ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

SCHOOL OF INFORMATION SCIENCES AND TECHNOLOGY

Department of Statistics – Postgraduate program

Thesis Title : Comparisons and applications of bootstrap methods for time series data

Stundent : Vlachogiorgos Apostolos

Supervisor : Pedeli Xanthi

Athens, Greece

June, 2021

Σχολή Επιστημών και Τεχνολογίας της Πληροφορίας

Τμήμα Στατιστικής – Μεταπτυχιακός πρόγραμμα

Τίτλος Διπλωματικής: Συγκρίσεις και εφαρμογές bootstrap μεθόδων σε δεδομένα χρονολογικών σειρών.

Φοιτητής : Βλαχογιώργος Απόστολος

Επιβλέπων : Πεντελή Ξανθή

Αθήνα, Ελλάδα

Ιούνιος, 2021

ABSTRACT

In this project we examine the performance of block bootstrap methods under different block length selection techniques. Our goal is to compare these techniques in order to get the one that produces the “optimal” block length. This problem is of crucial importance, due to the fact that block bootstrap methods much depend on the selection of block length parameter. We conducted an extensive Monte Carlo simulation with different models and various values for their parameters. We used the well known methods HHJ and NPPI that are MSE-optimal as well as a method of Politis and White (2004) for two of the block bootstrap methods as they proposed. However, we did not stop there. We proposed four new methods to use as block length selection techniques and the results were clearly in favor of them. The two of these methods using a version of Quadratic loss function were in most cases the dominant ones. Furthermore, we compared the block bootstrap methods as well as the block length selection techniques with INAR models, which are time series modelling count data. The estimation of the parameters for these models is cumbersome so we wanted to know how well they are able to perform. Lastly, we performed the analysis in a real dataset which we modelled with as INAR(4) and the results in this case were mixed, yet again in favor of the new proposed methods.

ΠΕΡΙΛΗΨΗ

Σε αυτή την εργασία εξετάσαμε την απόδοση block bootstrap μεθόδων με διαφορετικές τεχνικές επιλογής της παραμέτρου block length. Ο σκοπός μας είναι να συγκρίνουμε αυτές τις τεχνικές έτσι ώστε να καταλήξουμε σε αυτή που δίνει το «ιδανικό» block length. Αυτό το πρόβλημα είναι πολύ μεγάλης σημασίας λόγω του γεγονότος ότι οι μέθοδοι block bootstrap εξαρτώνται πολύ από την επιλογή της παραμέτρου block length. Φέραμε εις πέρας μία εκτενή προσομοίωση Monte Carlo με διαφορετικά μοντέλα και διάφορες τιμές για τις παραμέτρους τους. Χρησιμοποιήσαμε τις γνωστές μεθόδους HHJ και NPPI που παρέχουν το ιδανικό block length μέσω του MSE όπως επίσης και μία μέθοδο των Politis και White (2004), για δύο από τις μεθόδους block length. Παρόλα αυτά, δε σταματήσαμε εκεί. Προτείναμε τέσσερις νέες μεθόδους ως τεχνικές επιλογής της παραμέτρου block length και τα αποτελέσματα ήταν ξεκάθαρα υπέρ τους. Τα δύο από αυτά τα κριτήρια που χρησιμοποιούν μία εκδοχή της συνάρτησης Quadratic loss ήταν στις περισσότερες περιπτώσεις αυτά που κυριάρχησαν. Επιπροσθέτως, συγκρίναμε τις μεθόδους block bootstrap όπως επίσης και τις τεχνικές επιλογής της παραμέτρου block length σε μοντέλα INAR, τα οποία είναι μοντέλα χρονολογικών σειρών για δεδομένα απαρίθμησης. Η εκτίμηση των παραμέτρων για αυτά τα μοντέλα είναι δύσκολη, έτσι θέλαμε να δούμε πόσο καλά μπορούν να αποδόσουν οι μέθοδοι μας. Τέλος, κάναμε την ανάλυση μας και σε ένα σετ πραγματικών δεδομένων όπου τα αποτελέσματα ήταν μοιρασμένα, αλλά και πάλι υπέρ των νέων προτεινόμενων κριτηρίων.

ACKNOWLEDGMENTS

I would like to express my thanks to my supervisor professor Xanthi Pedeli, whose precious advices lead me to the completion of this Thesis. Her guidance was constant. In addition, I would like to thank all of my professors in the MSc in Statistics program, who managed to provide us high quality knowledge. Last but not least, I owe much to my family and my friends whose support was non-negotiable.

CONTENTS

List of abbreviations

1. MBB : Moving Block Bootstrap
2. NBB : Non-Overlapping Block Bootstrap
3. CBB : Circular Block Bootstrap
4. SBB : Stationary Block Bootstrap
5. HHJ : Hall's et al. block length selection method
6. NPPI: Non Parametric Plug In block length selection method
7. PW : Politis and White's block length selection method.

LIST OF TABLES AND FIGURES

1. INTRODUCTION

In this project we will try to make clear the connection of bootstrap methods and time series. Time series are used in many scientific fields and we can apply their methods in various aspects of our lives and economy. Likewise most of statistical models our goal is to investigate their properties and estimate the parameters we need. In this place among other methods come bootstrap. Bootstrap methods have been extremely popular for two main reasons. Bootstrap is able to give valid estimations of sampling distribution and standard error in difficult situations, where the asymptotic distribution is intractable or very difficult to deal with. In addition, even in cases where, it is easy to estimate the parameters of interest and the standard errors of the estimators research has shown that bootstrap competes alternative estimation methods. The second reason, is the range of applicability. Bootstrap is a powerful tool, used widely in many estimation problems and despite being an intensive procedure is rather easy to implement, without the need of dealing with cumbersome distributions. Efron's pioneering article in 1979 expanded the horizons of statistical procedures and since then a quite large part of the literature has been devoted to bootstrap applications in various concepts. One of these concepts is bootstrap application to data showing a dependent structure. Time series are such type of data. It was rather easy for researchers to understand that Efron's bootstrap principle was unable to be used in time series. Many alternatives that are able to present the structure of time series data have been proposed in order to deal with this problem. The most popular among them are the Block Bootstrap methods. Block Bootstrap methods do not use every data point as an individual one, but rather makes use of blocks of neighboring data points in order to implement bootstrap.

The most popular block bootstrap methods are Non-Overlapping Block Bootstrap (NBB), Moving Block Bootstrap (MBB), Circular Block Bootstrap (CBB) and Stationary Block Bootstrap (SBB). However, block bootstrap methods critically depend on the size each block. The same methods may perform very differently if one uses different block sizes. In order to deal with this problem and be able to implement block bootstrap methods in the best possible way, there have been introduced various block length selection techniques.

In this particular project, we examine some block length selection methods via simulation studies and we also introduce four new methods (D-methods as we will call them from now on) that are rather intuitive. They make use of correlation structure of the series as well as some well-known loss functions. Our goal is to implement each of the block bootstrap methods in extensive simulation studies with different block length selection techniques in order to determine if any of the proposed methods is able to outperform the others.

For the simulation studies we used the classic time series models AR(1), MA(1) and ARMA(1,1) as well as the INAR(2) (We will talk about these models in the next chapter) model which has only integer values. Finally, we implement the aforementioned methods to a real dataset consisting of count data.

The rest of the thesis is structured as follows. In Sections 2.1-2.2 we describe the classic bootstrap principle in both parametric and non-parametric implementation. In Section 2.3 we briefly describe the reason i.i.d. bootstrap does not work for time series data and in Section 2.4 we present an extensive description of what literature has came up with in block bootstrap methods so far. Section 3 includes a description of block bootstrap methods and block length selection techniques. In Section 4, we describe INAR models which have some interesting definition. In Section 5, the simulation study is being carried out and the results are presented. Finally, in Section 6 there are some concluding remarks.

2. TIME SERIES

In the evolution of Statistics, many statistical methods were developed regarding independent or uncorrelated data. However, in many practical situations the case is that the data appear to be correlated. We may find out that in some situations data are autocorrelated or they are collected over time. A sequence of data that is collected at successive (discrete) time points is defined as time series. In this project, we confine ourselves with observations made at regularly spaced time-points without loss of generality, taking the interval between such points to be one (1). We shall write the series as $\{X_t, t \in \mathbb{Z}\}$. The main problem with time series is that observations are not independent and therefore their analysis comes in a different point of view than this of independent data. Time series come with their own unique concepts, such as stationarity, and models based on stationarity. We will make a brief introduction to such concepts as well as there will be a report to the concept of spectral desity and an introduction to INAR models whose observations are integer valued.

2.1. STATIONARITY

In order to deal with the problems arising with time series analysis, we often make assumptions about the structure of time dependence to make the analysis easier. The most common assumption is the assumption of stationarity. A stationary process has the property that the mean, variance and autocorrelation structure do not change over time. By this, we mean that although the series change over time, their statistical properties remain the same. So, we easily understand that stationarity has a significant role in time series analysis and that it is an important feature. Stationarity comes with two different definitions, strict stationarity and weak stationarity.

A time series is called strictly stationary when X_1^k and X_t^{t+k-1} have the same distribution for all k and t, i.e. the distribution of blocks of length k is time invariant. However, in most cases strict stationarity is too strong to be satisfied. Thus, we have the definition of weak stationarity which allows us to work in cases when strict stationarity can not be satisfied. Under weak stationarity, we imply that the shift invariance in time has to be only applied in the first moment and the cross-moment (the autocovariance). This means the process has the same mean at all time points, and that the covariance between the values at any two time points, t and $t-k$, depend only on k, the difference between the two times, and not on the location of the points along the time axis. Formally, a process $\{X_t, t \in \mathbb{Z}\}$ is weakly stationary if:

1. The first moment of X_t is constant, i.e. for every t $E[X_t] = \mu$
2. The second moment of X_t is finite for all t, i.e. for every t $E[X_t^2] < \infty$. Note that this also implies $E[X_t^2 - \mu] < \infty$ and therefore variance is finite.
3. Autocovariance depends only on the difference $u - v$, i.e. for evey $u, v, a, Cov(X_u, X_v) = Cov(X_{u+a}, X_{v+a})$.

So from the third statement, we get that $Cov(X_t, X_{t+k}) = \gamma_k$, where γ_k is indepentent of t. In addition, for the autocovariance function we have $\gamma_0 = Var(X_t)$. We also define the autocorrelation function of X_t at lag k as

$$\rho_k = \frac{\gamma_k}{\gamma_0} = corr(X_t, X_{t+k})$$

Before continuing with the presentation of models used in this project, we have to define the concept of White Noise (WN) which has an essential role in time series analysis. As White Noise, we define a stationary stochastic process ε_t with the following properties

1. $E[\varepsilon_t] = 0$, for every t,
2. $\gamma_o = E[\varepsilon_t]^2 = \sigma_\varepsilon^2$, for every t and
3. $\gamma_\kappa = E[\varepsilon_t, \varepsilon_{t-k}] = 0$, for every t and $k \neq 0$.

As we are discussing about stationary time series, another characteristic of a population stationary time series is the spectral density function. The spectral density is a frequency domain representation of a time series that is directly related to the autocovariance time domain representation. In essence, the spectral density and the autocovariance function contain the same information, but express it in different waves. Many time series show periodic behavior. This behavior can be very complex. Spectral analysis is a technique that allows us to discover underlying periodicities. The spectral density can be estimated using an object called periodogram, which is the squared correlation between our time series and sine/cosine waves at different frequencies spanned by the series. Technical details lie far beyond the scope of this project and we will not go deeper into spectral analysis.

2.2. Models based on stationarity

The class of models we mainly focus in this project is the class of autoregressive moving average (ARMA) models which is a subclass of linear time series models. In fact, every weakly stationary process is either a linear process or can be transformed to a linear process by subtracting a deterministic component. This result is known as Wold's decomposition (see Brockwell and Davis (1991), pp. 187-191). The time series X_t is a linear process if it has the representation

$$X_t = \sum_{i=-\infty}^{\infty} \psi_i \varepsilon_{t-i} \qquad (1.0)$$

for all t, where $\varepsilon_t \sim WN(0, \sigma^2)$ and ψ_i is a sequence of constants that are absolute summable, i.e. $\sum_{-\infty}^{\infty} |\psi_i| < \infty$.

Based on the represantation we got in (1.0), if only a finite number of ψ_i are nonzero, then we get the Moving Average process of order q (MA(q)) with th following definition

$$X_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \qquad (1.1)$$

where, $\varepsilon_t \sim WN(0, \sigma^2)$ and $\theta_1, \dots, \theta_q$ are constatnts. We can prove that in general, every MA(q) process is weakly stationary.

Similarly, we have that X_t represents an autoregressive process of order p if it can be defined as

$$X_t = \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \cdots + \varphi_p X_{t-p} + \varepsilon_t \qquad (1.2)$$

where, $\varepsilon_t \sim WN(0, \sigma^2)$ and $\varphi_1, \dots, \varphi_p$ are constants. It is important to note that unlike MA(q) process, AR(p) is not automatically stationary. For the special case of AR(1) model, we can prove that the process is stationary if $|\varphi| < 1$.

Another common process in time series modelling compining the aforementioned two is the Autoregressive Moving Average (ARMA). ARMA process models the observed time dependence using both autoregressive and moving average represantation. ARMA process is defined as follows

$$X_t = \varphi_1 X_{t-1} + \cdots + \varphi_p X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \quad (1.3)$$

where, for $\varphi_i{}'s$, $\theta_i's$ and ε_t we have the same proporties as in MA(q) and AR(p) process. Due to the fact that ARMA process is a compination of AR and MA processes we get that ARMA's properties will be a compination of the ones of AR and MA properties. So, in the case of the simple ARMA(1,1) model we get that it will be considered as stationary if the AR part is stationary.

In the next phase, we make an introduction to INAR models, which will also be used in our project. As we know, AR models are one of the most common approaches when we come up with modelling time series data. However, models of this type are suitable only in the case of continuous data. Its modification is the integer – valued autoregressive (INAR) process which is one of the most famous model in order to deal with dependent count data. The definition of an INAR(1) process $\{X_t : t = 0, \pm1, \pm2, \pm3, \dots\}$ is the following:

$$X_t = a \circ X_{t-1} + \varepsilon_t \qquad (1.4)$$

where,

$$a \circ X = \sum_{i=1}^{X} Y_i$$

In the above definition, X is a non-negative integer-valued random variable, while Y_i is a sequence of i.i.d. random variables – independent of X – such that $P_r(Y_i = 1) = 1 - P_r(Y_i = 0) = a$, ($a \in [0,1]$) and " $\circ$ " is the *binomial thinning operator* introduced by Steutal and Van Harn (1979). Lastly, ε_t is a sequence of uncorrelated non-negative integer-valued random variables having mean μ and finite variance σ^2. The most common approach is that ε_t

follows the Poisson distribution and then (and only then) X_t also follows the same distribution function. INAR(1) model was indepedently introduced by Al-Osh and Alzaid (1987)

One natural extension of INAR(1) model is the INAR(p) model, defined as

$$X_t = a_1 \circ X_{t-1} + \cdots + a_p \circ X_{t-p} + \varepsilon_t \qquad (1.5)$$

INAR(p) model was introduced by Alzaid and Al-Osh (1990) and Du and Li (1991), with different specifications of the thinning operators. Alzaid & Al-Osh (1990) assume a conditional multinomial distribution while Du and Li (1991) require conditional independence. Here we focus on the specification of Du and Li (1991). The assumptions for $\{\varepsilon_t\}$ are the same as we described above. In addition, for an INAR(p) model to be assumed as stationary we need to have that the sum of the thinning parameters is less than one.

In this part, in order to explain a little bit more INAR models a more precise definition of "$\circ$", which is conditional in X_{t-k} is that $a_k \circ X_{t-k}$ is equal to $\sum_{i=1}^{X_{t-k}} B_{i,k}$,where each collection $\{B_{i,k}, i = 1, 2, \dots, X_{t-k}\}$ consists of independently distributed Bernoulli random variables with parameter a_k. So, we could state that $a_k \circ X_{t-k}$ is the number of individuals that would have independently survived a Binomial experiment in a given period, where each of $X_{(t-k)}$ individuals has identical surviving probability a_k.

Furthermore, INAR(p) models strongly depends on the parametric assumption for the error term and unlike standard AR(1) model, one finds for the INAR(1) process that given X_{t-1} and ε_t, X_t is still a random variable. In addition, the estimation problem connected with the INAR(p) process is more complicated than that of the AR(p) process. The complication arises from the fact that the conditional distribution of X_t given $(X_{t-1}, \dots, X_{t-p})$ is the convolution of the distribution of the innovation process ε_t, and that of p binomial distributions with parameters X_{t-i} and a_i. Under these circumstances we will try to evaluate the block bootstrap methods we have described in order to estimate the parameters of some INAR(2) models and we will try to identify the most accurate method of detecting the block length parameter over the ones we have referred in this project. INAR models have various methods to estimate thinning parameters. In this project we used Conditional Least Squares with initial parameters for the algorithm those displayed by Y-W technique. These methods are described in detail in Du and Li (1991).

3. BOOTSTRAPPING

3.1. Introduction

Let's assume that we observe a finite sample $\boldsymbol{X} = (X_1, X_2, \dots, X_n)$ from a population with distribution function F and we wish to make inference about a parameter θ of the distribution in the basis of the sample we have in hand. Here is where bootstrap comes into play. The main idea is to make inferences about F by using an estimate of F based on the sample we have in hand. A well known estimate of F is the empirical distribution $\hat{F}_n$ we will see in the next section. The reason we can use $\hat{F}_n$ is that it can be shown that $\hat{F}_n$ is a consistent estimator of F and that $\sup_x |\hat{F}_n - \mathrm{F}| \xrightarrow{a.s.} 0$ according to Gilvenko-Cantelli Theorem. Moreover, the density function $\hat{f}_n$ that is associated with $\hat{F}_n$ will give probability $\frac{1}{n}$ to all observed points. So, as we will describe in the next section the idea of bootstrap is to use the observed sample to make inferences about F using $\hat{F}_n$ as an estimate of F. Generally, with the term *bootstrap* we can describe any statistical procedure which uses random resampling with replacement from an observed finite sample. Bootstrap is a computationally intensive method which was first introduced by Efron (1979) for i.i.d. data and managed to become very popular in the following years, as computers came into play for statisticians. In the next section, we will describe the bootstrap idea which consists of two main components, *parametric* and *non-parametric bootstrap*. The main difference between these two approaches is that in the first one we make some assumptions about the distributional form, whereas in the latter one we make use of the empirical distribution to make assumptions about the parameters of interest.

3.2. Parametric and non-parametric bootstrap

Let's first describe the non-parametric bootstrap. The main idea is the following. Suppose we observe a finite sample $\boldsymbol{X} = (X_1, X_2, \dots, X_n)$ of n independent and identically distributed (i.i.d.)

data points, where X_i are univariate random variables for $i = 1, \dots, n$, and we have no knowledge of the generating process, i.e. the probability distribution $F(\cdot)$ from which they came from. Therefore, we cannot simulate from F. Let us now denote by θ the parameter of interest we wish to estimate on the basis of $\mathbf{X}$. To do so, we calculate $\hat{\theta}$ from $T_n(\boldsymbol{X})$, where $T_n(\cdot)$ is the statistic of interest. Now, with the aim to access the accuracy of our estimation we denote with $\hat{F}_n$ the empirical distribution defined as

$$\hat{F}_n(x) = \frac{\#\, observations \leq x}{n} = \frac{1}{n}\sum_{i=1}^{n} I(x_i \leq x)$$

and we resample with replacement from $\hat{F}_n$. As mentioned in the previous section, $\hat{F}_n$ is a consistent estimator of F. The reason we do so, is that we have no knowledge of the distributional form our sample was generated. Once we obtain *n* resampled data points we can say that we have a bootstrap sample $X^* = ({X_1}^*, {X_2}^*, \dots, {X_n}^*)$, which has been generated from the resampling based on $\hat{F}_n(x)$.As we mentioned, the resampling was carried out with replacement and therefore some data points of the original sample have appeared more than one times in the bootstrap sample whereas other do not appear. Now, we estimate the parameter of interest θ from the bootstrap sample and get the bootstrap estimation ${\hat{\theta}_1}^*$. Therefore, the bootstrap estimate of $T(\hat{\theta})$ is a plug-in estimate that uses the empirical distribution function $\hat{F}$ is place of F. We repeat this process for a large number of times, say *B*, and get the bootstrap estimates ${\hat{\theta}_1}^*, {\hat{\theta}_2}^*, \dots, {\hat{\theta}_B}^*$ from $T_n(\boldsymbol{X}^*)$ corresponding to the B distinguished bootstrap samples. So, in the end of the process we have a random sample $\hat{\theta}^* = ({\hat{\theta}_1}^*, {\hat{\theta}_2}^*, \dots, {\hat{\theta}_B}^*)$ from the distribution of the parameter of interest and hence we are able to estimate the sampling distribution of parametric functions of interest.

The above can be summarized in the following steps:

1. Select B independent bootstrap samples ${\boldsymbol{X_1}}^*, {\boldsymbol{X_2}}^*, \dots, {\boldsymbol{X_B}}^*$, each consisting of *n* data points drawn with replacement from the observed sample.

2. Estimate the parameter of interest corresponding to each bootstrap sample

$$\widehat{\theta_t}^* = T_n(\boldsymbol{X_t}^*), \qquad t = 1, \dots , B$$

3. Approximate any quantity of interest from the random sample $\hat{\theta}^* = (\widehat{\theta_1}^*, \widehat{\theta_2}^*, \dots , \widehat{\theta_B}^*)$ of the distribution of the parameter of interest.

. On the other hand, the parametric bootstrap technique suggests that we have knowledge of the distribution form from which the observed data have been generated. We then estimate the unknown parameters of interest with a suitable technique (MLE for example) and in the next step we simulate a large number of samples from the known distribution using the estimated parameters obtained from the observed sample. These simulated samples are the bootstrap samples from which we derive

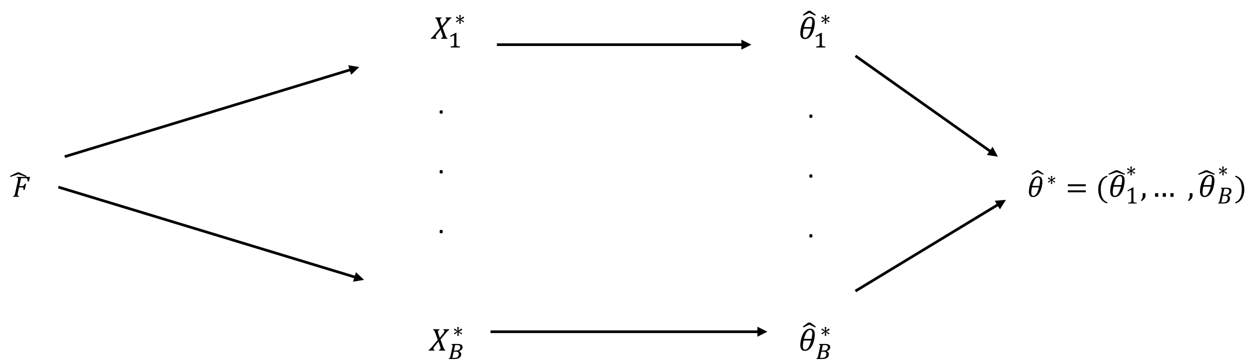


Figure 1 : Schematic representation of non-parametric bootstrap approach.

the required statistic $\hat{\theta}$. Parametric bootstrap could be considered as a simulation technique rather than a resampling one. The parametric bootstrap algorithm is summarized in the next steps:

i. From the observed data determine the unknown parameter(s) that best fit(s) the data from the known distribution family using appropriate estimation method.
ii. Generate B bootstrap samples $\{X_1^*, \dots, X_n^*\}$ by randomly resampling from the fitted distribution.
iii. For each bootstrap sample $\{X_1^*, \dots, X_n^*\}$ calculate the required statistic $\hat{\theta}_i$, $i = 1,2, \dots , n$.

For both cases, the bootstrap estimate of the parameter(s), denoted by $\hat{T}_n^*$, is defined by the mean of the bootstrap replications

$$T_n^* = \frac{1}{B}\sum_{i=1}^{B} T_n(X_t^*)$$

and the variance is estimated by the empirical variance of the replications

$$V(T_n^*) = \frac{1}{B-1}\sum_{i=1}^{B}(T_n(X_t^*) - T_n^*)^2$$

3.3. Why i.i.d. bootstrap does not work

Not long after Efron introduced bootstrap method it became clear that in cases where the data are dependent his method does not work sufficiently. If someone tries to implement i.i.d. bootstrap to time series the structure of the series will not be preserved in the bootstrap sample and the estimators provided will not be consistent. For example, if i.i.d bootstrap is directly applied to all sample points any correlation between the neighboring data will be lost.

3.4. Literature review to bootstrap methods for time series

After the realization that Efron's i.i.d. bootstrap does not work sufficiently for dependent data there have been developed many bootstrap and subsampling techniques in the literature in order to overcome this problem. These methods consist of both parametric and non-parametric approaches.

One of the most well-known and used approaches is the AR-sieve Bootstrap. This idea belongs to the category of model-based resampling. The key factor, is that the time series we have in hand is invertible and allows a representation by an autoregressive model of order infinity [(AR(∞)]. If this is the case, we choose an autoregressive order $\hat{p}$ for the original data and then –

since there is not serial autocorrelation between the residuals of an autoregressive model - we randomly resample the rescaled residuals like the classic i.i.d. bootstrap approach. The proposed way to select $\hat{p}$ is via the Akaike Information Criterion (AIC) due to the fact that AIC has the nice property of automatically selecting higher orders for more dependent models. In addition, Shibata (1980) show the optimality of AIC in prediction of AR(∞) models. Selection of $\hat{p}$ could be considered a tuning parameter problem, like the selection of the block length in Block Bootstrap methods we will discuss in the following section. In both cases, there is no standard method to plug-in the "best" estimation of the tuning parameter and it rather depends on the true underlying process, the statistic to be bootstrapped and the purpose the bootstrap is used for. In addition, according to B$\ddot{u}$lhmann (2002) the tuning parameter in AR-sieve bootstrap has a straightforward interpretation and it allows for back-testing. Apart from the fact that we will not examine AR-sieve bootstrap in this project we have to clarify that in many researches done by now it has been proved that in many cases -and if AR-sieve is suitable- this method has better performance than block bootstrap methods. However, it is not easy to judge if a specific dataset can be represented as an AR(∞) model and this is the main disadvantage of this approach. In the same manner works the *Residual Bootstrap*. This method can be implemented to both parametric and non-parametric models. As the method's name suggests the idea is that we either fit a known parametric model to the raw data (in the parametric situation) or if the data generating process has a non-parametric equation (in the non-parametric situation) and then resample the residuals assuming that they are i.i.d. in contrast to the original data . An alternative approach, for bootstrapping dependent data is the *Frequency Domain Bootstrap (FDB).* Kreiss and Lahiri (2012) have shown that this method is used in order to make inferences for population parameters of a second-order stationary process that can be expressed as a function of its spectral density. This method has the advantage of avoiding tuning parameters selection. However, it is rather sensitive to deviations from the model assumptions.

A rather extensive part of the literature concentrates in a variety of methods called block-bootstraps. The general idea of these methods is that since we are not able to directly apply i.i.d. bootstrap to the whole set of data we have to adjust the idea in a different manner in order to preserve the dependent nature of the data. In order to achieve that, we can bootstrap not single values, but blocks of consecutive data points. If we do so, we accomplish to preserve the dependence within the bootstrapped blocks and finally get consistent estimators. Block bootstrap methods have advantages and disadvantages compared to other bootstrap methods. Here, we will refer to the main components of these pros and cons, but we won't examine them thoroughly, since this is not the aim of this research.

The first well known approach was the so-called *Non-overlapping Block Bootstrap – NBB*, as we will call this method from now on - (Carlstein 1986) which uses blocks of consecutive data points with the restriction that these blocks are not allowed to have any identical point. Some years later, *Künch* (1989) , as well as *Liu and Sigh* (1992) in two independent researches came up with a similar procedure which is called *Moving Block Bootstrap (MBB)* which in contrast to NBB uses overlapping blocks in order to perform the bootstrap method. Lahiri (1999) noted that MBB estimators manage to achieve smaller MSEs than their NBB counterparts at any block size l – we will discuss about block sizes later. However, MBB gives an uneven weighting to the data points forcing the end points to be chosen less times than the others. The need to overcome this burden gave rise to *Circular Block Bootstrap (CBB)* and *Stationary Block Bootstrap (SBB)*, introduced both by Politis and Romano (1992, 1994) . Both these methods use an extended version of the dataset we have in hand by periodically extending the observed segment of the time series. CBB and SBB methods first take the observed series and reform them in a circle where the first one uses a fixed block length for the bootstrap procedure and the latter uses blocks of random length. Other popular variants of the block bootstrap are the *Matched Block Bootstrap (MaBB)* introduced by Carlstein et al. (1998) and *Tappered Block Bootstrap (TBB)* of Paparoditis and Politis (2001). The first one, tries to deal with the enhanced bias that can be presented in variance estimators, especially if the dependence in strong, due to the fact that neighboring blocks may not appear simultaneously in the bootstrap phase. To do so, it uses a stochastic mechanism that gives favor to blocks that are a priori more likely to be close to one another. In the same manner, TBB tries to reduce the edge effects caused by joining independent (with no same elements) blocks together by shrinking the boundary values in a block towards a common value (sample mean, for example). Both MaBB and TBB are more cumbersome to implement than MBB or CBB for example but achieve better variance estimators with a minor effect on the bias estimators. To understand the difference, let n be the length of the time series and l the block length. While, NBB, MBB, CBB and SBB have biases of the variance estimators of order $O(l^{-1})$ and variances of order $O(n^{-1}l)$, MaBB and TBB have the equivalent biases and variances of the same estimators being of order $O(l^{-2})$ and $O(n^{-1}l)$ respectively.

Another variant that worths mentioning is the *Local Block Bootstrap (LBB)* of Paparoditis and Politis (2002). This method tries to address the fact that in real life situations time series are usually not stationary, but rather they change their stochastic structure over time. In order to deal with it, LBB method resamples blocks that are close to each other. There are also some other variants of block bootstrap, like blocks of blocks bootstrap, but for the purposes of this research we will stick with the examination of the most popular NBB, MBB, CBB and SBB, which will be explicated in the following chapters.

In the previous paragraph, we referred at some points to a parameter that has a fundamental role in block bootstrap, the block length. Almost immediately after the arise of block bootstrap methods researchers had to answer the following question. Which number should define how many consecutive data points should we have in one block? This problem plays a crucial role in the performance of every variant of block bootstrap. In this project we try to address this problem, by examining the performance of various block length selection methods that already exist, as well as four new methods. Block length could be viewed as a tuning parameter (like the bandwidth in LBB or p in AR-sieve bootstrap) we have to choose in order to implement block bootstrap. The main disadvantage of block length selection problem is that this parameter cannot be interpreted and in addition, we do not have – in contrast to AR-sieve's bootstrap $\hat{p}$ selection – any diagnostic tools to check if our choice is suitable. However, since block bootstrap methods are the most famous and widely-used for dependent data there have been developed various methods for the selection of block length.

4. BLOCK BOOTSTRAP

4.1. Block Bootstrap Methods

4.1.1. Moving Block Bootstrap

The Moving Block Bootstrap (MBB) is a bootstrap procedure that mimics Efron's i.i.d. bootstrap by resampling of blocks $\boldsymbol{X_{t+1}}, \boldsymbol{X_{t+2}}, \dots, \boldsymbol{X_{t+l}}$ of consecutive observations. The MBB process resamples the aforementioned blocks in order to construct an estimator $\hat{\theta}$ of the parameter of interest. MBB was first introduced by Künsch (1989) and Liu and Singh (1992) in two separate researches aiming to overcome the burden of the failure of single-value bootstrap to produce valid approximations in the presence of dependence. By resampling blocks of consecutive observations we are able to preserve the dependence structure at short lag distances and carry this information in the bootstrap estimators.

The MBB scheme is the following. Suppose that $\{\boldsymbol{X_t}\}_{\boldsymbol{t \in N}}$ is a stationary weakly dependent time series and $\boldsymbol{X_n} \equiv \{\boldsymbol{X_1}, \dots, \boldsymbol{X_n}\}$ are observed. Next, let l be an integer satisfying $1 \leq l \leq n$. Now, define the overlapping blocks $\boldsymbol{B_1}, \dots, \boldsymbol{B_N}$ of length l contained in $\boldsymbol{X_n}$ as

$$B_1 = (X_1, X_2, \dots, X_l),$$

$$B_2 = \qquad (X_2, \dots, X_l, X_{l+1}),$$

$$\dots \qquad\qquad \dots$$

$$B_N = \qquad\qquad\qquad (X_{n-l+1}, \dots, X_{n-1}, X_n).$$

where $N = n - l + 1$. Then, if l divides n set $b = n/l$. In order to create the MBB samples we select b blocks at random with replacement from the full selection $\boldsymbol{B} = (\boldsymbol{B_1}, \dots, \boldsymbol{B_N})$ of blocks and then concentrate them serially in the order we resampled them. Since, each block has l elements we now have $n = b \cdot l$ bootstrap observations $X_1^*, X_2^*, \dots, X_n^*$. If the number of blocks N is not a multiple of l we resample $b = [N/l] + 1$ blocks, but we only use some portion of the last block so to have n bootstrap observations. Some authors propose to use the whole bootstrap sample, but we will stick with our choice, because we want every bootstrap sample having the same length with the original series.

In this phase we have to point out that if we choose $l = 1$ then we have the classic Efron's bootstrap. However, for a valid approximation in the MMB it is usually requested for dependent data to choose $l \to \infty$ and $n^{-1} \cdot l \to 0$ as $n \to \infty$. To say in other words, l is assumed to increase as the length of the observations is increased, but not very fast.

4.1.2. Non-Overlapping Block Bootstrap

The next blocking rule to be considered is based on Carlstein's (1986) paper. In contrast to MBB's methodology, NBB uses distinct only blocks. Every block has it's own consecutive observations and every next or previous block does not share the same information. It is immediate that under this context, NBB uses less blocks compared to MBB to generate the bootstrap replicates.

Considering again the scheme for the underlying process and block length parameter l as in MBB method, we define the non-overlapping blocks of consecutive observations $\boldsymbol{B_1}, \dots, \boldsymbol{B_N}$ of length l as:

$$
\begin{aligned}
B_1 &= (X_1, X_2, \dots, X_l), \\
B_2 &= (X_{l+1}, \dots, X_{2l-1}, X_{2l}), \\
&\dots \\
B_N &= \left(X_{(b-1)l+1}, \dots, X_{n-1}, X_n\right).
\end{aligned}
$$

where b is set to be the result of division between n and l ($b = n/l$) and N (= b) is the total number of blocks constructed. Now, we select b blocks at random, with replacement, from the full selection $\boldsymbol{B} = (\boldsymbol{B_1}, \dots, \boldsymbol{B_N})$ and likewise MBB we concentrate them serially to get a bootstrap replicate of the series.

At this point, we have to state that due to the non-overalapping principle that NBB uses to construct the bootstrap observations it is easier to analyze the theoretical properties of NBB estimators of population parameters compared to MBB's ones. However, NBB estimators have typically higher MSE's compared to their MBB counterparts at any block size (Lahiri, (1999)). Furthermore, NBB methodology has the disadvantage of being inflexible with block size choices. In order to construct the blocks collection, one has to specify a block length parameter l that divides exactly the number n of observations. This characteristic restricts anyone who want to analyze a time series dataset with NBB, as he gets a limited set of values for the block length parameter to choose.

4.1.3. Circular Block Bootstrap

Circuclar Block Bootstrap (CBB) is another block bootstrap procedure which extends MBB in the manner of correcting MBB's incapability to assign the same probability in every data point to get resampled. In the way MBB's blocks are constructed the very few first and last observations appear in less blocks in the full collection of blocks $\boldsymbol{B} = \{\boldsymbol{B_1}, \dots, \boldsymbol{B_N}\}$.

So as to explain the process, let's assume again that we have a $\{\boldsymbol{X_t}\}_{\boldsymbol{t \in N}}$ being a weakly dependent stationary process and we observe $\boldsymbol{X_n} \equiv \{\boldsymbol{X_1}, \dots, \boldsymbol{X_n}\}$. We first "wrap" the data around a "circle", that is, to define for $i > n$ $X_i \equiv X_{i_n}$, where $i_n = i(mod n)$ and $X_0 \equiv X_n$. Now, we again define the set of blocks $\boldsymbol{B} = \{\boldsymbol{B_1}, \dots, \boldsymbol{B_N}\}$. Note that in CBB we have $N = n$ such blocks, regardless the integer l we choose for block length. In the next phase, if n is a multiple of l we let $b = N/l$ blocks be resampled with replacement from the full set of B of blocks. Otherwise, we choose $b = [N/l] + 1$ blocks to be resampled.

As we see, CBB method is able to give the same selection probability for every data point we have which is a very convenient characteristic.

4.1.4. Stationary Block Bootstrap

The last block booststrap procedure we are going to examine in detail is the Stationary Block Bootstrap. The aforementioned block bootstrap procedures share the same scheme. Divide the observed series into blocks (overlapping or non-overlapping) of data points and then resample with replacement from the full collection of blocks ending up with a pseudo-series. The fact is, that neither of these methods is able to come up with stationary block bootstrap samples which is a property we would naturally wish to have. This problem is addressed with Stationary Block Bootstrap (SBB), introduced by Politis and Romano (1994).

With the SBB procedure the pseudo-series produced try to mimic the behavior of the observed sample retaining the stationarity property. In order to do so, SBB has the following scheme.

Let $\boldsymbol{B_{i,l}} = \{\boldsymbol{X_i}, \dots, \boldsymbol{X_{i+l-1}}\}$ be a block consisting of l observations starting from $\boldsymbol{X_i}$. In addition, we assume the same "wrap" of the observed sample as in CBB method, so in the case of $i > n$ we denote $X_i \equiv X_{i_n}$, where $i_n = i(mod n)$ and $X_0 \equiv X_n$. Let also p be a fixed number in $[0,1]$. Independent of $X = \{X_1, \dots, X_n\}$ let $L_1, L_2, \dots$ be a sequence of i.i.d random variables having the geometric distribution, so that the probability of the event $\{L_i = m\}$ is $(1-p)^{m-1} \cdot p$ for $m = 1,2, \dots$. Independent of the L_i and the X_i, let $I_1, I_2, \dots$ be a sequence of i.i.d random variables having the discrete Uniform distribution of $\{1, \dots, N\}$. Now, in order to produce a bootstrap sample $X^* = \{{X_1}^*, \dots, {X_n}^*\}$ repeatedly sample blocks of random length by the prescription $B_{I_1,L_1}, B_{I_2,L_2}, \dots$ and place them in a sequence until n observations in the pseudo-series have been generated. The first L_1 observations of the bootstrap sample $X_1^*, \dots, X_n^*$ are determined by the first block B_{I_1,L_1} of observations $X_{I_1}, \dots, X_{I_1+L_1-1}$ and so on the next blocks. As we mentioned before, we stop when $N = n$ observations have been generated, because SBB enables the generation of arbitrary length blocks.

In order to give an example of how SBB works let's assume we have a set of observations X being the numbers from 1 to 10, so $X = \{1,2, \dots, 10\}$. We first take a number at random between 1 and 10, let's assume this number is 3, and then we obtain an observation from the geometric distribution with the probability of success we have set. In our case we assume that this observation is the number 4. Then the first block we obtain is $B_1 = \{3,4,5,6\}$. In our second try we suppose that the random number between 1 and 10 is taken to be 6 and the corresponding observation from the geometric distribution is 2. Then the second block is $B_2 = \{6,7\}$. So, we have six observations already and we need another four in order to get a bootstrap sample. The next pair of numbers appearing from the random selection of a number between 1 and 10 and from the geometric distribution is 2 and 5 respectively. Then the third block will be $B_3 = \{2,3,4,5,6\}$. So, the bootstrap sample will be $B^* = \{3,4,5,6,6,7,2,3,4,5\}$. We have excluded the number 6 appearing in the third block due to the fact that we needed the first ten out of the total of eleven numbers produced.

4.2. Block Length Selection Methods.

4.2.1. Introduction

As we have already mentioned, the block length parameter l plays a crucial role in block bootstrap methods. However, it is rather cumbersome to choose the optimal one. In order to deal with this problem there have been developed several techniques. Block length parameter is the tuning parameter in block bootstrap and unfortunately we do not have a straightforward interpretation for it. According to several literature reviews (see $Bülhmann\ and\ Künch$ (1999) for example) the block length has to increase with sample size and the asymptotic formula for selecting the optimal block length for estimating standard errors is $l = Cn^{\frac{1}{3}}$. The constant C depends on the statistic to be bootstrapped as well as the dependence among the observations and it is typically very difficult to be computed. For this reason there have been developed some techniques which try to estimate block length l and provide a suitable environment for block bootstrap implementation.

In general, as a tuning parameter, selecting the optimal block length is achieved by balancing the error about the mean against bias, so to minimize mean square error (MSE). In particular, as the block length l increases, the bias of a block bootstrap estimator decreases whereas the variance increases. However, Hall et al. (1995) state that this is not always the case, as for large values of l one may observe an increase in both bias and variance of an estimator.

In addition, the optimal block length is rather intractable and in most cases the proposed block length parameter estimator is sub-optimal.

A common approach to the problem is to use $l = n^{\frac{1}{3}}$ to be the block length parameter. Hall et al. (1995) admit that optimal block length should be considered $n^{\frac{1}{3}}, n^{\frac{1}{4}}$ and $n^{\frac{1}{5}}$ depending on the context with $n^{\frac{1}{3}}$ being the optimal block length for the bias-variance situation we are examining in this project. However, it is not validated that we may always use $C = 1$ from the asymptotic formula. We will examine though, if this method is able to compare or even outperform more sophisticated methods with theoretical background.

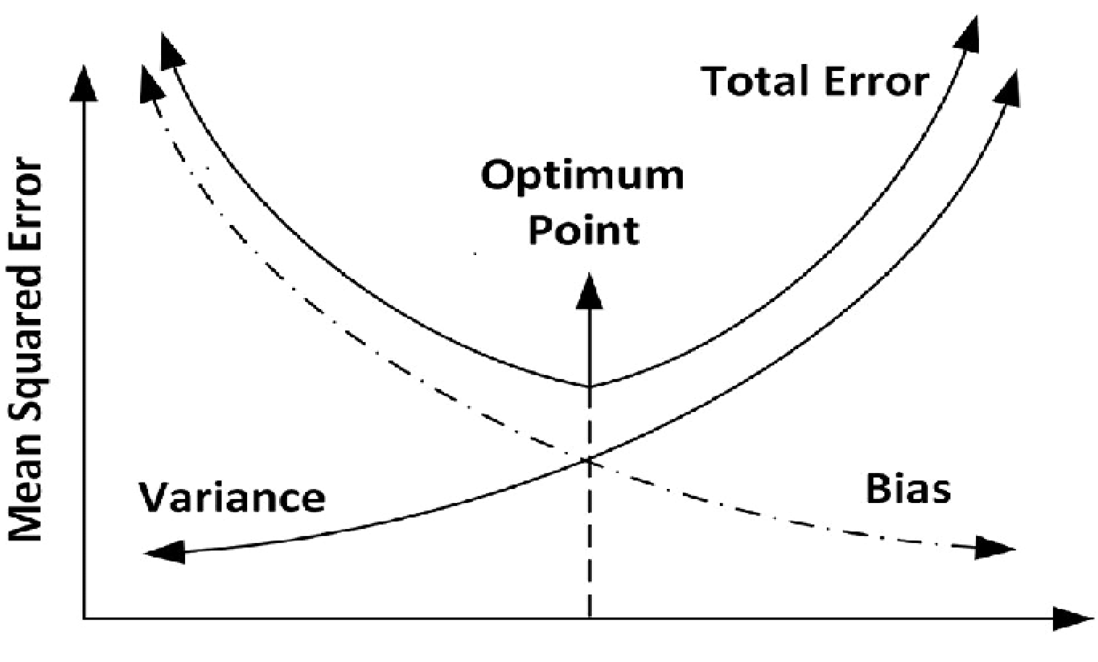


Figure 2. :Schematic representation of Bias-Variance trade-off to get the optimal MSE. (source : Gungor, Osman Erman & Al-Qadi, Imad & Mann, Justan. (2018). Detect and charge: Machine learning based fully data-driven framework for computing overweight vehicle fee for bridges. Automation in Construction. 10.1016/j.autcon.2018.09.007.)

4.2.2. Basic framework for MSE-optimal block size

In the case of MSE optimal block length holds the following principle according to Lahiri (2003). Suppose that $\hat{\varphi}_n \equiv \hat{\varphi}_n(l)$ is a block bootstrap estimator of a population parameter φ_n with the corresponding block length l. It is widely accepted that for many $\varphi_n{}'s$, the variance of the bootstrap estimator of $\hat{\varphi}_n(l)$ is an increasing function of the block length l, while its bias is a decreasing function of l. Under some regularity conditions, variance and bias admit expansions of the form

$$n^{2a}Var\big(\hat{\varphi}_n(l)\big) = C_1 n^{-1} l^r + o(n^{-1} l^r), \qquad as\ n \to \infty \qquad (2.1)$$

and

$$n^{a}Bias\big(\hat{\varphi}_n(l)\big) = C_2 l^{-1} + o(l^{-1}), \qquad as\ n \to \infty \qquad (2.2)$$

over a suitable set of possible block lengths l, where C_1 and C_2 are population parameters, $r \geq 1$ is an integer and $a \in [0, \infty)$ is a known constant. In the case of $\hat{\varphi}_n = Var\big(\hat{\theta}_n\big)$ where $\hat{\theta}_n$ is the sample mean it holds that r = 1 and α = 0. The problem here is that is very difficult to get exact expressions of C_1 and C_2. However, some of the proposed methods do not need these exact expressions. Using the above expansions for variance and bias we can get the expansion of the MSE of $\hat{\varphi}_n(l)$ as follows,

$$n^{2a}MSE(\hat{\varphi}_n(l) = C_1 n^{-1} l^r + C_2^2 l^{-2} + o(n^{-1} l^r + l^{-2}), \qquad as\ n \to \infty$$

So, the MSE-optimal block length $l^0 \equiv l_n^0$ has the form

$$l^0 \sim \left[\frac{2C_2^2}{rC_1}\right]^{\frac{1}{r+2}} n^{\frac{1}{r+2}}$$

At his point we have to add some explanation in some of the symbols we use. As may someone have observed in some equations (for example 2.1,2.2 and 2.3) we add a term of form $o(.)$, while in others we add a term of the form $O(.)$. The difference between these two symbols is that when we use the “big-O” we mean that the added term is of the same order as the term inside the parentheses, while when we use the “little-o” we mean that the added term is genuinely smaller than the term inside the parentheses.

4.2.3. Non Parametric Plug-In Rule

Lahiri et al. (2007) proposed a new alternative to the empirical choice of the optimal block length in block bootstrap for various population quantities. Thus, they stated that this method may be referred as a generalized «plug-in» rule. One key advantage of this method is that although it is considered a plug-in rule differs from other plug-in principles in the sense that we do not need to derive explicit analytical expressions of the constants in the leading term of optimal block length. In order to estimate these constants NPPI method uses nonparametric resampling techniques. Another important advantage is that due to the fact that this method – in contrast to HHJ – does not require repeated resampling, thus presenting computation efficacy. In their paper Lahiri et al. proposed and evaluated the method for the MBB. However, with some modifications the method is applicable also with the CBB. So, we will present the method in the basis of MBB, but in the application we will also use CBB.

In order to get through with this method, having in mind (2.1) and (2.2) we get for the population parameters that

$$C_1 \sim [nl^{-r}]\{n^{2a}Var\left(\hat{\varphi}_n(l)\right)\}$$

and

$$C_2 \sim [l]\{n^{a}Bias(\hat{\varphi}_n(l))$$

Now, suppose that we could estimate consistently the population quantities $Var\left(\hat{\varphi}_n(l)\right)$ and $Bias(\hat{\varphi}_n(l))$ by $\widehat{VAR}$ and $\widehat{BIAS}$. Then we would get consistent estimators of C_1 and C_2 in the form

$\hat{C}_1 = [nl^{-r}]\{n^{2a}\widehat{VAR}\}$ and $\hat{C}_2 = [l]\{n^{a}\widehat{BIAS}\}$.

Then, the NPPI estimator of the MSE-optimal block length l^0 is given by

$$\hat{l}^{\,0} = \left[\frac{2\hat{C}_2^2}{r\hat{C}_1}\right]^{\frac{1}{r+2}} n^{\frac{1}{r+2}}$$

For the variance estimator $\widehat{VAR}$, NPPI method uses the JAB (Jackknife after Bootstrap) variance estimator.

To describe the JAB method, let $m \equiv m_n$ be an integer such that $m \to \infty$ and $\frac{m}{n} \to 0$ as $n \to \infty$ and let $M = N - m + 1$, where $N = n - l + 1$, ($N = n \; for \; CBB$). The parameter m denotes the number of blocks that will be deleted for every i in $\{1, \dots, M\}$. So, for every i in $\{1, \dots, M\}$ we set $I_i = \{1, \dots, N\} \setminus \{i, \dots, i + m - 1\}$ to be the set of all possible blocks considered under a suitable pilot block length parameter once we have deleted the m blocks $\{B_i, \dots, B_{i+m-1}\}$. So, for every i in $\{1, \dots, M\}$ we define a jackknife version $\hat{\varphi}_n^{(i)} \equiv \hat{\varphi}_n^{(i)}(l)$ of $\hat{\varphi}_n(l)$. Once we have done that, we just continue with the computation of the i-th block-deleted jackknife point value $\hat{\varphi}_n^{(i)}$ by resampling b blocks (again, b $\approx n/l$) randomly from the reduced collection of blocks $\{B_j, j \in I_i\}$ and then compute the corresponding block bootstrap estimator. In our case we use the population variance as the bootstrap estimator. So, once we have applied the suitable block bootstrap method to the whole set of blocks, we return to this set and for every i in $\{1, \dots, M\}$ we delete the corresponding blocks as we have described and again perform calculations without the need of additional resampling. This is where the key advantage of this method stands. With the jackknife point values we have extracted with the above method we get the variance estimator :

$$\widehat{VAR}_{JAB}(\hat{\varphi}_n) = \frac{m}{N - m} \frac{1}{M} \sum_{i=1}^{M} \left(\tilde{\varphi}_n^{(i)} - \hat{\varphi}_n\right)^2$$

where, $\tilde{\varphi}_n^{(i)} = m^{-1}[N\hat{\varphi}_n - (N - m)\hat{\varphi}_n^{(i)}]$

For the bias estimator, Lahiri et.al. (2007) suggest the following scheme. Suppose that the expected value of $\hat{\varphi}_n$ admits an expansion of the form $E[\hat{\varphi}_n(l)] = \varphi_n + \frac{C_2}{n^a l} + o(n^{-a} l^{-1})$ with

$C_2 \in \mathbb{R}$ and $C_2 \neq 0$. Then the leading term in the bias of $\hat{\varphi}_n$ is given by $\frac{C_2}{n^a l}$. Then, we consider the bootstrap estimators $\hat{\varphi}_n(l)$ and $\hat{\varphi}_n(2l)$. Then, by using the above expansion we have that

$$2E\left[\hat{\varphi}_n(l) - \hat{\varphi}_n(2l)\right] = 2\left[\left\{\varphi_n + \frac{C_2}{n^a l} + o(n^{-a} l^{-1})\right\} - \left\{\varphi_n + \frac{C_2}{2n^a l} + o(n^{-a} l^{-1})\right\}\right] =$$

$$= \frac{C_2}{n^a l} + o(n^{-a} l^{-1}) = Bias\big(\hat{\varphi}_n(l)\big) + o(n^{-a} l^{-1})$$

they concluded to $\widehat{BIAS} \equiv \widehat{BIAS}(\hat{\varphi}(l) \equiv 2(\hat{\varphi}_n(l) - \hat{\varphi}_n(2l))$ as an estimator of $Bias\big(\hat{\varphi}_n(l)\big)$. In section 4 of the same article they show the consistency of both $\widehat{VAR}$ and $\widehat{BIAS}$.

To sum up, the proposed NPPI algorithm can be described as following :

1. Compute the block bootstrap estimator of $\hat{\varphi}_n(l)$ and $\hat{\varphi}_n(2l)$ with l being the pilot block length and $l = n^{\frac{1}{4}}$.
2. Use the bootstrap replicates from the computation of $\hat{\varphi}_n(l)$ to compute the JAB variance estimator $\widehat{VAR}$ for $\hat{\varphi}_n(l)$ with $m = n^{\frac{1}{3}} \cdot l^{\frac{2}{3}}$
3. Compute $\hat{C}_1 = n l^{-1} \widehat{VAR}$ and $\hat{C}_2 = 2l(\hat{\varphi}_n(l) - \hat{\varphi}_n(2l))$

The NPPI optimal block bootstrap estimator is given by $\hat{l}^0 = \left[\frac{2\hat{C}_2^2}{\hat{C}_1}\right]^{\frac{1}{2}} n^{\frac{1}{2}}$

4.2.4. Hall's et al. Method

Hall, Horowitz and Jing with their paper in 1995 tried to address the problem of optimal block length selection for block bootstrap in dependent data via a new MSE-optimization method. As we have already mentioned block size is taken to be equal to $n^{\frac{1}{3}}, n^{\frac{1}{4}}$ and $n^{\frac{1}{5}}$ in the cases of bias or variance estimation, one-sided distribution function estimation and two-sided distribution function estimation, respectively. In the next paragraphs, we will describe HHJ method in detail and we will outline the computational boundaries this method displays.

Let ψ denote the population parameter of interest. In our case ψ is the population variance. The optimal block size in this method will be first computed for a time series of length $m < n$, where m is the length of the reduced series and n is the length of the original time series with a user-defined pilot block length l. So, in the first step we have to select m with the requirement $0 < m \ll n$ and $\frac{m}{n} \to 0$ as $n \to \infty$. Then, let L denote all the $m - n + 1$ of length m obtainable of the original time series, and apply the block bootstrap in every subseries of L. Each application of block bootstrap produces a point estimate of ψ, say $\widehat{\psi_i}$ $1 \leq i \leq n - m$. If we denote with $\hat{\psi}$ the point estimate of ψ from the entire dataset computed with a plausible pilot block length l' we estimate the mean squared error in every sample of size m of L by computing the averages of the square of the differences $\widehat{\psi_i} - \hat{\psi}$. We now have to apply the block bootstrap over a series of possible block lengths in every subseries of L and the block length which achieved the smaller MSE value will be denoted as $\hat{l}_m$. Then, in order to choose the MSE-optimal block length for the entire dataset we will get l to be $\left(\frac{n}{m}\right)^{\frac{1}{3}} \cdot \hat{l}_m$ (or $\left(\frac{n}{m}\right)^{\frac{1}{k}} \cdot \hat{l}_m$ with k = 1,2 if we were interested for one-sided distribution function estimation and two-sided distribution function estimation, respectively) . It is proposed by the authors to iterate the above procedure in order to get an updated version of the block length choice and also to minimize the effect of the initial pilot block length selection. In our application we seek for convergence via the iterations of the algorithm in the sense of selecting the same block length for the entire series in two consecutive iterations. However, due to these iterations and the cross-validation nature of the algorithm this method suffers from computational inefficiency. In addition, it is possible for the iteration scheme to not converge and in this case, we will not be able to get the proposed optimal block length. The above statements will be examined via extensive Monte Carlo simulation. This method requires from us to provide a value to another parameter besides the pilot block length and this is m (the length of the subseries MBB will be implemented). However, researches have shown that the choice of m is not crucial.

At this point it would be useful to mention that according to Nordman et al. (2007) if we denote by $\hat{b}_{n,OL}$ the MSE-optimal block size proposed by the HHJ method for MBB and by $\hat{b}_{n,NOL}$ the corresponding block size for NBB, then the relationship between those two estimates is

$\hat{b}_{n,OL} = \hat{b}_{n,NOL}\left(\frac{3}{2}\right)^{\frac{1}{3}}\left(1+o(1)\right)$. So in order to implement the NBB method we will first find the corresponding MSE-optimal block size estimation from the HHJ method and then we will implement NBB with block length being less than or equal to $l_{hhj}\left(\frac{2}{3}\right)^{\frac{1}{3}}$, where l_{hhj} is the block length proposed by HHJ. We will call this method Nordman's rule.

4.2.5. Politis and White's Method

The last well-known method is due to Politis and White (2004). This method deals only with the sample mean estimator for the cases of CBB and SBB. Lahiri (1996b) gave the following theorem.

Theorem 1 (Lahiri, 1996b): *Assume* $E[X_t]^{6+\delta} < \infty$, *and* $\sum_{k=1}^{\infty} k^2 \left(a_\chi(k)\right)^{\frac{\delta}{6+\delta}} < \infty$ *for some* $\delta > 0$. *If* $b \to \infty$ *as* $N \to \infty$ *but with* $b = o(N^{\frac{1}{2}})$ *then we have :*

1. $Bias(\hat{\sigma}^2_{b,CB}) = E\hat{\sigma}^2_{b,CB} - \sigma^2_\infty = -\frac{1}{b}G + o(\frac{1}{b})$
2. $Var(\hat{\sigma}^2_{b,CB}) = \frac{b}{N}D_{CB} + o\left(\frac{b}{N}\right)$
3. $Bias(\hat{\sigma}^2_{b,SB}) = E\hat{\sigma}^2_{b,SB} - \sigma^2_\infty = -\frac{1}{b}G + o(\frac{1}{b})$
4. $Var(\hat{\sigma}^2_{b,SB}) = \frac{b}{N}D_{SB} + o\left(\frac{b}{N}\right)$

where, $\hat{\sigma}^2_{b,CB}$ and $\hat{\sigma}^2_{b,SB}$ are the estimators of σ^2_∞ under CBB and SBB respectively, $D_{CB} = \frac{4}{3}g(0)$, $D_{SB} = (4g(0) + \frac{2}{\pi}\int_{-\pi}^{\pi}(1+\cos w)g^2(w)dw$ and $G = \sum_{k=-\infty}^{\infty}|k|R_k$, $R(s)$ is the autocovariance sequence and $g(w) := \sum_{s=-\infty}^{\infty} R(s)\cos(ws)$ is the spectral density function.

From the above theorem it stands that for the SBB we have

$$MSE\left(\hat{\sigma}^2_{b,SB}\right) = \frac{G^2}{b^2} + D_{SB}\frac{b}{N} + o(b^{-2})$$

and for the CBB we have

$$MSE\left(\hat{\sigma}^2_{b,CB}\right) = \frac{G^2}{b^2} + D_{CB}\frac{b}{N} + o(b^{-2}) + o\left(\frac{b}{N}\right)$$

In order to minimize these two quantities we have to choose the block length parameter to be

- $l_{opt,SB} = \left(\frac{2G^2}{D_{SB}}\right)^{\frac{1}{3}} N^{\frac{1}{3}}$, where N is the total number of observations, and
- $l_{opt,CB} = \left[\left(\frac{2G^2}{D_{CB}}\right)^{\frac{1}{3}} N^{\frac{1}{3}}\right]$, where [x] indicates the closest integer number to x.

In order to get the estimates of these block length parameters, one has to estimate accurately the parameters D_{CB}, D_{SB} and G and plug-in these estimators into the above equations. However, the estimation of these parameters lies beyond this project's target. For this reason, we will only use the algorithm from the *blocklength* library in R and we will compare the results with the other methods.

It would be useful to note that Politis and White came with the following result in the same article. First, they introduced the asymptotic relative efficiency (ARE) of SBB relative to CBB as

$$ARE_{\frac{CB}{SB}} := \lim_{N\to\infty}\frac{MSE_{opt,CB}}{MSE_{opt,SB}}$$

where, $MSE_{opt,CB} := \inf_b MSE(\hat{\sigma}^2_{b,CB})$ and $MSE_{opt,SB} := \inf_b MSE(\hat{\sigma}^2_{b,SB})$.

Then, according to some assumptions they made they concluded that

$$0.331 \leq ARE_{\frac{CB}{SB}} \leq 0.481$$

We can interpret this result as the price we must pay in accuracy in order to get a stationary bootstrap sample.

4.2.6. New Methods

In addition to the already proposed block length selection methods we have already mentioned, we introduce in this project four (4) new methods for selecting the optimal block length according to some characteristics of the dataset we have in hand. These four methods utilize some well-known functions and use a cross-validation type approach to get the estimation of optimum *l*.

In the first step we introduce the functions we use.

1. Entropy loss function : $\Delta_1^{(l)} = tr\left(\hat{\Gamma}_l^{-1}\tilde{\Gamma}_l\right) - \log\left|\hat{\Gamma}_l^{-1}\tilde{\Gamma}_l\right| - l$
2. Kullback – Leibler function : $\Delta_2^{(l)} = tr\left(\tilde{\Gamma}_l^{-1}\hat{\Gamma}_l\right) - \log\left|\tilde{\Gamma}_l^{-1}\hat{\Gamma}_l\right| - l$
3. Quadratic loss function 1. : $\Delta_3^{(l)} = tr\left(\hat{\Gamma}_l^{-1}\tilde{\Gamma}_l - I_l\right)^2$
4. Quadratic loss function 2. : $\Delta_4^{(l)} = tr\left(\tilde{\Gamma}_l^{-1}\hat{\Gamma}_l - I_l\right)^2$

where

$$\hat{\Gamma}_l = \begin{bmatrix} \gamma(0) & \gamma(1) & \cdots & \gamma(l-1) \\ \gamma(1) & \gamma(0) & \cdots & \gamma(l-2) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma(l-1) & \gamma(l-2) & \cdots & \gamma(0) \end{bmatrix}$$

is the estimated autocovariance matrix (up to lag *l*-1) based on all data, and

$$\tilde{\Gamma}_l = \frac{1}{n-l+1}\sum_{j=1}^{n-l+1} \tilde{\Gamma}_l^{(j)}$$

is the estimated autocovariance matrix based on moving blocks of l observations. Here, $\tilde{\Gamma}_l^{(j)}$ stands for the autocovariance matrix of the j – th block.

Now, let $l_1, \dots, l_k, \dots, l_K$, with K being a suitable number that we have set – in our case we start with $l_1 = 2$ and end up in $l_K = 100$, be the ordered series of l – values (the block lengths under consideration). For each value l_k, we compute the loss function $\Delta_i^{(l_k)}$, $i = 1, 2, 3, 4$. Define the optimal block length $\hat{l}$ as the value l_k which provides the maximum $\Delta_i^{(l_k)}$ in the interval between min $(\Delta_i^{(l_1)}, \Delta_i^{(l_2)}, \dots, \Delta_i^{(l_K)})$ and the 15% percentile of the empirical distribution of $\Delta_i^{(l_1)}, \Delta_i^{(l_2)}, \dots, \Delta_i^{(l_K)}$. This approach is inspired by Chang and Tsay (2010) and has been used in the past by Pedeli at al. (2015). Actually, the functions described above are used as measures of the accuracy of the autocovariance matrix estimate and our choice of the block length is driven by this criterion (i.e. the accuracy). For the rest of the thesis, each one of these methods will be called as D1, D2, D3, D4 respectively.

5. SIMULATION STUDY

In order to evaluate our block length selection methods we run an extensive series of Monte Carlo simulations over four different models (with various parameters) and an application to a real dataset. The models used were :

1. AR(1) model

$$X_t = \varphi X_{t-1} + \varepsilon_t$$

2. MA(1) model

$$X_t = \varepsilon_t + \theta\varepsilon_{t-1}$$

3. ARMA(1,1) model

$$X_t = \varphi X_{t-1} + \theta\varepsilon_{t-1} + \varepsilon_t$$

In both AR(1) and MA(1) models we used the values $-0.9, -0.5, -0.1, 0.1, 0.5, 0.9$ for the parameters φ_1 and θ_1 respectively and ε_t are i.i.d. normally distributed , i.e. $\varepsilon_t \sim N(0, \sigma^2)$. For the ARMA(1,1) we investigate two models with the values $(\varphi, \theta) = (-0.8, 0.5)$ for the first model and the values $(\varphi, \theta) = (0.5, -0.8)$ for the second one. Again, we have $\varepsilon_t \sim N(0, \sigma^2)$. For the parameter estimation we used maximum likelihood estimation. Another point we have to add is that all the simulations for AR(1), MA(1) and ARMA(1,1) models used $\sigma^2 = 1$.

4. INAR(2) model

$$X_t = a_1 \circ X_{t-1} + a_2 \circ X_{t-2} + \varepsilon_t$$

Likewise ARMA(1,1) we again use two different INAR(2) models with thinning parameters a_1 and a_2 take the values $(a_1, a_2) = (0.3, 0.5)$ for both models and for the innovations ε_t we have $\varepsilon_t \sim Poisson(2)$ for the first model and $\varepsilon_t \sim Poisson(5)$ for the second one.

For all models we evaluate the block length selection methods for both $n = 200$ and $n = 500$ with the parameters stated above. In addition, the comparison results are based on Bias and MSE of the parameters of interest. Finally, we used 500 Monte Carlo simulation and 500 Bootstrap samples from each simulation in order to get our results. All statistical analysis was performed using R Statistical Software (version 4.0.2.; R Foundation for Statistical Computing, Vienna, Austria).

In total, eight different methods were compared. The first one is that block length parameter l was selected in the basis of $l = n^{\frac{1}{3}}$. Then, we examined how each one of the four new methods performs. For the MBB (and only for the MBB) method we also evaluated the HHJ method. NPPI method was used for both MBB and CBB methods, while PW method was used for SBB and CBB.

Finally, in order to also implement NBB method we used Nordman's rule and we compare the results with those of MBB with block length parameter coming from the HHJ block length selection technique. In Table 1 we have in the first column the block length selection methods we examined and in the second column the Block Bootstrap methods we used in each case. As we have already mentioned, for the NBB method we first estimated the block length for the MBB via HHJ and then applied the Nordman's rule to compare the two methods. So, as we can see in Table 1, under MBB and CBB we are comparing seven block length selection methods and under SBB we have six block length selection methods.

Table A : Block length selection methods examined and the corresponding Block Bootstrap methods they were applied to

Block length selection method	**Block Bootstrap method**
$l = n^{\frac{1}{3}}$	MBB, CBB, SBB
NPPI	MBB, CBB
HHJ	MBB
D1	MBB, CBB, SBB
D2	MBB, CBB, SBB
D3	MBB, CBB, SBB
D4	MBB, CBB, SBB
PW	CBB, SBB

The first interesting fact to observe is that if we set the value of φ being "close" to zero then all methods achieve MSE's of the same magnitude. This fact gives a great boost in the choice of block length parameter according to the rule $l = n^{\frac{1}{3}}$, due to the fact that this "method" in the only one of those examined that does not need any estimation procedure of block length. In fact, as we can see in Table 1, this choice of block length is the one that gives the lower MSE's for the estimations of φ in the case where $\varphi = -0.1$ ($n = 200$) or 0.1 (both $n = 200$ and $n = 500$) . However, as this is not always the case, another obvious observation is that in most occasions one of the new proposed methods is the one that prevails in performance over the other ones. Specifically, in the case of $\varphi = -0.9$ we have that the first of the methods using the Quadratic loss functions has clearly the best performance independently of the length of the series. The same superiority is observed when $\varphi = -0.5, 0.5$ for both $n = 200$ and $n = 500$ and for $\varphi = 0.9$ and $n = 200$ in the case of the second method using Quadratic loss functions (D4). In the latter case, for $n = 500$ methods D2 and D4 have almost identical performance, with D2 just slightly outperforming D4.

Another interesting observation is that HHJ – one of the first block length selection methods introduced- has a very bad performance especially in the cases of $\varphi = -0.9$. It is not confirmed by our results that HHJ or NPPI should be favored against the choice $l = n^{\frac{1}{3}}$. If we also take into considerations that these algorithms may not converge to a plausible block length we immediately understand that they are sub-optimal.

However, for n = 500 we observe that HHJ is able to catch the best Bias and MSE of the variance when φ = -0.1. When φ = 0.1 NPPI method outperforms the other methods in the same category.

The same pattern is observed in the case of CBB and SBB in Tables 11 and 14 respectively. Again, D-methods have the majority of the best results with D4 being observed more frequently as the one with the greatest performance. In both CBB and SBB there is introduced the method of Politis and White.

Table 2 : Bias and MSE for $\hat{\varphi}$ and $\widehat{\sigma^2}$ in each block-length selection method under MBB – AR(1)

Method		n = 200		n = 500	
		φ = -0.9	σ^2 = 1	φ = -0.9	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	0.1604	1.2534	1.8356	1.3103
	MSE	0.0264	1.8141	0.0238	1.8141
D1	Bias	0.1041	0.6902	0.0466	0.3590
	MSE	0.0188	0.7083	0.0033	0.1485
D2	Bias	0.2538	1.9603	0.1766	1.4358
	MSE	0.0728	5.1472	0.0464	3.3834
D3	Bias	**0.0454**	**0.2258**	**0.0228**	**0.1484**
	MSE	**0.0045**	**0.0952**	**0.0010**	**0.0315**
D4	Bias	0.0965	0.7060	0.0915	0.7690
	MSE	0.0212	1.7872	0.0193	1.7233
NPPI	Bias	0.1249	0.9317	0.1296	1.0352
	MSE	0.0240	1.4986	0.0252	1.5832
HHJ	Bias	0.4498	3.0067	0.3014	2.3713
	MSE	0.2040	10.4449	0.0912	5.9785

Method		n = 200		n = 500	
		φ = -0.5	σ^2 = 1	φ = -0.5	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	0.0894	0.0948	0.0860	0.0980
	MSE	0.0108	0.0224	0.0084	0.0148
D1	Bias	0.1851	0.6903	0.1680	0.3590
	MSE	0.0307	0.0514	0.0289	0.1864
D2	Bias	0.1714	0.1839	0.1677	0.1858
	MSE	0.0305	0.0511	0.0288	0.0415
D3	Bias	0.0975	0.1088	0.0447	0.0527
	MSE	0.0116	0.0326	0.0029	0.0092
D4	Bias	**0.0545**	**0.0609**	**0.0205**	**0.0237**
	MSE	**0.0070**	**0.0184**	**0.0023**	**0.0061**
NPPI	Bias	0.0827	0.0703	0.0862	0.0864
	MSE	0.0124	0.0209	0.0115	0.0160
HHJ	Bias	0.2546	0.2367	0.1689	0.1790
	MSE	0.0659	0.0806	0.0292	0.0383

Table 2 continued

Method		n = 200		n = 500	
		φ = -0.1	σ^2 = 1	φ = -0.1	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	0.0178	-0.0050	0.0202	-0.0017
	MSE	**0.0036**	**0.0086**	0.0019	0.0046
D1	Bias	0.0345	0.0013	0.0283	-0.0017
	MSE	**0.0036**	0.0108	0.0019	0.0039
D2	Bias	0.0307	0.0009	0.0239	-0.0023
	MSE	**0.0036**	0.0108	**0.0018**	0.0039
D3	Bias	0.0119	-0.0099	0.0043	-0.0085
	MSE	0.0047	0.0115	0.0020	0.0039
D4	Bias	**0.0096**	**-0.0005**	**0.0026**	-0.0059
	MSE	0.0050	0.0115	0.0021	0.0040
NPPI	Bias	0.0189	0.0011	0.0207	-0.0029
	MSE	0.0043	0.0110	**0.0018**	0.0045
HHJ	Bias	0.0544	0.0040	0.0349	**-0.0009**
	MSE	0.0042	0.0096	0.0021	**0.0037**

Method		n = 200		n = 500	
		φ = 0.1	σ^2 = 1	φ = 0.1	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	-0.0167	-0.0029	-0.0179	0.0010
	MSE	**0.0036**	0.0101	**0.0016**	0.0046
D1	Bias	-0.0177	-0.0106	0.0159	**0.0002**
	MSE	0.0038	0.0111	0.0017	0.0039
D2	Bias	-0.0117	-0.0120	-0.0114	-0.0004
	MSE	0.0041	0.0113	0.0018	0.0040
D3	Bias	-0.0053	-0.0194	-0.0054	-0.0049
	MSE	0.0045	0.0116	0.0019	0.0040
D4	Bias	**-0.0036**	-0.0123	**-0.0043**	-0.0019
	MSE	0.0049	0.0115	0.0020	0.0041
NPPI	Bias	-0.0198	0.0703	-0.0179	-0.0006
	MSE	0.0040	**-0.0081**	0.0018	**0.0038**
HHJ	Bias	-0.0377	**-0.0028**	-0.0283	0.0056
	MSE	0.0043	0.0095	0.0021	0.0042

Table 2 continued

Method		n = 200		n = 500	
		φ	$\sigma^2 = 1$	φ = 0.5	$\sigma^2 = 1$
$l = n^{\frac{1}{3}}$	Bias	-0.0856	0.0954	-0.0862	0.0968
	MSE	0.0100	0.0233	0.0085	0.0153
D1	Bias	-0.0549	0.0522	-0.0350	0.0372
	MSE	0.0076	0.0152	0.0030	0.0059
D2	Bias	-0.0350	0.0281	-0.0242	0.0234
	MSE	0.0056	0.0122	0.0023	0.0050
D3	Bias	-0.0477	0.0403	-0.0268	0.0250
	MSE	0.0058	0.0151	0.0020	0.0051
D4	Bias	**-0.0231**	**0.0143**	**-0.0128**	**0.0092**
	MSE	**0.0047**	**0.0113**	**0.0017**	**0.0042**
NPPI	Bias	-0.1006	0.7194	-0.0819	0.0879
	MSE	0.0120	0.0216	0.0106	0.0159
HHJ	Bias	-0.0914	0.0760	-0.0639	0.0668
	MSE	0.0193	0.0259	0.0094	0.0138

Method		n = 200		n = 500	
		φ = 0.9	$\sigma^2 = 1$	φ = 0.9	$\sigma^2 = 1$
$l = n^{\frac{1}{3}}$	Bias	-0.1480	1.1464	-0.1487	**0.0010**
	MSE	0.0232	1.5457	0.0225	1.2589
D1	Bias	-0.0988	0.8455	-0.0334	0.2695
	MSE	0.0149	1.6797	0.0017	0.1282
D2	Bias	-0.0327	0.1621	**-0.0187**	0.1283
	MSE	0.0026	0.0450	**0.0008**	**0.0242**
D3	Bias	-0.2276	1.8441	-0.2662	2.1570
	MSE	0.0597	4.6184	0.0749	5.2522
D4	Bias	**-0.0293**	**0.1308**	-0.0201	0.1366
	MSE	**0.0023**	**0.0397**	0.0009	0.0310
NPPI	Bias	-0.1480	1.1464	-0.1135	0.8952
	MSE	0.0165	0.8948	0.0201	1.2002
HHJ	Bias	-0.0388	0.2201	-0.0257	0.1596
	MSE	0.0041	0.1107	0.0030	0.1566

This method is able to achieve very good performance. However, as we can observe in Tables 23 and 24 there is a great chance that we will not be able to get a useful block length especially in the cases where the value of φ in our model is close enough to zero. In the case of SBB this fact is more pronounced making this method less usable than the others. As the length of the series grows we have that PW method is able to produce plausible block lengths more often. Yet again, in the case of $\varphi = -0.1$ or 0.1 we have that for $n = 500$ PW method was not able to "converge" 49 and 57 times respectively, which is approximately 10% of the occasions. It is worth mentioning though that in the cases of $\varphi = -0.1$ or 0.1 and $n = 200$ PW has the lowest MSE for the estimation of φ despite the fact that there were used a lot less simulations due to the fact of lack of convergence. This observation means that this method is able to produce a nearly optimal block size versus the other methods for values of φ near zero. In combination with the fact that this algorithm produces a result really fast, it would be worth trying in cases of CBB and SBB implementation. As for the comparison of PW algorithm for CBB and SBB we can find that SBB produces better results. This is not a surprise as SBB is less sensitive to misspecification of the block length (or the expected block length in this case) than CBB.

Another interesting comment is that in the cases of $l = n^{\frac{1}{3}}$ and the 4 new methods, where the three block bootstrap methods are applied with the same block length parameters we have that MBB and CBB display similar results. This is an expected result as it is known in the literature that both these methods are asymptotically equivalent. On the other hand, SBB exhibits higher MSE's and Bias in the corresponding estimations. Again, this is an expected result, due to the extra randomization induced in the length of each block. Corresponding results can be found in Tables 2, 11, 12, 23, 24,25 of the Appendix.

In total, we have that for all three methods D-methods have the best performance in almost all cases. When φ lies near zero all methods produce similar results.

Table 3: Bias and MSE for $\hat{\theta}$ and $\widehat{\sigma^2}$ in each block-length selection method under MBB – MA(1)

Method		n = 200		n = 500	
		θ = -0.9	$\sigma^2 = 1$	θ = -0.9	$\sigma^2 = 1$
$l = n^{\frac{1}{3}}$	Bias	0.3706	0.3906	0.3675	0.4107
	MSE	0.1380	0.1765	**0.1353**	0.1777
D1	Bias	0.5197	0.5768	0.5194	0.5839
	MSE	0.2712	0.3640	0.2701	0.3550
D2	Bias	0.5196	0.5768	0.5194	0.5840
	MSE	0.2711	0.3642	0.2694	0.3550
D3	Bias	0.5124	0.5696	0.5183	0.5831
	MSE	0.2662	0.3602	0.2694	0.3545
D4	Bias	0.4830	0.5366	0.4932	0.5545
	MSE	0.2411	0.3281	0.2535	0.3328
NPPI	Bias	**0.3195**	**0.3471**	**0.3509**	**0.3902**
	MSE	**0.1148**	**0.1569**	0.1362	**0.1762**
HHJ	Bias	0.6313	0.6658	0.5187	0.5852
	MSE	0.3997	0.4794	0.2694	0.3544

Method		n = 200		n = 500	
		θ = -0.5	$\sigma^2 = 1$	θ = -0.5	$\sigma^2 = 1$
$l = n^{\frac{1}{3}}$	Bias	0.1207	0.0857	0.1161	0.0899
	MSE	0.0165	0.0198	0.0142	0.0135
D1	Bias	0.2103	0.1561	0.2115	0.1413
	MSE	0.0456	0.0408	0.0452	0.0257
D2	Bias	0.2103	0.1561	0.2114	0.1408
	MSE	0.0456	0.0408	0.0451	0.0255
D3	Bias	0.1140	0.0931	0.0641	0.0391
	MSE	0.0154	0.0255	0.0050	0.0066
D4	Bias	**0.0887**	**0.0723**	**0.0440**	**0.0230**
	MSE	**0.0113**	0.0205	**0.0040**	**0.0055**
NPPI	Bias	0.1027	0.0755	0.1172	0.0837
	MSE	0.0160	**0.0195**	0.0185	0.0139
HHJ	Bias	0.2932	0.1846	0.2114	0.1494
	MSE	0.0869	0.0502	0.0452	0.0283

Table 3 continued

Method		n = 200		n = 500	
		θ = -0.1	$\sigma^2 = 1$	θ = -0.1	$\sigma^2 = 1$
$l = n^{\frac{1}{3}}$	Bias	0.0186	-0.0077	0.0177	-0.0037
	MSE	**0.0034**	0.0102	**0.0015**	0.0039
D1	Bias	0.0323	0.0020	0.0274	-0.0028
	MSE	0.0036	0.0103	0.0018	0.0037
D2	Bias	0.0275	0.0011	0.0225	-0.0039
	MSE	0.0038	0.0103	0.0017	**0.0037**
D3	Bias	0.0074	-0.0033	0.0029	-0.0079
	MSE	0.0049	0.0102	0.0020	0.0037
D4	Bias	**0.0043**	-0.0037	**0.0009**	-0.0081
	MSE	0.0054	0.0101	0.0021	0.0037
NPPI	Bias	0.0179	**0.0001**	0.0194	**0.0027**
	MSE	0.0043	**0.0092**	0.0016	0.0041
HHJ	Bias	0.0470	0.0095	0.0324	0.0045
	MSE	0.0036	0.0095	0.0019	0.0042

Method		n = 200		n = 500	
		θ = 0.1	$\sigma^2 = 1$	θ = 0.1	$\sigma^2 = 1$
$l = n^{\frac{1}{3}}$	Bias	-0.0183	-0.0029	-0.0162	-0.0014
	MSE	**0.0037**	0.0101	**0.0016**	0.0043
D1	Bias	-0.0160	-0.0077	-0.0119	-0.0006
	MSE	0.0038	0.0095	0.0017	0.0037
D2	Bias	-0.0117	-0.0120	-0.0071	-0.0013
	MSE	0.0041	**0.0095**	0.0018	0.0038
D3	Bias	-0.0080	-0.0093	-0.0016	-0.0026
	MSE	0.0050	0.0095	0.0019	0.0038
D4	Bias	**-0.0044**	-0.0094	**0.0002**	-0.0027
	MSE	0.0052	0.0096	0.0021	0.0038
NPPI	Bias	-0.0136	-0.0045	-0.0219	-0.0030
	MSE	**0.0037**	0.0101	0.0018	0.0037
HHJ	Bias	-0.0373	**0.0004**	-0.0266	**-0.0003**
	MSE	**0.0037**	0.0110	0.0020	**0.0036**

Table 3 continued

Method		n = 200		n = 500	
		θ = 0.5	σ^2 = 1	θ = 0.5	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	-0.1148	0.0917	-0.1168	0.0932
	MSE	0.0149	0.0199	0.0143	0.0135
D1	Bias	-0.1009	0.0665	-0.0675	0.0516
	MSE	0.0155	0.0176	0.0070	0.0080
D2	Bias	-0.0679	0.0435	-0.0494	0.0382
	MSE	0.0100	0.0147	0.0049	0.0067
D3	Bias	-0.0683	0.0445	-0.0373	0.0292
	MSE	0.0078	0.0142	0.0023	0.0053
D4	Bias	**-0.0242**	**0.0133**	**-0.0105**	**0.0088**
	MSE	**0.0045**	**0.0106**	**0.0014**	**0.0042**
NPPI	Bias	-0.1120	0.3499	-0.1150	0.0800
	MSE	0.0190	0.0201	0.0180	0.0127
HHJ	Bias	-0.1835	0.1170	-0.1449	0.1115
	MSE	0.0465	0.0334	0.0257	0.0204

Method		n = 200		n = 500	
		θ = 0.9	σ^2 = 1	θ = 0.9	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	-0.3670	0.4063	-0.3666	0.4024
	MSE	0.1354	0.1894	0.1346	0.1711
D1	Bias	-0.4952	0.5495	-0.5121	0.5756
	MSE	0.2520	0.3423	0.2651	0.3479
D2	Bias	-0.3030	0.3339	-0.2801	0.3119
	MSE	0.1040	0.1491	0.0899	0.1202
D3	Bias	-0.4740	0.5280	-0.5039	0.5674
	MSE	0.2386	0.3279	0.2595	0.3419
D4	Bias	**-0.1699**	**0.1883**	-0.1467	**0.1622**
	MSE	**0.0336**	**0.0567**	0.0244	**0.0358**
NPPI	Bias	-0.3307	1.1464	-0.3492	0.3904
	MSE	0.1221	0.1650	0.1330	0.1773
HHJ	Bias	-0.4393	0.4669	**0.0044**	0.4388
	MSE	0.2295	0.2818	**0.0213**	0.2306

Things change a little bit when we have the Moving Average model (MA(1)). In most cases D-methods (except D4) display greater bias in the estimation of φ compared to the other methods.

In fact, for the case of $\theta = -0.9$ we observe in Table 3 that NPPI method gives the best results. However, as we can see the difference between NPPI and $l = n^{\frac{1}{3}}$ is minor, considering the fact that NPPI is a rather computationally intensive method. In addition, likewise AR(1) model, we have that when θ takes values $-0.9, -0.5, 0.5, 0.9$ D4 method dominates the other methods, whereas in the cases of $\theta = -0.1$ or 0.1 we have that for both $n = 200$ and $n = 500$ choosing block length with the rule $l = n^{\frac{1}{3}}$ is the best option due to lack of additional computation in the cases of similar results. Nonetheless, we have to state here that this block length option does not seem to have the same great performance in bias estimation.

With the introduction of PW method in CBB we have from Table 5 that PW is able to prevail in all categories for values of θ being -0.9 or -0.5, independently of the length of the series. Furthermore, for cases of $\theta = -0.1$ and $\theta = 0.1$ with $n = 200$ PW produces the lowest MSE. When n grows we again observe that $n^{\frac{1}{3}}$ is the optimal choice. Similar results are observed in the case of SBB implementation with PW algorithm. However, we have to state that again PW produces too many non-feasible results, especially in the cases of $\theta = -0.1, 0.1$. In Tables 27 and 28 we can see that PW does not converge to a plausible estimation for the block length parameter 106 and 134 respectively for CBB and SBB when $\theta = -0.1$ and $n = 200$ and 143 and 194 are the numbers when $\theta = 0.1$. The fact that in both cases PW is able to demonstrate the best results is rather surprising. Furthermore, the fact that PW algorithm does converge to choices of $\hat{l} \geq 2$ for all the cases of models with $\theta = -0.9$ and $\theta = -0.5$ and that for these cases the results are greatly in favor of this method makes us think again that this algorithm is able to catch very sufficiently the structure of the underlying process.

In general, the results are similar with those of AR(1) modelling. D4 method seem to be the best choice in most cases. In addition, the introduction of PW algorithm changes the allocation of the results. HHJ method performs better in contrast to AR(1) modelling, were the results were really disappointing in some cases. Yet again, HHJ is not able to outperform significantly any other technique and in the case this happens ($\theta = 0.9$ and $n = 500$) the results are slightly better than those displayed by D4 with the drawback of needing significantly more time to give an estimation.

Additionally, the fact that MBB and CBB produce similar results does not change in MA(1) modelling. However, we have to state that the performance of NPPI is clearly better in the case of MBB. Maybe this is the case due to the fact that NPPI was introduced for MBB in the first place and not for CBB. Corresponding results for the MA(1) model can be found in Tables 3, 13, 14, 26, 27, 28 of the Appendix.

For the ARMA(1,1) model we used two different approaches. The two models used are :

- $X_t = -0.8X_{t-1} + 0.5\varepsilon_{t-1} + \varepsilon_t$
- $X_t = 0.5X_{t-1} - 0.8\varepsilon_{t-1} + \varepsilon_t$

The results for the block length selection techniques under MBB can be found in Table 4. The outcomes are undeniably in favor of D4 for both $n = 200$ and $n = 500$. Again, the HHJ method performs the worst for all estimations and also needed a great amount of time to give results.

Table 4: Bias and MSE for $\hat{\varphi}, \hat{\theta}$ and $\hat{\sigma}^2$ in each block-length selection method under MBB – ARMA(1,1) with parameters $\varphi = -0.8, \theta = 0.5$

Method		n = 200			n = 500		
		φ= -0.8	θ = 0.5	$\sigma^2 = 1$	φ = -0.8	θ = 0.5	σ^2=1
$l = n^{\frac{1}{3}}$	Bias	0.2335	-0.2244	**0.0201**	0.1446	-0.1407	0.0695
	MSE	0.0639	0.0608	0.0801	0.0233	0.0241	0.0098
D1	Bias	0.1938	-0.3847	0.1405	0.4102	-0.3723	0.1475
	MSE	0.4287	0.1586	0.0394	0.1766	0.1470	0.0294
D2	Bias	0.3890	-0.3506	0.1283	0.3730	-0.3398	0.1358
	MSE	0.1712	0.1424	0.0361	0.1573	0.1327	0.0267
D3	Bias	0.2011	-0.1939	0.0674	0.0895	-0.0921	0.0383
	MSE	0.0537	0.0577	0.0242	0.0117	0.0158	0.0078
D4	Bias	**0.1107**	**-0.1087**	0.0212	**0.0395**	**-0.0406**	**0.0082**
	MSE	**0.0269**	**0.0326**	**0.0132**	**0.0057**	**0.0086**	**0.0043**
NPPI	Bias	0.2402	-0.2204	0.0288	0.2540	-0.2256	0.0138
	MSE	0.0962	0.0766	0.0476	0.1170	0.0819	0.0770
HHJ	Bias	0.7659	-0.6757	0.1838	0.4343	-0.3936	0.1585
	MSE	0.5889	0.4613	0.0544	0.1898	0.1570	0.0322

Interestingly, $l = n^{\frac{1}{3}}$ rule seems to work better than NPPI also independently of the length of the series. In addition, HHJ and NPPI methods need the more time to implement due to the cross-validation nature of their approach. Interestingly, HHJ method converges in all cases under both ARMA models.

The interesting fact here, is that in contrast to AR(1) and MA(1) models, SBB method works better in ARMA(1,1) versus MBB and CBB , which once again have similar performance. This is the fact also for the case of PW technique. Additionally, for this method we observe that for both models when $n = 500$ is able to converge in all cases except only one and that when $n = 200$ the cases of non-convergence are less than 5%. This is the case also, for the NPPI method where for both MBB and CBB the percentage of non-convergence is less than 5%.

Table 5: Bias and MSE for $\hat{\varphi}, \hat{\theta}$ and $\hat{\sigma}^2$ in each block-length selection method under MBB – ARMA(1,1) with parameters $\varphi = 0.5, \theta = -0.8$

Method		n = 200			n = 500		
		φ = 0.5	θ = -0.8	$\sigma^2 = 1$	φ = 0.5	θ = -0.8	$\sigma^2 = 1$
$l = n^{\frac{1}{3}}$	Bias	-0.2139	0.2949	0.0511	-0.1118	0.1786	0.0508
	MSE	0.0638	0.1016	0.0144	0.0203	0.0365	0.0074
D1	Bias	-0.3303	0.4593	0.0759	-0.1913	0.2960	0.0753
	MSE	0.1222	0.2224	0.0196	0.0460	0.0952	0.0099
D2	Bias	-0.2971	0.4150	0.0706	-0.1787	0.2775	0.0719
	MSE	0.1032	0.1845	0.0187	0.0413	0.0847	0.0094
D3	Bias	-0.1814	0.2433	0.0360	-0.0714	0.1113	0.0338
	MSE	0.0533	0.0777	0.0152	0.0119	0.0159	0.0052
D4	Bias	**-0.1501**	**0.1957**	**0.0306**	**-0.0586**	**0.0866**	0.0294
	MSE	**0.0421**	**0.0549**	**0.0139**	**0.0099**	**0.0113**	**0.0049**
NPPI	Bias	-0.2283	0.3115	-0.0132	-0.1824	0.2791	**-0.0005**
	MSE	0.0793	0.1404	0.0675	0.0591	0.1201	0.0618
HHJ	Bias	-0.5149	0.7021	0.0980	-0.3146	0.4591	0.0891
	MSE	0.2698	0.5000	0.0206	0.1063	0.2163	0.0129

It is clear from Table 5, that for the second ARMA(1,1) model again D4 method gives the best results. However, for the case of $n = 200$ we can observe that $n^{\frac{1}{3}}$ does not have much worse performance. It is very interesting so far, how well this "method" works in contrast to the other D-methods and also NPPI and HHJ. The case of SBB performing better than MBB and CBB continues also with the second model. Corresponding results can be found in tables 4, 5, 15, 17, 16, 18, 29 of the Appendix.

In this phase, we will continue with the simulation studies over INAR models.

From Table 6, we can observe that regardless the length of the series the proposed methods share performance of the same range for the estimation of a_1. The differences between the methods begin to be more clear when it comes to the estimation of a_2. Again, D4 is able to give the best results. In addition, D4 achieves the best results for estimating the parameter of the Poisson distribution the innovations follow. Overall, this method is the one to come nearest to the original model the simulated samples came from.

Table 6. Bias and MSE for $\hat{a}_1$, $\hat{a}_2$ and $\hat{\lambda}$ in each block-length selection method under MBB – INAR(2) with parameters $a_1 = 0.5$,$a_2 = 0.3$,$lambda = 2$

Method		n = 200			n = 500		
		a_1	a_2	λ	a_1	a_2	λ
$l = n^{\frac{1}{3}}$	Bias	0.0025	-0.2244	1.6939	0.0125	-0.1380	1.2234
	MSE	0.0041	0.0349	3.0985	**0.0014**	0.0196	1.5950
D1	Bias	0.0051	-0.0890	0.8184	0.0038	-0.0476	0.4189
	MSE	**0.0010**	0.0119	1.0216	**0.0014**	0.0036	0.2950
D2	Bias	-0.0046	-0.0623	0.6195	**0.0009**	-0.0338	0.3118
	MSE	0.0054	0.0087	0.7171	**0.0014**	0.0025	0.2183
D3	Bias	**-0.0004**	-0.1311	1.2350	0.0090	-0.0996	0.8803
	MSE	0.0045	0.0228	1.9650	0.0016	0.0135	1.0163
D4	Bias	-0.0062	**-0.0524**	**0.5493**	**-0.0009**	**-0.0286**	**0.2827**
	MSE	0.0059	**0.0082**	**0.6420**	0.0015	**0.0022**	**0.2030**
NPPI	Bias	-0.0163	-0.1241	1.0475	-0.0088	-0.1250	1.1037
	MSE	0.0128	0.0219	2.0173	0.0070	0.0206	2.1054
HHJ	Bias	-0.0099	-0.0692	0.7405	-0.0006	-0.0325	0.3185
	MSE	0.0064	0.0109	1.3356	**0.0014**	0.0032	0.3330

Table 7 presents the results of MBB method implementation with each method for estimating the block length parameter. Again, (for $n = 200$) we have that the MSE's are of the same order of magnitude for all methods for a_1 with the choice of $l = n^{\frac{1}{3}}$ presenting the lowest one. Changes begin to become clear in the estimation of a_2, where D4 method is able to succeed the lowest MSE and Bias, but without major differences from the other methods. However, as we can observe none of the methods is able to estimate sufficiently enough λ from Poisson distribution. A typical example is the choice of $n^{\frac{1}{3}}$, where the MSE of λ is 20.7023 with a Bias of 4.4096. The best choice seems to be D4 again, with some minor differences from D2, NPPI and HHJ. As for the case where $n = 500$, we get that D4, NPPI and HHJ have the lowest biases for a_1 with HHJ just slightly outperforming. Taking into account though that HHJ needs a vast amount of time to be implemented and D4 being the least computationally intensive method from these three, we conclude that D4 is the prevalent choice. As for the MSE's all methods give almost the same results.

Table 7: Bias and MSE for $\hat{a}_1$, $\hat{a}_2$ and $\hat{\lambda}$ in each block-length selection method under MBB – INAR(2) with parameters $a_1 = 0.5$,$a_2 = 0.3$, $\lambda = 5$

Method		n = 200			n = 500		
		a_1	a_2	λ	a_1	a_2	λ
$l = n^{\frac{1}{3}}$	Bias	0.0083	-0.1905	4.4096	0.0194	-0.1399	2.9383
	MSE	**0.0032**	0.0374	20.7023	0.0019	0.0203	9.0802
D1	Bias	0.0013	-0.1109	2.6770	0.0100	-0.0765	1.6099
	MSE	0.1222	0.0161	9.9792	**0.0017**	0.0078	3.6359
D2	Bias	-0.0009	-0.0828	2.0530	0.0047	-0.0495	1.0804
	MSE	0.0050	0.0108	6.6886	**0.0017**	0.0041	1.7737
D3	Bias	**-0.0005**	-0.0956	2.3406	0.0124	-0.1214	2.6597
	MSE	0.0046	0.0136	7.9783	0.0018	0.0188	9.3175
D4	Bias	-0.0013	**-0.0771**	**1.9400**	0.0017	-0.0392	0.9069
	MSE	0.0053	**0.0099**	**6.3574**	0.0018	**0.0032**	**1.4702**
NPPI	Bias	-0.0009	-0.0865	2.1207	0.0022	-0.0776	1.5832
	MSE	0.0049	0.0110	6.4226	0.0043	0.0089	4.7121
HHJ	Bias	0.0009	-0.0823	2.0159	**0.0016**	**-0.0341**	**0.8007**
	MSE	0.0051	0.0104	6.4424	0.0018	0.0033	1.6292

Finally, for the estimation of λ and a_2 we observe that D4 and HHJ display very similar results. As for CBB and SBB, as expected CBB and MBB demonstrate very similar performance, with one being slightly better in some occasions and the other in some others. The introduction of PW method in this phase does not seem to make any significant augmentation in the performance. It's interesting though that for both INAR models, regardless of the length of the series PW converges always in a plausible block length. In, addition both NPPI and HHJ do converge almost all times. This fact is in contrast with the other models, where we had that PW, NPPI and HHJ were not able to converge in many occasions. Furthermore, again SBB seems to perform better than MBB and CBB. Especially, for the PW method, SBB display better results confirming that SBB is less sensitive to misspecification of the block length parameter. Corresponding results can be found in Tables 6, 7, 19, 21, 20, 22, 30 of the Appendix.

Before continuing with the application of the methods to a real dataset I would like to briefly demonstrate the results of the comparison between NBB and MBB under Nordman's rule. There were a few surprises but in general the rule that MBB performs better than NBB was confirmed. The only exceptions to this rule were in the first ARMA(1,1) model with parameters $\varphi = -0.8$ and $\theta = 0.5$ NBB was able to estimate better the parameters, except in the case of θ and $n = 500$. In the MA(1) models MBB outperformed NBB in all categories and values of θ except when θ took the value -0.9. The same interesting fact was in someway observed for AR(1) model and the value $\varphi = -0.9$ However, it was only for $n = 200$ that this phenomenon occurred. Lastly, in INAR modelling things were more clear as MBB was by far better than NBB in all categories. Corresponding results can be found in Tables 31, 32, 33, 34 of the Appendix.

6. APPLICATION TO A REAL DATASET

In this section, we examine the performance of the proposed block bootstrap methods, as well as the performance of each block length selection technique to a real dataset consisting of weekly number of reported disease cases caused by Escherichia coli in the state of North Rhine Westphalia (Germany) from January 2001 to May 2013, excluding cases of EHEC and HUS for a total of 646 observations. The series can be downloaded by the R package *tscount*. In order to examine which INAR model would be more suitable to model this count time series we examined the sample autocorrelation function. As we can see in Figure 3, a more suitable model for this series would be an INAR(3) or INAR(4). We will follow the choice of Bisaglia and Gerolimetto (2019) who examined the AIC displayed by these two models and concluded that INAR(4) is the more suitable choice.

The estimation technique via CSS gave the that the model is the following:

$$X_t = 0.44 \circ X_{t-1} + 0.21 \circ X_{t-2} + 0.05 \circ X_{t-2} + 0.07 \circ X_{t-3} + \varepsilon_t$$

with $\varepsilon_t \sim Poisson$ (4.68). So, in order to evaluate the methods we used the INAR(4) model with the estimated thinning parameters via CLS and performed all the bootstrap methods we have examined as well as with the corresponding block length selection techniques.

From Table A7 we observe that the NPPI method has the best performance in terms of bias and MSE of a_3 and a_4. Again, like the simulation studies for the two INAR(2) models, we have that all methods perform similarly in the estimation of a_1. However, NPPI performed the worst in estimating λ, where D4 had clearly the best result.

For the CBB case we have that two D-methods, D2 and D4 were able to estimate significantly better the parameters a_1 and a_2. For parameters a_3 and a_4 all methods gave MSEs and biases of the same order of magnitude, with D3 being the one that was able to perform slightly better. The main differences were observed, on the estimation of λ where D4 and PW gave very similar results.

Table 8 : Bias and MSE for the parameters of INAR(4) modelling of the real dataset - MBB

Method	a_1		a_2		a_3		a_4		λ	
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE
$l = n^{\frac{1}{3}}$	0.0699	0.0249	-0.0615	0.0038	0.0638	0.0041	0.0439	0.0019	-2.3037	5.6570
D1	0.0038	**0.0100**	-0.0346	0.0012	0.0769	0.0059	0.0572	0.0033	-2.0818	4.6839
D2	0.0097	0.0101	-0.0351	0.0056	0.0751	0.0056	0.0555	0.0031	-2.1029	4.8322
D3	0.0480	0.0123	-0.0481	0.0023	0.0669	0.0045	0.0466	0.0022	-2.2775	5.5270
D4	**-0.0035**	**0.0100**	-0.0433	0.0821	0.0821	0.0067	0.0626	0.0039	**-1.9772**	**4.2593**
NPPI	0.1824	0.0833	**-0.1138**	0.0130	**0.0376**	**0.0014**	**0.0361**	**0.0013**	-2.6794	7.6092
HHJ	0.0048	**0.0100**	-0.0332	**0.0011**	0.0762	0.0058	0.0562	0.0032	-2.1253	4.8569

Table 9 : Bias and MSE for the parameters of INAR(4) modelling of the real dataset - CBB

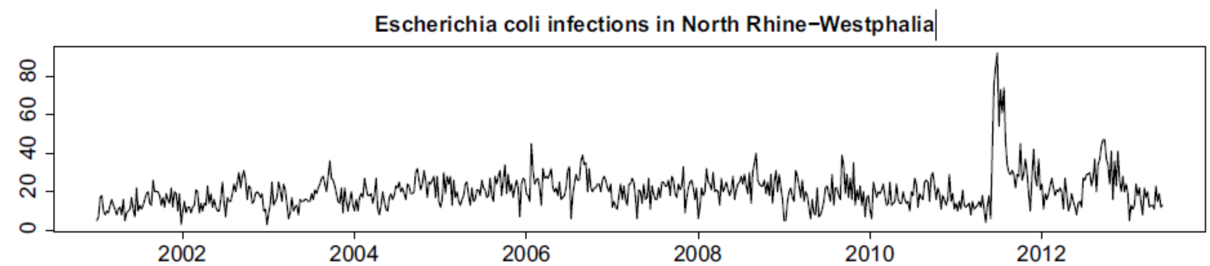
Method	a_1		a_2		a_3		a_4		λ	
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE
$l = n^{\frac{1}{3}}$	0.0739	0.0255	-0.0650	0.0042	0.0640	0.0041	0.0436	0.0019	-2.3256	5.7484
D1	-0.0031	0.0200	-0.0312	0.0010	0.0779	0.0061	0.0586	0.0034	-2.1012	4.8150
D2	-0.0092	**0.0101**	**-0.0264**	**0.0007**	0.0781	0.0061	0.0596	0.0036	-2.1303	4.8582
D3	0.0828	0.0269	-0.0581	0.0034	**0.0585**	**0.0034**	**0.0384**	**0.0015**	-2.4241	6.2063
D4	-0.0089	**0.0101**	-0.0267	**0.0007**	0.0783	0.0061	0.0590	0.0035	-2.0795	4.7343
PW	**0.0024**	0.0200	-0.0384	0.0015	0.0788	0.0062	0.0588	0.0035	**-2.0664**	**4.6600**

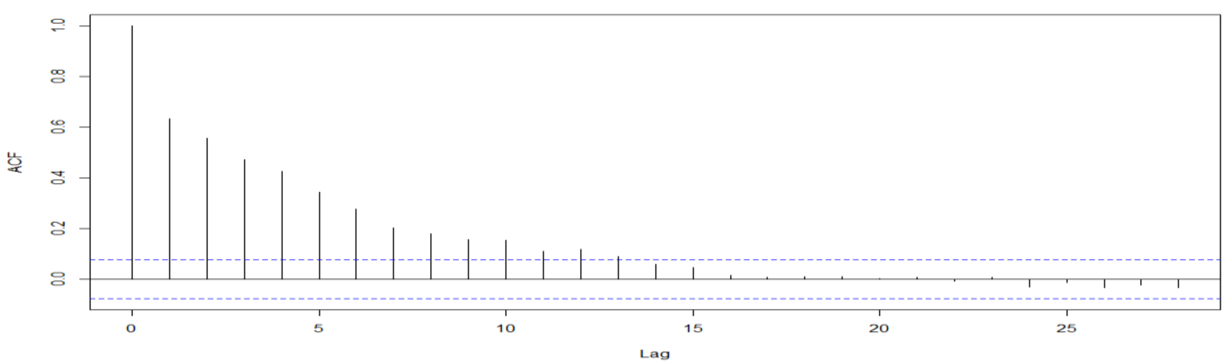
However, this is in favor of PW which is computationally much less intensive. Interestingly, we observe that the block bootstrap methods demonstrate many differences in the block length selection method they perform better. For instance, let's continue with SBB. Despite the fact that the rule $l = n^{\frac{1}{3}}$ is not able to sufficiently estimate a_1 and a_2, it outperforms the other methods in the estimation of a_3 and a_4. Unfortunately, it displays the poorest performance in the estimation of λ.

In general, for the real dataset, the proposed method should be MBB with block length selection technique being D4, as it the one that estimates sufficiently enough the thinning parameters of the model and the one to estimate better the λ parameter of the Poisson innovations.

Table 10: Bias and MSE for the parameters of INAR(4) modelling of the real dataset - SBB

Method	a_1		a_2		a_3		a_4		λ	
	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE
$l = n^{\frac{1}{3}}$	0.0608	0.0337	-0.0609	0.0037	**0.0669**	**0.0045**	**0.0466**	**0.0022**	-2.2516	5.4897
D1	-0.0150	0.0102	-0.0320	0.0010	0.0823	0.0068	0.0625	0.0039	**-1.9983**	4.3632
D2	**-0.0059**	**0.0100**	-0.0324	0.0010	0.0790	0.0062	0.0601	0.0036	-2.0739	4.6211
D3	0.0303	0.0209	-0.0499	0.0025	0.0732	0.0054	0.0534	0.0029	-2.1640	4.9829
D4	-0.0154	0.0102	**-0.0205**	**0.0004**	0.0786	0.0062	0.0587	0.0034	-2.0957	4.7720
PW	-0.0114	0.0101	-0.0367	0.0013	0.0829	0.0069	0.0628	0.0039	-1.9894	**4.3177**





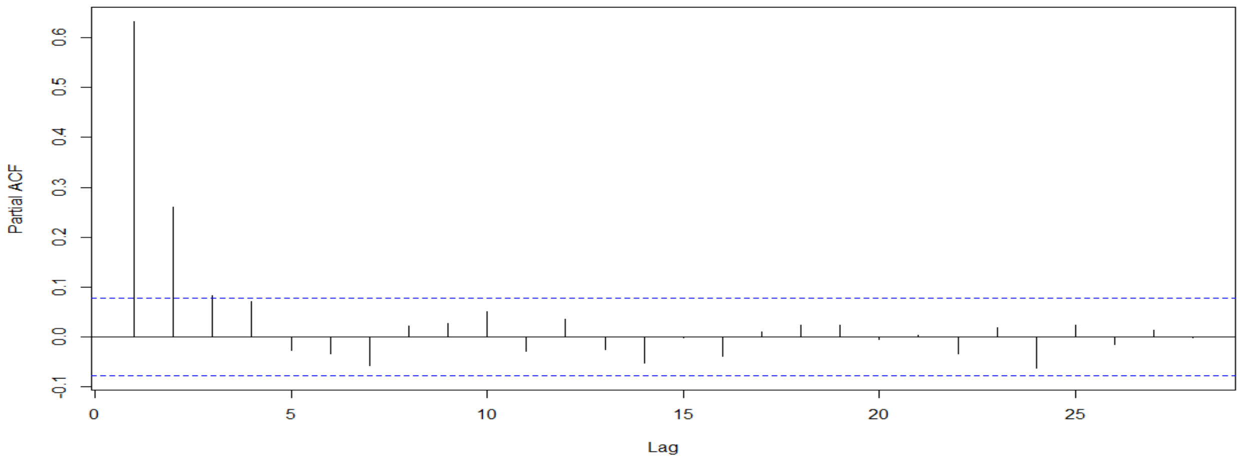


Figure 3 : Plot of the series, acf and pacf functions.

7. CONCLUSION

In this project, we examined the performance of various block length selection methods for three block bootstrap methods (MBB-CBB-SBB). We introduced four new methods to estimate the block length parameter along with methods already proposed (HHJ-NPPI-PW). All these methods try to deal with the crucial problem of correctly estimating the block length parameter. For this purpose, we carried out an extensive Monte Carlo simulations, under different four different models with various values for their parameters. The models were AR(1), MA(1), ARMA (1,1) and INAR(2). In the end, we also evaluated the methods to a real dataset downloaded from tscount library in R Core Team(2020).

In general, the results regarding the performance of the suggested D-methods, and especially D4, are promising. It seems that the way this method is able utilize the Quadratic Loss function along with the correlation estimators is indeed very effective. In most cases this method was able to demonstrate the best results in terms of both Bias and MSE.

A rather interesting observation is that HHJ method displayed poor performance in general. There were some occasions that this method managed to perform well and in fact outperform other methods, but the general results showed that this method is unable to estimate optimally the block length parameter in most cases and in addition it is so computationally intensive that makes this method ineffective.

Now, the good news are that if someone used the "naive" rule $l = n^{\frac{1}{3}}$, would in most cases not be disappointed and this fact can be supported by the results in the real dataset. Furthermore, theoretical results supporting that MBB and CBB are the methods that perform better were confirmed in most cases. The extra variation added by the random block lengths in SBB and the fact that NBB does not allow blocks to overlap made these two methods to perform worse.

8. REFERENCES

Al-Osh, M. A., & Alzaid, A. A. (1987). First-order integer-valued autoregressive (INAR (1)) process. Journal of Time Series Analysis, 8(3), 261-275.

Bisaglia, L., & Gerolimetto, M. (2016). Estimation of INAR (p) models using bootstrap.

Bisaglia, L., & Gerolimetto, M. (2019). Model-based INAR bootstrap for forecasting INAR (p) models. Computational Statistics, 34(4), 1815-1848.

Brockwell, P. J., & Davis, R. A. (1991). Stationary time series. In *Time Series: Theory and Methods* (pp. 1-41). Springer, New York, NY.

Bühlmann, P. (2002). Bootstraps for time series. *Statistical science*, 52-72.

Bühlmann, P., & Künsch, H. R. (1999). Block length selection in the bootstrap for time series. *Computational Statistics & Data Analysis*, *31*(3), 295-310.

Carlstein, E., Do, K. A., Hall, P., Hesterberg, T., & Künsch, H. R. (1998). Matched-block bootstrap for dependent data. *Bernoulli*, *4*(3), 305-328.

Carlstein, E. (1986). The use of subseries values for estimating the variance of a general statistic from a stationary sequence. The annals of statistics, 14(3), 1171-1179.

Dimitris N. Politis (2003). The Impact of Bootstrap Methods on Time Series Analysis. Statistical Science 2003, Vol. 18, No. 2, 219–230 Institute of Mathematical Statistics, 2003.

Dimitris N. Politis and Joseph P. Romano (1994). The Stationary Bootstrap. Journal of the American Statistical Association Vol. 89, No. 428 (Dec., 1994), pp. 1303-1313

Daniel J. Nordman and Soumendra N. Lahiri (2014). Convergence rates of empirical block length selectors for block bootstrap. Bernoulli 20(2), 2014, 958–978.

Efron, B. (1992). Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics* (pp. 569-593). Springer, New York, NY.

Gungor, O. E., Al-Qadi, I. L., & Mann, J. (2018). Detect and charge: Machine learning based fully data-driven framework for computing overweight vehicle fee for bridges. *Automation in Construction*, *96*, 200-210.

Nordman, D. J., Lahiri, S. N., & Fridley, B. L. (2007). Optimal block size for variance estimation by a spatial block bootstrap method. Sankhyā: The Indian Journal of Statistics, 468-493.

Härdle, W., Horowitz, J., & Kreiss, J. P. (2003). Bootstrap methods for time series. International Statistical Review, 71(2), 435-459.

Hans R. Kunsch. The Jackknife and the Bootstrap for General Stationary Observations. The Annals of Statistics , Sep., 1989, Vol. 17, No. 3 (Sep., 1989), pp. 1217-1241.

Jin-Guan, D., & Yuan, L. (1991). The integer-valued autoregressive (INAR (p)) model. Journal of time series analysis, 12(2), 129-142.

Kreiss, J. P., & Lahiri, S. N. (2012). Bootstrap methods for time series. In *Handbook of statistics* (Vol. 30, pp. 3-26). Elsevier.

Lahiri, S. N. (2003). Selecting optimal block lengths for block bootstrap methods. In Proceedings of the 35th Symposium of the Interface. Salt lake city.

Liu, R. Y., & Singh, K. (1992). Moving blocks jackknife and bootstrap capture weak dependence. Exploring the limits of bootstrap, 225, 248.

Paparoditis, E., & Politis, D. N. (2001). Tapered block bootstrap. *Biometrika*, *88*(4), 1105-1119.

Paparoditis, E., & Politis, D. N. (2002). Local block bootstrap. *Comptes Rendus Mathematique*, *335*(11), 959-962.

Hall, P., Horowitz, J. L., & Jing, B. Y. (1995). On blocking rules for the bootstrap with dependent data. *Biometrika*, *82*(3), 561-574.

Politis, D. and Romano, J.P. (1992) A Circular Block Resampling Procedure for Stationary Data. In: Lepage, R. and Billard, L., Eds., Exploring the Limits of Bootstrap, Wiley, New York, 263-270.

Politis, D. N., & White, H. (2004). Automatic block-length selection for the dependent bootstrap. *Econometric reviews*, *23*(1), 53-70.

Radovanov, B., & Marcikić, A. (2014). A comparison of four different block bootstrap methods. *Croatian Operational Research Review*, 189-202.

Regina Y. Liu and Kesar Singh (1992). Efficinecy and robusteness in resampling. The Annals of Statistics, 1992, Vol. 20, No.1, 370 – 384.

S. N. Lahiri (1999). Theoretical Comparisons of Block Bootstrap Methods. The Annals of Statistics , Feb., 1999, Vol. 27, No. 1 (Feb., 1999), pp. 386-404.

S.N. Lahiri, K. Furukawa, Y.-D. Lee (2007). A nonparametric plug-in rule for selecting optimal block lengths for block bootstrap methods. Statistical Methodology Volume 4, Issue 3, July 2007, Pages 292-321.

Shibata, R. (1980). Asymptotically efficient selection of the order of the model for estimating parameters of a linear process. *The annals of statistics*, 147-164.

Steutel, F. W., & van Harn, K. (1979). Discrete analogues of self-decomposability and stability. *The Annals of Probability*, 893-899.

9. APPENDIX

Table 11 : Bias and MSE for $\hat{\varphi}$ and $\widehat{\sigma^2}$ in each block-length selection method under CBB – AR(1)

Method		n = 200		n = 500	
		φ = -0.9	$\sigma^2 = 1$	φ = -0.9	$\sigma^2 = 1$
$l = n^{\frac{1}{3}}$	Bias	0.1604	1.2828	0.1550	1.3232
	MSE	0.0276	1.8882	0.0243	1.8709
D1	Bias	0.1081	0.7303	0.0483	0.3747
	MSE	0.0195	0.7567	0.0035	0.1600
D2	Bias	0.2538	1.9604	0.1779	1.4475
	MSE	0.0742	5.1884	0.0468	3.3925
D3	Bias	**0.0486**	**0.2801**	**0.0242**	**0.1675**
	MSE	**0.0047**	**0.1174**	**0.0010**	**0.0363**
D4	Bias	0.0985	0.7542	0.0924	0.7873
	MSE	0.0214	1.7946	0.0194	1.7242
NPPI	Bias	0.1515	1.1947	0.1447	1.1488
	MSE	0.0331	2.1380	0.0303	1.8583
PW	Bias	0.0638	0.4337	0.0392	0.2751
	MSE	0.0082	0.3582	0.0038	0.1535

Method		n = 200		n = 500	
		φ = -0.5	$\sigma^2 = 1$	φ = -0.5	$\sigma^2 = 1$
$l = n^{\frac{1}{3}}$	Bias	0.0909	0.0959	0.0866	0.0988
	MSE	0.0111	0.0227	0.0086	0.0150
D1	Bias	0.1718	0.1876	0.1685	0.1872
	MSE	0.0313	0.0523	0.0291	0.0421
D2	Bias	0.1712	0.1865	0.1684	0.1869
	MSE	0.0311	0.0519	0.0291	0.0420
D3	Bias	0.0990	0.1149	0.0453	0.0559
	MSE	0.0119	0.0327	0.0029	0.0092
D4	Bias	**0.0554**	**0.0602**	**0.0209**	**0.0235**
	MSE	**0.0070**	0.0185	**0.0022**	**0.0059**
NPPI	Bias	0.1066	0.1036	0.0996	0.1143
	MSE	0.0174	0.0270	0.0142	0.0227
PW	Bias	0.0761	0.0759	0.0546	0.0551
	MSE	0.0107	**0.0183**	0.0053	0.0082

Table 11 continued

Method		n = 200		n = 500	
		φ = -0.1	σ^2 = 1	φ = -0.1	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	0.0177	-0.0050	0.0204	-0.0011
	MSE	0.0035	**0.0085**	0.0019	0.0045
D1	Bias	0.0351	0.0008	0.0281	-0.0017
	MSE	0.0036	0.0107	0.0018	0.0038
D2	Bias	0.0307	**0.0000**	0.0237	-0.0024
	MSE	0.0035	0.0107	0.0018	**0.0038**
D3	Bias	0.0120	-0.0042	0.0041	-0.0064
	MSE	0.0042	0.0105	0.0018	0.0038
D4	Bias	**0.0101**	-0.0045	**0.0028**	-0.0067
	MSE	0.0043	0.0104	0.0019	0.0038
NPPI	Bias	0.0201	-0.0064	0.0214	0.0010
	MSE	0.0039	0.0109	0.0020	0.0041
PW	Bias	0.0153	-0.0046	0.0235	**-0.0005**
	MSE	**0.0029**	0.0111	**0.0017**	0.0039

Method		n = 200		n = 500	
		φ = 0.1	σ^2 = 1	φ = 0.1	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	-0.0167	-0.0029	-0.0178	0.0011
	MSE	0.0036	0.0100	**0.0016**	0.0045
D1	Bias	-0.0181	-0.0101	-0.0161	0.0009
	MSE	0.0037	0.0109	0.0017	0.0039
D2	Bias	-0.0125	-0.0118	-0.0113	**0.0000**
	MSE	0.0040	0.0108	0.0017	0.0039
D3	Bias	-0.0057	-0.0130	-0.0056	-0.0015
	MSE	0.0042	0.0108	0.0017	0.0039
D4	Bias	**-0.0031**	-0.0133	**-0.0043**	-0.0013
	MSE	0.0044	0.0108	0.0039	0.0041
NPPI	Bias	-0.0208	**0.0013**	-0.0208	-0.0003
	MSE	0.0035	**0.0099**	0.0018	**0.0039**
PW	Bias	-0.0264	0.0022	-0.0295	0.0073
	MSE	**0.0027**	0.0106	0.0019	0.0040

Table 11 continued

Method		n = 200 φ = 0.5	n = 200 σ^2 = 1	n = 500 φ = 0.5	n = 500 σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	-0.0856	0.0969	-0.0872	0.0976
	MSE	0.0103	0.0233	0.0087	0.0154
D1	Bias	-0.0578	0.0547	-0.0361	0.0383
	MSE	0.0078	0.0152	0.0030	0.0059
D2	Bias	-0.0372	0.0296	-0.0249	0.0244
	MSE	0.0056	0.0116	0.0023	0.0049
D3	Bias	**-0.0057**	0.0477	-0.0277	0.0283
	MSE	0.0058	0.0144	0.0021	0.0050
D4	Bias	-0.0239	**0.0138**	**-0.0133**	**0.0097**
	MSE	**0.0044**	**0.0101**	**0.0016**	**0.0040**
NPPI	Bias	-0.0980	0.0895	-0.0901	0.0985
	MSE	0.0157	0.0252	0.0123	0.0178
PW	Bias	-0.0819	0.0879	-0.0522	0.0576
	MSE	0.0108	0.0211	0.0044	0.0078

Method		n = 200 φ = 0.9	n = 200 σ^2 = 1	n = 500 φ = 0.9	n = 500 σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	-0.1519	1.1748	-0.1504	1.2740
	MSE	0.0244	1.6206	0.0242	1.7337
D1	Bias	-0.1023	0.8834	-0.0354	0.2856
	MSE	0.0155	1.7157	0.0019	0.1357
D2	Bias	-0.0358	0.2018	**-0.0208**	**0.1435**
	MSE	0.0027	0.0638	**0.0009**	**0.0287**
D3	Bias	-0.2304	1.8753	-0.2674	2.1667
	MSE	0.0608	4.6777	0.0755	5.2750
D4	Bias	**-0.0318**	**0.1727**	-0.0219	0.1552
	MSE	**0.0023**	**0.0529**	0.0009	0.0350
NPPI	Bias	-0.1150	0.8323	-0.1240	1.0015
	MSE	0.0206	1.2060	0.0227	1.4276

Table 12 : Bias and MSE for $\hat{\varphi}$ and $\widehat{\sigma^2}$ in each block-length selection method under SBB – AR(1)

Method		n = 200		n = 500	
		φ = -0.9	$\sigma^2 = 1$	φ = -0.9	$\sigma^2 = 1$
$l = n^{\frac{1}{3}}$	Bias	0.1713	1.2801	0.1566	1.3225
	MSE	0.0285	1.8812	0.0248	1.8681
D1	Bias	0.1109	0.7458	0.0494	0.3838
	MSE	0.0203	0.7753	0.0036	0.1668
D2	Bias	0.2612	1.9863	0.1795	1.4496
	MSE	0.0767	5.1893	0.0474	3.3863
D3	Bias	**0.0499**	**0.2992**	**0.0248**	**0.1757**
	MSE	**0.0048**	**0.1287**	**0.0011**	**0.0393**
D4	Bias	0.1008	0.7670	0.0939	0.7959
	MSE	0.0223	1.7994	0.0198	1.7292
PW	Bias	0.0717	0.5010	0.1447	1.1488
	MSE	0.0099	0.4439	0.0303	1.8583

Method		n = 200		n = 500	
		φ = -0.5	$\sigma^2 = 1$	φ = -0.5	$\sigma^2 = 1$
$l = n^{\frac{1}{3}}$	Bias	0.0912	0.0955	0.0868	0.0987
	MSE	0.0111	0.0226	0.0086	0.0150
D1	Bias	0.1729	0.1862	0.1689	0.1867
	MSE	0.0317	0.0518	0.0292	0.0418
D2	Bias	0.1721	0.1857	0.1690	0.1870
	MSE	0.0314	0.0517	0.0292	0.0420
D3	Bias	0.1003	0.1157	0.0459	0.0564
	MSE	0.0122	0.0328	0.0030	0.0092
D4	Bias	**0.0563**	**0.0621**	**0.0215**	**0.0246**
	MSE	**0.0071**	**0.0186**	**0.0023**	**0.0060**
PW	Bias	0.0852	0.0873	0.0616	0.0631
	MSE	0.0125	0.0207	0.0062	0.0094

Table 12 Continued

Method		n = 200		n = 500	
		φ = -0.1	σ^2 = 1	φ = -0.1	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	0.0178	-0.0048	0.0205	-0.0011
	MSE	0.0036	**0.0085**	0.0019	0.0045
D1	Bias	0.0351	0.0009	0.0283	-0.0016
	MSE	0.0035	0.0107	0.0018	0.0039
D2	Bias	0.0311	**0.0001**	0.0238	-0.0023
	MSE	0.0035	0.0107	**0.0018**	0.0039
D3	Bias	0.0123	-0.0040	0.0043	-0.0061
	MSE	0.0042	0.0104	0.0018	**0.0038**
D4	Bias	**0.0102**	-0.0039	**0.0028**	-0.0064
	MSE	0.0043	0.0105	0.0043	0.0038
PW	Bias	0.0153	-0.0030	0.0266	**0.0003**
	MSE	**0.0026**	0.0111	0.0018	0.0039

Method		n = 200		n = 500	
		φ = 0.1	σ^2 = 1	φ = 0.1	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	-0.0169	**-0.0025**	-0.0179	0.0011
	MSE	0.0036	**0.0100**	**0.0016**	0.0046
D1	Bias	-0.0185	-0.0102	-0.0163	0.0009
	MSE	0.0037	0.0108	0.0017	0.0039
D2	Bias	-0.0125	-0.0114	-0.0115	**0.0000**
	MSE	0.0040	0.0108	0.0017	0.0039
D3	Bias	-0.0060	-0.0125	-0.0057	-0.0011
	MSE	0.0042	0.0109	0.0017	**0.0039**
D4	Bias	**-0.0036**	-0.0126	**-0.0043**	-0.0012
	MSE	0.0044	0.0108	0.0017	0.0039
PW	Bias	-0.0265	0.0055	-0.0321	0.0084
	MSE	**0.0022**	0.0104	0.0018	0.0040

Table 12 Continued

Method		n = 200		n = 500	
		φ = 0.5	$\sigma^2 = 1$	φ = 0.5	$\sigma^2 = 1$
$l = n^{\frac{1}{3}}$	Bias	-0.0878	0.0972	-0.0872	0.0980
	MSE	0.0103	0.0234	0.0087	0.0155
D1	Bias	-0.0585	0.0557	-0.0365	0.0386
	MSE	0.0078	0.0152	0.0030	0.0059
D2	Bias	**-0.0382**	0.0316	-0.0253	0.0248
	MSE	0.0057	0.0118	0.0023	0.0050
D3	Bias	-0.0508	0.0490	-0.0280	0.0289
	MSE	0.0059	0.0146	**0.0021**	0.0050
D4	Bias	-0.0249	**0.0156**	**-0.0137**	**0.0106**
	MSE	**0.0044**	**0.0102**	0.0017	**0.0040**
PW	Bias	-0.0940	0.1016	-0.0599	0.0668
	MSE	0.0130	0.0252	0.0053	0.0091

Method		n = 200		n = 500	
		φ = 0.9	$\sigma^2 = 1$	φ = 0.9	$\sigma^2 = 1$
$l = n^{\frac{1}{3}}$	Bias	-0.1545	1.1748	-0.1517	1.2721
	MSE	0.0252	1.6195	0.0234	1.7280
D1	Bias	-0.1047	0.8937	-0.0361	0.2929
	MSE	0.0161	1.7231	0.0019	0.1395
D2	Bias	-0.0367	0.2205	**-0.0215**	**0.1528**
	MSE	0.0027	0.0725	**0.0009**	**0.0315**
D3	Bias	-0.2342	1.8776	-0.2694	2.1663
	MSE	0.0627	4.6810	0.0766	5.2683
D4	Bias	**-0.0326**	**0.1897**	-0.0226	0.1652
	MSE	**0.0024**	**0.0599**	0.0010	0.0381
PW	Bias	-0.0734	0.4984	-0.0446	0.3243
	MSE	0.0071	0.2967	0.0025	0.1180

Table 13: Bias and MSE for $\hat{\theta}$ and $\widehat{\sigma^2}$ in each block-length selection method under CBB – MA(1)

Method		n = 200		n = 500	
		θ = -0.9	σ^2 = 1	θ = -0.9	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	0.3749	0.3957	0.3693	0.4125
	MSE	0.1412	0.1809	0.1366	0.1793
D1	Bias	0.5223	0.5746	0.5202	0.5836
	MSE	0.2739	0.3618	0.2710	0.3546
D2	Bias	0.5221	0.5749	0.5202	0.5834
	MSE	0.2736	0.3622	0.2709	0.3550
D3	Bias	0.5155	0.5650	0.5194	0.5819
	MSE	0.2690	0.3567	0.2705	0.3533
D4	Bias	0.4866	0.5309	0.4948	0.5509
	MSE	0.2440	0.3248	0.2544	0.3319
NPPI	Bias	0.3676	0.3941	0.3726	0.4037
	MSE	0.1467	0.1950	0.1515	0.1861
PW	Bias	**0.1228**	**0.1210**	**0.0887**	**0.0840**
	MSE	**0.0162**	**0.0293**	**0.0082**	**0.0123**

Method		n = 200		n = 500	
		θ = -0.5	σ^2 = 1	θ = -0.5	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	0.1229	0.0879	0.1171	0.0905
	MSE	0.0170	0.0198	0.0145	0.0136
D1	Bias	0.2132	0.1546	0.2123	0.1403
	MSE	0.0468	0.0402	0.0455	0.0254
D2	Bias	0.2103	0.1544	0.2120	0.1404
	MSE	0.0463	0.0402	0.0454	0.0254
D3	Bias	0.1172	0.0874	0.0653	0.0360
	MSE	0.0160	0.0256	0.0051	0.0066
D4	Bias	0.0921	0.0722	0.0453	0.0233
	MSE	0.0118	0.0209	0.0041	0.0057
NPPI	Bias	0.1297	0.0874	0.1271	0.0907
	MSE	0.0230	0.0231	0.0209	0.0154
PW	Bias	**0.0430**	**0.0288**	**0.0336**	**0.0174**
	MSE	**0.0054**	**0.0109**	**0.0028**	**0.0046**

Table 13 continued

Method		n = 200		n = 500	
		θ = -0.1	σ^2 = 1	θ = -0.1	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	0.0189	-0.0022	0.0178	-0.0034
	MSE	0.0035	0.0095	**0.0015**	0.0039
D1	Bias	0.0324	0.0016	0.0274	-0.0025
	MSE	0.0036	0.0103	0.0018	**0.0037**
D2	Bias	0.0277	0.0015	0.0226	-0.0034
	MSE	0.0036	0.0104	0.0017	0.0037
D3	Bias	**0.0079**	-0.0099	0.0031	-0.0095
	MSE	0.0046	0.0106	0.0019	0.0039
D4	Bias	0.0107	-0.0023	**0.0015**	-0.0066
	MSE	0.0049	**0.0051**	0.0020	0.0039
NPPI	Bias	0.0175	**0.0005**	0.0209	**0.0015**
	MSE	0.0038	0.0107	0.0017	0.0040
PW	Bias	0.0143	-0.0018	0.0324	0.0045
	MSE	**0.0028**	0.0095	0.0019	0.0042

Method		n = 200		n = 500	
		θ = 0.1	σ^2 = 1	θ = 0.1	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	-0.0194	-0.0057	-0.0163	-0.0011
	MSE	0.0037	0.0101	**0.0016**	0.0043
D1	Bias	-0.0218	-0.0062	-0.0121	**-0.0006**
	MSE	0.0037	**0.0097**	0.0017	**0.0037**
D2	Bias	-0.0153	-0.0075	-0.0069	-0.0015
	MSE	0.0040	0.0098	0.0018	0.0038
D3	Bias	-0.0081	-0.0153	-0.0009	-0.0051
	MSE	0.0046	0.0102	0.0018	0.0038
D4	Bias	**-0.0057**	-0.0078	**0.0005**	-0.0027
	MSE	0.0048	0.0105	0.0019	0.0039
NPPI	Bias	-0.0244	-0.0806	-0.0231	0.0019
	MSE	0.0042	0.0878	0.0018	0.0039
PW	Bias	-0.0268	**0.0025**	-0.1463	0.4418
	MSE	**0.0026**	0.0100	0.0022	0.0047

Table 13 continued

Method		n = 200		n = 500	
		θ = 0.5	$\sigma^2 = 1$	θ = 0.5	$\sigma^2 = 1$
$l = n^{\frac{1}{3}}$	Bias	-0.1166	0.0929	-0.1177	0.0938
	MSE	0.0153	0.0200	0.0145	0.0136
D1	Bias	-0.104	0.0650	-0.0693	0.0503
	MSE	0.0162	0.0177	0.0072	0.0079
D2	Bias	-0.0716	0.0408	-0.0511	0.0369
	MSE	0.0103	0.0148	0.0050	0.0066
D3	Bias	-0.0705	0.0371	-0.0389	0.0292
	MSE	0.0079	0.0151	0.0024	0.0261
D4	Bias	**-0.0289**	**0.0124**	**-0.0124**	**0.0076**
	MSE	**0.0043**	**0.0114**	**0.0013**	**0.0044**
NPPI	Bias	-0.1303	0.0862	-0.1274	0.0888
	MSE	0.0236	0.0234	0.0208	0.0140
PW	Bias	-0.1761	0.1295	-0.1463	0.4418
	MSE	0.0370	0.0319	0.0242	0.0166

Method		n = 200		n = 500	
		θ = 0.9	$\sigma^2 = 1$	θ = 0.9	$\sigma^2 = 1$
$l = n^{\frac{1}{3}}$	Bias	-0.3714	0.4099	-0.3687	0.4044
	MSE	0.1387	0.1921	0.1362	0.1728
D1	Bias	-0.4989	0.5460	-0.5133	0.5742
	MSE	0.2550	0.3397	0.2660	0.3468
D2	Bias	-0.3092	0.3254	-0.2839	0.3083
	MSE	0.1071	0.1447	0.0914	0.1187
D3	Bias	-0.4781	0.5182	-0.5053	0.5652
	MSE	0.2413	0.3253	0.2605	0.3409
D4	Bias	**-0.1797**	**0.1723**	**-0.1522**	**0.1539**
	MSE	**0.0368**	**0.0526**	**0.0258**	**0.0345**
NPPI	Bias	-0.3708	0.3925	-0.3764	0.4103
	MSE	0.1502	0.1919	0.1543	0.1940
PW	Bias	-0.4646	0.5187	-0.3965	0.2791
	MSE	0.2254	0.3069	0.1619	0.2088

Table 14: Bias and MSE for $\hat{\theta}$ and $\widehat{\sigma^2}$ in each block-length selection method under SBB – MA(1)

Method		n = 200		n = 500	
		θ = -0.9	σ^2 = 1	θ = -0.9	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	0.3702	0.3927	0.3676	0.4115
	MSE	0.1376	0.1785	0.1354	0.1785
D1	Bias	0.5758	0.7458	0.5196	0.5835
	MSE	0.0203	0.3633	0.2702	0.3545
D2	Bias	0.5765	1.9863	0.5198	0.5833
	MSE	0.0767	0.3643	0.2704	0.3543
D3	Bias	0.5688	0.2992	0.5186	0.5819
	MSE	**0.0048**	0.3594	0.2696	0.3531
D4	Bias	0.5362	0.7670	0.4944	0.5545
	MSE	0.0223	0.3276	0.2536	0.3326
PW	Bias	**0.1287**	**0.1311**	**0.0900**	**0.0862**
	MSE	0.0179	**0.0325**	**0.0085**	**0.0127**

Method		n = 200		n = 500	
		θ = -0.5	σ^2 = 1	θ = -0.5	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	**-0.0297**	0.3927	0.1167	0.0900
	MSE	**0.0014**	0.1785	0.0144	0.0135
D1	Bias	0.2131	0.1567	0.2121	0.1407
	MSE	0.0465	0.0409	0.0453	0.0255
D2	Bias	0.2117	0.1559	0.2119	0.1408
	MSE	0.0459	0.0407	0.0453	0.0255
D3	Bias	0.1180	0.0937	0.0657	0.0393
	MSE	0.0161	0.0257	0.0052	0.0066
D4	Bias	0.0928	0.0730	0.0459	0.0234
	MSE	0.0118	0.0205	0.0041	0.0055
PW	Bias	0.0504	**0.0350**	**0.0391**	**0.0218**
	MSE	0.0061	**0.0114**	**0.0032**	**0.0049**

Table 14 continued

Method		n = 200		n = 500	
		θ = -0.1	σ^2 = 1	θ = -0.1	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	0.0189	-0.0025	0.0181	-0.0034
	MSE	0.0035	**0.0095**	**0.0016**	0.0039
D1	Bias	0.0326	0.0019	0.0276	-0.0031
	MSE	0.0036	0.0102	0.0018	**0.0037**
D2	Bias	0.0275	0.0010	0.0226	-0.0039
	MSE	0.0036	0.0102	0.0017	**0.0037**
D3	Bias	0.0081	-0.0029	0.0032	-0.0076
	MSE	0.0046	0.0101	0.0019	0.0037
D4	Bias	**0.0054**	-0.0033	**0.0028**	-0.0080
	MSE	0.0049	0.0101	0.0020	**0.0037**
PW	Bias	0.0147	**-0.0002**	0.0277	**0.0008**
	MSE	**0.0023**	0.0097	0.0020	0.0043

Method		n = 200		n = 500	
		θ = 0.1	σ^2 = 1	θ = 0.1	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	-0.0193	-0.0061	-0.0164	-0.0012
	MSE	0.0037	0.0101	**0.0016**	0.0043
D1	Bias	-0.0221	-0.0060	-0.0120	**-0.0003**
	MSE	0.0038	**0.0095**	0.0017	**0.0037**
D2	Bias	-0.0160	-0.0072	-0.0071	-0.0013
	MSE	0.0040	0.0096	0.0018	0.0038
D3	Bias	**-0.0085**	-0.0088	0.0038	-0.0024
	MSE	0.0046	**0.0095**	0.0018	0.0038
D4	Bias	-0.0089	-0.0089	**0.0003**	-0.0024
	MSE	0.0048	**0.0095**	0.0019	**0.0037**
PW	Bias	-0.0274	**0.0036**	-0.0348	0.0066
	MSE	**0.0021**	0.0096	0.0022	0.0047

Table 14 continued

Method		n = 200		n = 500	
		θ = 0.5	$\sigma^2 = 1$	θ = 0.5	$\sigma^2 = 1$
$l = n^{\frac{1}{3}}$	Bias	-0.1158	0.0931	-0.1176	0.0935
	MSE	0.0151	0.0201	0.0145	0.0135
D1	Bias	-0.1043	0.0674	-0.0694	0.0520
	MSE	0.0160	0.0178	0.0072	0.0080
D2	Bias	-0.0725	0.0444	-0.0518	0.0387
	MSE	0.0103	0.0146	0.0050	0.0066
D3	Bias	-0.0308	0.0456	-0.0394	0.0298
	MSE	0.0080	0.0144	0.0024	0.0054
D4	Bias	**-0.0249**	**0.0153**	**-0.0134**	**0.0095**
	MSE	**0.0044**	**0.0107**	**0.0013**	**0.0043**
PW	Bias	-0.1917	0.1414	-0.1672	0.1153
	MSE	0.0420	0.0355	0.0331	0.0198

Method		n = 200		n = 500	
		θ = 0.9	$\sigma^2 = 1$	θ = 0.9	$\sigma^2 = 1$
$l = n^{\frac{1}{3}}$	Bias	-0.3669	0.4070	-0.3668	0.4032
	MSE	0.1354	0.1899	0.1348	0.1718
D1	Bias	-0.4972	0.5489	-0.5128	0.5752
	MSE	0.2530	0.3419	0.2653	0.3475
D2	Bias	-0.3060	0.3329	-0.2827	0.3123
	MSE	0.1049	0.1477	0.0906	0.1203
D3	Bias	-0.4769	0.5267	-0.5050	0.5672
	MSE	0.2396	0.3267	0.2600	0.3418
D4	Bias	**-0.1785**	**0.1907**	**-0.1520**	**0.1635**
	MSE	**0.0360**	**0.0575**	**0.0257**	**0.0360**
PW	Bias	-0.4890	0.5511	-0.4216	0.4707
	MSE	0.2478	0.3430	0.1823	0.2351

Table 15: Bias and MSE for $\hat{\varphi}, \hat{\theta}$ and $\hat{\sigma}^2$ in each block-length selection method under CBB – ARMA(1,1) with parameters $\varphi = -0.8, \theta = 0.5$

Method		n = 200			n = 500		
		φ = -0.8	θ = 0.5	σ^2 = 1	φ = -0.8	θ = 0.5	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	0.2363	-0.2262	0.0832	0.1461	-0.1411	0.0703
	MSE	0.0649	0.0612	0.0207	0.0237	0.0240	0.0098
D1	Bias	0.4308	-0.3858	0.1413	0.4108	-0.3724	0.1478
	MSE	0.1952	0.1590	0.0395	0.1771	0.1469	0.0294
D2	Bias	0.3917	-0.3526	0.1283	0.3730	-0.3402	0.1359
	MSE	0.1727	0.1432	0.0358	0.1578	0.1327	0.0267
D3	Bias	0.2056	-0.1978	0.0730	0.0895	-0.0942	0.0415
	MSE	0.0556	0.0591	0.0236	0.0119	0.0158	0.0076
D4	Bias	0.1134	-0.1106	0.0222	0.0408	-0.0427	0.0090
	MSE	0.0276	0.0323	0.0124	0.0056	0.0085	0.0040
NPPI	Bias	0.3116	-0.2740	0.0116	0.2680	-0.2359	0.0241
	MSE	0.1519	0.1084	0.0869	0.1188	0.0839	0.0747
PW	Bias	0.2566	-0.2391	0.0723	0.1887	-0.1772	0.0718
	MSE	0.1113	0.0938	0.0226	0.0704	0.0597	0.0127

Table 17: Bias and MSE for $\hat{\varphi}, \hat{\theta}$ and $\hat{\sigma}^2$ in each block-length selection method under CBB – ARMA(1,1) with parameters $\varphi = 0.5, \theta = -0.8$

Method		n = 200			n = 500		
		φ = 0.5	θ = -0.8	σ^2 = 1	φ = 0.5	θ = -0.8	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	-0.2156	0.2967	0.0515	-0.1234	0.1812	0.0523
	MSE	0.0645	0.1054	0.0147	0.0212	0.0369	0.0079
D1	Bias	-0.3322	0.4619	0.0764	-0.1928	0.2981	0.0751
	MSE	0.1232	0.2251	0.0193	0.0466	0.0965	0.0099
D2	Bias	-0.2972	0.4160	0.0717	-0.1787	0.2780	0.0717
	MSE	0.1032	0.1857	0.0184	0.0419	0.0852	0.0093
D3	Bias	-0.1845	0.2466	0.0412	-0.0714	0.1124	0.0360
	MSE	0.0538	0.0789	0.0146	0.0118	0.0159	0.0052
D4	Bias	-0.1539	0.1997	0.0312	-0.0570	0.0871	0.0276
	MSE	0.0424	0.0557	0.0134	0.0098	0.0113	0.0048
NPPI	Bias	-0.2709	0.3817	-0.0341	-0.2179	0.3332	-0.0377
	MSE	0.1026	0.1938	0.0965	0.0754	0.1601	0.0956
PW	Bias	-0.2323	0.2947	0.0395	-0.0552	0.0795	0.0180
	MSE	0.0723	0.1186	0.0131	0.0104	0.0115	0.0044

Table 16 : Bias and MSE for $\hat{\varphi}, \hat{\theta}$ and $\hat{\sigma}^2$ in each block-length selection method under SBB – ARMA(1,1) with parameters $\varphi = -0.8, \theta = 0.5$

Method		n = 200			n = 500		
		φ = -0.8	θ = 0.5	σ^2 = 1	φ = -0.8	θ = 0.5	σ^2 = 1
$l = n^{\frac{1}{3}}$	Bias	0.1988	-0.1785	0.0780	0.1194	-0.1067	0.0667
	MSE	0.0505	0.0442	0.0194	0.0171	0.0162	0.0092
D1	Bias	0.3256	-0.2671	0.1341	0.2757	-0.2228	0.1415
	MSE	0.1162	0.0813	0.0369	0.0813	0.0550	0.0273
D2	Bias	0.2993	-0.2472	0.1225	0.2538	-0.2066	0.1300
	MSE	0.1047	0.0752	0.0338	0.0738	0.0511	0.0248
D3	Bias	0.1739	-0.1575	0.0700	0.0787	-0.0775	0.0398
	MSE	0.0413	0.0410	0.0226	0.0092	0.0120	0.0072
D4	Bias	0.1050	-0.0992	0.0236	0.0394	-0.0406	0.0097
	MSE	0.0247	0.0286	0.0123	0.0056	0.0081	0.0041
PW	Bias	0.2139	-0.1844	0.0783	0.1488	-0.1263	0.0760
	MSE	0.0711	0.0542	0.0235	0.0361	0.0261	0.0136

Table 18: Bias and MSE for $\hat{\varphi}, \hat{\theta}$ and $\hat{\sigma}^2$ in each block-length selection method under SBB – ARMA(1,1) with parameters $\varphi = 0.5, \theta = -0.8$

Method		n = 200			n = 500		
		φ = 0.5	θ = -0.8	σ^2 = 1	φ = 0.5	θ = -0.8	σ^2 =1
$l = n^{\frac{1}{3}}$	Bias	-0.2171	0.3085	0.0780	-0.1293	0.1893	0.0592
	MSE	0.0679	0.1096	0.0194	0.0223	0.0412	0.0085
D1	Bias	-0.2762	0.4005	0.0732	-0.1426	0.2444	0.0094
	MSE	0.0906	0.1726	0.0187	0.0290	0.0663	0.0721
D2	Bias	-0.2538	0.3682	0.0686	-0.1322	0.2285	0.0689
	MSE	0.0799	0.1484	0.0179	0.0259	0.0586	0.0089
D3	Bias	-0.1665	0.2267	0.0397	-0.0638	0.1043	0.0351
	MSE	0.0466	0.0680	0.0144	0.0105	0.0140	0.0051
D4	Bias	-0.1423	0.1871	0.0302	-0.0522	0.0820	0.0277
	MSE	0.0392	0.0506	0.0132	0.0092	0.0103	0.0047
PW	Bias	-0.2190	0.2863	0.0420	-0.0548	0.0830	0.0204
	MSE	0.0648	0.1095	0.0133	0.0103	0.0119	0.0045

Table 19: Bias and MSE for $\hat{a}_1$, $\hat{a}_2$ and $\hat{\lambda}$ in each block-length selection method under CBB – INAR with parameters $a_1 = 0.5$, $a_2 = 0.3, \lambda = 2$

Method		n = 200			n = 500		
		$a_1 = 0.5$	$a_2 = 0.3$	λ = 2	$a_1 = 0.5$	$a_2 = 0.3$	λ =2
$l = n^{\frac{1}{3}}$	Bias	**0.0040**	-0.1881	1.7061	0.0164	-0.1440	1.2293
	MSE	**0.0037**	0.0369	3.1498	0.0015	0.0213	1.6107
D1	Bias	0.0090	-0.0978	0.7792	0.0085	-0.0533	0.4117
	MSE	0.0049	0.0136	0.9433	0.0013	0.0043	0.2883
D2	Bias	0.0060	-0.0726	0.5645	0.0054	-0.0400	0.3092
	MSE	0.0048	0.0099	0.6062	0.0013	0.0030	0.2103
D3	Bias	0.0076	-0.1397	1.2026	0.0135	-0.1048	0.8741
	MSE	0.0038	0.0244	1.8528	0.0015	0.0143	0.9917
D4	Bias	0.0059	**-0.0627**	**0.4656**	0.0039	**-0.0356**	**0.2750**
	MSE	0.0048	**0.0088**	**0.4925**	**0.0012**	**0.0027**	**0.1924**
NPPI	Bias	-0.0056	-0.1561	1.4928	-0.0040	-0.1606	1.5892
	MSE	0.0049	0.0303	3.4518	0.0027	0.0317	3.6605
PW	Bias	0.0082	-0.1038	0.8479	**-0.0016**	-0.0384	0.3216
	MSE	0.0046	0.0150	1.0717	0.0022	0.0041	0.3471

Table 21: Bias and MSE for $\hat{a}_1$, $\hat{a}_2$ and $\hat{\lambda}$ in each block-length selection method under CBB – INAR(2) with parameters $a_1 = 0.5$, $a_2 = 0.3, \lambda = 5$

Method		n = 200			n = 500		
		$a_1 = 0.5$	$a_2 = 0.3$	λ = 5	$a_1 = 0.5$	$a_2 = 0.3$	λ =5
$l = n^{\frac{1}{3}}$	Bias	0.0177	-0.2042	4.4337	0.0270	-0.1480	2.9131
	MSE	**0.0031**	0.0427	20.8070	0.0022	0.0226	8.8922
D1	Bias	0.0273	-0.1279	2.3313	0.0216	-0.0862	1.5157
	MSE	0.0042	0.0198	8.2878	0.0020	0.0093	3.3585
D2	Bias	0.0304	-0.0984	1.5367	0.0192	-0.0592	0.9100
	MSE	0.0046	0.0129	3.9983	0.0020	0.0051	1.3335
D3	Bias	0.0286	-0.1118	1.9056	0.0218	-0.1298	2.5914
	MSE	0.0042	0.0158	5.5530	0.0021	0.0210	9.1559
D4	Bias	0.0312	**-0.0928**	**1.3818**	0.0170	**-0.0492**	**0.7208**
	MSE	0.0049	**0.0118**	**3.6406**	**0.0019**	**0.0039**	**1.0251**
NPPI	Bias	**0.0081**	-0.1816	4.1240	0.0206	-0.0683	1.1020
	MSE	0.0044	0.0369	22.9463	0.0020	0.0060	1.8502
PW	Bias	0.0273	-0.1277	2.3348	0.0250	-0.0958	1.6754
	MSE	0.0039	0.0187	7.1933	0.0022	0.0106	3.3098

Table 20 : Bias and MSE for $\hat{a}_1$, $\hat{a}_2$ and $\hat{\lambda}$ in each block-length selection method under SBB – INAR(2) with parameters $a_1 = 0.5$, $a_2 = 0.3$, $\lambda = 2$

Method		n = 200			n = 500		
		$a_1 = 0.5$	$a_2 = 0.3$	λ = 2	$a_1 = 0.5$	$a_2 = 0.3$	λ =2
$l = n^{\frac{1}{3}}$	Bias	-0.0107	-0.1657	1.6273	0.0048	-0.1278	1.1835
	MSE	**0.0035**	0.0290	2.8738	0.0011	0.0169	1.4976
D1	Bias	0.0061	-0.0942	0.7641	0.2757	-0.2228	0.1415
	MSE	0.0048	0.0128	0.9119	0.0813	0.0550	0.0273
D2	Bias	0.0064	-0.0709	0.5444	0.2538	-0.2066	0.1300
	MSE	0.0048	0.0097	0.5854	0.0738	0.0511	0.0248
D3	Bias	**-0.0026**	-0.1264	1.1722	0.0787	-0.0775	0.0398
	MSE	**0.0035**	0.0199	1.7539	0.0092	0.0120	0.0072
D4	Bias	0.0073	**-0.0612**	0.4413	0.0394	-0.0406	**0.0097**
	MSE	0.0049	**0.0088**	0.4712	0.0056	0.0081	**0.0041**
PW	Bias	0.0034	-0.1080	0.9304	**-0.0006**	**-0.0325**	0.3185
	MSE	0.0043	0.0153	1.2241	**0.0014**	**0.0032**	0.3330

Table 22: Bias and MSE for $\hat{a}_1$, $\hat{a}_2$ and $\hat{\lambda}$ in each block-length selection method under SBB – INAR(2) with parameters $a_1 = 0.5$, $a_2 = 0.3$, $\lambda = 5$

Method		n = 200			n = 500		
		$a_1 = 0.5$	$a_2 = 0.3$	λ = 5	$a_1 = 0.5$	$a_2 = 0.3$	λ = 5
$l = n^{\frac{1}{3}}$	Bias	**0.0024**	-0.1821	4.2639	**0.0146**	-0.1325	2.8308
	MSE	**0.0026**	0.0342	19.3721	**0.0015**	0.0182	8.4199
D1	Bias	0.0250	-0.1256	2.3389	-0.1426	0.2444	0.0094
	MSE	0.0044	0.0189	8.2312	0.0290	0.0663	0.0721
D2	Bias	0.0303	-0.0985	1.5420	-0.1322	0.2285	0.0689
	MSE	0.0046	0.0128	4.0823	0.0259	0.0586	0.0089
D3	Bias	0.0293	-0.1112	1.8751	-0.0638	0.1043	0.0351
	MSE	0.0043	0.0158	5.4174	0.0105	0.0140	0.0051
D4	Bias	0.0328	**-0.0935**	**1.3630**	-0.0522	**0.0820**	**0.0277**
	MSE	0.0048	**0.0121**	**3.6035**	0.0092	**0.0103**	**0.0047**
PW	Bias	0.0249	-0.1322	2.4997	0.0192	-0.0977	1.8644
	MSE	0.0040	0.0199	8.0909	0.0018	0.0109	4.0021

Table 23 : Frequencies of non-convergence - MBB-AR

n = 200	-0.9	-0.5	-0.1	0.1	0.5	0.9
Method						
HHJ	0	0	4	11	27	7
NPPI	12	15	26	23	22	12
n = 500						
Method						
HHJ	0	0	2	10	26	6
NPPI	11	31	29	19	25	14

Table 24 : Frequencies of non-convergence - CBB-AR

n = 200	-0.9	-0.5	-0.1	0.1	0.5	0.9
Method						
NPPI	18	33	38	31	24	16
PW	5	13	102	151	1	0
n = 500						
Method						
NPPI	19	30	28	36	32	13
PW	3	3	41	39	0	0

Table 25 : Frequencies of non-convergence - SBB-AR

n = 200	-0.9	-0.5	-0.1	0.1	0.5	0.9
Method						
PW	7	18	132	207	3	0
n = 500						
Method						
PW	3	5	49	57	0	0

Table 26 : Frequencies of non-convergence - MBB-MA

n = 200	-0.9	-0.5	-0.1	0.1	0.5	0.9
Method						
HHJ	1	0	8	8	11	14
NPPI	25	19	21	12	18	18
n = 500						
Method						
HHJ	0	0	0	5	20	11
NPPI	22	23	23	35	29	29

Table 27 : Frequencies of non-convergence - CBB-MA

n = 200	-0.9	-0.5	-0.1	0.1	0.5	0.9
Method						
NPPI	39	24	46	39	45	37
PW	0	0	106	143	45	53
n = 500						
Method						
NPPI	32	30	35	30	30	26
PW	0	8	29	49	13	4

Table 28 : Frequencies of non-convergence - SBB-MA

n = 200	-0.9	-0.5	-0.1	0.1	0.5	0.9
Method						
PW	0	0	134`	194	57	62
n = 500						
Method						
PW	0	0	36	73	14	6

Table 29 : Frequencies of non-convergence - ARMA

n = 200	$(\varphi,\theta) = (-0.8,0.5)$			$(\varphi,\theta) = (0.5,-0.8)$		
Method	MBB	CBB	SBB	MBB	CBB	SBB
HHJ	0	-	-	0	-	-
NPPI	14	33	-	27	42	-
PW	-	32	38	-	0	1
n = 500						
HHJ	0	-	-	0	-	-
NPPI	32	30		27	44	-
PW	-	21	24	-	0	0

Table 30 : Frequencies of non-convergence - INAR

n = 200	$(a_1,a_2,\lambda) = (0.5,0.3,2)$			$(a_1,a_2,\lambda) = (0.5,0.3,5)$		
Method	MBB	CBB	SBB	MBB	CBB	SBB
HHJ	15	-	-	14	-	-
NPPI	15	30	-	0	15	-
PW	-	0	0	-	0	0
n = 500						
HHJ	22	-	-	20	-	-
NPPI	10	25		5	0	-
PW	-	0	0	-	0	0

Table 31: Bias and MSE for $\hat{\varphi}$ and $\hat{\sigma}$ under NBB using Nordman's rule AR(1)

		n = 200		n = 500	
		φ	s^2	φ	s^2
$\varphi = -0.9$	Bias	0.4108	2.9054	0.4324	3.0571
	MSE	0.1732	9.7154	0.1881	9.9334
$\varphi = -0.5$	Bias	0.2473	0.2287	0.2469	0.2482
	MSE	0.0628	0.0178	0.0615	0.0703
$\varphi = -0.1$	Bias	0.0417	0.0068	0.0473	0.0020
	MSE	0.0046	0.0106	0.0033	0.0040
$\varphi = 0.1$	Bias	-0.0363	-0.0036	-0.0397	0.0085
	MSE	0.0044	0.1050	0.0028	0.0041
$\varphi = 0.5$	Bias	-0.0979	0.1031	-0.0958	0.0962
	MSE	0.0194	0.0332	0.0180	0.0242
$\varphi = 0.9$	Bias	-0.0463	0.2905	-0.0335	0.2198
	MSE	0.0046	0.2228	0.0057	0.2543

Table 32 : Bias and MSE for $\hat{\theta}$ and $\widehat{\sigma^2}$ under NBB using Nordman's rule – MA(1)

		n = 200		n = 500	
		θ	σ^2	θ	σ^2
$\theta = -0.9$	Bias	0.6270	0.6827	0.6316	0.6833
	MSE	0.3956	0.5063	0.3997	0.4812
$\theta = -0.5$	Bias	0.2857	0.1923	0.2916	0.1908
	MSE	0.0841	0.0548	0.0860	0.0434
$\theta = -0.1$	Bias	0.0436	0.0048	0.0477	0.0028
	MSE	0.0045	0.0101	0.0034	0.0043
$\theta = 0.1$	Bias	-0.0393	-0.0044	-0.0397	-0.0003
	MSE	0.0053	0.0112	0.0030	0.0037
$\theta = 0.5$	Bias	-0.1928	0.1247	-0.1992	0.1386
	MSE	0.0494	0.0342	0.0486	0.0284
$\theta = 0.9$	Bias	-0.4554	0.4946	-0.4778	0.5142
	MSE	0.2407	0.3143	0.2590	0.3137

Table 33: Bias and MSE for $\hat{\varphi}$, $\hat{\theta}$ and $\widehat{\sigma^2}$ under NBB using Nordman's rule – ARMA(1)

Model		n = 200			n = 500		
		φ	θ	σ^2	φ	θ	σ^2
$\varphi = -0.8$	Bias	0.6654	-0.5743	0.1800	0.7451	-0.6549	0.1914
$\theta = 0.5$	MSE	0.4685	0.3656	0.0537	0.5595	0.4352	0.0460
$\varphi = 0.5$	Bias	-0.5554	0.7398	0.0919	-0.5302	0.7208	0.0940
$\theta = -0.8$	MSE	0.3135	0.5558	0.0224	0.2830	0.5231	0.0137

Table 34: Bias and MSE for $\hat{\alpha}_1$, $\hat{\alpha}_2$ and $\widehat{\sigma^2}$ under NBB using rule – INAR(2)

Model		n = 200			n = 500		
		a_1	a_2	λ	a_1	a_2	λ
$a_1 = 0.3$	Bias	-0.0236	-0.0958	0.9732	-0.0078	-0.0617	0.4568
$a_2 = 0.5$	MSE	0.0125	0.0198	1.7423	0.0087	0.0094	0.5349
$\lambda = 2.$							
$a_1 = 0.3$	Bias	0.0028	-0.1001	2.5632	0.0096	-0.0643	1.0069
$a_2 = 0.5$	MSE	0.0101	0.0195	6.8324	0.0074	0.0102	1.7348
$\lambda = 5.$							