterates with mean close to the full-batch optimum; the added variance is $O(N^{-1/2})$ and does not alter the ordering in expectation.

Support Vector Machines with a Polynomial and Gaussian (RBF) Kernels

Let $k(\mathbf{z}, \mathbf{z}')$ be a positive-definite kernel positive–definite kernel with feature map ϕ and Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$. Given a training set $\{(\mathbf{z}_i, \tilde{y}_i)\}_{i=1}^N$ with

$\tilde{y}_i \in \{\pm 1\}$ from $y \in \{0,1\}$ via $\tilde{y} = 2y - 1$, the minority-only re-weighting uses sample weights ω_i (majority: $\omega_i \equiv 1$; minority $i \in \mathcal{I}_1$: $\omega_i \propto \delta_i$, $\sum_{i\in\mathcal{I}_1} \delta_i = N_1^*$; NO: $\delta_i \equiv 1$; BO: uniform; MB: $\delta_i \propto d_i$). The weighted soft-margin SVM (hinge loss, penalty $C > 0$) solves

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \tfrac{1}{2}\|\mathbf{w}\|_{\mathcal{H}}^2 + C\sum_{i=1}^{N} \omega_i\, \xi_i \\ \text{s.t.} \quad & y_i\big(\langle \mathbf{w}, \phi(\mathbf{z}_i)\rangle + b\big) \geq 1 - \xi_i, \quad \xi_i \geq 0,\ i = 1,\dots,N. \end{aligned} \tag{4.64}$$

The decision function is

$$s_{\widehat{\mathbf{w}},\widehat{b}}(\mathbf{z}) = \sum_{i=1}^{N} \widehat{\alpha}_i\, y_i\, k(\mathbf{z}_i, \mathbf{z}) + \widehat{b}, \qquad \widehat{f}(\mathbf{z}) = \mathbb{I}\{\, s_{\widehat{\mathbf{w}},\widehat{b}}(\mathbf{z}) \geq 0\,\}, \tag{4.65}$$

with Karush-Kuhn-Tucker (KKT) multipliers $\widehat{\alpha}_i$.

Let $(\mathbf{w}^*, b^*)$ minimise the regularised hinge risk $\mathcal{R}_\lambda(f) = \mathbb{E}[\max\{0,\, 1 - Y f(\mathbf{Z})\}] + \frac{\lambda}{2}\|w\|_{\mathcal{H}}^2$ (with $\lambda = 1/(NC)$). Assume standard regularity so $(\widehat{\mathbf{w}}, \widehat{b})$ is a Hadamard-differentiable functional of the empirical measure (strong convexity via $\|w\|_{\mathcal{H}}^2$). Then, a standard influence–function expansion yields, for a small change in minority weights ω_i,

$$\begin{aligned} \widehat{\mathbf{w}} = & -\mathbf{H}^{-1}\left[\frac{C}{N}\sum_{i\in\mathcal{I}_1} (\omega_i - 1)\, \nabla_{\mathbf{w}}\, \ell_{\text{hinge}}\Big(y_i, \langle \mathbf{w}^*, \phi(\mathbf{z}_i)\rangle + b^*\Big)\right] \\ & + \mathbf{w}^* + o(\|\omega - 1\|), \end{aligned} \tag{4.66}$$

$$\begin{aligned} \widehat{b} = & -h_b^{-1}\left[\frac{C}{N}\sum_{i\in\mathcal{I}_1} (\omega_i - 1)\, \partial_b\, \ell_{\text{hinge}}\Big(y_i, \langle \mathbf{w}^*, \phi(\mathbf{z}_i)\rangle + b^*\Big)\right] \\ & + b^* + o(\|\omega - 1\|), \end{aligned} \tag{4.67}$$

where $\mathbf{H} \succ 0$ is the generalised Hessian at $(\mathbf{w}^*, b^*)$, and

$$\nabla_{\mathbf{w}}\, \ell_{\text{hinge}}(y,t) = \begin{cases} -\,y\,\phi(\mathbf{z}), & 1 - yt > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{4.68}$$

$$\partial_b\, \ell_{\text{hinge}}(y,t) = \begin{cases} -\,y, & 1 - yt > 0, \\ 0, & \text{otherwise}. \end{cases} \tag{4.69}$$

Thus only support vectors / margin violators contribute at first order.

Let

$$m_1 := \mathbb{E}_{P_1}[s_{\widehat{\mathbf{w}},\widehat{b}}(\mathbf{Z})] = \left\langle \boldsymbol{\mu}_{\boldsymbol{\phi},\mathbf{1}}, \widehat{\mathbf{w}} \right\rangle + \widehat{b}, \tag{4.70}$$

$$m_{\phi,1} := \mathbb{E}_{P_1}[\phi(\mathbf{Z})]. \tag{4.71}$$

From (4.67),

$$m_1 = m_1^{\mathrm{NO}} + \sum_{i \in \mathcal{I}_1} \beta_i \, (\omega_i - 1) + o(\|\omega - 1\|), \tag{4.72}$$

$$\beta_i := \frac{C}{N} \left\langle \boldsymbol{\mu}_{\boldsymbol{\phi},\mathbf{1}}, \mathbf{H}^{-1} \, y_i \, \phi(\mathbf{z}_i) \, \mathbf{1}\{1 - y_i t_i^* > 0\} \right\rangle + h_b^{-1} \, y_i \, \mathbf{1}\{1 - y_i t_i^* > 0\}, \tag{4.73}$$

with $t_i^* = \langle \mathbf{w}^*, \phi(\mathbf{z}_i) \rangle + b^*$. Hence, to first order, the change of m_1 is a linear functional of the minority weights, driven by support vectors/margin violators.

Theorem 4.1.9 (Ordering of SVM power and Type-I error in an RKHS)**.** *Assume:*

i. *Fixed kernel k and C; the primal* (4.64) *is locally strongly convex at the population optimum so that* (4.67) *holds.*

ii. *On class $y = 1$, the Mahalanobis distance d_i is positively associated with the hinge deficit $\xi_i^* = \max(0, 1 - t_i^*)$; i.e., larger d_i indicates points closer to/inside the margin.*

iii. *The decision threshold is* 0.

Then, taking expectation over the resampling mechanism (minority-only NO, BO, MB),

$$\mathbb{E}_{\mathrm{MB}}[\mathrm{Power}] \;\geq\; \mathbb{E}_{\mathrm{BO}}[\mathrm{Power}] \;\geq\; \mathbb{E}_{\mathrm{NO}}[\mathrm{Power}],$$

and

$$\mathbb{E}_{\mathrm{MB}}[\alpha] \;\leq\; \mathbb{E}_{\mathrm{BO}}[\alpha] \;\leq\; \mathbb{E}_{\mathrm{NO}}[\alpha].$$

Proof. Order minority points by d_i. By (ii), this coincides (in expectation) with ordering by larger hinge deficits ξ_i^*, hence with larger coefficients β_i in (4.73). The first–order change of m_1 is $L(\omega) = \sum_i \beta_i \omega_i$. An upward transfer of expected weight from a smaller–d index to a larger–d index increases $L(\omega)$; iterating transfers gives $L(\bar{\omega}^{\mathrm{MB}}) \geq L(\bar{\omega}^{\mathrm{BO}}) \geq L(\bar{\omega}^{\mathrm{NO}})$, hence $\mathbb{E}[m_1]$ increases in the same order. Any monotone link from m_1 to the rejection probability under $y{=}1$ gives the power ordering; under the same link (or a no–drift assumption for the class-0 mean score), the Type-I error $\alpha = \Pr\{s_{\widehat{\mathbf{w}},\widehat{b}}(\mathbf{Z}) \geq 0 \mid y = 0\}$ is non-increasing, yielding the α ordering. □

The argument applies to any fixed RKHS; the Polynomial and RBF kernels are used here for concreteness. For the polynomial kernel, $k(\mathbf{z}, \mathbf{z}') = (\gamma\, \mathbf{z}^\top \mathbf{z}' + c_0)^p$ with $\gamma > 0$, $c_0 \geq 0$, $p \in \mathbb{N}$; for the Gaussian (RBF) kernel, $k(\mathbf{z}, \mathbf{z}') = \exp\!\big(-\|\mathbf{z} - \mathbf{z}'\|_2^2/(2\sigma^2)\big)$ with $\sigma > 0$. Assumption (i) holds for fixed kernel parameters and C (strong convexity via $\|w\|_{\mathcal{H}}^2$); Assumption (ii) is the empirical alignment we verify in Subsection 4.1.5.

4.1.4 Case Study: Intracranial EEG

Let $x[n]$, $n = 1, \ldots, N$, denote a single-channel EEG segment of length N uniformly sampled at $f_s = 100$ Hz. We compute 12 standard time- and frequency-domain features per channel and concatenate them across all channels (in our case, 15 channels).

Feature Extraction

The following feature extraction procedures are applied to each of the 15 channels. For simplicity, the superscript that indicates different channels is omitted.

The time domain features derived are as follow.

1. The first measure is the energy,

$$\mathcal{E} = \sum_{n=1}^{N} x[n]^2, \tag{4.74}$$

   which emphasises high-amplitude events such as epileptic spikes [51].

2. We also calculate the Root Mean Square (RMS) that measures the signal's power,

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{n=1}^{N} x[n]^2}. \tag{4.75}$$

3. The line length measures the cumulative absolute change between consecutive samples. It quantifies the waveform "ruggedness" [52],

$$\text{LL} = \sum_{n=1}^{N-1} |x[n+1] - x[n]|\,. \tag{4.76}$$

4. Hjorth Activity is simply the variance of the signal to capture signal variability and dynamic changes [53]

$$\text{Activity} \;=\; \sigma_x^2 \;=\; \frac{1}{N-1}\sum_{n=1}^{N}\left(x[n]-\bar{x}\right)^2, \tag{4.77}$$

where $\bar{x}$ is the sample mean of $x[n]$.

5. Hjorth Mobility is given by

$$\text{Mobility} = \sqrt{\frac{\sigma_{\Delta x}^2}{\sigma_x^2}}, \tag{4.78}$$

where the first difference is

$$\Delta x[n] = x[n+1] - x[n], \tag{4.79}$$

and $\sigma_{\Delta x}^2$ denotes the variance of $\Delta x[n]$ [53].

6. Hjorth Complexity is given by

$$\text{Complexity} \;=\; \frac{\sqrt{\sigma_{\Delta^2 x}^2/\sigma_{\Delta x}^2}}{\sqrt{\sigma_{\Delta x}^2/\sigma_x^2}} \;=\; \frac{\sqrt{\sigma_{\Delta^2 x}^2/\sigma_{\Delta x}^2}}{\text{Mobility}}. \tag{4.80}$$

where $\sigma_{\Delta^2 x}^2$ is the variance of the second difference of the signal [53],

$$\Delta^2 x[n] = x[n+2] - 2x[n+1] + x[n]. \tag{4.81}$$

Spectral analysis provides insight into the distribution of power across canonical EEG bands, which are known to change during seizures. Let $\hat{X}(f)$ be the estimate of the power spectral density (PSD) obtained, for instance, via Welch's method:

$$\hat{P}_{xx}(f) \;=\; \frac{1}{K}\sum_{k=1}^{K}\left|\text{DFT}\{x_k[n]\}\right|^2, \tag{4.82}$$

where $x_k[n]$ are overlapped windows of the signal and K is the number of segments [54]. The total power in the physiologically relevant range $[0.5,\ f_s/2]$ [55] is

$$\hat{P}_{\text{total}} = \text{bandpower}\left(x, f, [0.5, f_s/2]\right). \tag{4.83}$$

For each frequency band, the band power is computed and then normalized by P_{total}. Relative delta $(0.5\text{–}4\,\text{Hz})$, theta $(4\text{–}8\,\text{Hz})$, alpha $(8\text{–}13\,\text{Hz})$, beta $(13\text{–}30\,\text{Hz})$, and gamma $(30\text{–}45\,\text{Hz})$ band powers have all been shown to be informative markers of seizure onset [56].

1. Relative Delta Band Power (0.5-4 Hz)

$$\hat{P}_{\text{rel},\delta} = \frac{\hat{P}_{\delta}}{\hat{P}_{\text{total}}} \tag{4.84}$$

where $\hat{P}_{\delta}$ is the power estimate in the delta band.

2. Relative Theta Band Power (4-8 Hz)

$$\hat{P}_{\text{rel},\theta} = \frac{\hat{P}_{\theta}}{\hat{P}_{\text{total}}} \tag{4.85}$$

where $\hat{P}_{\theta}$ is the power estimate in the theta band.

3. Relative Alpha Band Power (8-13 Hz)

$$\hat{P}_{\text{rel},\alpha} = \frac{\hat{P}_{\alpha}}{\hat{P}_{\text{total}}} \tag{4.86}$$

where $\hat{P}_{\alpha}$ is the power estimate in the alpha band.

4. Relative Beta Band Power (13-30 Hz)

$$\hat{P}_{\text{rel},\beta} = \frac{\hat{P}_{\beta}}{\hat{P}_{\text{total}}} \tag{4.87}$$

where $\hat{P}_{\beta}$ is the power estimate in the beta band.

5. Relative Gamma Band Power (30-45 Hz)

$$\hat{P}_{\text{rel},\gamma} = \frac{\hat{P}_{\gamma}}{\hat{P}_{\text{total}}} \tag{4.88}$$

where $\hat{P}_{\gamma}$ is the power estimate in the gamma band.

6. To quantify the randomness of the spectral distribution, the spectral entropy is computed by first normalizing the PSD into a probability mass function

$$p\left(f\right) = \frac{\hat{P}\left(f\right)}{\sum_{f} \hat{P}\left(f\right)}, \tag{4.89}$$

and then we calculate the Shannon entropy

$$H = -\sum_{i} p(f_i)\, \log_2\left(p(f_i)\right). \tag{4.90}$$

Higher spectral entropy indicates a more uniform (less predictable) spectrum, often associated with ictal dynamics [57].

Dimensionality Reduction via Autoencoders

In order to learn a compact representation of our high–dimensional feature vectors $\mathbf{x}_i \in \mathbb{R}^{180}$, we employ an undercomplete autoencoder [58] to obtain $\mathbf{z}_i \in \mathbb{R}^3$.

A three–dimensional autoencoder

$$f_{\boldsymbol{\theta}} : \mathbb{R}^d \longrightarrow \mathbb{R}^n, \qquad n = 3, \tag{4.91}$$

parameterised by weights $\boldsymbol{\theta}$, is trained unsupervised to reconstruct $\mathbf{x}_i$ from a latent code $\mathbf{z}_i$:

$$\boldsymbol{\theta} = \arg\min_{\theta} \sum_i \|\mathbf{x}_i - g_\theta(f_\theta(\mathbf{x}_i))\|_2^2, \tag{4.92}$$

with $\mathbf{z}_i = f_{\boldsymbol{\theta}}(\mathbf{x}_i)$ and $\hat{\mathbf{x}}_i = g_{\boldsymbol{\theta}}(\mathbf{z}_i)$.

Writing the latent matrix $\mathbf{Z} = [\mathbf{z}_1^\intercal \ \dots \ \mathbf{z}_N^\intercal]^\intercal \in \mathbb{R}^{N \times n}$, we define the latent space

$$\mathcal{Z} = \left\{ \mathbf{z} = f_{\boldsymbol{\theta}}(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^d \right\} \subset \mathbb{R}^n. \tag{4.93}$$

Let $\mathbf{x}_i$, $i = 1, \dots, N$, be the ith training sample. The encoder defines a nonlinear mapping

$$\mathbf{z}_i = f\left(\mathbf{x}_i\right) = \sigma\left(\mathbf{W}_e\, \mathbf{x}_i + \mathbf{b}_e\right), \tag{4.94}$$

where $\mathbf{W}_e \in \mathbb{R}^{n \times p}$ and $\mathbf{b}_e \in \mathbb{R}^n$ are learnable weights and biases, and $\sigma(\cdot)$ is an element–wise activation (e.g. ReLU or sigmoid). The decoder then attempts to reconstruct the original input via

$$\hat{\mathbf{x}}_i = g\left(\mathbf{z}_i\right) = \phi\left(\mathbf{W}_d\, \mathbf{z}_i + \mathbf{b}_d\right), \tag{4.95}$$

with $\mathbf{W}_d \in \mathbb{R}^{p \times n}$, $\mathbf{b}_d \in \mathbb{R}^p$, and decoder activation $\phi(\cdot)$.

The training minimises the mean squared error (MSE) plus a weight-decay regularisation on the parameters:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} \ + \ \lambda \left(\|\mathbf{W}_e\|_F^2 + \|\mathbf{W}_d\|_F^2 \right), \tag{4.96}$$

where

$$\mathcal{L}_{\text{rec}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{x}^{(i)} \ - \ \hat{\mathbf{x}}^{(i)} \right\|_2^2, \tag{4.97}$$

$\|\cdot\|_F$ is the Frobenius norm and $\lambda > 0$ is the regularization coefficient.

4.1.5 Results

Refer to Tables 4.1-4.2 for the per-subject and per-classifier scores. All experiments used the public intracranial EEG datasets (dogs 1–5, human patients P1–P2) from [59].

For each of the seven subjects, we trained seven classifiers: QDA, CART, NB, LR, $\text{MLP}_{8\text{–}4}$, SVM_{poly}, and SVM_{rbf}, under three resampling strategies: no resampling (NO), baseline random oversampling (BO), and the proposed MS-Boot (MB). To capture run-to-run variability, we repeat the train/test procedure 100 times for every ⟨subject, classifier, strategy⟩ triple, yielding $7 \times 7 \times 3 \times 100 = 14{,}700$ scored runs.

Tables 4.3-4.4 summarise the key findings. For each metric we form paired differences $\Delta^{\text{MB–NO}}$ and $\Delta^{\text{MB–BO}}$ inside every subject–classifier block (49 blocks total) and pool them. On the patient datasets (P1, P2), MB raises F_1 by $+21.3$ pp (P1) and $+20.1$ pp (P2) vs. NO, with additional gains of $+3.3$ pp and $+2.7$ pp vs. BO, respectively. Averaged over P1+P2, power (TPR) increases by $+32.5$ pp vs. NO and $+6.5$ pp vs. BO, while maintaining competitive α (FPR) and overall accuracy (ACC). Under extreme class skew (Dog1–Dog5), MB consistently outperforms NO in F_1 (e.g., Dog2 $+26.9$ pp, Dog4 $+23.5$ pp) and is broadly competitive with BO (dataset-dependent deltas), with notable pockets of win for specific model families (e.g., Dog3 with CART). Model-level highlights on the patient cohorts indicate that these gains are not confined to a particular hypothesis class: CART on P1, LR on P2, and a compact $\text{MLP}_{8\text{–}4}$ on P2 all exhibit substantial increases in power (reduced β) under MB without inflating α.

4.1.6 Conclusion and Future Work

We introduced the Mahalanobis–distance Stratified Bootstrap (MS-Boot), a hyperparameter-free, distance-aware resampling scheme that preserves a specified training prior while reallocating sampling mass toward informative minority instances. On the theory side, a first-order influence-function analysis and a simple monotone–transfer argument show that MS-Boot increases the class-1 mean margin, and hence the statistical powers $(1 - \beta)$, at a fixed operating threshold relative to both uniform oversampling and no resampling, under mild regularity. A Jensen step further explains the systematic improvement over the no-resampling baseline.

Empirically, MS-Boot is classifier-agnostic and effective on intracranial EEG seizure prediction. Across seven subjects and seven model families, it consistently raises power/sensitivity without inflating Type-I error (α) or sacrificing precision, with especially strong F_1 and TPR (power) gains on the human patient datasets (P1–P2) and competitive performance under the extreme skews of Dog1–Dog5. Notably, these benefits arise without adding model complexity or tuning extra hyperparameters.

Table 4.1: Results Dogs 1-5

	Dog 1			Dog 2			Dog 3			Dog 4			Dog 5		
	NO	BO	MB	NO	BO	MB	NO	BO	MB	NO	BO	MB	NO	BO	MB
TPR	0	.250	.083	0	.847	.707	.178	.578	.483	.006	.533	.478	0	.600	.600
TNR	1	.527	.166	1	.607	.712	.996	.853	.778	1	.772	.763	1	.464	.427
ACC	.950	.514	.162	.925	.625	.711	.955	.840	.764	.888	.745	.731	.938	.473	.438
ERR	.050	.487	.838	.075	.375	.289	.045	.160	.236	.112	.255	.270	.063	.527	.563
PREC	0	.027	.009	0	.150	.169	.558	.184	.105	.050	.229	.203	0	.070	.065
F1	0	.049	.016	0	.254	.272	.264	.273	.172	.010	.320	.285	0	.125	.118

(a) QDA

	Dog 1			Dog 2			Dog 3			Dog 4			Dog 5		
	NO	BO	MB	NO	BO	MB	NO	BO	MB	NO	BO	MB	NO	BO	MB
TPR	0	.068	0	.262	.353	.277	0	.103	.453	.007	.263	.347	.086	.220	.400
TNR	1	.901	.299	.965	.921	.914	1	.978	.990	.986	.862	.862	.986	.932	.907
ACC	.950	.860	.284	.912	.878	.866	.950	.934	.963	.876	.794	.804	.929	.887	.875
ERR	.050	.140	.716	.088	.122	.134	.050	.066	.037	.124	.206	.196	.071	.113	.125
PREC	0	.025	0	.313	.266	.210	0	.126	.698	.007	.197	.243	.148	.159	.222
F1	0	.036	0	.280	.302	.225	0	.111	.501	.007	.222	.284	.090	.181	.286

(b) CART

	Dog 1			Dog 2			Dog 3			Dog 4			Dog 5		
	NO	BO	MB	NO	BO	MB	NO	BO	MB	NO	BO	MB	NO	BO	MB
TPR	0	0	0	.025	.833	.453	.023	.178	.108	.038	.222	.218	0	.208	.200
TNR	.991	.883	.282	.995	.857	.849	1	.950	.919	.995	.978	.977	1	.860	.867
ACC	.941	.839	.268	.923	.855	.819	.951	.911	.878	.887	.893	.892	.937	.819	.825
ERR	.059	.161	.732	.077	.145	.181	.049	.089	.122	.113	.107	.109	.063	.181	.175
PREC	0	0	0	.018	.323	.192	.070	.225	.068	.131	.596	.582	0	.105	.091
F1	0	0	0	.021	.465	.269	.033	.194	.082	.055	.320	.314	0	.134	.125

(c) NB

Table 4.1: Results Dogs 1-5

	Dog 1			Dog 2			Dog 3			Dog 4			Dog 5		
	NO	BO	MB	NO	BO	MB	NO	BO	MB	NO	BO	MB	NO	BO	MB
TPR	0	.250	.145	.003	.833	.447	.165	.253	.250	0	.560	.481	0	.600	.600
TNR	1	.530	.159	.999	.585	.820	1	.824	.795	1	.742	.741	1	.545	.573
ACC	.950	.516	.158	.924	.604	.792	.958	.780	.768	.888	.721	.712	.938	.548	.575
ERR	.050	.485	.842	.076	.396	.208	.042	.205	.232	.113	.279	.289	.063	.452	.425
PREC	0	.027	.014	.005	.140	.174	.660	.071	.061	0	.217	.193	0	.082	.086
F1	0	.049	.025	.004	.240	.240	264	.111	.098	0	.312	.274	0	.144	.150

(d) LR

	Dog 1			Dog 2			Dog 3			Dog 4			Dog 5		
	NO	BO	MB	NO	BO	MB	NO	BO	MB	NO	BO	MB	NO	BO	MB
TPR	.015	.170	.060	.023	.775	.445	.108	.183	.115	.006	.412	.342	.006	.396	.400
TNR	.997	.826	.263	.990	.846	.887	.996	.941	.947	.997	.863	.871	1	.686	.653
ACC	.948	.793	.253	.918	.841	.854	.952	.903	.905	.886	.812	.811	.938	.668	.638
ERR	.052	.207	.747	.083	.159	.147	.048	.097	.095	.114	.188	.189	.063	.332	.363
PREC	.017	.046	.016	.025	.326	.277	.347	.293	.170	.006	.280	.246	.020	.078	.071
F1	.016	.065	.025	.023	.441	.317	.160	.183	.110	.006	.322	.280	.009	.129	.121

(e) $MLP_{8\text{-}4}$

	Dog 1			Dog 2			Dog 3			Dog 4			Dog 5		
	NO	BO	MB	NO	BO	MB	NO	BO	MB	NO	BO	MB	NO	BO	MB
TPR	0	0.030	.018	.037	.788	.460	.098	.010	.010	.036	.344	.293	.014	.392	.200
TNR	.998	.928	.282	.990	.867	.922	.991	.978	.982	.980	.844	.846	.996	.727	.867
ACC	.948	.884	.269	.918	.861	.888	.946	.930	.933	.874	.788	.783	.934	.706	.825
ERR	.052	.117	.731	.082	.139	.113	.054	.070	.067	.126	.212	.217	.066	.294	.175
PREC	0	.014	.007	.024	.340	.342	.370	.004	.011	.015	.229	.193	.024	.092	.091
F1	0	.019	.009	.024	.469	.371	.153	.006	.009	.020	.271	.229	.016	.146	.125

(f) SVM_{poly}

Table 4.1: Results Dogs 1-5

	Dog 1			Dog 2			Dog 3			Dog 4			Dog 5		
	NO	BO	MB	NO	BO	MB	NO	BO	MB	NO	BO	MB	NO	BO	MB
TPR	0	0	0	.010	.620	.395	.108	0	.003	.018	.056	.071	.002	0	0
TNR	1	.988	.314	.997	.993	.999	1	1	1	.995	.997	.991	.999	.954	.947
ACC	.950	.939	.299	.923	.965	.954	.955	.950	.950	.885	.891	.888	.936	.894	.888
ERR	.050	.061	.701	.077	.035	.046	.045	.050	.050	.115	.109	.112	.064	.106	.113
PREC	0	0	0	.015	.940	.986	.415	0	.010	.053	.247	.333	.010	0	0
F1	0	0	0	.012	.733	.553	.170	0	.004	.024	.084	.102	.003	0	0

(g) SVM_{rbf}

Our analysis assumes a fixed operating threshold and a no–FPR–drift condition; characterizing MS-Boot under explicit cost asymmetries and calibrated threshold selection for deployment priors is a natural extension. The Mahalanobis metric relies on an estimated minority covariance; studying shrinkage/robust estimators in very small N_1 regimes, adaptive choices of representation, e.g., supervised embeddings, and extensions to multiclass or sequence-aware settings (temporal EEG structure) are promising directions. Finally, combining MS-Boot with cost-sensitive training or curated synthetic augmentation may further improve power in the hardest regimes while retaining the plug-and-play simplicity demonstrated here.

Table 4.2: Results Patients 1-2

	P1			P2		
	NO	BO	MB	NO	BO	MB
TPR	.753	1	1	.285	.530	.550
TNR	.821	.800	.800	.673	.515	.534
ACC	.805	.846	.846	.543	.520	.539
ERR	.195	.154	.154	.457	.480	.461
PREC	.572	.600	.600	.209	.355	.376
F1	.658	.750	.750	.400	.421	.436

(a) QDA

	P1			P2		
	NO	BO	MB	NO	BO	MB
TPR	.847	.993	.993	.440	.470	.470
TNR	.795	.815	.875	.715	.566	.521
ACC	.807	.856	.902	.623	.534	.504
ERR	.193	.144	.098	.377	.466	.496
PREC	.545	.637	.754	.445	.337	.321
F1	.664	.768	.842	.508	.419	.385

(b) CART

	P1			P2		
	NO	BO	MB	NO	BO	MB
TPR	.800	1	1	0	.405	.478
TNR	.751	.652	.601	.908	.595	.538
ACC	.762	.732	.693	.605	.532	.518
ERR	.238	.267	.307	.395	468	.483
PREC	.491	.479	.452	0	.278	.327
F1	.617	.644	.617	0	.412	.408

(c) NB

Table 4.2: Results Patients 1-2 (Continued)

	P1			P2		
	NO	BO	MB	NO	BO	MB
TPR	.593	.997	1	0	.670	.695
TNR	.911	.435	.419	.949	.596	.563
ACC	.838	.565	.553	.633	.621	.607
ERR	.162	.435	.447	.368	.379	.393
PREC	.667	.349	.344	0	.464	.446
F1	.667	.516	.511	0	.538	.546

(d) LR

	P1			P2		
	NO	BO	MB	NO	BO	MB
TPR	.730	.943	.970	.465	.508	.503
TNR	.898	.791	.794	.556	.523	.520
ACC	.859	.826	.835	.526	.518	.514
ERR	.141	.174	.165	.474	.483	.486
PREC	.687	.635	.630	.342	.342	.344
F1	.734	.742	.753	.410	.416	.410

(e) $MLP_{8\text{-}4}$

	P1			P2		
	NO	BO	MB	NO	BO	MB
TPR	.319	.673	.650	.415	.716	.765
TNR	.888	.764	.773	.821	.666	.638
ACC	.740	.740	.741	.686	.683	.680
ERR	.260	.260	.259	.314	.317	.320
PREC	.547	.515	.520	.434	.535	.543
F1	.348	.572	.563	.392	.594	.621

(f) SVM_{poly}

Table 4.2

	P1			P2		
	NO	BO	MB	NO	BO	MB
TPR	.329	.361	.396	.194	.323	.370
TNR	.961	.961	.947	.969	.937	.913
ACC	.797	.806	.804	.711	.732	.732
ERR	.203	.194	.196	.289	.268	.268
PREC	.699	.832	.812	.470	.881	.890
F1	.416	.461	.483	.266	.413	.443

(g) SVM_{rbf}

Table 4.3: Average F1 change on Dog1–Dog5 (extreme class imbalance). Values are absolute percentage points (pp).

Dataset	**vs NO (pp)**	**vs BO (pp)**
Dog1	+0.8	-2.1
Dog2	+26.9	-9.4
Dog3	-1.0	+1.4
Dog4	+23.5	-1.2
Dog5	+11.5	+0.9

Table 4.4: MB (MS-Boot) gains on patients (P1, P2). Values are absolute percentage points (pp).

Metric	**vs NO (pp)**	**vs BO (pp)**
F1 (P1)	+21.3	+3.3
F1 (P2)	+20.1	+2.7
Mean TPR uplift (P1+P2)	+32.5	+6.5

Table 4.5: Dog3 with CART (Decision Tree): detailed gains. Values are absolute percentage points (pp).

Metric	vs NO (pp)	vs BO (pp)
F1	+50.1	+39.1
ACC	+1.3	+2.8
PREC	+69.8	+57.2
TPR	+45.3	+35.0

Table 4.6: Model-level highlights on patient cohorts. Values are absolute percentage points (pp).

Model / Dataset / Metric	vs NO (pp)	vs BO (pp)
P1 • CART • F1	+41.8	+13.5
P1 • CART • ACC	+20.7	+12.0
P1 • CART • PREC	+43.3	+25.0
P1 • CART • TNR	+13.9	+16.4
P1 • CART • TPR	+40.0	-0.4
P2 • LR • F1	+30.6	+10.4
P2 • LR • TPR	+62.5	+27.4
P2 • $MLP_{8\times4}$ • F1	+43.3	+5.0
P2 • $MLP_{8\times4}$ • TPR	+55.4	+13.3

4.2 Heterogeneity-Stratified Bootstrap (HS-Boot)

We propose the Heterogeneity-Stratified Bootstrap (HS-Boot), a stratification method that gives higher resampling probabilities to the sample points in the less homogeneous regions. We demonstrate its advantage in the case of training a detector by oversampling the under-represented class in an imbalanced data set. We took a case study of a spoiled food detector in form of an electronic nose. The performance metrics were calculated on the out-of-bag test set as well as on measurements collected from another sensor.

4.2.1 Introduction

It is often the case that the class distribution in a dataset is not equal. The learning process to develop effective decision boundaries to support the decision-making process is called imbalanced learning [60]. Some machine learning algorithms such as naive Bayes classifier [61, Chapter 4], decision trees [62], quadratic discriminant analysis [63], and neural network [64] are often biased towards the majority classes than the minority target class, such that there is a higher misclassification rate in the target instances [35]. Using a balanced training set may help. This may be created with an artificially equal class distribution by oversampling the instances in the minority class.

The simplest oversampling method is by randomly duplicating instances in the minority class [65]. Other techniques include the SMOTE [5] and its variations [66], the ADASYN [35], and data augmentation [67].

In this section, we propose an oversampling method based on the stratified resampling with replacement, or stratified bootstrapping. It is well known that stratification reduces variance [40]. In regression problems, the stratified bootstrap was proven to be robust against outliers [68]. The proposed stratification is not only done based on the class labels but also the heterogeneity in the feature space. The proposed method is applied to train a spoiled food detector.

Anosmic or visually impaired people may fail to recognise spoiled food in their fridge or pantry. A smart detector that can be trained to recognise such hazards is needed. Intelligently choosing a method to train the algorithm may reduce dependencies on the complexity and the amount of data collected. IoT devices do not always support collection of long data segments. Bootstrap techniques help in estimating statistical characteristics of interest in the case of limited samples [12], [13], [14].

The measurements in this study were collected using the BME688 (Bosch Sensortec) [69]. It is a metal oxide-based sensor that detects gases by adsorption and subsequent oxidation or reduction on its sensitive layer. It is capable of measuring volatile organic compounds (VOCs) in the surrounding air. The metal oxide layer of the gas sensor at

different temperatures allows measurements with different sensitivities, thereby creating unique fingerprints for different gas compositions. In other words, the sensor acts as an electronic nose that can distinguish different gas compositions by their unique digital fingerprints. However, it needs to first learn about the different gases. A trained classification model can then be deployed on a microcontroller that will take the readings from the BME688, which in our case is the Adafruit HUZZAH32 - ESP32 Feather board [70].

Given the measurement set $\mathcal{X} = \{(x_j, y_j)\}_{j=1}^{N}$, where the total number of samples N can be broken down according to the class labels,

$$N = N_0 + N_1, \quad N_0 >> N_1, \tag{4.98}$$

and the subscripts 0 and 1 indicate fresh and spoiled, respectively, we would like to report the performance metrics with the proposed oversampling method on the Out-of-Bag Bootstrap (or Out-of-Bootstrap, OOB) test set [71], [72] as well as on measurements resulting from another sensor.

Next, Subsection 4.2.2 elaborates the proposed oversampling algorithm, a dummy example to illustrate it, and its application on the case study. Subsection 4.2.3 includes the qualitative results and finally Subsection 4.2.4 concludes the section and discusses the future direction.

4.2.2 Methodology

Suppose the feature space can be divided into K cells. We propose using the heterogeneity measure

$$H\left(C,k\right) = -\sum_{c=1}^{C} \frac{a_{ck}}{N} \log\left(\frac{a_{c,k}}{\sum_{c=1}^{C} a_{c,k}}\right), \quad k = 1, \ldots, K, \tag{4.99}$$

where a_{ck} is the number of sample points that belong to class c in cell k such that the cells with no sample points or with homogeneous sample points will have $H(k) = 0$. This was inspired by the homogeneity score in clustering [73] and here we treat each cell as one cluster. The resampling probability distribution is then modified such that the homogenous area becomes less likely to be selected and the samples in the heterogeneous area are more favored. Contrary to the other oversampling methods previously discussed, where the treatments are focused on the minority class, this method also affects the majority class.

Algorithm 2 summarises the proposed method applied for oversampling. A simple example with two classes (Class $*$ and Class $\circ$) with 6 and 8 instances is illustrated in Table 4.7. The two-dimensional feature space is divided into $K = 18$ cells. Originally, the stratified resampling probability distribution is uniform with $p_* \left(k\right) = 1/6$ and $p_\circ \left(k\right) = 1/8$,

$k = 1, \ldots, K$ (4.8a). The heterogeneity measure is calculated (4.8b) and as we can see the probabilities for both classes in the more heterogeneous area increased while those in the more homogenous area are reduced (4.8c).

Algorithm 2: Oversampling with Heterogeneity-Stratified Bootstrap (HSBoot)

Step 1 Divide the sample domain into K cells.

Step 2 For each cell k, $k = 1, \ldots, K$, calculate the heterogeneity measure $H(k)$ (Eq. 4.99)

Step 3 Modify the resampling probability distribution $p_c\,(k)$ for each class c, $c = 1, \ldots, C$, in cluster k,

$$p'_c\,(k) = \frac{\frac{1}{N_c} + \gamma H(k)}{1 + \gamma \sum_k a_{ck} H(k)}, \tag{4.100}$$

where γ is a hyperparameter constant that determines the influence of the heterogeneity, a_{ck} is the number of samples in cell k that belongs to class c, and the denominator is to make sure that the sum equals 1.

Step 4 HSBoot resampling and oversampling: Sample N_0 instances with replacement from Class 0 (non-target) and likewise N_0 instances from Class 1 (target) with the updated distribution p'_c. Let this sample set with $2N_0$ instances be $\mathcal{X}_{\text{train}}$.

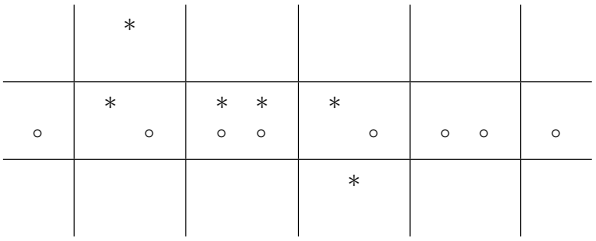
Step 5 Out-of-Bootstrap (OOB) test set: take the observations from $\mathcal{X}$ that do not make it to $\mathcal{X}_{\text{train}}$ to be the bootstrap test set, $\mathcal{X}_{\text{test}}$.

The problem with using the OOB test set is that the remaining minority instances that did not make it to the training will be even smaller in proportion to the majority instances. Therefore, we also took measurements from a different sensor that was measuring the same specimens at around the same time, as an additional test set.

The gas compositions of the following specimens were measured:

1. Fresh Chicken: a piece of fresh, raw chicken
2. Yoghurt: fresh yoghurt
3. Beef: a piece of fresh raw beef
4. Coffee: a handful of coffee beans

Table 4.7: Dummy example of $C = 2$ classes with respectively 6 and 8 sample points, and $K = 18$ cells.



	.167				
	.167		.167 .167	.167			
.125	.125		.125 .125	.125	.125 .125	.125
				.167		

(a) Random resampling probability distribution $p_c\,(k)$

0	0	0	0	0	0
0	.099	.198	.099	0	0
0	0	0	0	0	0

(b) Heterogeneity measure $H(k)$

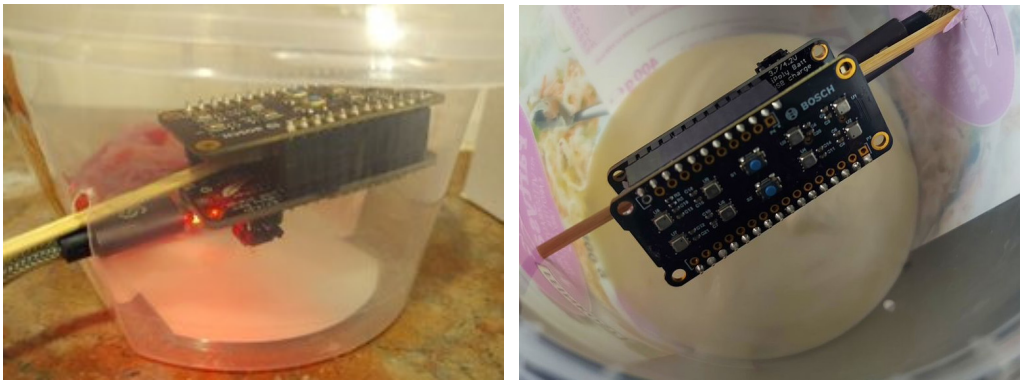
	.105				
	.167		.229 .229	.167			
.078	.140		.202 .202	.140	.078 .078	.078
				.105		

(c) Updated resampling probability distribution $p'_c\,(k)$

5. Mix (target): a piece of spoiled raw chicken, mixed with some fresh vegetables

6. Rotten Chicken (target): a piece of spoiled raw chicken

Each specimen was placed in a plastic container together with the board that was suspended a few centimeters above the base of the container (see Fig. 4.1). The containers were not completely sealed off, hence the sensor still has a slight exposure to the atmosphere in the well-ventilated room where the experiments took place.



(a) **(b)**

Figure 4.1: Training the electronic nose with different specimens.

For the purpose of this work, we took only 500 measurements for each of the specimens mentioned. We used two predictive features, the temperature and gas resistance. The original data were in degree Celcius and Ohms, respectively, then normalised such that they have 0 to 1 range.

Fig. 4.2 shows the gas measurements of the 6 specimens. The first 4 specimens are non-target (fresh) and the last 2 are the target (spoiled) classes. Fig. 4.3 shows the gas measurements from a different sensor. The new test set is left as is (i.e. not resampled, hence unbalanced).

The random resampling probability for Class 0 is $1/(4\times500)$ and for Class 1 (target) is $1/(2\times500)$. After taking the heterogeneity into account, the probability distribution was updated, as illustrated in Fig. 4.4.

In the experiment, we tried different values of λ: $\lambda = 0.1$, $\lambda = 1$, and $\lambda = 5$, and compared the performance of our method with the random oversampling. In the next subsection we report the performance of the trained models based on different algorithms: naive Bayes classifier[61, Chapter 4], decision trees [62], quadratic discriminant analysis

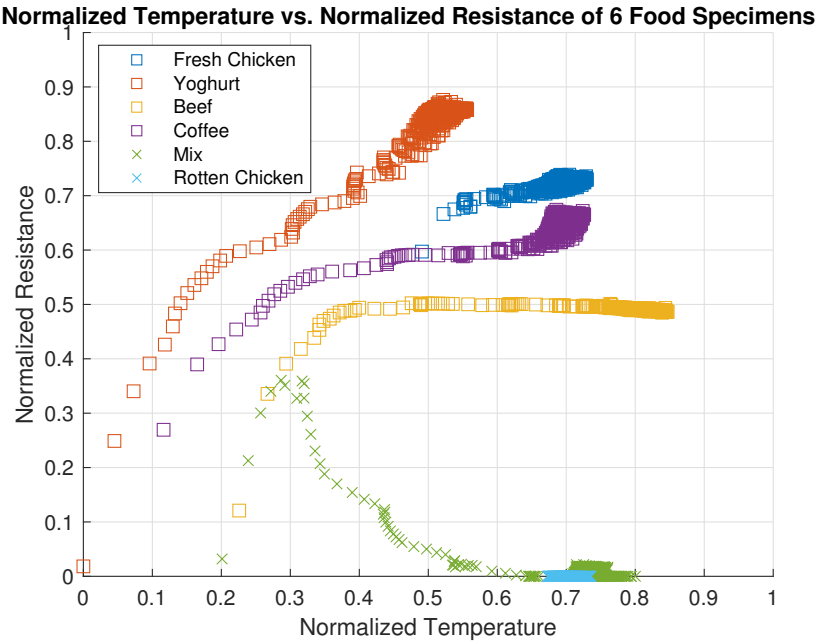


Figure 4.2: The measurements for the training set

[63], and neural network [64]. For each of the algorithm, the hyperparameters were optimised with leave-out 0.3 cross validation.

4.2.3 Experiment and Results

We list in Table 4.8 the following classification metrics, where TP is the true positives, TN the true negatives, FP the false positive, and FN the false negatives [74]:

1. Sensitivity (SN) or true positive rate (TPR) or recall (REC) = TP / (TP + FN)
2. Specificity (SP) or true negative rate (TNR) = TN / (TN + FP)
3. Accuracy (ACC) = (TP + TN) / (TP + TN + FN + FP)
4. Error rate (ERR) = (FP + FN) / (TP + TN + FN + FP)
5. Precision (PREC) = TP / (TP + FP)
6. F1 score (F1) = 2 * PREC * REC / (PREC + REC)

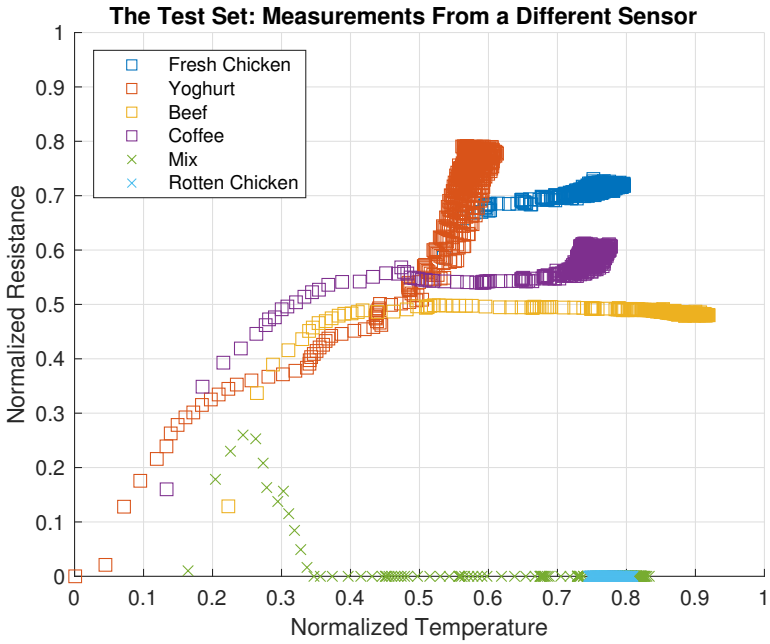


Figure 4.3: The measurements as the additional test set

We observe that the proposed resampling method might slightly reduce the sensitivity and increase the error rate, but the specificity, accuracy, precision, and F1 score are constantly better, which is favorable in an unbalanced data set. We also observe that higher λ seems to improve the metrics up to certain point, then there is a diminishing return.

4.2.4 Conclusion and Future Work

We proposed a novel stratified bootstrap method based on the heterogeneity of instances in certain regions of the feature space. This was then applied for oversampling a minority class in an unbalanced dataset when training a spoiled food detector. The proposed method constantly improved the specificity, accuracy, precision, and F1 score of the detector as compared to those by using random oversampling.

In the future, it would be interesting to replace the cells with adaptive region of interests as far as the heterogeneity is concerned, as well as to develop a method to optimise the value of λ.

Table 4.8: Average out of 500 trials for SN (sensitivity, or true positive rate or recall), SP (specificity, or true negative rate), ACC (accuracy), ERR (error rate), PREC (precision), F1 (F1 score, harmonic mean of precision and recall)

	Random, on OOB test set	**Proposed HS-Boot** on OOB test set			Random, new sensor test set	**Proposed HS-Boot** new sensor		
		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$
SN	**.9945**	.9917	.9917	0.9905	.9842	**.9850**	.9797	.9701
SP	.9990	.9991	.9994	**.9995**	.9944	.9945	.9960	**.9961**
ACC	.9983	.9966	**.9968**	.9965	.9910	**.9914**	.9906	.9874
ERR	**.0017**	.0034	.0032	.0035	.0090	**.0086**	.0094	.0126
PREC	.9944	.9982	.9988	**.9990**	.9888	.9890	.9920	**.9921**
F1	.9944	.9949	**.9952**	.9947	.9863	**.9868**	.9855	.9806

(a) Naive Bayes

	Random, on OOB test set	**Proposed HS-Boot** on OOB test set			Random, new sensor test set	**Proposed HS-Boot** new sensor		
		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$
SN	.9989	.9978	.9982	**.9990**	**.9991**	.9982	.9980	.9983
SP	.9980	.9984	**.9987**	**.9987**	.9932	.9933	.9950	**.9957**
ACC	.9981	.9982	.9985	**.9988**	.9952	.9952	.9960	**.9965**
ERR	.0019	.0018	.0015	**.0012**	.0048	.0048	.0040	**.0035**
PREC	.9893	.9967	.9973	**.9974**	.9867	.9869	.9901	**.9914**
F1	.9940	.9972	.9977	**.9982**	.9928	.9929	.9940	**.9948**

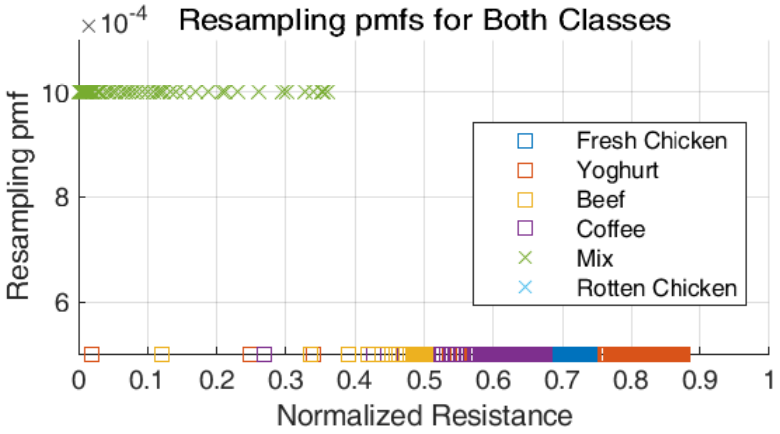
(b) Decision Tree

	Random, on OOB test set	Proposed HS-Boot on OOB test set			Random, new sensor test set	Proposed HS-Boot new sensor		
		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$
SN	.9960	.9914	.9957	**.9976**	.9978	.9930	**.9956**	.9951
SP	.9967	.9975	**.9979**	.9978	.9909	**.9937**	.9932	.9921
ACC	.9966	.9955	.9971	**.9977**	.9932	.9934	**.9940**	.9931
ERR	.0034	.0045	.0029	**.0023**	.0068	.0066	.0060	**.0069**
PREC	.9826	.9951	**.9958**	.9957	.9822	.9821	**.9868**	.9846
F1	.9891	.9942	.9957	**.9966**	.9899	.9900	**.9911**	.9897

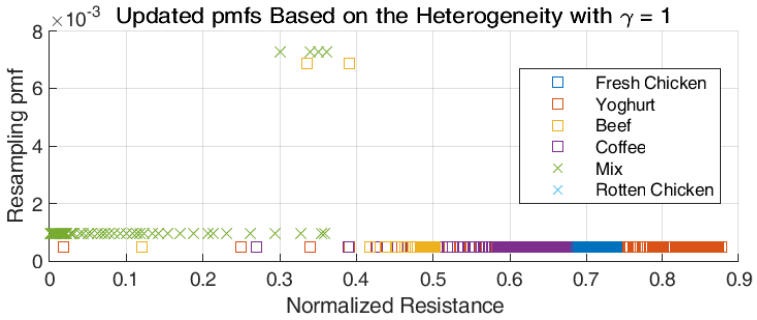
(c) Quadratic Discriminant

	Random, on OOB test set	Proposed HS-Boot on OOB test set			Random, new sensor test set	Proposed HS-Boot new sensor		
		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$
SN	**.9803**	.9433	.9440	.9086	**.9761**	.9642	.9364	.9021
SP	.9723	.9985	.9989	**.9990**	.9449	.9540	.9698	**.9717**
ACC	.9735	.9801	**.9806**	.9688	.9553	.9574	**.9587**	.9485
ERR	.0265	.0199	**.0194**	.0312	.0447	.0426	**.0413**	.0515
PREC	.9663	.9969	.9977	**.9978**	.9338	.9377	.9504	**.9518**
F1	.9751	.9970	.9978	**.9982**	.9576	.9614	.9666	**.9683**

(d) Neural Network with 2 hidden layers, with 8 and 4 fully connected outputs for each hidden layer, respectively.



(a)



(b)

Figure 4.4: Illustration of the updated resampling PMFs

4.3 The Heterogeneity-Intensified and Heterogeneity Ratio-Stratified Bootstrap (HiS- and HeRS-Boot)

We investigated two variations of the previously proposed heterogeneity-stratified bootstrap (HSBoot) oversampling method, namely the improved Heterogeneity-Stratified (HiS-) and Heterogeneity Ratio-Stratified (HeRS-) Boot, for balancing a data set by assigning higher resampling probabilities to sample points in less homogeneous regions. Our study focused on two detection cases: spoiled food and allergen. Results demonstrate the effectiveness and generalizability of our method across different sensors, highlighting its potential for real-world applications and positive impact on daily life.

4.3.1 Introduction

Building on our previous work [29], we introduce two variations: the Heterogeneity-Intensified Stratified-Bootstrap (HiS-Boot) and the Heterogeneity Ratio-Stratified Bootstrap (HeRS-Boot), offering an advanced approach to harnessing heterogeneity within the feature domain.

In this section, our exploration extends to two detection cases: spoiled food and allergens. These two oversampling methods, HiS-Boot and HeRS-Boot, are applied to improve the performance of classification algorithms in the context of food safety detection, which is often challenged by unbalanced datasets. Our ultimate aim is to demonstrate the potential of HiS-Boot and HeRS-Boot in enhancing food safety detection capabilities by improving the performance of the chosen classification algorithm.

Bootstrap techniques help in estimating statistical characteristics of interest in the case of limited samples [12], [13], [14]. Oversampling introduces bias to compensate for the class imbalance. On the other hand, it is also well known that stratified bootstrapping reduces variance [40]. This is easily shown by the law of total variance $\mathrm{Var}\left[H\left(U\right)\right] \geq \mathrm{E}\left[\mathrm{Var}\left[H\left(U\right)|\, S\right]\right]$ for all random variables U and S. Let S be a random variable taking values in $\{1,\ldots,C\}$ with the probabilities $\{p_c, c=1,\ldots,C\}$. Let $\ell = \mathrm{E}\left[H\left(U\right)\right]$, The stratified sampling estimator is then given by $\hat{\ell}^s = \sum_{c=1}^{C} p_c \frac{1}{N_c} \sum_{j=1}^{N_c} H\left(U_{cj}|\, S=c\right)$, where U_{cj} is the jth sample from the conditional distribution of U given $S=c$.

Denote the estimator as a result of double stratification with respect to the sample classes and the K strata as $\hat{\ell}^{c,k}$. With the law of iterated expectations, it can further be shown that

$$\mathrm{Var}\left[\hat{\ell}^c\right] = \sum_{c=1}^{C} \frac{N_c \sigma_c^2}{N^2} \;\geq\; \mathrm{Var}\left[\hat{\ell}^{c,k}\right] = \sum_{k=1}^{K}\sum_{c=1}^{C} \frac{N_{c,k}}{N^2} \sigma_{c,k}^2. \tag{4.101}$$

where we used $\sigma_c^2 = \mathrm{Var}\left[H\left(U\right)|\, S=c\right]$ and correspondingly for the subscript $_{c,k}$.

4.3.2 Methodology

We introduce refinements to the heterogeneity-stratified bootstrap oversampling methods and expand their applications beyond food spoilage detection to include allergen detection. Our methodology is summarized in Algorithm 1. The process begins with segmenting the sample domain into K cells and computing the heterogeneity measure for each cell, $H(C, k)$, following the formula as shown in Table 4.9. Next, the resampling probability distribution for each class in the cell is adjusted, and oversampling is performed, drawing N_0 instances with replacement from both Class 0 (non-target) and Class 1 (target), utilizing the updated distribution. Table 4.8 gives the example calculation.

The problem with the Out-of-Bag Bootstrap (OOB) test set lies in the decreased proportion of remaining minority instances compared to the majority instances. To counter this, measurements from a separate sensor serve as an additional test set. This method consolidates the effectiveness of the HS-, HiS-, and HeRS-Boot oversampling methods.

We ensure continuity with our previous work by employing the same dataset from our previous work [29], recorded using the BME688 sensor and Adafruit HUZZAH32 - ESP32 Feather board [69], [70], which includes the following food specimens:

1. Fresh Chicken: a piece of fresh, raw chicken
2. Yoghurt: fresh yoghurt
3. Beef: a piece of fresh raw beef
4. Coffee: a handful of coffee beans
5. Mix: a piece of spoiled raw chicken, mixed with some fresh vegetables
6. Rotten Chicken: a piece of spoiled raw chicken.

The specimens were placed in a non air-tight plastic containers along with a suspended board (Fig. 4.1).

4.3.3 Experiment and Results

The measurements of the specimens are shown in Figure 4.5. Tables 4.10 and 4.11 demonstrate the efficacy of the HiS-, and HeRS-Boot oversampling methods in both food spoilage and allergen detection. We assessed the performance of the algorithms based on six detection metrics, averaged over 500 trials, where TP is the true positives, TN the true negatives, FP the false positive, and FN the false negatives [74]:

Algorithm 3: Oversampling with the HS-, HiS, or HeRS-Boot

Step 1 Divide the sample domain into K cells.

Step 2 For each cell k, $k = 1, \ldots, K$, calculate the heterogeneity measure $H(k)$ and modify the resampling probability distribution $p_c(k)$ for each class c in cluster k (Table 4.9).

Step 3 Oversampling: Sample N_0 instances with replacement from Class 0 (non-target) and likewise N_0 instances from Class 1 (target) with the updated distribution p'_c.

1. Sensitivity (SN) or true positive rate (TPR) or recall (REC) = TP / (TP + FN)
2. Specificity (SP) or true negative rate (TNR) = TN / (TN + FP)
3. Accuracy (ACC) = (TP + TN) / (TP + TN + FN + FP)
4. Error rate (ERR) = (FP + FN) / (TP + TN + FN + FP)
5. Precision (PREC) = TP / (TP + FP)
6. F1 score (F1) = 2 * PREC * REC / (PREC + REC)

The performance of three classifiers: Naive Bayes, Quadratic Discriminant, and a Neural Network (with 2 hidden layers, with 8 and 4 fully connected outputs for each hidden layer, respectively) are presented. We see some improvements compared to the results presented in [29]. We also observe that in most cases, the sensitivity is not compromised while the other metrics are improved.

4.3.4 Conclusion and Future Work

We introduced HiS- and HeRS-Boot, two variations of HS-Boot oversampling to counter the imbalance in a dataset. The results indicate that the refinements introduced to the oversampling methods not only improve upon traditional training set selection methods but also successfully expand their application to new domains, such as allergen detection.

Table 4.8: Dummy example of $C = 2$ classes with respectively 6 and 8 sample points, and $K = 18$ cells.

	*				
	*	* *	*		
∘	∘	∘ ∘	∘	∘ ∘	∘
			*		

	.167						
	.167	.167	.167	.167			
.125	.125	.125	.125	.125	.125	.125	.125
			.167				

(a) $p_c\,(k)$

0	0	0	0	0	0
0	.099	.198	.099	0	0
0	0	0	0	0	0

0	0	0	0	0	0
0	.0495	.0495	.0495	0	0
0	0	0	0	0	0

(b) $H(k)$ for HSBoot and HiSBoot (left) and HeRSBoot (right)

	.1046						
	.167	.2287	.2287	.167			
.0783	.1404	.2024	.2024	.1404	.0783	.0783	.0783
				.1046			

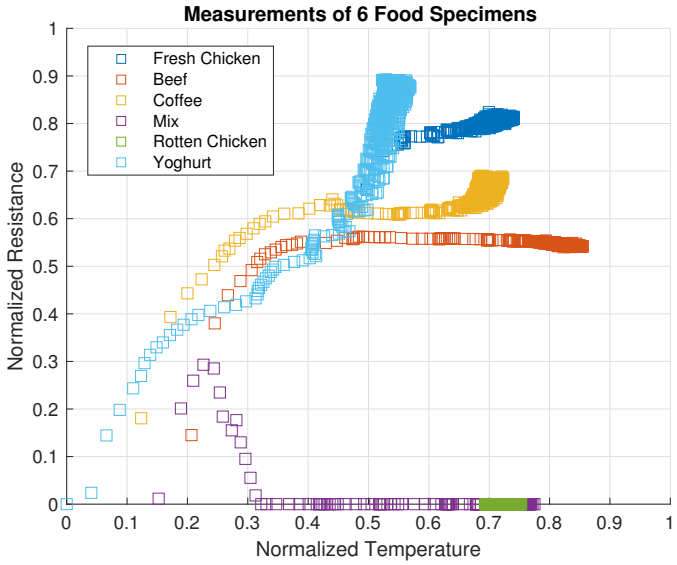
(c) $p'_c\,(k)$ HSBoot

	0.0443						
	.228	.228	.228	.228	.228		
.125	.125	.125	.125	.125	.125	.125	.125
				.0443			

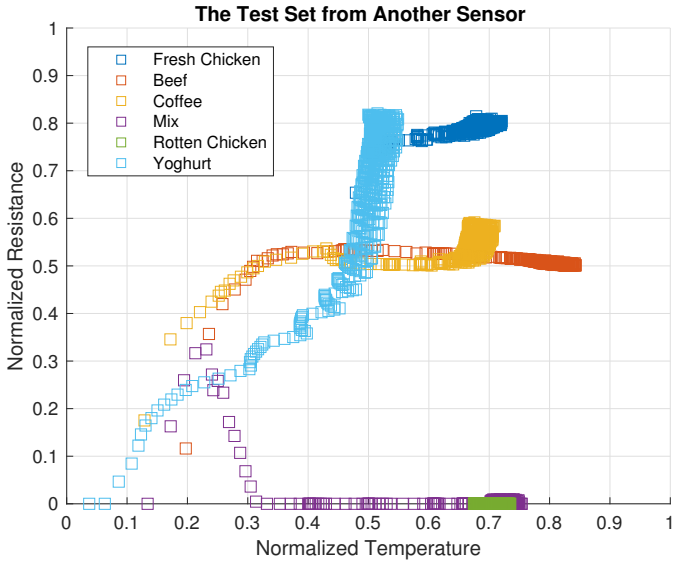
(d) $p'_c\,(k)$ HiSBoot

	.1394						
	.1807	.1807	.1807	.1807			
.1043	.1457	.1457	.1457	.1457	.1043	.1043	.1043
				.1394			

(e) $p'_c\,(k)$ HeRSBoot



(a) Training set allergen



(b) Test set allergen

Figure 4.5: The measurements of the six specimens collected from different sensors around the same time.

Table 4.9: Summary of the three algorithms, where γ is a hyperparameter constant that determines the influence of the heterogeneity, a_{ck} is the number of samples in cell k that belongs to class c, $\rho_{ck} = a_{ck}/a_k$ the ratio of members of class c in cell k, and the denominator is to make sure that the sum equals 1.

HSBoot, **HiSBoot**	**HeRSBoot**
$H(k) = -\sum_{c=1}^{C} \frac{a_{ck}}{N} \log\left(\frac{a_{c,k}}{\sum_{c=1}^{C} a_{c,k}}\right)$	$H(k) = -\sum_{c=1}^{C} \frac{\rho_{ck}}{K} \log\left(\rho_{c,k}\right)$

HSBoot	**HiSBoot**, **HeRsBoot**
$p'_c(k) = \frac{\frac{1}{N_c} + \gamma H(k)}{1 + \gamma \sum_k a_{ck} H(k)}$	$p'_c(k) = \begin{cases} \frac{\frac{1}{N_c} + \gamma H(k)}{1 + \gamma \sum_k a_{ck} H(k)} & \text{for target class} \\ p_c(k) & \text{for non target class} \end{cases}$

Table 4.10: **Spoiled Food** Detection Results

	Random	HiSBoot			HeRSBoot		
		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$
SN	.9899	.9885	.9922	.9902	.9911	**.9924**	**.9924**
SP	.9924	**.9953**	.9945	.9927	.9946	.9923	.9917
ACC	.9916	.9930	**.9937**	.9919	.9934	.9923	.9920
ERR	.0084	.0070	**.0063**	.0081	.0066	.0077	.0080
PREC	.9850	**.9906**	.9891	.9856	.9892	.9848	.9837
F1	.9872	.9894	**.9906**	.9877	.9901	.9885	.9879

(a) Naive Bayes

	Random	HiSBoot			HeRSBoot		
		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$
SN	.9968	.9941	.9948	.9959	.9944	.9961	**.9978**
SP	.9887	**.9932**	.9920	.9908	.9929	.9903	.9875
ACC	.9914	**.9935**	.9929	.9925	.9934	.9923	.9909
ERR	.0086	**.0065**	.0071	.0075	.0066	.0077	.0091
PREC	.9779	**.9866**	.9844	.9819	.9859	.9810	.9757
F1	.9872	**.9903**	.9895	.9888	.9901	.9885	.9866

(b) Logistic Regression

	Random	HiSBoot			HeRSBoot		
		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$
SN	**.9969**	.9957	.9960	.9962	.9961	.9965	.9963
SP	.9926	**.9941**	.9938	.9934	.9938	.9931	.9934
ACC	.9941	**.9946**	**.9946**	.9944	.9945	.9942	.9944
ERR	.0059	**.0054**	**.0054**	.0056	.0055	.0058	.0056
PREC	.9855	.9883	**.9878**	.9870	.9877	.9864	.9870
F1	.9912	**.9920**	.9919	.9916	.9918	.9914	.9916

(c) Tree

Table 4.10: **Spoiled Food** Detection Results (Continued)

	Random	HiSBoot			HeRSBoot		
		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$
SN	.9941	.9916	.9931	**.9953**	.9925	.9944	.9948
SP	.9913	**.9947**	.9930	.9901	.9938	.9911	.9917
ACC	.9922	**.9937**	.9930	.9918	.9933	.9922	.9928
ERR	.0078	**.0063**	.0070	.0082	.0067	.0078	.0072
PREC	.9831	**.9898**	.9863	.9807	.9879	.9826	.9837
F1	.9885	**.9906**	.9896	.9879	.9901	.9884	.9892

(d) Quadratic Discriminant

	Random	HiSBoot			HeRSBoot		
		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$
SN	**.9869**	.9303	.9446	.9309	.9208	.9531	.9352
SP	.9253	.9654	**.9664**	.9569	.9605	.9535	.9518
ACC	.9459	.9537	**.9591**	.9482	.9473	.9534	.9462
ERR	.0541	.0463	**.0409**	.0518	.0527	.0466	.0538
PREC	.9094	.9406	**.9429**	.9274	.9325	.9244	.9194
F1	.9449	.9652	**.9667**	.9579	.9607	.9562	.9536

0

(e) Neural Network with 2 hidden layers, with 8 and 4 fully connected outputs for each hidden layer, respectively.

Table 4.11: **Allergen Detection** Results

	Random	HiSBoot			HeRSBoot		
		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$
SN	.6053	.7992	.7583	.7128	.8122	**.8157**	.7823
SP	.9810	.9762	.9793	**.9816**	.9744	.9728	.9735
ACC	.9184	.9467	.9425	.9368	**.9474**	.9466	.9417
ERR	.0816	.0533	.0575	.0632	**.0526**	.0534	.0583
PREC	.8655	.8723	.8819	**.8877**	.8653	.8590	.8576
F1	.7020	.8296	.8117	.7870	**.8338**	.8325	.8140

(a) Naive Bayes

	Random	HiSBoot			HeRSBoot		
		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$
SN	**.7436**	.7055	.7196	.7195	.6984	.7428	.7342
SP	.9656	**.9704**	.9694	.9693	**.9708**	.9687	.9693
ACC	.9286	.9262	.9277	.9277	.9254	**.9310**	.9301
ERR	.0714	.0738	.0723	.0723	.0746	**.0690**	.0699
PREC	.8122	.8266	.8248	.8245	**.8275**	.8263	.8273
F1	.7752	.7601	.7665	.7669	.7567	**.7802**	.7763

(b) Logistic Regression

	Random	HiSBoot			HeRSBoot		
		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$
SN	**.8221**	.8097	.7826	.7805	.8115	.7984	.7927
SP	.9810	.9817	.9840	**.9845**	.9810	.9821	.9823
ACC	**.9545**	.9530	.9504	.9505	.9528	.9515	.9507
ERR	**.0455**	.0470	.0496	.9117	.0472	.0485	.0493
PREC	.8977	.8998	**.9083**	.8392	.8968	.9003	.9008
F1	**.8568**	.8510	.8397	.8392	.8506	.8451	.8418

(c) Tree

Table 4.11: **Allergen Detection** Results (Continued)

	Random	HiSBoot			HeRSBoot		
		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$
SN	.8057	.7591	.7931	.7509	.7503	.8310	**.8474**
SP	.9653	.9701	.9697	**.9762**	.9704	.9653	.9631
ACC	.9387	.9349	.9402	.9387	.9337	.9429	**.9438**
ERR	.0613	.0651	.0598	.0613	.0663	.0571	**.0562**
PREC	.8228	.8353	.8401	**.8647**	.8353	.8273	.8211
F1	.8141	.7952	.8154	.8030	.7904	.8289	**.8334**

(d) Quadratic Discriminant

	Random	HiSBoot			HeRSBoot		
		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$
SN	**.7252**	.6395	.6399	.6276	.6228	.6506	.6474
SP	.9620	.9805	.9811	**.9824**	.9801	.9768	.9764
ACC	.9225	.9237	**.9242**	.9233	.9205	.9225	.9216
ERR	.0775	.0763	**.0758**	.0767	.0795	.0775	.0784
PREC	.8315	.8681	.8707	**.8782**	.8627	.8496	.8465
F1	**.7737**	.7548	.7656	.7690	.7524	.7653	.7702

(e) Neural Network with 2 hidden layers, with 8 and 4 fully connected outputs for each hidden layer, respectively.

5 Complex Seasonal Bootstrap Methods

This chapter introduces a family of bootstrap procedures designed to address the challenges posed by time series that exhibit multiple and overlapping seasonal patterns. Classical block bootstrap methods assume a single dominant correlation structure and often treat temporal dependence as locally homogeneous. However, many real-world time series violate these simplifying assumptions. Data from energy systems, transportation networks, climatology, and sensor-based monitoring often contain several interacting seasonal components that vary in amplitude, phase, and persistence. These characteristics render conventional block bootstraps inadequate, as they struggle to replicate simultaneous daily, weekly, annual, or domain-specific cyclicities.

To meet these challenges, this chapter presents the *Complex Seasonal Circular Block Bootstrap (XSCBB)*, a generalisation of the circular block bootstrap that aligns the resampling mechanism with multiple seasonal structures. Instead of relying on fixed contiguous blocks, XSCBB constructs season-aware blocks that respect the combined periodicities present in the data. This allows the bootstrap samples to preserve dependencies across different seasonal scales and ensures that generated trajectories reflect realistic temporal dynamics. As such, XSCBB represents a significant step toward more flexible resampling tools for time series with rich and intertwined seasonal patterns.

While XSCBB effectively captures complex seasonality, many real-world applications also exhibit volatility fluctuations, heavy-tailed behaviour, or structural irregularities that demand greater robustness. To address these issues, the chapter introduces two extensions: the *Block Robust Estimates Standardized Influence Function - Complex Seasonal Circular Block Bootstrap (BRESIF-XSCBB)* and the *Block Average Complex Seasonal Circular Block Bootstrap (BA-XSCBB)*. These variants incorporate robust statistical principles, including influence functions and block averaging, to mitigate the effects of outliers or shifts in signal variability. The result is a set of seasonal bootstrap procedures capable of producing more stable uncertainty estimates in environments where noise distributions deviate from Gaussian assumptions or where measurement artefacts are present.

Throughout the chapter, the proposed methods are evaluated through simulations and applications to real-world time-series problems. The results show that incorporating multi-seasonality and robustness into the bootstrap framework improves the quality of

resampled trajectories and the reliability of inference. This chapter thus expands the bootstrap toolkit to better reflect the complexity of modern time series, offering adaptable, structure-aware methods that bridge the gap between theoretical resampling principles and practical forecasting and analysis needs.

5.1 The Complex Seasonal Circular Block Bootstrap (XSCBB)

We propose the XSCBB, a variation of seasonal (circular) block bootstrap that caters for multiple seasonality components in a time series. Electricity consumption (load) prediction is important to balance the supply and load demand, to plan facilities construction and maintenance, to plan distribution, and avoid outages or excess loss. We apply the XSCBB method parametrically to calculate the prediction interval of future electricity consumption given a relatively small amount of historical sample points using the composite ARMA(p, q) – GARCH(r, s) model.

5.1.1 Introduction

Short term and long term forecasting of electricity consumption (load) are essential for planning the infrastructure of an electrical power system. However, the electricity load data can be non-linear, non-stationary, and have a time-varying variance. Such characteristics can be attributed to the variation in the climate as well as the unpredictable shifts in consumer behaviors. However, there are also expected consistencies in the fluctuation of consumption such as during days versus nights or weekdays versus weekends. Such an effect is called seasonality and causes the data to be non-stationary.

Many existing forecasting methods do not only assume that the data to be stationary, but also that the variance to be constant over time. These two assumptions cannot be reasonably applied to the electricity consumption data for the reasons mentioned above. Therefore, we consider a model that may explain the time-varying volatility in the residuals, i.e. the Autoregressive Conditional Heteroskedasticity (ARCH) model [75] and its generalised version, Generalised ARCH (GARCH) model [76]. Some related applications include forecasting energy [77], sea surface temperature anomalies [78], electricity price [79], and wind power [80].

Electricity load forecasting had been performed with some variations of Autoregressive (AR) and Moving Average (MA) models that take seasonalities into account, such as the Seasonal Autoregressive Integrated Moving Average (SARIMA) model, with some degree of success [81], [82], [83]. One method to estimate the SARIMA parameters is by maximum likelihood estimation.

To construct a likelihood function, the probability distribution of the random observations need to be specified. However, often such distribution is not available. A quasi-likelihood function is the likelihood function constructed to infer about a parameter with insufficient information. The Quasi Maximum Likelihood (QML) can jointly parameterise conditional means, conditional variances, and conditional covariances [84].

The bootstrap method replaces an unknown distribution function by its empirical

estimator [13], [14]. Bootstrapping for a GARCH model had been studied in [22], [23], [24], [25], [26], [27], [85], [86].

The Seasonal Block Bootstrap (SBB) was proposed for bootstrapping a time series where a seasonality effect is present [15]. Consider observations $X_1, \ldots, X_N$ arising from a time series $\{X_t, t \in \mathbb{Z}\}$, with

$$X_t = \mu_t + Y_t \text{ and } \mu_t = \mu_{t-d},\ \forall t \in \mathbb{Z}, \tag{5.1}$$

where d is an integer denoting the period of the deterministic, unknown μ_t. Usually d is known or obvious from the data and $\{Y_t, t \in \mathbb{Z}\}$ a strictly stationary sequence with mean zero. If μ_t is not a constant and not stationary, the Block Bootstrap (BB) and its variations [2] is not directly applicable.

The SBB process can be summarised as follows:

- Assume the sample size $N = nd$ for some integer n
- Choose a positive integer $b < n$
- Draw i.i.d $i_0, i_1, \ldots, i_{k-1}$ with uniform distribution on the set $\{1, 2, \ldots, n - b + 1\}$
- We may take $k = [n/b]$ (different choices are also possible)
- The SBB constructs a bootstrap pseudo-series $X_1^*, X_2^*, \ldots, X_l^*$ where $l = kbd$: $X_{mbd+j}^* := X_{i_md+j-1}$ for $m = 0, 1, \ldots, k-1$ and $j = 1, 2, \ldots, bd$.
- SBB is a version of the BB with blocks size bd and starting points integer multiples of the period d: $i_0 d, i_1 d, \ldots, i_{k-1} d$.

The SBB poses a restriction on the relative size of the period and block size, where the blocks' size and starting points are restricted to be integer multiples of the period d. To resolve this restriction, Dudek *et al.* proposed the Generalised Seasonal Block Bootstrap (GSBB) and its circular version, GSCBB, [16], and the Generalised Seasonal Tapered Block Bootstrap (GSTBB).

The GSCBB can be summarised as follows:

- Choose a positive integer block size $b < N$, let $l = [N/b]$.
- For $t = 1, b + 1, 2b + 1, \ldots, lb + 1$, let

$$\left(X_t^*, X_{t+1}^*, \ldots, X_{t+b-1}^*\right) = \left(X_{k_t}, X_{k_t+1}, \ldots, X_{k_t+b-1}\right)$$

where k_t is a discrete uniform random variable taking values in the set

$$S_{t,N} = \{t - dR_{1,N}, t - d\left(R_{1,N} - 1\right), \ldots, t - d, t, \\ t + d, \ldots, t + d\left(R_{2,N} - 1\right), t + dR_{2,N}\}$$

with $R_{1,N} = \left[(t-1)/d\right]$ and $R_{2,N} = \left[(N-b-t)/d\right]$

- Join the $l+1$ blocks $(X_{k_t}, X_{k_t+1}, \ldots, X_{k_t+b-1})$ to obtain a new series of bootstrap pseudo-observations $X_1^*, X_2^*, \ldots, X_N^*, \ldots, X_{(l+1)b}^*$ from which only the first N points $X_1^*, X_2^*, \ldots, X_N^*$ are retained.

The illustrations of the SBB and the GSCBB are shown in Figures 5.1a and 5.1b, respectively.

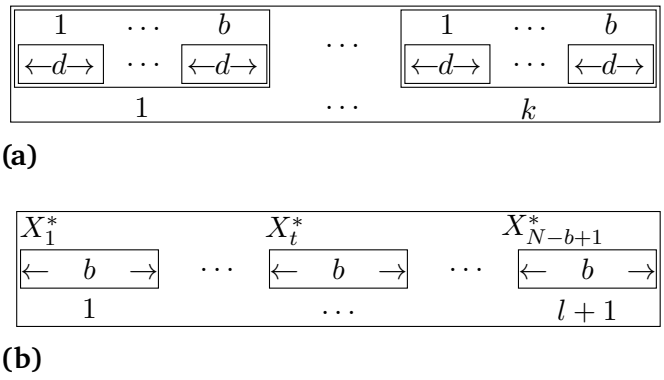


Figure 5.1: (a)The illustrations of the Seasonal Block Bootstrap (SBB) [15] and (b) the Generalised Seasonal Circular Bootstrap (GSCBB) [16]

The aforementioned methods deal with the assumption of a single known seasonality. We propose the Complex Seasonal Circular Block Bootstrap (XSCBB), that may take a signal with multiple seasonalities. Section 5.1.2 elaborates the proposed method and its application on a time series that can be modeled with the composite ARMA(p, q) – GARCH(r, s) model. Section 5.1.3 gives the results when the proposed method is applied to a real life data set including the comparison with those from a residual bootstrap method and the published results from the ENTSO-E Transparency Platform [87]. Lastly, Section 5.1.4 gives a brief conclusion and the future direction.

5.1.2 Methodology

Algorithm 4 describes the proposed XSCBB method. The application of interest is finding the prediction interval of a time series that can be modeled with ARMA(p, q) – GARCH(r, s) model. The summary of the parametric application of XSCBB is given in Algorithm 5.

Here we use the composite conditional mean and variance ARMA(p, q) – GARCH(r, s) model,

$$X_t = a_0 + \varepsilon_t + \sum_{i=1}^{p} a_i X_{t-i} + \sum_{j=1}^{q} \alpha_j \varepsilon_{t-j} \tag{5.2}$$

where the residual $\varepsilon_t = \sqrt{h_t}\eta_t$ and the conditional variance process has the form

$$h_t = b_0 + \sum_{j=1}^{s} b_j \varepsilon_{t-j}^2 + \sum_{i=1}^{r} \beta_i h_{t-i}, \tag{5.3}$$

$b_0 > 0, b_j \geq 0, \beta_i \geq 0$, and the innovations $\{\eta_t\}$ are independent and identically distributed (i.i.d) random variables such that $\mathrm{E}[\eta_t] = 0$, $\mathrm{E}\left[\eta_t^2\right] = 1$, follow a symmetric distribution $\mathrm{E}\left[\eta_t^3\right] = 0$, and $\mathrm{E}\left[\eta_t^4\right] < \infty$.

Let $\boldsymbol{\theta} = (a_0, \ldots, a_p, \alpha_1, \ldots, \alpha_q, b_0, \ldots, b_s, \beta_1, \ldots, \beta_r)'$.

The log likelihood function for a set of N observations is

$$L_N(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=0}^{N-1} l_{t-i}(\boldsymbol{\theta}) \tag{5.4}$$

where

$$l_t(\boldsymbol{\theta}) = -\frac{1}{2} \log h_t(\boldsymbol{\theta}) - \frac{\varepsilon_t^2}{2h_t}(\boldsymbol{\theta}). \tag{5.5}$$

A Quasi Maximum Likelihood (QML) estimator $\hat{\boldsymbol{\theta}}$ is any measurable solution of

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Phi}} L_N(\boldsymbol{\theta}), \tag{5.6}$$

with the parameter space $\boldsymbol{\Phi} = \boldsymbol{\Phi_a} \times \boldsymbol{\Phi_b}$, where $\boldsymbol{\Phi_a} \subset \mathbb{R}^{p+q+1}$, $\boldsymbol{\Phi_b} \subset \mathbb{R}_0^{s+r+1}$, $\mathbb{R} = (-\infty, \infty)$, and $\mathbb{R}_0 = [0, \infty)$ [84].

Since the predictions are done after deseasonalising, the original seasonalities need to be added back. Let $A(L)$ be the lag operator polynomial,

$$\begin{aligned} A(L) &= \left(1 - L^{d_1}\right) \cdots \left(1 - L^{d_M}\right) \\ &= 1 + \phi_1 L^1 + \ldots + \phi_\kappa L^\kappa, \quad \phi_{1,\ldots,\kappa} \in \{-1, 1\}, \end{aligned} \tag{5.14}$$

Algorithm 4: The XSCBB

Step 1 Given the multiple seasons $d_1, \ldots, d_M$, calculate their least common multiple $d = \text{lcm}\,(d_1, \ldots, d_M)$.

Step 2 Choose an integer b such that the resampled block length is $b \cdot d$. There are $k = \lceil N/(b \cdot d) \rceil$ blocks of samples that will be "stitched" together, where $\lceil \cdot \rceil$ is the ceiling notation.

Step 3 For $j = 0, \ldots, k-1$, let

$$\begin{aligned}\left(X^*_{jbd+1}, X^*_{jbd+2}, \ldots, X^*_{(j+1)bd-1}\right) \\ = \left(X_{i_j}, X_{i_j+1}, \ldots, X_{i_j+bd-1}\right)\end{aligned} \tag{5.7}$$

where i_j is a discrete uniform random variable taking values in the set

$$\begin{aligned}\mathcal{S} = \{\tau, \phi(\tau + d), \phi(\tau + 2d), \ldots, \\ \phi(\tau + \lceil N/d \rceil d)\}\end{aligned} \tag{5.8}$$

where $1 \leq \tau \leq N$, and

$$\phi(t) = \begin{cases} t, & t \leq N \\ t \mod N, & t > N. \end{cases} \tag{5.9}$$

Step 4 Join the k blocks to obtain a new series of bootstrap pseudo-observations $X^*_1, X^*_2, \ldots, X^*_{kbd}$, $kbd \geq N$, and retain only the first N points, $X^*_1, X^*_2, \ldots, X^*_N$.

then the inverse seasonal difference operator is

$$A^{-1}(L) = 1 - \phi_1 L^1 - \ldots - \phi_\kappa L^\kappa. \tag{5.15}$$

The predicted value in the original domain is then

$$\begin{aligned}\hat{x}_{t+1} = &[-\phi_\kappa, -\phi_{\kappa-1}, \ldots, -\phi_1, 1]^\intercal \\ &[x_{t-\kappa+1}, x_{t-\kappa+2}, \ldots, x_t, \tilde{x}_{t+1}]\,.\end{aligned} \tag{5.16}$$

For the application, we used the hourly electricity consumption in Megawatt from TransnetBW, one of four transimission grid operators in Germany. The sample points

are hourly from January 1st 2015 until October 1st 2020 (Figures 5.2a, 5.2b). We calculated the one-hour-ahead and one-day-ahead predictions. For both cases, the number of historical sample points used to estimate the parameters is $N = 840$. We compared the forecasted values to the day-ahead load forecast in TransnetBW as published on the ENTSO-E Transparency Platform. The dataset is publicly available in Open Power System Data [87]. We also compared the prediction intervals obtained by XSCBB to those obtained by a residual bootstrap method [88] that can be summarised as follows:

- Deseason the past N samples similarly to Step 2 of Algorithm 5.
- Obtain the QML estimator $\hat{\boldsymbol{\theta}}$, calculate the residuals $\hat{\varepsilon}_\tau$, variance $\hat{h}_\tau$ and the standardised residuals $\tilde{\varepsilon}_\tau$ in a similar way to Equations (5.11)-(5.13), except with the original deseasonalised observations.
- Resample $\tilde{\varepsilon}_\tau$ with replacement to obtain $\tilde{\varepsilon}^\star_\tau,\ \tau = t-N+1,\dots,t$. With $\hat{\boldsymbol{\theta}}$, x_t, $\hat{h}_\tau$, and $\tilde{\varepsilon}^\star_\tau$ as the disturbance path, estimate the predicted value at time $t+1$, $\tilde{x}$, according to Eq. (5.2).
- Repeat B times to obtain the bootstrapped predictions, and the rest is as in Step 6 and 7 of Algorithm 5.

5.1.3 Experiment and Results

The deseasonalised sample is depicted in Fig. 5.2c. The Engle's test [75] to the resulting residuals indicated that they are heteroskedastic. Both the Akaike Information Criterion (AIC) [22] and Bayesian Information Criterion (BIC) [89] confirmed the most parsimonious model to be ARMA$(1,1)$ – GARCH$(1,1)$.

As shown in Figures 5.3 and 5.4, the prediction intervals of XSCBB are consistently narrower, which indicates that it is more stable than the residual bootstrap method. Due to the narrower intervals, there is naturally a trade-off in the coverage probability (CP) as shown in the results in Table 5.1. Nevertheless, the Root Mean Square Errors (RMSE) and the Mean Absolute Percentage Error (MAPE) of the proposed method are consistently better than those of the residual bootstrap and the published predictions by ENTSO-E Transparency Platform.

5.1.4 Conclusion and Future Work

We proposed the XSCBB, a variation of the seasonal block bootstrap that may take multiple seasonalities, and its parametric application to a time series that may be modeled by the

composite conditional mean and variance ARMA(p, q) – GARCH(r, s). As compared to the residual bootstrap method, the proposed XSCBB is more stable, as evident from the narrower prediction intervals. The accuracy metrics also show the XSCBB is better than the other 2 methods.

In our example, the block size $b = 2$ was arbitrarily determined. In the future, this value should be optimised, such that the coverage percentage and the accuracy can be further improved.

Algorithm 5: Parametric XSCBB for ARMA(p,q) – GARCH(r,s) Model

Step 1 With N past observations $\{X_\tau, \tau = t-N+1,\ldots,t\}$, obtain the bootstrapped observations $X_1^*, X_2^*, \ldots, X_N^*$ using the XSCBB.

Step 2 Deseasonalising of the bootstrapped sample: Let L be the lag (backshift) operator such that $L^j x_\tau = x_{\tau-j}$. The stationary (deseasonalised) bootstrap observations $\tilde{X}_1^*, \tilde{X}_2^*, \ldots, \tilde{X}_N^*$ are obtained by multiplying them with the suitable lag operator polynomials,

$$\tilde{X}_\tau^* = \left(1 - L^{d_1}\right) \cdots \left(1 - L^{d_M}\right) X_\tau^*. \tag{5.10}$$

Step 3 With $\tilde{X}_1^*, \tilde{X}_2^*, \ldots, \tilde{X}_N^*$, obtain the QML estimator of the parameters $\hat{\boldsymbol{\theta}}^* = \left(\hat{a}_0^*, \ldots, \hat{a}_p^*, \hat{\alpha}_1^*, \ldots, \hat{\alpha}_q^*, \hat{b}_0^*, \ldots, \hat{b}_s^*, \hat{\beta}_1^*, \ldots, \hat{\beta}_r^*\right)$ and calculate the residuals

$$\hat{\varepsilon}_\tau^* = \tilde{X}_\tau - \hat{a}_0^* - \hat{\varepsilon}_0^* - \sum_{i=1}^{p} \hat{a}_i^* \tilde{X}_{\tau-i} - \sum_{j=1}^{q} \hat{\alpha}_j^* \hat{\varepsilon}_{\tau-j}^*, \tag{5.11}$$

$\tilde{X}_\tau$ being the deseasonalised original signal equivalent to Eq. 5.26, $\tau = t-N+1,\ldots,t$.

Step 4 Compute the variance

$$\hat{h}_\tau^* = \hat{b}_0^* + \hat{b}_1^* \hat{\varepsilon}_{\tau-1}^{*2} + \hat{\beta}_1^* \hat{h}_{\tau-1}^*, \quad \tau = t-N+1,\ldots,t \tag{5.12}$$

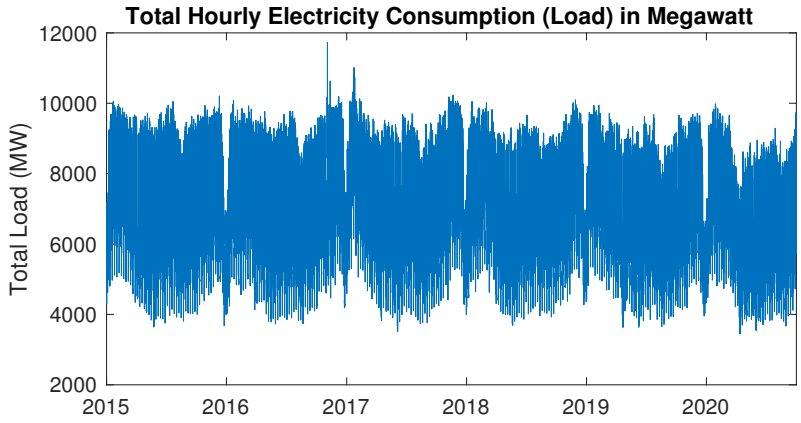
and the standardised residuals

$$\tilde{\varepsilon}_\tau^* = \frac{\hat{\varepsilon}_\tau^*}{\sqrt{\hat{h}_\tau^*}}, \ \tau = t-N+1,\ldots,t. \tag{5.13}$$

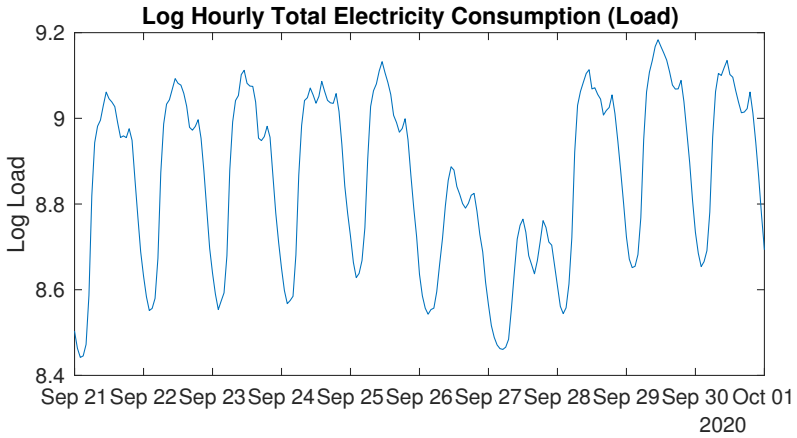
Step 5 With $\hat{\boldsymbol{\theta}}^*$, $\hat{h}_\tau^*$, and $\tilde{\varepsilon}_\tau^*$ as the disturbance path, estimate the predicted value at time $t+1$, $\tilde{x}$.

Step 6 Repeat **Step 1-5** B times to obtain the bootstrapped predictions, $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_B$ and sort these into $\tilde{x}_{(1)} \leq \tilde{x}_{(2)}, \ldots \leq \tilde{x}_{(B)}$.

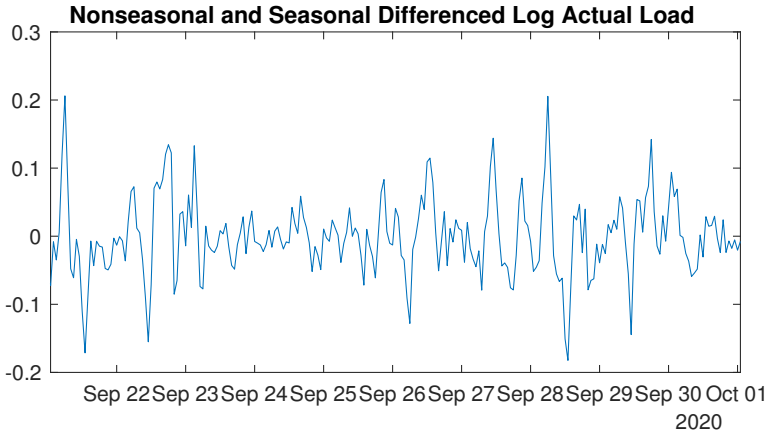
Step 7 Obtain the $100\,(1-\alpha)\,\%$ confidence interval from $\tilde{x}_{(1)} \leq \tilde{x}_{(2)}, \ldots \leq \tilde{x}_{(B)}$ and the estimated point prediction from the average $\sum_{j=1}^{B} \tilde{x}_j / B$, or the median $\text{Med}\,(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_B)$.



(a)



(b)



(c)

Figure 5.2: (a) The full dataset of hourly electricity consumption in Megawatt (MW), (b) the natural log of the last 10 days (240 hours) of the dataset, and (c) the corresponding non-seasonal and multiple seasonal differenced log sample.

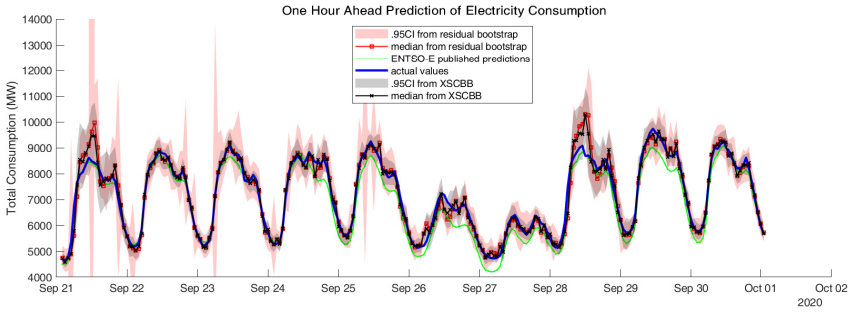


Figure 5.3: One hour ahead predictions from hourly training sample with $N = 840$, $b = 2$, $B = 500$, $d = \text{lcm}(1, 7, 24) = 168$, noting that $d_1 = 1$ will remove the linear trend in the time-series.

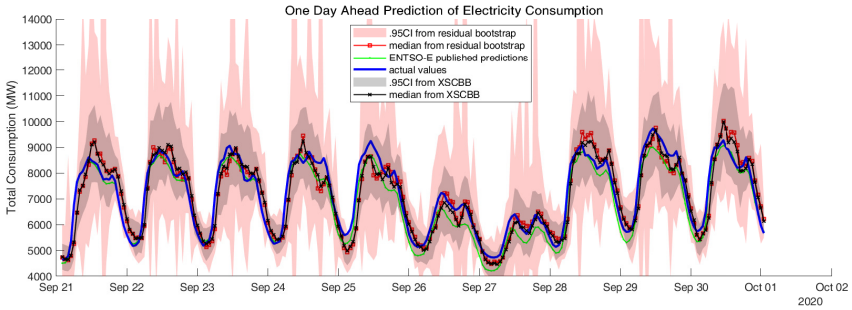


Figure 5.4: One day ahead predictions from hourly training sample with $N = 840$, $b = 2$, $B = 500$, $d = \text{lcm}(1, 7, 12) = 84$.

Table 5.1: Prediction metrics for the experiment setting as described in Figures 5.3 and 5.4.

		RMSE (MW)		MAPE (%)	
	CP	Median	Mean	Median	Mean
Residual Bootstrap	**96.25**	330.38	384.10	2.85	2.98
XSCBB	95	**271.749**	**325.04**	**2.48**	**2.47**

(a) Hour-ahead-predictions

		RMSE (MW)		MAPE (%)	
	CP	Median	Mean	Median	Mean
Residual Bootstrap	**97.9**	368.64	366.83	3.71	3.72
XSCBB	90.42	**323.15**	**325.04**	**3.43**	**2.97**

Published RMSE: 375.12 MW, MAPE: 4.52%

(b) Day-ahead-predictions

5.2 The Block Robust Empirical Standardised Influence Function (BRESIF)-XSCBB and the Block Average (BA)-XSCBB

We propose two robust seasonal bootstrap techniques, based on our prior work, the XSCBB. These robustifications are implemented through two methods, the Block Robust Empirical Standardised Influence Function (BRESIF or Block RESIF) and the Block Average XSCBB (BA-XSCBB) methods. The BRESIF-XSCBB method employs an adaptive weighting function to adjust the XSCBB technique for time series with heavy-tailed distributions or extreme values. Conversely, the BA-XSCBB strategy seeks to reduce the influence of outliers by incorporating either the block average or median to restore the seasonal characteristics. Through simulation studies, we showcase the efficacy of these methods in enhancing the XSCBB's performance when outliers are present. Our results indicate that these robustified versions of XSCBB not only preserve the original method's capacity to detect complex seasonal patterns but also significantly enhance resilience and accuracy in challenging data scenarios.

5.2.1 Introduction

Bootstrap methods provide a powerful statistical tool for estimating key characteristics from small data sets. By repeatedly resampling the original dataset with replacement, these techniques can approximate the distribution of a statistic of interest [12], [13], [14]. The Stratified Bootstrap approach refines the standard bootstrap method by altering the resampling probability distribution to better reflect the structure within the dataset. This modification involves dividing the dataset into distinct strata or groups. Several variations of the stratified bootstrap technique have been developed, each has demonstrated that these variations can enhance the effectiveness of established detection and estimation methodologies [29], [30], [40]

The Seasonal Block Bootstrap (SBB) method [15] and its variations [16], [17], [31] are resampling techniques that address the challenges of analyzing time series data that exhibits seasonal patterns. Such patterns are common in various fields, including economics, environmental science, and meteorology, where variables can display regular fluctuations that repeat over a defined period, such as weekly, monthly, or annually. In integrating the stratified and seasonal block bootstrap methods, each resampling block, designed to retain the seasonal cycle, is regarded as a distinct stratum.

Outliers in time series data represent unexpected and unexplained deviations that introduce additional complexity beyond what typical analysis predicts. These may manifest as sudden spikes, crashes, gaps, and days of extreme data fluctuations. Such outliers

can significantly undermine the validity, accuracy, and reliability of statistical estimation. Addressing these challenges often requires the application of robust statistical methods, effective outlier detection techniques, and meticulous model selection and specification. This approach ensures that the impact of outliers on the estimation process is adequately managed. In this context, our work seeks to enhance our previous work, the Complex Seasonal Circular Block Bootstrap (XSCBB) [31], by incorporating methodologies that robustify the analysis of time series with multiple seasonal patterns and extreme data points.

The structure of this section is as follows. In Subsection 5.2.2, we present the formulation of the problem at hand. Following this, Subsection 4.3.2 explains the proposed methodologies, including the Block RESIF and the Block Average-XSCBB, which are discussed in 5.2.3 and 5.2.3, respectively. Subsection 5.2.4 describes the experimental setup and discusses the obtained results. Finally, Subsection 5.2.5 summarises the main findings and suggest directions for future research.

5.2.2 Problem Formulation

We are analyzing a time series characterised by multiple seasonal patterns, often referred to as complex seasonalities [90, Chapter 11],

$$X_t = X_t^{(1)} + \ldots + X_t^{(M)}, \quad \text{where} \tag{5.17}$$

$$X_t^{(m)} = X_{t-d_m}^{(m)}, \tag{5.18}$$

$$t \in \mathbb{Z}, m = 1, \ldots, M.$$

The observed signal may have been affected by random fluctuations $\eta_t \sim \mathcal{N}(0, \sigma_\eta)$ and outliers at specific times t_i, each with a magnitude $A_{t_i} \sim \mathcal{N}(\mu_{\text{out}}, \sigma_{\text{out}})$,

$$Y_t = X_t + \eta_t + \sum_{i=1}^{Q} A_{t_i} \delta(t - t_i), \quad t = 1, \ldots, N, \tag{5.19}$$

where $Q = Np_{\text{out}}/100$ represents the total count of outlier points, p_{out} is the outlier percentage with respect to the total number of data points N in the dataset, t_i, $i = 1, \ldots, Q$, denotes the time instances at which outliers are observed, and $\delta(t - t_i)$ is Kronecker's delta, which equals 1 if $t = t_i$ and 0 otherwise. The Signal to Noise Ratio (SNR) in dB is defined as $\text{SNR}_{\text{dB}} = 10 \log_{10}(P_X/\sigma_\eta^2)$, where P_X is the power of the ideal time series X.

Assume that the multiple seasonalities $d_1, \ldots, d_M$ can be reliably deduced from the sample, and their least common multiple (lcm) is $d = \text{lcm}\,(d_1, \ldots, d_M)$. In the XSCBB

method [31], the block length can be chosen to be an integer multiple of d, say bd, $b \in \mathbb{Z}^+$. Suppose that N observations are available, $\mathcal{Y} = \{Y_1, \ldots, Y_N\}$. To simplify the notations and without loss of generality, let N be an integer multiple of the chosen block length $N = kbd$, $k \in \mathbb{Z}^+$. The XSCBB [31] is summarised in Algorithm 6. Our objective is to have a reliable estimate of the future values of X_t, $t = N+1, N+2, \ldots$.

Algorithm 6: The Complex Seasonal Circular Block Bootstrap (XSCBB) [31]

Step 1 Given the multiple seasons $d_1, \ldots, d_M$, calculate the least common multiple $d = \text{lcm}\,(d_1, \ldots, d_M)$.

Step 2 With the block length $b \cdot d$, there are $k = N/(b \cdot d)$ blocks of samples. Each block is

$$\begin{aligned}\left(Y^*_{jbd+1}, Y^*_{jbd+2}, \ldots, Y^*_{(j+1)bd-1}\right) \\ = \left(Y_{i_j}, Y_{i_j+1}, \ldots, Y_{i_j+bd-1}\right),\end{aligned} \tag{5.20}$$

$j = 0, \ldots, k-1$, where i_j is a discrete uniform random variable taking values in the set

$$\begin{aligned}\mathcal{S} = \{\tau, \Phi(\tau + d), \Phi(\tau + 2d), \ldots, \\ \Phi(\tau + (k-1)d)\}\end{aligned} \tag{5.21}$$

where $1 \leq \tau \leq N$, and

$$\Phi(t) = \begin{cases} t, & t \leq N \\ t \mod N, & t > N. \end{cases} \tag{5.22}$$

Step 3 Join the k blocks to obtain a new series of bootstrap pseudo-observations $Y^*_1, Y^*_2, \ldots, Y^*_{kbd}$.

5.2.3 Methodology

In this section, we explore our two proposed methods, the Block Robust Estimates Standardized Influence Function (BRESIF) and the BA-XSCBB, in more detail.

Block RESIF XSCBB

Assume that the actual distribution of the data is in a contaminated vicinity around a central model $F_{\boldsymbol{\theta}}$. With $\mathcal{Y} = (Y_1, \ldots, Y_N)$ the available sample and $\hat{\Theta}$ an estimator for the parameter $\boldsymbol{\theta}$, the Standardised Influence Function (SIF) [18] of $\hat{\Theta}$ is defined as

$$\begin{aligned} &\text{SIF}\left(\mathcal{Y}; \hat{\Theta}, F_{\boldsymbol{\theta}}\right) = \\ &\left[\text{InfluenceFunction(IF)}\left(\mathcal{Y}; \hat{\Theta}, F_{\boldsymbol{\theta}}\right)^{\mathsf{T}} \mathrm{V}^{-1}\left(\hat{\Theta}, F_{\boldsymbol{\theta}}\right) \text{IF}\left(\mathcal{Y}; \hat{\Theta}, F_{\boldsymbol{\theta}}\right)\right]^{1/2}, \end{aligned} \tag{5.23}$$

where IF is the theoretical influence function [19] and $\mathrm{V}\left(\hat{\Theta}, F_{\boldsymbol{\theta}}\right)$ is the asymptotic variance of $\hat{\Theta}$. The Robust Estimates Standardized Influence Function (RESIF) [18] was defined by plugging in the estimates of the unknown parameters $\boldsymbol{\theta}$ that shape $F_{\boldsymbol{\theta}}$, from either a non-robust or robust estimator. The proposed Block RESIF (BRESIF) uses (non-robust) standardised residuals for outlier detection, based on a central model with a standard normal distribution.

The calculation of BRESIF for each block j is as follows:

$$\text{BRESIF}_j = \frac{1}{\sqrt{bd}} \sum_{i=1}^{bd} \tilde{\varepsilon}_{(j-1)bd+i}, \quad j = 1, \ldots, k, \tag{5.24}$$

where each $\tilde{\varepsilon}$ is an *appropriate* standardised residual estimate.

To demonstrate the application of BRESIF in complex seasonal forecasting models, we consider a simplification of electricity consumption patterns as follows. Following the (ideal) signal model in Eq. (5.18), let $X_t^{(1)}$ be a constant ρ, $X_t^{(2)} = D_t$ be a daily seasonality component, and $X_t^{(3)} = W_t$ be a weekly seasonality component,

$$X_t = \rho + D_t + W_t. \tag{5.25}$$

Let L be the lag (backshift) operator such that $L^j y_t = y_{t-j}$. The stationary (deseasonalised) observations $\tilde{Y}_1, \tilde{Y}_2, \ldots, \tilde{Y}_N$ are obtained by multiplying $Y_1, \ldots, Y_N$ with the suitable lag operator polynomials that correspond to the linear, daily, and weekly trends at hourly spot:

$$\tilde{Y}_t = \left(1 - L^1\right)\left(1 - L^{24}\right)\left(1 - L^{24\times 7}\right) Y_t. \tag{5.26}$$

The resulting deseasonalised component is thus a random fluctuation centered around zero (for example as shown in Fig. 5.5).

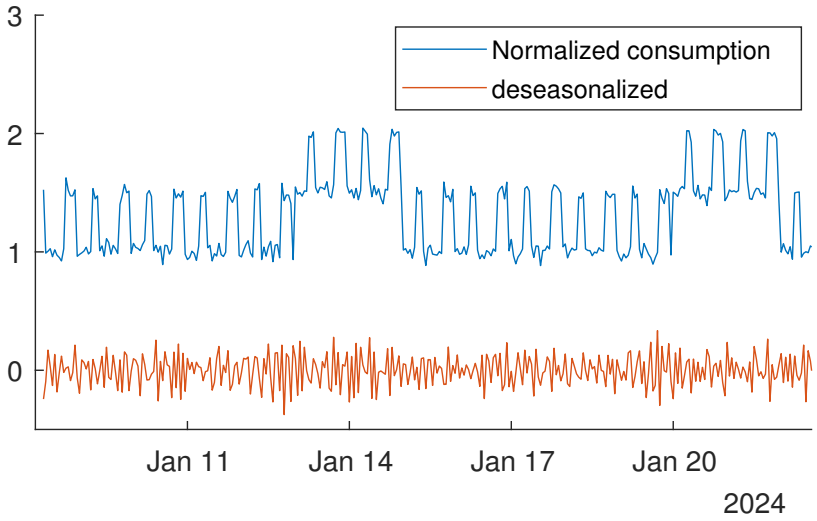


Figure 5.5: Simulated data with daily and weekly seasonalities at SNR of 30 dB and the deseasonalised version, no outliers added.

An estimate of $\tilde{\rho}$, denoted as $\hat{\tilde{\rho}}$, can be calculated by taking the average over $\tilde{Y}_t$, thus the appropriate standardised residual is

$$\tilde{\varepsilon}_t = \left(\tilde{Y}_t - \hat{\tilde{\rho}}\right) / \hat{\sigma}_\eta. \tag{5.27}$$

with $\hat{\sigma}_\eta$ being an estimate of σ_η from $\tilde{Y}_t$.

Similarly to the weights distribution proposed in [18], our proposed block weights are calculated as follows:

$$\begin{aligned} w_j =& \mathbb{I}_{[0,c]}\left(|\text{BRESIF}_j|\right) \\ &+ \psi\left(c, |\text{BRESIF}_j|\right) \times \mathbb{I}_{[c,+\infty]}\left(|\text{BRESIF}_j|\right), \\ & j = 1, 2, \ldots, k, \end{aligned} \tag{5.28}$$

where $\mathbb{I}$ is the indicator function with value of 1 inside the interval and 0 otherwise, c is a tuning constant, ψ a non-negative function with the following conditions [18]:

1) $\lim_{t\to+\infty} t^2\psi\left(c,t\right) = 0$

2) $\frac{\partial}{\partial t}\psi\left(c,t\right)|_{t=c} = 0$.

We chose the function ψ following [18],

$$\psi(x; c, \sigma, \gamma) = \begin{cases} \left[1 + \frac{(x-c)^2}{\gamma\sigma^2}\right]^{-\frac{\gamma+1}{2}} & 0 < \gamma < \infty \\ \exp\left[-\frac{(sx-c)^2}{2\sigma^2}\right] & \gamma = \infty, \end{cases} \tag{5.29}$$

with c being the location parameter, σ the scale parameter, and γ a shape parameter. Condition 1) is satisfied if $\gamma > 1$. We also set $\sigma = c$ to reduce the number of parameters. Then, the BRESIF-XSCBB can be summarised as listed in Algorithm 7.

Algorithm 7: Proposed Method: BRESIF-XSCBB

Step 1 With the deseasonalised observations $\tilde{Y}_1, \ldots, \tilde{Y}_N$, obtain the standardised residuals $\tilde{\varepsilon}_t$ following Eq. (5.27).

Step 2 Select $k = N/(bd)$ blocks of length bd following the indices selection in equations (5.21)–(5.22). Note that the starting indices do not have to be an integer multiple of bd and the blocks may have overlapping instances.

Step 3 For each block $j = 1, \ldots, k$, calculate BRESIF$_j$ according to Eq. (5.24) and the block weights according to Eq. (5.28).

Step 4 Calculate the block probabilities $\boldsymbol{P} = (p_1, p_2, \ldots, p_k)$ where $p_j = w_j / \sum_{i=1}^{k} w_i$, $j = 1, \ldots, k$.

Step 5 Obtain the bootstrapped observations $\mathcal{Y}^* = Y_1^*, Y_2^*, \ldots, Y_N^*$ by resampling with replacement and stitching together the k blocks following the probability distribution $\boldsymbol{P}$.

Step 6 Repeat **Step 5** to obtain B sample sets $\mathcal{Y}_1^*, \ldots, \mathcal{Y}_B^*$ that will be used in the prediction step.

Block Average – XSCBB

Previously [31], we proposed the prediction in the original domain to be obtained from

$$\begin{aligned} \hat{x}_{t+1} = &[-\phi_\kappa, -\phi_{\kappa-1}, \ldots, -\phi_1, 1]^\intercal \cdot \\ &[y_{t-\kappa+1}, y_{t-\kappa+2}, \ldots, y_t, \tilde{x}_{t+1}]\,. \end{aligned} \tag{5.30}$$

where $\phi_k, k = 1, \ldots, \kappa$ correspond to the coefficients in the lag operator polynomial,

$$\begin{aligned}\Pi(L) = &\left(1 - L^{d_1}\right) \cdots \left(1 - L^{d_M}\right) \\ =&1 + \phi_1 L^1 + \ldots + \phi_\kappa L^\kappa, \\ &\phi_{1,\ldots,\kappa} \in \{-1, 1\},\end{aligned} \tag{5.31}$$

with its inverse seasonal difference operator

$$\Pi^{-1}(L) = 1 - \phi_1 L^1 - \ldots - \phi_\kappa L^\kappa. \tag{5.32}$$

Let I be the random index shift, $0 \leq I < b \cdot d$, implied in Eq. (5.21) when randomly choosing for $\tau \leq N$. Recall that generally $kbd \geq N$ hence the bootstrapped sample $Y_1^*, \ldots, Y_{kbd}^*$ is truncated to $Y_1^*, \ldots, Y_N^*$ to retain its original sample size. The one-step prediction corresponds to the time index $N + 1$. To robustify the prediction in the original domain, we propose to replace Eq. (5.30) such that the prediction is now

$$\begin{aligned}\hat{x}_{t+1} = &[-\phi_\kappa, -\phi_{\kappa-1}, \ldots, -\phi_1, 1]^\intercal \cdot \\ &\left[\bar{Y}_{\Psi(\theta-\kappa+1)}, \bar{Y}_{\Psi(\theta-\kappa+2)}, \ldots, \bar{Y}_{\Psi(\theta-1)}, \bar{Y}_\theta, \tilde{x}_{t+1}\right],\end{aligned} \tag{5.33}$$

where

$$\Psi(u) = \begin{cases} u & u \mod (bd) = 0 \\ u \mod (bd) & \text{otherwise,} \end{cases} \tag{5.34}$$

and the index θ corresponds to

$$\theta = \begin{cases} bd & (N - I) \mod (bd) = 0 \\ (N - I) \mod (bd) & \text{otherwise,} \end{cases} \tag{5.35}$$

and $\bar{Y}$ may denote the positional block average or alternatively the median. This idea is illustrated in Fig. 5.6.

5.2.4 Experiment and Results

In the first part of the experiment, we generated a hypothetical normalised (unit-free) hourly electricity consumption with base consumption of 1 and increased by 0.5 everyday during busy hours between 6 to 8 am and between 6 to 10 pm as well as on the weekends. The dataset was generated for the whole January and February months, with 24 hours $\times$ 60 days $= 1440$ instances. The block length with $b = 1$ and $d = \text{lcm}(1, 24, 24 \times 7) = 168$ is $bd = 168$.

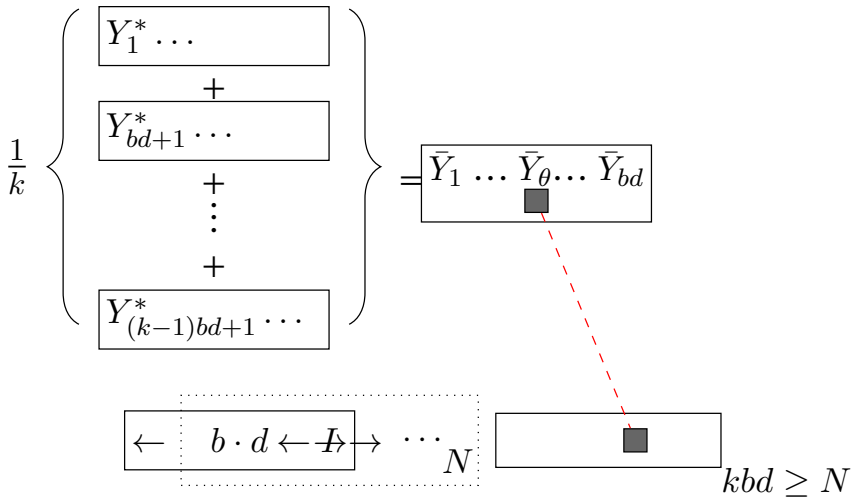


Figure 5.6: The illustration of Block Average XSCBB.

To test the robustness of the proposed method, the simulated signal is corrupted by Gaussian noise with SNR of -20 up to 20 dB. Additive outliers with magnitude centered around 5 were also added at the amount of $\{0, 0.1, 1, 5, 10, 20\}$ percent of the dataset. The number of bootstrap resampling is $B = 1000$.

The values of the hyperparameters in Eq. (5.29) are $c = \sigma = 1.645$ and $\gamma = 4$. These values can be understood through Fig. 5.7. As illustrated, the design decision of c follows the 90th percentile of the absolute standard normal distribution, while the value of γ that decides the shape of the weighting function was chosen arbitrarily.

Table 5.2 lists the Mean Absolute Error (MAE) of the base value predictions from the proposed BRESIF-XSCBB method in comparison with those obtained by simply averaging and robust fitting with Huber and Tukey's biweight (or bisquare) methods [20], [21]. The first column, the proposed method with the short-hand **B-X**, gives the MAEs after the original sample was resampled 1000 times. This comparison highlights the improvement introduced by resampling the data set by BRESIF-XSCBB method.

In the next performance analysis for BA-XSCBB, we used a publicly available dataset from the Open Power System Data [87], more precisely the hourly electricity consumption in Megawatt from TransnetBW, one of the transmission grid operators in Germany. We computed the one-step ahead all the way to ten-steps ahead predictions. We compare the performance of the proposed BA-XSCBB, the XSCBB, and the residual bootstrap method [88]. For all methods, the number of bootstrap resamples is $B = 100$ and the number of

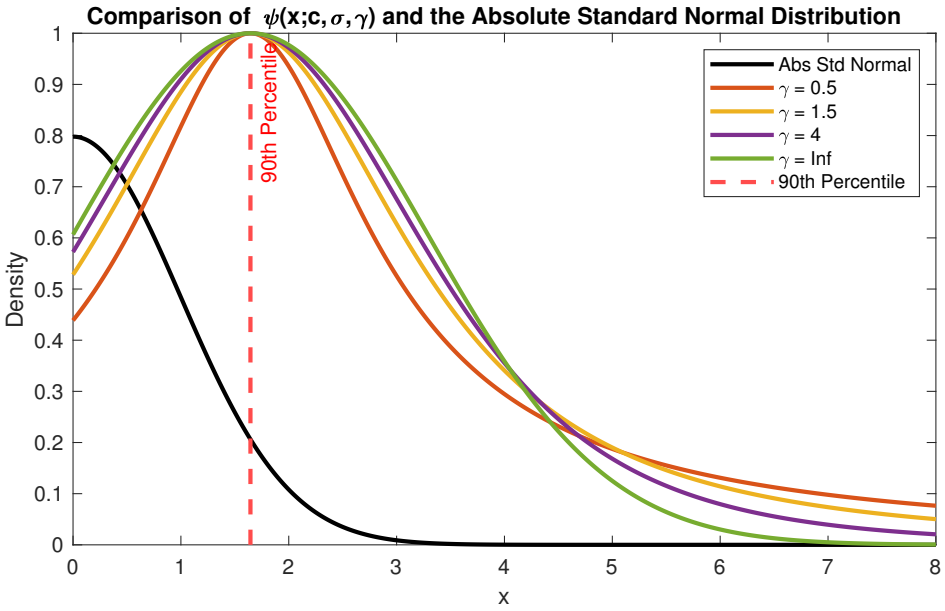


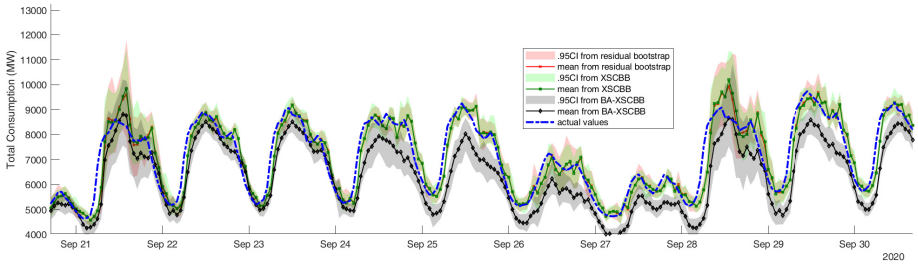
Figure 5.7: Comparison of $\psi(x; c, \sigma, \gamma)$ and the absolute Standard Normal distribution.

Table 5.2: Performance comparison in the Mean Absolute Deviation (MAD) of the estimates by the proposed method, BRESIF-XSCBB (short-hand B-X in this table), averaging, and the Huber and bisquare methods with the default tuning constants of 1.345 and 4.685, respectively, to maximise robustness and efficiency [20], [21].

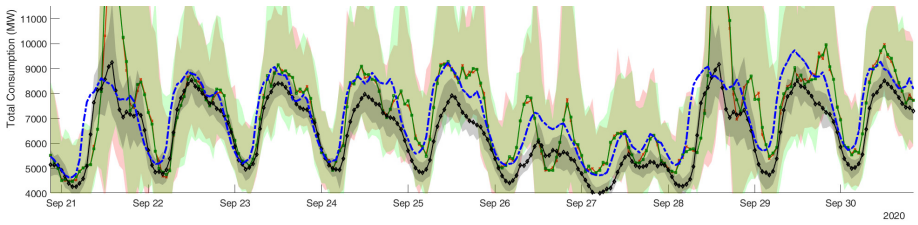
MAD	SNR = -20dB				SNR = -10dB			
Outlier (%)	**B-X**	Mean	Huber	BiSq	**B-X**	Mean	Huber	BiSq
0	**.0003**	.0176	.0308	.0174	**.0001**	.0029	.0339	.0278
0.1	**.0014**	.0132	.2457	.2345	**.0002**	.0122	.0288	.0170
1	**.0004**	.0625	.2069	.1936	**.0002**	.0156	.0109	.0107
5	**.0012**	.0156	.1357	.0959	**.0003**	.0095	.0702	.0599
10	**.0007**	.0271	.3903	.3037	**.0001**	.0036	.0658	.0777
20	**.0003**	.0257	.1985	.2042	**.0004**	.0030	.1131	.0374
MAD	SNR = 0dB				SNR = 10 dB			
Outlier (%)	**B-X**	Mean	Huber	BiSq	**B-X**	Mean	Huber	BiSq
0	**.0001**	.0047	.0170	.0131	**.0000**	.0015	.0016	.0005
0.1	**.0001**	.0009	.0262	.0371	**.0000**	.0014	.0072	.0064
1	**.0000**	.0003	.0140	.0155	**.0001**	.0012	.0008	.0027
5	**.0003**	.0029	.0217	.0236	**.0000**	.0005	.0084	.0113
10	**.0002**	.0004	.0245	.0282	**.0000**	.0057	.0040	.0254
20	**.0003**	.0027	.0120	.0182	**.0001**	.0037	.0067	.0089

historical data points for the training is $N = 840$. For both BA-XSCBB and XSCBB, the hyperparameter $b = 2$ and the seasonalities are again 1, 24, and 7×24 such that $d = 168$. For all methods, the Quasi Maximum Likelihood (QML) [84] were used to estimate the conditional means, conditional variances, and conditional covariances.

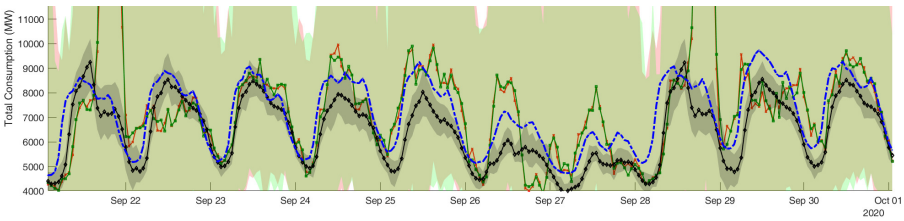
As demonstrated in Fig. 5.8, the BA-XSCBB successfully stabilises the prediction intervals for multi-step prediction in time series where the other methods start to widely deviate. The proposed method managed to follow the trend of the time series. However, the tendency to under-estimate the predictions can also be observed.



(a) 1-step ahead predictions



(b) 5-step ahead predictions



(c) 10-step ahead predictions

Figure 5.8: Comparison of the proposed BA-XSCBB (grey), XSCBB (green), residual bootstrap (pink) over multi-step predictions.

5.2.5 Conclusion and Future Work

We proposed two methods to robustify the XSCBB, namely the Block RESIF (BRESIF) – and the Block Average (BA) – XSCBB. We demonstrated the robustness of BRESIF-XSCBB over lower SNR and higher outlier percentages. With BA-XSCBB, we aim to stabilise time series predictions over multiple-steps by modifying the vector that is multiplied with the coefficients of the inverse seasonal difference operator. Thereby, we improved XSCBB, the bootstrap method that can handle multiple seasonalities, such that it may give more robust and stable predictions over many steps ahead. In the future, more analysis is needed in the hyperparameters selection such as the c, σ, and γ in Eq. (6.29) and the b to decide the block length in order to optimise the performance of the proposed methods.

6 Other Special Bootstrap Methods

This chapter presents a collection of specialised bootstrap procedures that extend the resampling framework to scenarios where classical methods are insufficient. These methods address settings characterised by changing variance dynamics, regime-switching behaviour, or complex nonlinear structures. While earlier chapters focused on oversampling strategies for imbalanced classification and on seasonal block structures for multi-periodic time series, the methods introduced in this chapter broaden the scope of bootstrap methodology by tackling problems motivated by real-world applications in communication and green energy.

Many practical environments require rapid uncertainty quantification, sometimes in near-real-time, where traditional bootstrap procedures become too computationally expensive. To meet this demand, the chapter introduces the *Block-Toeplitz Bootstrap (Blitz-Boot)*, a fast and approximate bootstrap + Finite Rate of Innovation (FRI) approach designed for situations in which only a limited number of resampling iterations can be computed. The Blitz-Boot explores the trade-off between speed and statistical fidelity, offering practitioners a tool for time-sensitive estimation tasks without sacrificing essential inferential reliability.

Another central challenge addressed in this chapter concerns signals and time series whose variability follows cyclical or regime-dependent patterns. Traditional block bootstrap methods treat volatility as stationary noise and therefore struggle to reproduce the alternating high- and low-variance phases that occur in financial time series, environmental processes, and industrial sensor readings. To capture these dynamics, the chapter introduces the *Cyclic Volatility Influence Function (CV-IF)*, a resampling framework that leverages influence functions to model and replicate volatility cycles more faithfully. The general CV-IF concept is further expanded into tailored variants such as the *GARCH-IF* and *Gated Sin-IF Bootstraps*, each designed to reflect distinct forms of cyclical or parametric volatility behaviour. These methods significantly enhance uncertainty quantification in contexts where volatility regime changes carry critical interpretive and predictive importance.

Collectively, the methods presented in this chapter demonstrate that specialised bootstrap schemes can outperform classical resampling techniques when the underlying data

exhibit dynamics that violate standard assumptions. By providing fast, volatility-aware, and structurally adaptive alternatives, this chapter completes the broader methodological framework developed throughout the dissertation. It shows that bootstrap procedures can be tailored not only to data imbalance and seasonal structure but also to computational constraints and intricate volatility patterns, thereby expanding the practical reach of resampling-based inference for modern signal analysis.

6.1 Blitz-Boot

We propose the Block-Toeplitz Bootstrap method, which we have named the Blitz-Boot, and apply it to improve the estimation of Direction of Arrival (DOA) and frequencies in compressed sensing applications. This method employs a joint frequency-DOA estimation approach based on the FRI principle. Empirical demonstrations reveal that Blitz-Boot enhances this estimation process. We present accuracy metrics for various block lengths and across different SNRs, demonstrating that the Blitz-Boot method notably improves performance, especially in environments characterized by lower SNRs. Furthermore, we provide a performance comparison with the Cramér-Rao Lower Bound (CRLB), underscoring the efficacy of our proposed method.

6.1.1 Introduction

In signal processing and statistical analysis, accurate parameter estimation is crucial for the performance of various applications, including communication systems, radar, biomedical engineering, and finance. Traditional estimation methods face challenges such as noise, sparse data, and computational limitations, especially in low SNR environments.

The challenge of determining the direction of signals is crucial in various fields such as radar, sonar, biomedical, and telecommunications. Conventional DOA estimation techniques typically require signals to be sampled at the Nyquist rate or demand a greater number of receivers than the actual number of signals to be discerned. However, sampling at the Nyquist becomes impractical or excessively expensive in scenarios involving high-frequency signals. This issue is especially pertinent in military or intelligence applications, where the frequencies of interest may not only be exceedingly high but also undisclosed, further compounding the difficulty of effective signal processing.

For DOA estimation and beamforming, there are techniques for sub-Nyquist rate sampling, but these involve extensive computational resources due to the need to sift through a dense dictionary of potential directions. Our work addresses the estimation of multiple unknown-frequency signals using a variation of block bootstrap method, which we call the Blitz-Boot in combination with a finite rate of innovation method proposed by [91]. We aim to identify both the frequency components and the directions of arrival of these signals.

This Section is structured as follows: Subsection 6.1.2 outlines the problem formulation, detailing the objective addressed in this study. Subsection 6.1.3 introduces our proposed method, providing a comprehensive overview of the techniques and algorithms developed. Subsection 6.1.4 discusses the simulation setup and presents the findings of our experiments. Lastly, Subsection 6.1.5 offers concluding remarks and summarizes the key

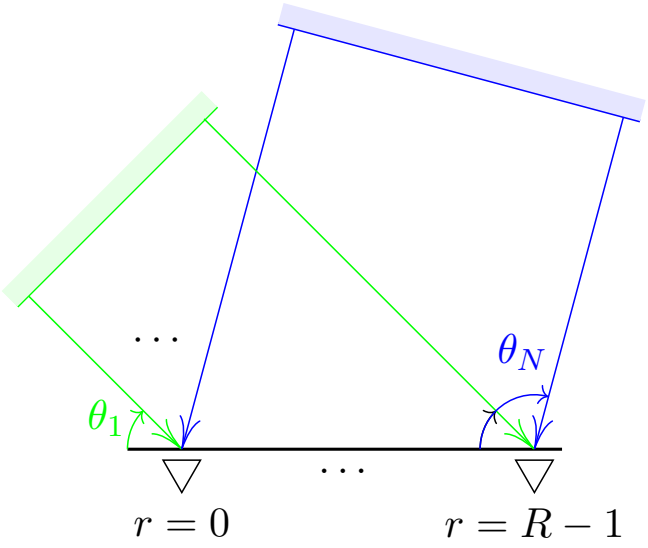


Figure 6.1: The uniform linear array (ULA) at the receiver and N transmitted plane waves

contributions of this work.

6.1.2 Problem Formulation

Refer to Fig. 6.1. The scenario involves N plane waves of sinusoidal signals transmitted and received by a ULA consisting of R receivers. The array is designed with an inter-element spacing of d. The N transmitted sinusoidal signals are represented as

$$s_n(t) = A_n e^{j2\pi f_n t},\ n = 1, \ldots, N, \tag{6.1}$$

where A_n and f_n are the amplitude and frequency of the nth signal, respectively, and are not known a priori. This scenario is typical in various applications such as radar, sonar, and wireless communications.

Each receiver has a known bandwidth limit of $f_{\max}$. The received signal at the rth receiver, corrupted by noise, is modeled as:

$$\tilde{x}_r(t) = \sum_{n=1}^{N} A_n e^{j2\pi f_n (t - r\Delta_n)} + \eta_r(t), \tag{6.2}$$

where Δ_n, the signal delay for the nth source, is a function of the receiver spacing d and

the direction of arrival θ_n, given by

$$\Delta_n = \frac{d}{c}\cos(\theta_n), \tag{6.3}$$

with c being the speed of light, $\eta_r(t) \sim \mathcal{N}\left(0, \sigma_\eta^2\right)$, and the SNR in dB is calculated as:

$$\text{SNR}_{\text{dB}} = 10\log_{10}\frac{P_x}{\sigma_\eta^2}, \tag{6.4}$$

where P_x is the power of the arriving signal.

Suppose that the signal $\tilde{x}_r(t)$ is then filtered through a filter $\varphi(t)$ and sampled at intervals $Ts = 1/F_s$, converting the continuous-time signal into a discrete form. The sampled values are represented as:

$$\tilde{y}_r[k] = \tilde{x}_r(t) * \varphi(t)\,|_{t=kT_s} + w_r[k] \tag{6.5}$$

where $w_r[k]$ denotes the digital noise. Suppose that $\tilde{y}_r[k]$ can also be modeled as

$$\tilde{y}_r[k] = y_r[k] + \tilde{w}_r[k], \tag{6.6}$$

where $y_r[k]$ is the noiseless part of the signal, and broken down as follows:

$$\begin{aligned} y_r[k] &= \int_{-\infty}^{\infty} x_r(kT_s - \tau)\,\varphi(\tau)\,d\tau \\ &= \sum_{n=1}^{N}\left(e^{-j2\pi f_n \Delta_n}\right)^r \cdot \left(A_n \int_{-\infty}^{\infty} e^{-j2\pi f_n \tau}\varphi(\tau)\,d\tau\right) \cdot \left(e^{j2\pi f_n T_s}\right)^k \qquad (6.7) \\ &\equiv \sum_{n=1}^{N} v_n^r \cdot a_n \cdot u_n^k. \qquad (6.8) \end{aligned}$$

The task is to recover the unknown parameters A_n and f_n using the sampled signals at different receivers, such as $y_0[k]$ and $y_1[k]$.

6.1.3 Methodology

An annihilating filter [92] $H(z)$, characterized by its coefficients $h[k]$, is used to recover the signal characteristics from $y_0[k]$. The filter satisfies the condition

$$H(z) = \prod_{n=1}^{N}\left(1 - u_n z^{-1}\right) = \sum_{k=0}^{N} h[k]\, z^{-k}. \tag{6.9}$$

In matrix form, the annihilating condition is represented as $\mathbf{Yh} = 0$, where $\mathbf{Y}$ is a Toeplitz matrix constructed from noiseless samples,

$$\mathbf{Y} = \begin{pmatrix} y_0[m+N] & y_0[m+N-1] & \cdots & y_0[m+0] \\ y_0[m+N+1] & y_0[m+N] & \cdots & y_0[m+1] \\ \vdots & \vdots & \ddots & \vdots \\ y_0[m+2N-1] & y_0[m+2N-3] & \cdots & y_0[m+N] \end{pmatrix}, \tag{6.10}$$

for an arbitrary $m \in \mathbb{Z}$, and $\mathbf{h} = [h[0], \ldots, h[N]]^T$.

We propose a resampling method which we denote as the Blitz-Boot. This method leverages the principles of the block bootstrap technique and the structure of Toeplitz matrices. The proposed method is summarised in Algorithm 8.

6.1.4 Experiments and Results

Table 6.1 presents a comparison of the RMSE for the estimated parameters $\hat{\theta}$ and $\hat{f}$ across various SNR levels and different estimation methods. It details the RMSE outcomes for the non-bootstrapped method (NB), which utilizes the entire data segment for calculations (about 9000 data points). In contrast, the results derived from the Blitz-boot method are characterized by the dimensions of the Toeplitz matrices; for instance, B1024 corresponds to a Toeplitz matrix with a height of 1024, equating to a block length of $2 \times 1024 + 1 = 2049$. This comparison aims to elucidate the accuracy and effectiveness of each method under different SNR conditions.

At low SNR of -20 dB, the RMSE values are relatively high for all bandwidths, suggesting a significant estimation error due to high noise levels. At moderate to high SNRs, the decreasing RMSE values indicate improved accuracy in the estimation of θ. The lowest RMSE values are observed at B64 across most SNR levels, suggesting that setting the block length to 129 may offer the best performance for estimating θ.

For the frequency estimation, notably, at an SNR of -20 dB, B1024 shows significantly better performance compared to other bandwidths. Beyond -20 dB, the RMSE values for $\hat{f}$ do not show a clear trend with changes in SNR, unlike $\hat{\theta}$, suggesting that the estimation of f might be less sensitive to SNR variations or that the noise level has a different impact on frequency estimation.

Figure 6.2 shows the comparison of the standard deviation values of the various estimators in comparison with those of the appropriate CRLB. The calculation of the CRLB will be detailed in the next paragraph. The standard deviation values of NB estimates with SNRs ranging from -20 to 5 dB were averaged from 1000 random runs using the whole

Algorithm 8: Blitz-Boot for joint frequency-DOA estimation

Step 1. Obtain $3 \times T$ noisy set of samples $\tilde{y}_r^{(i)}[k]$, $k = 0, \ldots, T-1$, $r = 0, \ldots, R-1$, from three different sampling rates $T_s^{(i)}$, $i = 1, 2, 3$, where $T_s^{(i)} = 1/F_s^{(i)}$, $F_s^{(i)} > F_s^{(i+1)}$ and $1/F_s^{(i+1)} - 1/F_s^{(i)} < 1/(2f_{\max})$. The use of 3 different sub-Nyquist sampling rates were proposed by [91].

Step 2. Choose an odd block length b, where $2N + 1 \leq b \leq T$. Draw B indices m_j, $j = 1, \ldots, B$, from $\mathcal{U}(0, T - b)$ to form B Toeplitz matrices of width $N + 1$ and height of at least $N + 1$ from the samples in the first receiver $(r = 0)$,

$$\tilde{\mathbf{Y}}_j^{*(i)} = \begin{pmatrix} \tilde{y}_0^{(i)}[m_j + N] & \tilde{y}_0^{(i)}[m_j + N - 1] & \cdots & \tilde{y}_0^{(i)}[m_j + 1] & \tilde{y}_0^{(i)}[m_j + 0] \\ \tilde{y}_0^{(i)}[m_j + N + 1] & \tilde{y}_0^{(i)}[m_j + N] & \cdots & \tilde{y}_0^{(i)}[m_j + 2] & \tilde{y}_0^{(i)}[m_j + 1] \\ \vdots & \vdots & & \vdots & \vdots \\ \tilde{y}_0^{(i)}[m_j + 2N] & \tilde{y}_0^{(i)}[m_j + 2N - 1] & \cdots & \tilde{y}_0^{(i)}[m_j + N + 1] & \tilde{y}_0^{(i)}[m_j + N] \end{pmatrix}.$$

Step 3. Minimize $\left\| \mathbf{Y}_j^{*(i)} \mathbf{h}_j^{*(i)} \right\|^2$ under the constraint that $\left\| \mathbf{h}_j^{*(i)} \right\|^2 = 1$ by performing a singular value decomposition of $\mathbf{Y}_j^{*(i)}$. If $\mathbf{Y}_j^{*(i)} = \mathbf{USV}^T$, then $\mathbf{h}_j^{*(i)}$ is the last column of $\mathbf{V}$ and $u_{n,j}^{*(i)}$ are the polynomial roots of $H(z)$ constructed from $\mathbf{h}_j^{*(i)}$.

Step 4. Solve for $f_{n,j}^{*(i)}$ from

$$2\pi \frac{f_{n,j}^{*(i)}}{F_s^{(i)}} = \arg\left(u_{n,j}^{*(i)}\right) + 2\pi L_n^{*(i)}, \; L_n \in \mathbb{Z}^+. \tag{6.11}$$

Sampling with the pairs $\left\{F_s^{(1)}, F_s^{(2)}\right\}$ and $\left\{F_s^{(2)}, F_s^{(3)}\right\}$ will result in $L_n|_{F_s = F_s^{(1)}} = L_n|_{F_s = F_s^{(2)}}$ and/or $L_n|_{F_s = F_s^{(2)}} = L_n|_{F_s = F_s^{(3)}}$ from

$$L_n = \frac{\left(F_s^{(j)} \arg\left(u_n\left(F_s^{(j)}\right)\right) - F_s^{(i)} \arg\left(u_n\left(F_s^{(i)}\right)\right)\right)}{2\pi\left(F_s^{(j)} - F_s^{(i)}\right)}, \; i \neq j. \tag{6.12}$$

Assign to the appropriate $L_n^{*(i)}$the *positive* L_n such that the a unique value for $f_{n,j}^{*(i)}$ is obtained [91]. From this point on, the superscript (i) will be dropped.

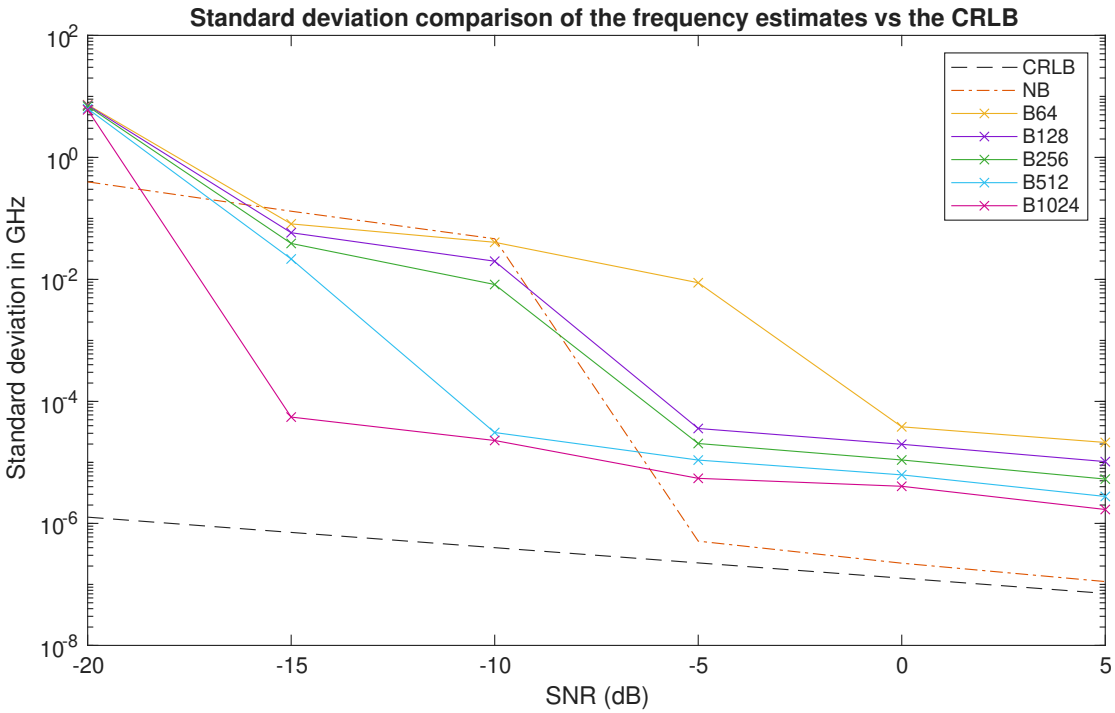


Figure 6.2: The comparison of standard deviation values for the estimates derived from the non-bootstrapped FRI estimator (NB) against those acquired through the Blitz-Boot matrices B64, B128, B256, B512, and B1024 was conducted, alongside an evaluation against the CRLB. This comparison aimed to assess the accuracy and efficiency of the bootstrapped estimators relative to the non-bootstrapped approach and the theoretical minimum variance bound indicated by the CRLB.

Algorithm 8: Blitz-Boot for joint frequency-DOA estimation (continued)

Step 5 Given the values of $u^*_{n,j}$, solve for $a^*_{n,j}$. Then substitute the values of $a^*_{n,j}$ and $u^*_{n,j}$ and solve for $v^*_{n,j}$. The delays are then given by

$$\Delta^*_{n,j} = -\frac{\arg\left(v^*_{n,j}\right)}{2\pi f^*_{n,j}}, \tag{6.13}$$

and the DOAs are given by

$$\theta^*_{n,j} = \frac{c}{d}\cos^{-1}\left(\Delta^*_{n,j}\right). \tag{6.14}$$

data segment (about 9000 data points). The standard deviation of the other estimators, B64 until B1024, were calculated from 1000 resampling of the 9000 data points by taking a shorter segment at a time (as before, B64 uses 129 data points per resampled block, B1024 uses 2049 data points per resampled block). We can observe that the Blitz-boot give more reliable estimates than using the non-bootstrapped method for for lower SNR range.

The CRLB is calculated as follows. The sampled signal from K sample points in the second receiver (subscript $r = 1$) is given by

$$\tilde{y}_1[k] = \sum_{n=1}^{N} a_n e^{j2\pi f_n(kT_s - \Delta_n)} + \tilde{w}_1[k],\ k = 0, \ldots, K-1. \tag{6.15}$$

With noise $\tilde{w}_1[k] \sim \mathcal{N}\left(0, \sigma^2\right)$, the pdf of $\tilde{y}_1$ is

$$p\left(\tilde{\mathbf{y}}_1; \mathbf{\Theta}\right) = \frac{1}{\left(2\pi\sigma^2\right)^{\frac{K}{2}}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{k=0}^{K-1}\left(\tilde{y}_1[k] - \mathcal{F}\left(\mathbf{\Theta}; k\right)\right)^2\right\}, \tag{6.16}$$

where for $N = 1$, the estimator is $\mathcal{F}\left(\mathbf{\Theta}; k\right) = ae^{j2\pi f(kT_s - \Delta)}$, where the parameter vector $\mathbf{\Theta} = (a, f, \phi)$ and $\Delta = d\cos(\phi)/c$. The log-likelihood function is given by

$$\ln p\left(\tilde{\mathbf{y}}; \mathbf{\Theta}\right) = -\frac{K}{2}\ln 2\pi - \frac{K}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}\sum_{k=0}^{K-1}\left(\tilde{y}_1[k] - \mathcal{F}\left(\mathbf{\Theta}; k\right)\right)^2. \tag{6.17}$$

The Fisher information matrix is given by

$$\mathbf{I}(\boldsymbol{\Theta}) = -\begin{pmatrix} \mathrm{E}\left[\frac{\partial^2 \ln p(\tilde{\mathbf{y}}_1;\boldsymbol{\Theta})}{\partial a^2}\right] & \mathrm{E}\left[\frac{\partial^2 \ln p(\tilde{\mathbf{y}}_1;\boldsymbol{\Theta})}{\partial a \partial f}\right] & \mathrm{E}\left[\frac{\partial^2 \ln p(\tilde{\mathbf{y}}_1;\boldsymbol{\Theta})}{\partial a \partial \theta}\right] \\ \mathrm{E}\left[\frac{\partial^2 \ln p(\tilde{\mathbf{y}}_1;\boldsymbol{\Theta})}{\partial a \partial f}\right] & \mathrm{E}\left[\frac{\partial^2 \ln p(\tilde{\mathbf{y}}_1;\boldsymbol{\Theta})}{\partial (f)^2}\right] & \mathrm{E}\left[\frac{\partial^2 \ln p(\tilde{\mathbf{y}}_1;\boldsymbol{\Theta})}{\partial f \partial \theta}\right] \\ \mathrm{E}\left[\frac{\partial^2 \ln p(\tilde{\mathbf{y}}_1;\boldsymbol{\Theta})}{\partial a \partial \theta}\right] & \mathrm{E}\left[\frac{\partial^2 \ln p(\tilde{\mathbf{y}}_1;\boldsymbol{\Theta})}{\partial f \partial \theta}\right] & \mathrm{E}\left[\frac{\partial^2 \ln p(\tilde{\mathbf{y}}_1;\boldsymbol{\Theta})}{\partial (\theta)^2}\right] \end{pmatrix} \tag{6.18}$$

Taking the negative expectation of the second derivative of $[\mathbf{I}(\boldsymbol{\Theta})]_{ij}$,

$$-E\left[\frac{\partial^2 \ln p\left(\tilde{\mathbf{y}}_1;\boldsymbol{\Theta}\right)}{\partial \Theta_i \partial \Theta_j}\right] = \frac{1}{\sigma^2}\sum_{k=0}^{K-1}\frac{\partial \mathcal{F}\left(\boldsymbol{\Theta};k\right)}{\partial \Theta_i}\frac{\partial \mathcal{F}\left(\boldsymbol{\Theta};k\right)}{\partial \Theta_j}, \tag{6.19}$$

where

$$\frac{\partial \mathcal{F}\left(\boldsymbol{\Theta};k\right)}{\partial a} = e^{j2\pi f\left(kT_s - \frac{d\cos(\theta)}{c}\right)} \tag{6.20}$$

$$\frac{\partial \mathcal{F}\left(\boldsymbol{\Theta};k\right)}{\partial f} = j2\pi a\left(kT_s - \frac{d\cos(\theta)}{c}\right)\cdot e^{j2\pi f\left(kT_s - \frac{d\cos(\theta)}{c}\right)} \tag{6.21}$$

$$\frac{\partial \mathcal{F}\left(\boldsymbol{\Theta};k\right)}{\partial \theta} = j2\pi\frac{d}{c}af\sin\left(\theta\right)e^{j2\pi f\left(kT_s - \frac{d\cos(\theta)}{c}\right)} \tag{6.22}$$

The CRLB of the estimate of the parameter Θ_i is then given by

$$\mathrm{CRLB}_{\Theta_i} = \left[\mathbf{I}^{-1}\left(\boldsymbol{\Theta}\right)\right]_{ii}. \tag{6.23}$$

6.1.5 Conclusion

We proposed the Blitz-Boot method, a resampling technique aimed at boosting the precision of estimation methods utilizing Toeplitz matrices. The results presented illustrate the improvement introduced by Blitz-Boot with various block lengths. The accuracy of estimating the frequency and DOA is influenced by both the SNR and the chosen block lengths which translates to the dimension of the Toeplitz matrices. While higher SNR generally leads to more accurate estimations of the DOAs, the estimation of the frequencies seems to be less sensitive to SNR variations beyond -20 dB. Furthermore, the optimal block length for the lowest RMSE in estimating DOAs changes with SNR, suggesting that the choice of block length should be context-dependent, based on the expected SNR conditions. These findings could guide the selection of block length and adjustments in signal processing strategies based on the expected SNR in various applications such as radar, sonar, or wireless communication systems.

Table 6.1: The RMSE values over different SNR and methods: the non-bootstrapped estimates (NB) and the estimates obtained by the Blitz-Boot method (B64-B1024).

SNR	RMSE of $\hat{\theta}$					
	NB	B1024	B512	B256	B128	B64
-20	**0.6844**	0.6929	0.6921	0.6920	0.6927	0.6939
-15	0.6515	0.6665	0.6181	0.5505	0.3125	**0.2408**
-10	0.4618	0.1788	0.1188	0.0788	**0.0761**	0.1106
-5	0.1284	0.0137	0.0126	0.0034	0.0071	**0.0032**
0	0.0486	0.0215	0.0275	0.0293	0.0236	**0.0180**
5	0.0011	0.0007	0.0019	0.0024	0.0006	**0.0005**
10	0.0044	0.0027	**0.0010**	0.0012	0.0034	0.0039
15	0.0034	**0.0031**	0.0043	0.0039	0.0048	**0.0031**
20	0.0017	0.0022	0.0023	**0.0015**	0.0022	0.0023

SNR	CRLB of $\hat{f}$					
	NB	B1024	B512	B256	B128	B64
-20	1.2825	**0.5456**	1.5831	3.1144	3.2131	4.3498
-15	0.0096	0.0096	0.0096	0.0096	0.0096	**0.0095**
-10	0.0094	0.0094	0.0094	0.0094	0.0094	0.0094
-5	0.0094	0.0094	0.0094	0.0094	0.0094	0.0094
0	0.0094	0.0094	0.0094	0.0094	0.0094	0.0094
5	0.0094	0.0094	0.0094	0.0094	0.0094	0.0094
10	0.0094	0.0094	0.0094	0.0094	0.0094	0.0094
15	0.0094	0.0094	0.0094	0.0094	0.0094	0.0094
20	0.0094	0.0094	0.0094	0.0094	0.0094	0.0094

6.2 The Cyclic Volatility Influence Function (CV-IF) Bootstrap

We propose the CV-IF Bootstrap – a resampling methodology that unifies periodic and stochastic components in time series. The framework addresses challenges in modeling signals with deterministic periodicity coupled with stochastic variability. Two implementations of the framework are presented: GARCH-IF, applied to heating consumption data, and Gated Sin-IF, applied to solar production data. By using an empirical influence function to weight residuals and correct for both seasonality and volatility, our method provides improved inference and uncertainty quantification compared to standard bootstrap techniques.

6.2.1 Introduction

One of the primary objectives of sustainability is to minimise energy waste. When heating demand and production are aligned, energy is used more efficiently. By matching heating demand to the availability of renewable energy sources, we can ensure their sustainable use. Reducing the gap between heating demand and production can also lead to energy independence. By relying on local, renewable energy sources, communities can reduce their dependence on energy imports, which are always subject to price volatility and geopolitical instability. These aspects motivated our work that started with more precise prediction of future heating demand and solar energy production based on historical data.

Time series in many domains exhibit a combination of strong periodic patterns and stochastic fluctuations. For example, heating consumption displays volatility influenced by weather variability, while solar production is governed by daily and seasonal cycles as well as random meteorological disturbances. Traditional bootstrap methods often fail to capture these intertwined features adequately.

To address these challenges, we introduce the CV-IF Residual Bootstrap. This unified framework builds upon the influence function (IF) concept to robustly resample residuals after decomposing the periodic time series into its stochastic components. We demonstrate two approaches; the GARCH-IF, tailored for time series that is well-modeled by GARCH-type dynamics, and the Gated Sin-IF, designed for time series that follows a gated sinusoidal pattern modulated by seasonal effects.

Bootstrap methods are a robust statistical approach for estimating essential characteristics from limited data sets. By repeatedly drawing samples with replacement from the original data, these techniques effectively approximate the distribution of a given statistic. [12], [13], [14].

Heating consumption data often exhibits volatility driven by weather conditions and usage patterns. We apply the GARCH-IF, a variant of CV-IF, where the stochastic component

is modeled using a GARCH model [76].

Solar production is naturally gated – zero during night hours – and its amplitude follows a seasonal (sinusoidal) envelope that can change over time (e.g., due to improved solar panel performance over the years). Our gated and enveloped residual bootstrap method first normalises the residuals by removing these systematic effects before resampling, then restores them to generate realistic prediction intervals. Results on hourly solar production data demonstrate improved uncertainty quantification and forecast performance over standard bootstrap methods.

The structure of this section is as follows. In Subsection 6.2.2, we present the formulation of the problem at hand. Following this, Subsection 6.2.3 discusses the two methods that fall under the CV-IF Bootstrap framework, i.e. the GARCH-IF and Gated Sin-IF methods. Subsection 6.2.4 presents the experimental results using real life datasets and we conclude in Subsection 6.2.5.

6.2.2 Problem Formulation

We consider a time series that may exhibit multiple sources of periodic behavior, which can be represented either as complex seasonalities or as a gated sinusoidal process. In the most general formulation, the true underlying signal can be decomposed as

$$X_t = \sum_{m=1}^{M} X_t^{(m)}, \tag{6.24}$$

where each component $X_t^{(m)}$ is periodic with period d_m,

$$X_t^{(m)} = X_{t-d_m}^{(m)}, \quad t \in \mathbb{Z}, m = 1, \ldots, M. \tag{6.25}$$

In the case of a gated sinusoidal process, the signal is further modulated by an envelope function and a gating function with varying length.

The observed time series Y_t is assumed to be affected by additive noise and occasional outliers. Specifically, letting $\eta_t \sim \mathcal{N}(0, \sigma_\eta)$ denote random fluctuations and

$$Y_t = X_t + \eta_t + \sum_i A_{t_i} \delta(t - t_i), \quad t \in \mathbb{Z}, \tag{6.26}$$

where $A_{t_i} \sim \mathcal{N}(\mu_{\text{out}}, \sigma_{\text{out}})$ is the magnitude of an outlier at time t_i, and $\delta(\cdot)$ is the Dirac delta function.

The proposed general framework, CV-IF Bootstrap, encompasses both time series with multiple seasonal components and those exhibiting a gated sinusoidal behavior, thereby providing a unified setting for further analysis and robust modeling.

6.2.3 Methodology

Suppose that the true distribution of the data lies in a neighborhood around a central model $F_{\boldsymbol{\theta}}$ that may subject to some contamination. Let $\mathcal{Y} = (Y_1, \ldots, Y_N)$ denote the available sample and let $\hat{\boldsymbol{\theta}}$ be an estimator for the parameter vector $\boldsymbol{\theta}$. Then, following [18], the Standardized Influence Function (SIF) of $\boldsymbol{\theta}$ is given by

$$\text{SIF}\left(\mathcal{Y}; \boldsymbol{\theta}, F_{\boldsymbol{\theta}}\right) = \left[\text{IF}\left(\mathcal{Y}; \boldsymbol{\theta}, F_{\boldsymbol{\theta}}\right)^{\intercal} \text{V}^{-1}\left(\boldsymbol{\theta}, F_{\boldsymbol{\theta}}\right) \text{IF}\left(\mathcal{Y}; \boldsymbol{\theta}, F_{\boldsymbol{\theta}}\right)\right]^{1/2}, \tag{6.27}$$

where IF represents the theoretical influence function [19] and $\text{V}\left(\boldsymbol{\theta}, F_{\boldsymbol{\theta}}\right)$ is the asymptotic variance of $\boldsymbol{\theta}$. The RESIF [18] is then defined by substituting into this expression the estimates of the unknown parameters that define $F_{\boldsymbol{\theta}}$.

For both GARCH-IF and Gated Sin-IF, we follow the general weights formula [18],

$$\begin{aligned} w_t =& \mathbb{I}_{[0,c]}\left(|\text{RESIF}_t|\right) + \psi\left(c, |\text{RESIF}_t|\right) \times \mathbb{I}_{[c,+\infty]}\left(|\text{RESIF}_t|\right), \\ & t = 1, 2, \ldots, N, \end{aligned} \tag{6.28}$$

where $\mathbb{I}$ is the indicator function with value of 1 inside the interval and 0 otherwise, c is a tuning constant, ψ a non-negative function with the following conditions [18]:

1) $\lim_{x \to +\infty} x^2 \psi\left(c, x\right) = 0$

2) $\frac{\partial}{\partial x} \psi\left(c, x\right) \mid_{x=c} = 0$.

We chose the function ψ following [18],

$$\psi\left(x; c, \sigma, \gamma\right) = \begin{cases} \left[1 + \frac{(x-c)^2}{\gamma \sigma^2}\right]^{-\frac{\gamma+1}{2}} & 0 < \gamma < \infty \\ \exp\left[-\frac{(x-c)^2}{2\sigma^2}\right] & \gamma = \infty, \end{cases} \tag{6.29}$$

where c is the location parameter, σ the scale parameter, and γ a shape parameter. Condition 1) is satisfied provided $\gamma > 1$. For parsimony, set $\sigma = c$ [18].

We modify the classical residual bootstrap by introducing influence-function-derived probabilities that reduce the chance of selecting outliers. The choice of the hyperparameters c and γ as well as the calculations of $|\text{RESIF}_t|$ for both cases are discussed in the following subsections.

GARCH-IF Bootstrap

Assume that the multiple seasonalities $d_1, \ldots, d_M$ in Eq. (6.25) can be reliably deduced from the sample. To detrend and remove the seasonal effects, linear and seasonal

differencing can be performed using the suitable lag operator polynomials $\Pi(L) = \left(1 - L^{d_1}\right) \cdots \left(1 - L^{d_M}\right) = 1 + \phi_1 L^1 + \ldots + \phi_\kappa L^\kappa, \phi_{1,\ldots,\kappa} \in \{-1, 1\}$. Then its inverse seasonal difference operator is $\Pi^{-1}(L) = 1 - \phi_1 L^1 - \ldots - \phi_\kappa L^\kappa$ [31], [32].

We model the detrended heating demand with a composite ARMA(p, q)-GARCH(r, s) model, where the ARMA part is

$$X_t = a_0 + \varepsilon_t + \sum_{i=1}^{p} a_i X_{t-i} + \sum_{j=1}^{q} \alpha_j \varepsilon_{t-j}, \tag{6.30}$$

where ε_t follows the standard normal distribution. The error term ε_t is

$$\varepsilon_t = \sqrt{h_t}\eta_t, \tag{6.31}$$

and the conditional variance process has the form

$$h_t = b_0 + \sum_{j=1}^{s} b_j \varepsilon_{t-j}^2 + \sum_{i=1}^{r} \beta_i h_{t-i} \tag{6.32}$$

where $b_0 > 0$, $b_j \geq 0$, $\beta_j \geq 0$, η_t are i.i.d., $\mathbb{E}[\eta_t] = 0$, $\mathbb{E}[\eta_t^2] = 1$, follow a symmetric distribution $\mathbb{E}[\eta_t^3] = 0$, and $\mathbb{E}[\eta_t^4] \leq \infty$. The proposed method, GARCH-IF Bootstrap, is described in Algorithm 9.

Gated Sin-IF Bootstrap

We propose to model the solar production as a composite function that has three parts:

1. Yearly amplitude envelope $g(t)$,

$$g(t) = \{C + D \cdot t\} \left\{ A + B \sin\left(\frac{2\pi}{T_{\text{year}}} \left(t_{\text{year}} - \xi \right) \right) \right\}, \tag{6.38}$$

   where $T_{\text{year}} = 8760$ hours, $t_{\text{year}} = \mod\left(t, T_{\text{year}}\right)$, $A, C \geq 0$, $B, D > 0$, and ξ is a time offset.

2. Daily gate with varying length $L(t)$,

$$L(t) = L_{\text{avg}} + L_{\text{amp}} \cdot \sin\left(\frac{2\pi}{T_{\text{year}}} \left(t_{\text{year}} - \chi \right) \right), \tag{6.39}$$

   where χ is a time shift, L_{avg} is the baseline length of the gate and L_{amp} the amplitude of sinusoidal modulation. For example, $L_{\text{avg}} = 10$ and $L_{\text{amp}} = 4$ to obtain a down time of 14 hours during winter time and 6 hours during summer time.

Algorithm 9: Proposed Method: GARCH-IF Bootstrap

Step 1 Remove the periodic (e.g., daily/weekly) pattern, $\tilde{X}_t = \Pi(L) \cdot Y_t$.

Step 2 Fit a GARCH(p, q) model to the de-seasonalised heating consumption data, e.g. by using its QML estimator from the past N

samples [31], [32],

$$\hat{\boldsymbol{\theta}} = (\hat{a}_0, \ldots, \hat{a}_p, \hat{\alpha}_1, \ldots, \hat{\alpha}_q, \hat{b}_0, \ldots, \hat{b}_s, \hat{\beta}_1, \ldots, \hat{\beta}_r), \tag{6.33}$$

Step 3 Calculate the residuals:

$$\hat{\varepsilon}_\tau = \tilde{X}_\tau - \hat{a}_0 - \sum_{i=1}^{p} \hat{a}_i \tilde{X}_{\tau-i} - \sum_{j=1}^{q} \hat{\alpha}_j \hat{\varepsilon}_{\tau-j}, \tag{6.34}$$

and the variance:

$$\hat{h}_\tau = \hat{b}_0 + \sum_{j=1}^{s} \hat{b}_j \hat{\varepsilon}^2_{\tau-1} + \sum_{i=1}^{r} \hat{\beta}_i \hat{h}_{\tau-i}, \tag{6.35}$$

for $\tau = t - N + 1, \ldots, t$.

Step 4 Compute RESIF for the GARCH model estimators:

$$\text{RESIF}_t = \tilde{\varepsilon}_t \equiv \frac{\hat{\varepsilon}_t}{\hat{\sigma}_\varepsilon}, \tag{6.36}$$

with $\hat{\sigma}_\varepsilon$ an estimate of the standard deviation of the residuals. Compute the weights w_i according to Eq. (6.28) using RESIF_t and tuning constants c and γ that can decided from a hyperparameter optimisation technique.

Step 5 Resample $\tilde{\varepsilon}_t$ using an IF-weighted scheme with probabilities $\mathbf{p} = (p_1, p_2, \ldots, p_N)^\intercal$ where:

$$p_t = \frac{w_t}{\sum_{n=1}^{N} w_n}. \tag{6.37}$$

Step 6 Combine the bootstrapped residuals with the estimated future predictions $\tilde{X}_{t+1}, \tilde{X}_{t+2}, \ldots$ and return its periodic components by using the coefficients of $\Pi^{-1}(L)$.

3. Half-sinusoid for the active signal: When the gate is “open”, i.e. during the active period, the signal follows a half sine curve that is positive,

$$f(t) = \begin{cases} \sin\left(\frac{\pi(t_d - \tau_{\text{start}}(t))}{L(t)}\right), & t_d \in [\tau_{\text{start}}(t), \tau_{\text{start}}(t) + L(t)] \\ 0 & \text{otherwise,} \end{cases} \tag{6.40}$$

where t_d is the time-of-day, $t_d = \mod(t, 24)$. For a given day, we assume the active period is centered around noon. The start time of the active window is

$$\tau_{\text{start}}(t) = \frac{24 - L(t)}{2}. \tag{6.41}$$

Putting it all together, the composite signal is

$$y(t) = f(t) \cdot g(t). \tag{6.42}$$

Algorithm 10 describes the proposed Gated-Sin IF Bootstrap method using this proposed signal model.

6.2.4 Experiment and Results

For the GARCH-IF method, we analysed the hourly heating demand in Megawatts (MW) in Germany [93]. Our objective was to perform one-hour-ahead predictions using a historical window of $N = 30$ hours, meaning each prediction was based on the past 30 hours of data. To preprocess the dataset, we addressed missing values by imputing them with the nearest available data point. Additionally, we standardised the number of data points per year by removing entries from February 29th in leap years. To improve the model’s ability to capture relative changes over time, we applied a logarithmic transformation to the data.

We compare three techniques: a non-bootstrap approach using the MATLAB command `forecast` [94] (Table 6.2), the classical residual bootstrap as described in [24] (Table 6.3), and the proposed GARCH-IF Bootstrap (Table 6.4). Figure 6.3 illustrates the prediction results for the last 100 hours of the dataset using the GARCH-IF Bootstrap method. The term ”Original” in the tables indicates that predictions were done with the original observations; in this case, outliers may have been naturally present (not artificially added). For all metrics, the GARCH-IF Bootstrap performed the best. Furthermore, to test the robustness of the GARCH-IF Bootstrap, we artificially added outliers randomly to 50% of the data points, increasing the mean values from approximately 8.6×10^4 to 1.3×10^5.

Algorithm 10: Proposed Method: Gated Sin-IF Bootstrap

Step 1 Assume the observations have been scaled to be between 0 and 1. Fit the parameters $\hat{\boldsymbol{\theta}} = \{A, B, C, D, \phi, \psi\}$ in the model described in equations (6.39) to (6.42) to obtain the estimates $\hat{y}(t)$, $t = 1, \ldots, N$, and compute the residuals,

$$\varepsilon_t = y_t - \hat{y}_t. \tag{6.43}$$

Step 2 Normalise the residuals by "removing" the envelope,

$$\tilde{\varepsilon}_t = \begin{cases} \varepsilon_t / g(t) & g(t) > \delta \\ \varepsilon_t / \delta & \text{otherwise,} \end{cases} \tag{6.44}$$

where δ is a small threshold to avoid division by zero.

Step 3 Since we have been working with scaled observations, we can take the values of $\text{RESIF}_t = \tilde{\varepsilon}_t$. Compute the weights w_t according to Eq. (6.28) using RESIF_t. Choose the tuning constant for example $c = 0.5$, to dampen the effects of residuals over half of the max normalised amplitude in the training data.

Step 4 Resample $\tilde{\varepsilon}_t$ using an IF-weighted scheme with probabilities according to Eq. (6.37) with replacement to generate bootstrap resamples $\tilde{\varepsilon}_t^*$.

Step 5 Reintroduce the envelope and the gating to obtain the final bootstrap residuals,

$$\varepsilon_t^* = \begin{cases} \tilde{\varepsilon}_t^* \cdot g(t), & t \in [\tau_{\text{start}}(t), \tau_{\text{start}}(t) + L(t)] \\ 0 & \text{otherwise.} \end{cases}, \tag{6.45}$$

Step 6 Combine the bootstrapped residuals with the estimates to obtain new pseudosamples, $y_t^* = \hat{y}_t + \varepsilon_t^*$ and obtain new bootstrapped parameter estimates $\hat{\boldsymbol{\theta}}^*$.

Table 6.2: Performance with Non-Bootstrap Approach

Data Set	RMSE (MW)	MAE (MW)	MAPE (%)
Original	9.0940e+03	5.6054e+03	0.0613
Additive Outliers	10.5456e+03	6.5789e+03	0.0851

Table 6.3: Performance with the Classical Residual Bootstrap

Data Set	RMSE (MW)	MAE (MW)	MAPE (%)
Original	8.9156e+03	5.3232e+03	0.0422
Additive Outliers	9.2355e+03	5.4599e+03	0.0512

Table 6.4: Performance with the GARCH-IF Bootstrap

Data Set	RMSE (MW)	MAE (MW)	MAPE (%)
Original	8.1273e+03	4.5631e+03	0.0390
Additive Outliers	8.5532e+03	4.7891e+03	0.0434

The tuning constants used were $c = 1.5$ and $\gamma = 10$. Likewise, we observed that the GARCH-IF Bootstrap performed the best.

For the solar energy production, we used the hourly total solar generation in Germany from the Open Power System Data [87]. We took a moving window of 365 days solar production starting from January 2019 as the training data to predict the subsequent 14 days solar production. We test 14 days windows in year 2020 with 7 days overlap.

Figure 6.5 shows the percentage coverage of the proposed bootstrap method with 100 resamples, which is defined as the percentage of when the actual output falls within the 95% confidence interval of the bootstrapped predictions. We observe that the actual observations fall well within the confidence intervals.

Figure 6.6 shows the percentage coverage of the proposed bootstrap method with 100 resamples, which is defined as the percentage of when the actual output falls within the 95% confidence interval of the bootstrapped predictions. We calculated that over 68% of the time, the bi-weekly bootstrapped predictions predict the solar production more accurately.

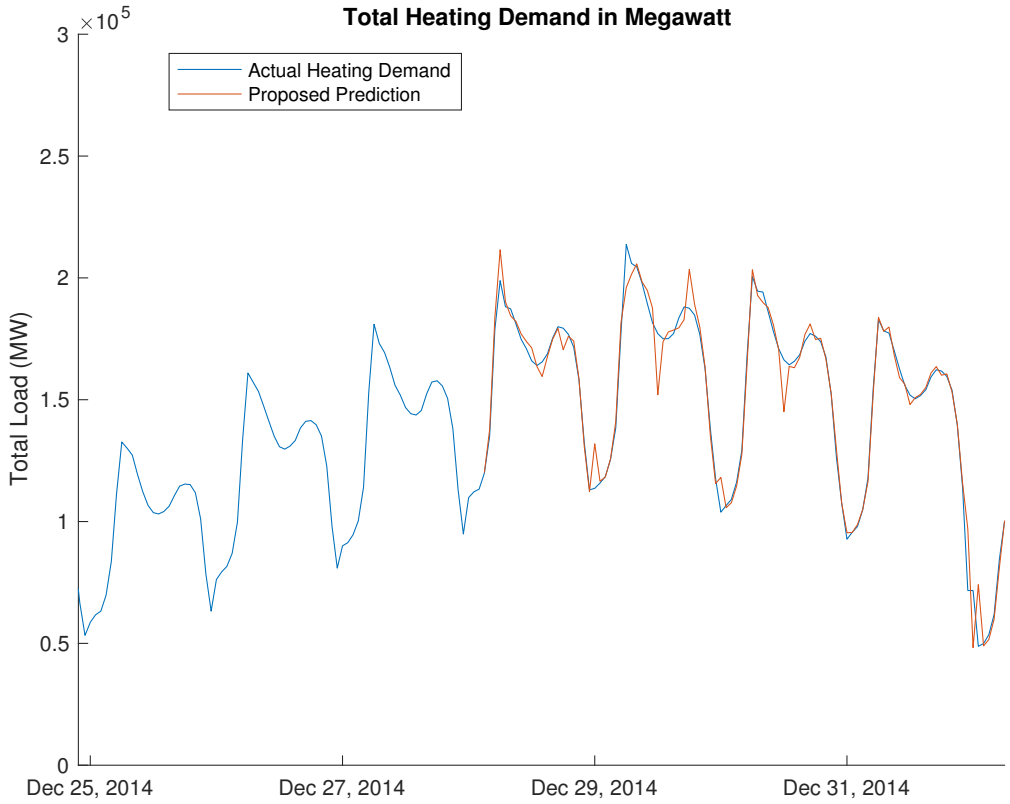


Figure 6.3: Predictions based on GARCH-IF Bootstrap

6.2.5 Conclusion and Future Work

We have introduced the CV-IF Bootstrap as a unified framework for resampling time series with mixed periodic and stochastic dynamics. By integrating influence function-based resampling with decomposition techniques, our approach effectively addresses the challenges posed by complex signals. The two case studies, GARCH-IF for heating consumption and Gated Sin-IF for solar energy production, demonstrate the versatility and effectiveness of our method. Future research will focus on extending the framework to other domains and refining the influence function computations for improved performance.

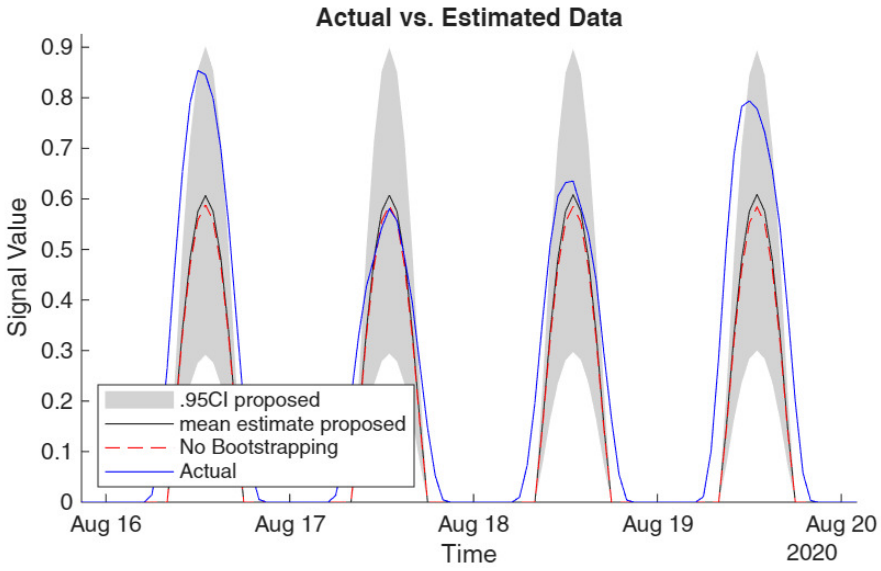


Figure 6.4: A snippet of the predictions using the proposed method.

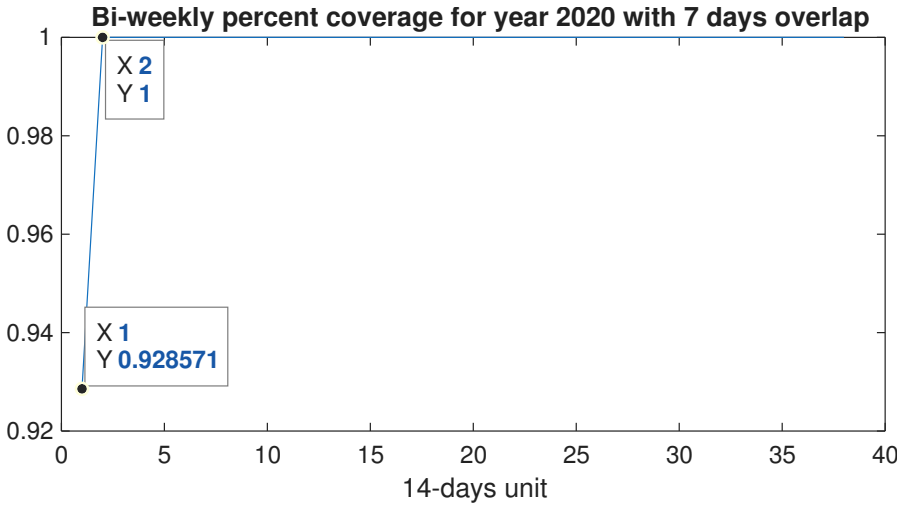


Figure 6.5: Coverage percentage of the biweekly predictions.

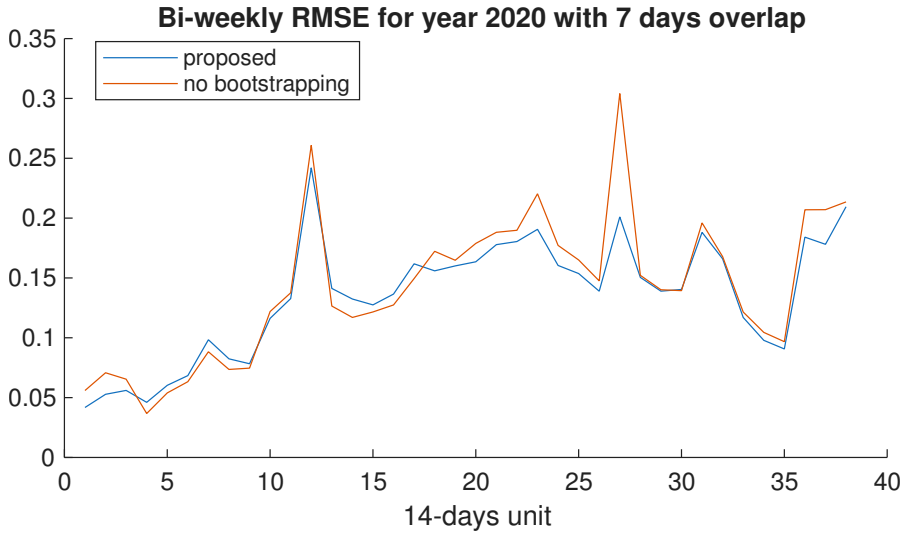


Figure 6.6: Bi-weekly RMSE comparison of the proposed model and non-bootstrapped technique.

7 Conclusion

This dissertation has presented a comprehensive set of methodological advances that extend the scope and applicability of bootstrap resampling to complex, heterogeneous, imbalanced, and volatile data environments. Motivated by the limitations of classical bootstrap procedures when applied to modern signal-processing and time-series applications, the work introduced adaptive frameworks that incorporate structural information, heterogeneity, multi-seasonality, and volatility dynamics directly into the resampling process. Through the development of oversampling stratified bootstrap methods, complex seasonal bootstrap procedures, and specialised influence-function and computationally efficient variants, this dissertation contributes a unified set of tools designed to meet the demands of contemporary data analysis.

Across the methods developed in Chapters 4-6, a central theme has been the need to respect underlying data structure rather than treating observations as interchangeable units. The oversampling stratified bootstrap family addressed the challenges posed by rare events and heterogeneous feature spaces, demonstrating how distance and heterogeneity measures can guide principled improvements in class balance and estimator stability. The complex seasonal bootstrap techniques showed that multi-seasonal behaviour, common in energy, environmental, and sensor-driven time series, can be effectively incorporated into resampling schemes, leading to more reliable inference in settings where classical block methods are insufficient. Finally, the specialised volatility-aware and computationally efficient bootstrap approaches illustrated how influence functions and fast approximations can extend bootstrap methodology to domains with rapidly changing variance dynamics or constrained resources.

The empirical studies conducted throughout the dissertation highlight the practical benefits of these contributions. The methods demonstrate improved robustness, more accurate uncertainty quantification, and enhanced performance in classification, forecasting, and detection tasks. Furthermore, the frameworks introduced here are broadly applicable and can be integrated into a wide range of signal-processing and statistical workflows, providing practitioners with flexible, data-aware alternatives to traditional resampling techniques.

While the contributions presented in this dissertation advance the state of the art in

bootstrap methodology, they also open several promising avenues for future research. One direction concerns the theoretical analysis of the proposed methods, including convergence guarantees, variance consistency, and conditions under which heterogeneity-based strata provide optimal performance. Another avenue involves the integration of bootstrap techniques with modern machine learning architectures, particularly in deep and representation-learning settings where structured resampling may enhance generalisation and robustness. Additionally, further exploration of adaptive block structures, automatic heterogeneity detection, and online or streaming bootstrap variants would extend the methods to even broader applications.

In summary, this dissertation demonstrates that carefully designed bootstrap procedures can significantly enhance inference in complex data environments. By embedding structural knowledge and data-driven adaptation into the resampling process, the methods developed here contribute to a more flexible and powerful resampling paradigm. It is my hope that these contributions will inspire further methodological development and support researchers and practitioners working with the increasingly rich and intricate data that define modern statistical practice.

Bibliography

[1] B. Efron, "Bootstrap methods: Another look at the jackknife," in *Breakthroughs in statistics: Methodology and distribution*, Springer, 1992, pp. 569–593.

[2] H. R. Kunsch, "The jackknife and the bootstrap for general stationary observations," *The annals of Statistics*, pp. 1217–1241, 1989.

[3] R. Y. Liu, K. Singh, et al., "Moving blocks jackknife and bootstrap capture weak dependence," *Exploring the limits of bootstrap*, vol. 225, p. 248, 1992.

[4] D. N. Politis and J. P. Romano, *A circular block-resampling procedure for stationary data*. Purdue University. Department of Statistics, 1991.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[6] E. M. Ronchetti and P. J. Huber, *Robust statistics*. John Wiley & Sons Hoboken, NJ, USA, 2009.

[7] B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*. Chapman and Hall/CRC, 1994.

[8] P. Hall, *The bootstrap and Edgeworth expansion*. Springer Science & Business Media, 2013.

[9] W. G. Cochran, *Sampling techniques*. john wiley & sons, 1977.

[10] J. Shao and D. Tu, "Theory for the bootstrap," in *The Jackknife and Bootstrap*, Springer, 1995, pp. 71–128.

[11] J. Neyman, "On the two different aspects of the representative method: The method of stratified sampling and the method of purposive selection," in *Breakthroughs in statistics: Methodology and distribution*, Springer, 1992, pp. 123–150.

[12] A. M. Zoubir and B. Boashash, "The bootstrap and its application in signal processing," *IEEE signal processing magazine*, vol. 15, no. 1, pp. 56–76, 1998.

[13] A. M. Zoubir and D. R. Iskander, *Bootstrap techniques for signal processing*. Cambridge University Press, 2004.

[14] A. M. Zoubir and D. R. Iskander, "Bootstrap methods and applications," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 10–19, 2007.

[15] D. N. Politis, "Resampling time series with seasonal components," in *Frontiers in data mining and bioinformatics: Proceedings of the 33rd symposium on the interface of computing science and statistics*, 2001, pp. 13–17.

[16] A. E. Dudek, J. Leśkow, E. Paparoditis, and D. N. Politis, "A generalized block bootstrap for seasonal time series," *Journal of Time Series Analysis*, vol. 35, no. 2, pp. 89–114, 2014.

[17] A. E. Dudek, E. Paparoditis, and D. N. Politis, "Generalized seasonal tapered block bootstrap," *Statistics & Probability Letters*, vol. 115, pp. 27–35, 2016.

[18] C. Amado and A. M. Pires, "Robust bootstrap with non random weights based on the influence function," *Communications in Statistics-Simulation and Computation*, vol. 33, no. 2, pp. 377–396, 2004.

[19] F. R. Hampel, "The Influence Curve and its Role in Robust Estimation," *Journal of the American Statistical Association*, vol. 69, no. 346, pp. 383–393, 1974.

[20] P. J. Huber, *Robust Statistics*. John Wiley & Sons, 2004, vol. 523.

[21] A. M. Zoubir, V. Koivunen, E. Ollila, and M. Muma, *Robust Statistics for Signal Processing*. Cambridge University Press, 2018.

[22] V. Corradi and E. M. Iglesias, "Bootstrap refinements for qml estimators of the garch (1, 1) parameters," *Journal of Econometrics*, vol. 144, no. 2, pp. 500–510, 2008.

[23] B. Chen, Y. R. Gel, N. Balakrishna, and B. Abraham, "Computationally efficient bootstrap prediction intervals for returns and volatilities in arch and garch processes," *Journal of Forecasting*, vol. 30, no. 1, pp. 51–71, 2011.

[24] K. Shimizu, *Bootstrapping stationary ARMA-GARCH models*. Springer, 2010.

[25] M. H. Neumann, "Bootstrap for integer-valued garch (p, q) processes," *Statistica Neerlandica*, 2021.

[26] Q. Zhu, G. Li, and Z. Xiao, "Quantile estimation of regression models with garch-x errors," *Statistica Sinica*, vol. 31, pp. 1261–1284, 2021.

[27] G. Cavaliere, I. Perera, and A. Rahbek, "Specification tests for garch processes," *arXiv preprint arXiv:2105.14081*, 2021.

[28] P. J. Bickel, F. Götze, and W. R. van Zwet, "Resampling fewer than n observations: Gains, losses, and remedies for losses," in *Selected works of Willem van Zwet*, Springer, 2011, pp. 267–297.

[29] P. J. Kunz and A. M. Zoubir, "Heterogeneity-stratified bootstrap oversampling for training a spoiled food detector," in *2023 24th International Conference on Digital Signal Processing (DSP)*, 2023, pp. 1–5. DOI: 10.1109/DSP58604.2023.10167943

[30] P. J. Kunz, S. b. Abid, and A. M. Zoubir, "The heterogeneity-intensified and heterogeneity ratio-stratified bootstrap (his- and hers-boot) oversampling to boost a detector performance," in *2023 IEEE SENSORS*, 2023, pp. 1–4. DOI: 10.1109/SENSORS56945.2023.10324861

[31] P. J. Kunz and A. M. Zoubir, "Complex seasonal circular block bootstrap for electricity load forecasting," in *2023 31st European Signal Processing Conference (EUSIPCO)*, 2023, pp. 2003–2007. DOI: 10.23919/EUSIPCO58844.2023.10289737

[32] P. J. Kunz and A. M. Zoubir, "Integrating block robust empirical standardised influence function and block average to robustify seasonal block bootstrap methods," in *2024 32nd European Signal Processing Conference (EUSIPCO)*, 2024, pp. 2742–2746. DOI: 10.23919/EUSIPCO63174.2024.10715332

[33] P. J. Kunz and A. M. Zoubir, "The blitz-boot approach: Improving the accuracy of joint direction-of-arrival and frequency estimation," in *2024 International Workshop on the Theory of Computational Sensing and its Applications to Radar, Multimodal Sensing and Imaging (CoSeRa)*, 2024, pp. 101–105. DOI: 10.1109/CoSeRa60846.2024.10720375

[34] P. J. Kunz, S. Frigiola, and A. M. Zoubir, "The cyclic-volatility influence function bootstrap – a resampling framework for periodic and volatility-driven signals," in *2025 25th International Conference on Digital Signal Processing (DSP)*, 2025, pp. 1–5. DOI: 10.1109/DSP65409.2025.11074840

[35] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[36] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Progress in artificial intelligence*, vol. 5, no. 4, pp. 221–232, 2016.

[37] Y. Sun, M. S. Kamel, and Y. Wang, "Boosting for learning multiple classes with imbalanced class distribution," in *Sixth international conference on data mining (ICDM'06)*, IEEE, 2006, pp. 592–602.

[38] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: A new over-sampling method in imbalanced data sets learning," in *International conference on intelligent computing*, Springer, 2005, pp. 878–887.

[39] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, IEEE, 2008, pp. 1322–1328.

[40] J. N. Rao and C. Wu, "Bootstrap inference with stratified samples.," WISCONSIN UNIV-MADISON MATHEMATICS RESEARCH CENTER, Tech. Rep., 1984.

[41] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural networks*, vol. 106, pp. 249–259, 2018.

[42] A. C. Davison and D. V. Hinkley, *Bootstrap methods and their application*. Cambridge university press, 1997.

[43] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone. Routledge, 2017.

[44] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*, 2nd. Cambridge University Press, 2017, One-sided Chebyshev (Cantelli) inequality; see Ch. 3, Exercise 3.18(a).

[45] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 138–156, 2006.

[46] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distribution," *Bulletin of the Calcutta Mathematical Society*, vol. 35, pp. 99–110, 1943.

[47] K. Fukunaga, *Introduction to statistical pattern recognition*. Elsevier, 2013.

[48] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[49] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[50] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *International conference on machine learning*, PMLR, 2017, pp. 1885–1894.

[51] U. R. Acharya, S. V. Sree, G. Swapna, R. J. Martis, and J. S. Suri, "Automated eeg analysis of epilepsy: A review," *Knowledge-Based Systems*, vol. 45, pp. 147–165, 2013.

[52] R. Esteller, J. Echauz, T. Tcheng, B. Litt, and B. Pless, "Line length: An efficient feature for seizure onset detection," in *2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, vol. 2, 2001, pp. 1707–1710.

[53] B. Hjorth, "Eeg analysis based on time domain properties," *Electroencephalography and clinical neurophysiology*, vol. 29, no. 3, pp. 306–310, 1970.

[54] P. Welch, "The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Transactions on audio and electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.

[55] E. Niedermeyer and F. L. da Silva, *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2005.

[56] A. Subasi and E. Ercelebi, "Classification of eeg signals using neural network and logistic regression," *Computer methods and programs in biomedicine*, vol. 78, no. 2, pp. 87–99, 2005.

[57] A. Mirzaei, A. Ayatollahi, P. Gifani, and L. Salehi, "Spectral entropy for epileptic seizures detection," in *2010 2nd International Conference on Computational Intelligence, Communication Systems and Networks*, IEEE, 2010, pp. 301–307.

[58] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[59] bbrinkm and W. Cukierski, *American epilepsy society seizure prediction challenge*, `https://kaggle.com/competitions/seizure-prediction`, Kaggle, 2014.

[60] Y. Ma and H. He, "Imbalanced learning: Foundations, algorithms, and applications," 2013.

[61] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

[62] B. Li, J. Friedman, R. Olshen, and C. Stone, "Classification and regression trees (cart)," *Biometrics*, vol. 40, no. 3, pp. 358–361, 1984.

[63] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[64] C. M. Bishop, "Neural networks and their applications," *Review of scientific instruments*, vol. 65, no. 6, pp. 1803–1832, 1994.

[65] C. X. Ling and C. Li, "Data mining for direct marketing: Problems and solutions.," in *Kdd*, vol. 98, 1998, pp. 73–79.

[66] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary," *Journal of artificial intelligence research*, vol. 61, pp. 863–905, 2018.

[67] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[68] S. Müller and A. Welsh, "Outlier robust model selection in linear regression," *Journal of the American Statistical Association*, vol. 100, no. 472, pp. 1297–1310, 2005.

[69] *Bme688 digital low power gas, pressure, temperature and humidity sensor with ai*, BME688, Rev. 1.1, Bosch Sensortec, Jul. 2022.

[70] *Esp32*, `https://www.espressif.com/en/products/socs/esp32`, Espressif System, 2023.

[71] J. S. Rao and R. Tibshirani, "The out-of-bootstrap method for model averaging and selection," *University of Toronto*, 1997.

[72] B. Efron and R. Tibshirani, "Improvements on cross-validation: The 632+ bootstrap method," *Journal of the American Statistical Association*, vol. 92, no. 438, pp. 548–560, 1997.

[73] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007, pp. 410–420.

[74] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PloS one*, vol. 10, no. 3, e0118432, 2015.

[75] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation," *Econometrica: Journal of the econometric society*, pp. 987–1007, 1982.

[76] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of econometrics*, vol. 31, no. 3, pp. 307–327, 1986.

[77] Y. Zhang, Y. Peng, X. Qu, J. Shi, and E. Erdem, "A finite mixture garch approach with em algorithm for energy forecasting applications," *Energies*, vol. 14, no. 9, p. 2352, 2021.

[78] F. Oktaviani, I. Setiawan, et al., "Forecasting sea surface temperature anomalies using the sarima arch/garch model," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1882, 2021, p. 012 020.

[79] Y. Yang, Y. Dong, Y. Chen, and C. Li, “Intelligent optimized combined model based on garch and svm for forecasting electricity price of new south wales, australia,” in *Abstract and Applied Analysis*, Hindawi, vol. 2014, 2014.

[80] H. Chen, Q. Wan, F. Li, and Y. Wang, “Garch in mean type models for wind power forecasting,” in *2013 IEEE Power & Energy Society General Meeting*, IEEE, 2013, pp. 1–5.

[81] N. Mohamed, M. H. Ahmad, and Z. Ismail, “Improving short term load forecasting using double seasonal arima model,” 2011.

[82] S. Noureen, S. Atique, V. Roy, and S. Bayne, “Analysis and application of seasonal arima model in energy demand forecasting: A case study of small scale agricultural load,” in *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*, IEEE, 2019, pp. 521–524.

[83] P. Amin, L. Cherkasova, R. Aitken, and V. Kache, “Automating energy demand modeling and forecasting using smart meter data,” in *2019 IEEE International Congress on Internet of Things (ICIOT)*, IEEE, 2019, pp. 133–137.

[84] H. Liu, S. Tan, and Q. Zhu, “Quasi-maximum likelihood inference for linear double autoregressive models,” *arXiv preprint arXiv:2010.06103*, 2020.

[85] L. Pascual, J. Romo, and E. Ruiz, “Bootstrap prediction for returns and volatilities in garch models,” *Computational Statistics & Data Analysis*, vol. 50, no. 9, pp. 2293–2312, 2006.

[86] H. Liu, “Robust estimation for garch models and varma models,” Ph.D. dissertation, Lancaster University, 2021.

[87] O. P. S. Data. “Data package time series. version 2020-10-06.” English.

[88] J. Fox, “Time-series Regression and Generalized Least Squares,” *An R S-PLUS Companion to Appl. Regres. Thousand Oaks, CA*, pp. 1–8, 2002.

[89] G. Schwarz, “Estimating the dimension of a model,” *The annals of statistics*, pp. 461–464, 1978.

[90] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[91] P. J. Kunz and P. Marziliano, “Finite rate of innovation method for DOA estimation of multiple sinusoidal signals with unknown frequency components,” in *2012 9th European Radar Conference*, IEEE, 2012, pp. 115–118.

[92] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE Trans. Signal Process.*, vol. 50, no. 6, pp. 1417–1428, 2002. DOI: 10.1109/TSP.2002.1003065

[93] O. P. S. Data, *Data package When2Heat Heating Profiles*, https://data.open-power-system-data.org/when2heat/2022-02-22, 2022.

[94] M. Forecast, https://it.mathworks.com/help/econ/arima.forecast.html, 2022.

List of Figures

4.1 Training the electronic nose with different specimens. 63
4.2 The measurements for the training set 64
4.3 The measurements as the additional test set 65
4.4 Illustration of the updated resampling PMFs 68
4.5 The measurements of the six specimens collected from different sensors around the same time. 73

5.1 (a)The illustrations of the Seasonal Block Bootstrap (SBB) [15] and (b) the Generalised Seasonal Circular Bootstrap (GSCBB) [16] 83
5.2 (a) The full dataset of hourly electricity consumption in Megawatt (MW), (b) the natural log of the last 10 days (240 hours) of the dataset, and (c) the corresponding non-seasonal and multiple seasonal differenced log sample. 89
5.3 One hour ahead predictions from hourly training sample with $N = 840$, $b = 2$, $B = 500$, $d = \text{lcm}(1, 7, 24) = 168$, noting that $d_1 = 1$ will remove the linear trend in the time-series. 90
5.4 One day ahead predictions from hourly training sample with $N = 840$, $b = 2$, $B = 500$, $d = \text{lcm}(1, 7, 12) = 84$. 90
5.5 Simulated data with daily and weekly seasonalities at SNR of 30 dB and the deseasonalised version, no outliers added. 96
5.6 The illustration of Block Average XSCBB. 99
5.7 Comparison of $\psi(x; c, \sigma, \gamma)$ and the absolute Standard Normal distribution. 100
5.8 Comparison of the proposed BA-XSCBB (grey), XSCBB (green), residual bootstrap (pink) over multi-step predictions. 102

6.1 The ULA at the receiver and N transmitted plane waves 108

6.2 The comparison of standard deviation values for the estimates derived from the non-bootstrapped FRI estimator (NB) against those acquired through the Blitz-Boot matrices B64, B128, B256, B512, and B1024 was conducted, alongside an evaluation against the CRLB. This comparison aimed to assess the accuracy and efficiency of the bootstrapped estimators relative to the non-bootstrapped approach and the theoretical minimum variance bound indicated by the CRLB. 112
6.3 Predictions based on GARCH-IF Bootstrap 124
6.4 A snippet of the predictions using the proposed method. 125
6.5 Coverage percentage of the biweekly predictions. 125
6.6 Bi-weekly RMSE comparison of the proposed model and non-bootstrapped technique. 126

List of Tables

4.1 Results Dogs 1-5 . . . 52
4.1 Results Dogs 1-5 . . . 53
4.1 Results Dogs 1-5 . . . 54
4.2 Results Patients 1-2 . . . 55
4.2 Results Patients 1-2 (Continued) . . . 56
4.2 . . . 57
4.3 Average F1 change on Dog1–Dog5 (extreme class imbalance). Values are absolute percentage points (pp). . . . 57
4.4 MB (MS-Boot) gains on patients (P1, P2). Values are absolute percentage points (pp). . . . 57
4.5 Dog3 with CART (Decision Tree): detailed gains. Values are absolute percentage points (pp). . . . 58
4.6 Model-level highlights on patient cohorts. Values are absolute percentage points (pp). . . . 58
4.7 Dummy example of $C = 2$ classes with respectively 6 and 8 sample points, and $K = 18$ cells. . . . 62
4.8 Average out of 500 trials for SN (sensitivity, or true positive rate or recall), SP (specificity, or true negative rate), ACC (accuracy), ERR (error rate), PREC (precision), F1 (F1 score, harmonic mean of precision and recall) . . . 66
4.8 Dummy example of $C = 2$ classes with respectively 6 and 8 sample points, and $K = 18$ cells. . . . 72
4.9 Summary of the three algorithms, where γ is a hyperparameter constant that determines the influence of the heterogeneity, a_{ck} is the number of samples in cell k that belongs to class c, $\rho_{ck} = a_{ck}/a_k$ the ratio of members of class c in cell k, and the denominator is to make sure that the sum equals 1. 74
4.10 **Spoiled Food** Detection Results . . . 75
4.10 **Spoiled Food** Detection Results (Continued) . . . 76
4.11 **Allergen Detection** Results . . . 77
4.11 **Allergen Detection** Results (Continued) . . . 78

5.1 Prediction metrics for the experiment setting as described in Figures 5.3 and 5.4. 91
5.2 Performance comparison in the MAD of the estimates by the proposed method, BRESIF-XSCBB (short-hand B-X in this table), averaging, and the Huber and bisquare methods with the default tuning constants of 1.345 and 4.685, respectively, to maximise robustness and efficiency [20], [21]. 101

6.1 The RMSE values over different SNR and methods: the non-bootstrapped estimates (NB) and the estimates obtained by the Blitz-Boot method (B64-B1024). 115
6.2 Performance with Non-Bootstrap Approach 123
6.3 Performance with the Classical Residual Bootstrap 123
6.4 Performance with the GARCH-IF Bootstrap 123

List of Algorithms

1 MS-Boot (Mahalanobis–distance Stratified Bootstrap) 32
2 Oversampling with Heterogeneity-Stratified Bootstrap (HSBoot) 61
3 Oversampling with the HS-, HiS, or HeRS-Boot 71

4 The XSCBB 85
5 Parametric XSCBB for ARMA(p, q) – GARCH(r, s) Model 88
6 The Complex Seasonal Circular Block Bootstrap (XSCBB) [31] 94
7 Proposed Method: BRESIF-XSCBB 97

8 Blitz-Boot for joint frequency-DOA estimation 111
8 Blitz-Boot for joint frequency-DOA estimation (continued) 113
9 Proposed Method: GARCH-IF Bootstrap 120
10 Proposed Method: Gated Sin-IF Bootstrap 122

List of Acronyms

ADASYN Adaptive Synthetic Sampling Approach for Imbalanced Learning 3, 28, 59

AIC Akaike Information Criterion 86

AR Autoregressive 20

ARCH Autoregressive Conditionally Heteroscedastic 17

ARMA Autoregressive Moving Average 16, 20, 81, 84, 86–88, 119, 141

BA-XSCBB Block Average Complex Seasonal Circular Block Bootstrap 79, 92, 94

BIC Bayesian Information Criterion 86

Blitz-Boot Block-Toeplitz Bootstrap 105, 107, 110–114, 138, 141

BO Uniform bootstrap oversampling 30, 35–46

BRESIF Block Robust Estimates Standardized Influence Function 94

BRESIF-XSCBB Block Robust Estimates Standardized Influence Function - Complex Seasonal Circular Block Bootstrap 79, 92, 97, 99, 101, 140, 141

CART Classification and Regression Trees 33, 38, 51, 52, 55, 58, 139

CBB Circular Block Bootstrap 14

CDF cumulative distribution function 39

CP coverage probability 86

CRLB Cramér-Rao Lower Bound 107, 110, 112–115, 138

CV-IF Cyclic Volatility Influence Function 105, 116, 117, 124

DOA Direction of Arrival 107, 113, 114, 141

EEG Electroenchephalogram 3, 28, 29

ERM empirical–risk minimization 30, 31, 35, 36, 41, 43, 44

FN false negatives 31

FP false positives 31

FRI Finite Rate of Innovation 105, 107

GARCH Generalised Autoregressive Conditionally Heteroscedastic 16, 17, 20, 81, 84, 86, 88, 105, 116–121, 123, 124, 138, 140, 141

GNB Gaussian Naïve Bayes 41

GSBB Generalised Seasonal Block Bootstrap 15–17

GSCBB Generalised Seasonal Circular Block Bootstrap 15–17

GSTBB Generalised Seasonal Tapered Circular Block Bootstrap 16, 17

IF Influence Function 95, 105, 116–124, 138, 140, 141

KKT Karush-Kuhn-Tucker 45

LLR log-likelihood ratio 36, 38

LR Logistic Regression 33

MAD Mean Absolute Deviation 101, 140

MAE Mean Absolute Error 99, 123

MAPE Mean Absolute Percentage Error 86, 123

MB MS-Boot 30, 35–46

MBB Moving Blocks Bootstrap 13, 14

MLE maximum likelihood estimation 42

MLEs maximum likelihood estimates 40

MLPs Multi Layer Perceptrons 33

MS-Boot Mahalanobis distance Stratified Bootstrap 28, 29, 51, 54

MSE mean squared error 50

MW Megawatts 121, 123

NB Naïve Bayes 33, 51, 52, 55

NO No Oversampling 30, 35–39, 41–46

PSD power spectral density 48

QDA quadratic discriminant analysis 33, 36, 51, 52, 55

QML Quasi Maximum Likelihood 84, 86, 88, 120

ReLU Rectified Linear Unit 43, 50

RESIF Robust Estimates Standardized Influence Function 95, 118

RMS Root Mean Square 47

RMSE Root Mean Square Errors 86, 110, 114, 115, 123, 126, 138

SBB Seasonal Block Bootstrap 14, 15, 17

SGD Stochastic Gradient Descent 44

SIF Standardized Influence Function 118

SMOTE synthetic minority oversampling technique 2, 3, 28, 59, 149

SNR Signal to Noise Ratio 93, 96, 99, 101, 103, 107, 109, 110, 113–115, 137

SVM Support Vector Machine 33, 35, 45, 46

TN true negatives 31

TP true positives 31

ULA uniform linear array 108, 137

XSCBB Complex Seasonal Circular Block Bootstrap 79, 81, 84–88, 91, 92, 141

Glossary

affine linear up to a constant shift 33

Bhattacharyya distance The Bhattacharyya distance measures class overlap: $B(P,Q) = -\log\left(\int \sqrt{p(x)q(x)}dx\right)$. Larger B means smaller Bayes error (via Chernoff/Bhattacharyya bounds). For Gaussians $\mathcal{N}(\mu_k, \Sigma_k)$, $B = \frac{1}{8}(\mu_1 - \mu_0)^\intercal\left(\frac{\Sigma_0+\Sigma_1}{2}\right)^{-1}(\mu_1 - \mu_0) + \frac{1}{2}\log\frac{\left|(\Sigma_0+\Sigma_1)/2\right|}{\sqrt{|\Sigma_0||\Sigma_1|}}$. 36

Cantelli's inequality Cantelli's inequality: for any $t > 0$, $P(X - \mu \leq -t) \leq \frac{\sigma^2}{\sigma^2+t^2}$. 33, 35, 36

Chebyshev's inequality Chebyshev's inequality: for any $t > 0$, $P(|X - \mu| \geq t) \leq \frac{\sigma^2}{t^2}$. 33

Chernoff/Bhattacharyya For two classes with densities p and q, the Bayes error P_e satisfies the Chernoff bound $P_e \leq \frac{1}{2}e^{-C}, C = -\log\left(\min_{0\leq s\leq 1}\int p(x)^s q(x)^{1-s}dx\right)$. Setting $s = \frac{1}{2}$ yields the Bhattacharyya bound $P_e \leq \frac{1}{2}e^{-B}, B = -\log\left(\int \sqrt{p(x)q(x)}dx\right)$. Larger C (or B) implies smaller Bayes error; both have closed forms for Gaussian classes. 36

Cramér-Rao Lower Bound The Cramér–Rao Lower Bound says that for any unbiased estimator $\hat{\theta}$ of a parameter θ, $\text{Var}(\hat{\theta}); \geq; \frac{1}{I(\theta)}$, where $I(\theta)$ is the Fisher information. It gives a theoretical minimum variance. 107, 143

Dirac delta function The Dirac delta function $\delta(x)$ is a mathematical idealization that is zero everywhere except at $x = 0$, but integrates to 1 over the whole real line, $\delta(x) = 0$ for $x \neq 0$, $\int_{-\infty}^{\infty}\delta(x), dx = 1$. Its key property (sifting property): $\int_{-\infty}^{\infty} f(x), \delta(x - a), dx = f(a)$. 117

Fisher consistent A surrogate loss $\ell(y, f(x))$ is Fisher consistent for 0–1 risk if, for every distribution of (X, Y), any population minimizer $f^* \in \arg\min_f \ \mathbb{E}\big[\ell(Y, f(X))\big]$ induces a Bayes–optimal classifier, i.e. $\operatorname{sign}\big(f^*(x)\big) = \operatorname{sign}\big(2\eta(x) - 1\big)$ with $\eta(x) = \Pr(Y = 1 \mid X = x)$. Equivalently, minimizing the surrogate risk recovers the Bayes decision rule. 36

Hadamard-differentiable A functional $T : \mathcal{P} \to \mathbb{R}^q$ is Hadamard differentiable at P (tangentially to a set $\mathcal{T}$) if there exists a continuous linear map $\dot{T}_P : \mathcal{T} \to \mathbb{R}^q$ such that for any $h \in \mathcal{T}$ and any $h_t \to h$, $\lim_{t \downarrow 0} \frac{T(P + t\,h_t) - T(P)}{t} = \dot{T}_P(h)$. This guarantees stable first-order expansions (influence functions) and justifies the functional delta method for ERM estimators under small re-weightings/perturbations of P. 45

Karush-Kuhn-Tucker In the constrained optimization $\min_x \ f(x) \quad \text{s.t.} \quad g_i(x) \le 0 \ (i = 1, \dots, m), \quad h_j(x) = 0 \ (j = 1, \dots, p)$, the Karush–Kuhn–Tucker (KKT) conditions (first-order optimality) state that there exist multipliers $(\lambda_i \ge 0)$, $(\nu_j \in \mathbb{R})$ such that:

Stationarity:	$\nabla f(x^\star) + \sum_{i=1}^{m} \lambda_i^\star \nabla g_i(x^\star) + \sum_{j=1}^{p} \nu_j^\star \nabla h_j(x^\star) = 0,$
Primal feasibility:	$g_i(x^\star) \le 0, \quad h_j(x^\star) = 0,$
Dual feasibility:	$\lambda_i^\star \ge 0,$
Complementary slackness:	$\lambda_i^\star, g_i(x^\star) = 0 \quad \forall i.$

Under suitable regularity (constraint qualification) and convexity, KKT conditions are necessary and sufficient for optimality. 45, 144

M-Estimation M-estimation refers to estimating a parameter by minimizing an empirical average loss, $\widehat{\theta} \in \arg\min_{\theta \in \Theta} \ \frac{1}{N} \sum_{i=1}^{N} \rho\left(Z_i; \theta\right)$, equivalently solving the estimating equation $\frac{1}{N} \sum_{i=1}^{N} \psi(Z_i; \theta) = 0$, $\psi = \nabla_\theta \rho$, and includes MLE as the special case $(\rho(z; \theta) = -\log p_\theta(z)$. 19, 21

Mahalanobis distance The Mahalanobis distance measures the distance between a point and the mean of a multivariate distribution, taking into account the correlations between variables: $D_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^\intercal \mathbf{S}^{-1} (\mathbf{x} - \boldsymbol{\mu})}$ 28, 30, 42, 145

Nyquist Nyquist (Nyquist–Shannon sampling theorem) states that, to perfectly reconstruct a band-limited signal, the sampling frequency f_s must be at least twice the highest frequency in the signal, $f_s \ge 2 f_{\text{max}}$. 107, 111

oversampling Oversampling is increasing the minority-class sample count to re-balance an imbalanced dataset either by duplicating existing minority points (e.g., random/bootstrap oversampling) or creating synthetic ones (e.g., SMOTE). ix, 2, 3, 23

positive-definite kernel A (symmetric) function $(k : \mathcal{X} \times \mathcal{X} \to \mathbb{R})$ is positive–definite if for any $(n \in \mathbb{N})$, any points $(x_1, \ldots, x_n \in \mathcal{X})$, and any coefficients $(c_1, \ldots, c_n \in \mathbb{R})$, $\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j, k(x_i, x_j) \geq 0$. Such (k) induces an RKHS $(\mathcal{H})$ and an embedding (ϕ) with $(k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}})$. 44

Rectified Linear Unit The rectified linear unit is the piecewise–linear activation

$$\mathrm{ReLU}(t) = \max\{0, t\} = \begin{cases} 0, & t \leq 0, \\ t, & t > 0, \end{cases}$$

with derivative

$$\mathrm{ReLU}'(t) = \begin{cases} 0, & t < 0, \\ 1, & t > 0, \end{cases}$$

(undefined at $t = 0$; use subgradient in $[0, 1]$). It induces sparsity (many zeros), is 1-Lipschitz, and preserves positive linear trends. 43, 145

RKHS In a reproducing kernel Hilbert space (RKHS) $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$, every function evaluation is a continuous linear functional and is "reproduced" by the kernel (k): $[\exists\, k(\cdot, x) \in \mathcal{H}$ s.t. $f(x) = \langle f,\ k(\cdot, x) \rangle_{\mathcal{H}} \quad \forall f \in \mathcal{H},\ x \in \mathcal{X}.]$ An RKHS is uniquely determined by its positive–definite kernel (k), with inner products satisfying $[\langle k(\cdot, x), , k(\cdot, x') \rangle_{\mathcal{H}} = k(x, x').]$ 44

von Mises expansion For a regular functional T at distribution P, small re-weightings of P admit the first–order expansion $T(P_\varepsilon) = T(P) + \varepsilon \int \mathrm{IF}(z; T, P)\{dP_\varepsilon(z) - dP(z)\} + o(\varepsilon)$. For the empirical measure P_n, this yields the asymptotic linearization $T(P_n) = T(P) + \frac{1}{n} \sum_{i=1}^{n} \mathrm{IF}(Z_i; T, P) + o_p(n^{-1/2})$, where $\mathrm{IF}(z; T, P)$ is the influence function of T at P. 18, 35, 41, 42