ANALYSING ROUNDING DATA USING RADIAL BASIS FUNCTION NEURAL NETWORKS MODEL

Submitted for the Degree of Master of Philosophy
At the University of Northampton

2007

ENDANG TRIASTUTI SUGIYARTO

ACKNOWLEDGMENTS

I am truly indebted to **Dr. Robin Crockett** and **Professor Phil Picton**, my internal supervisors, for their continuous guidance and competent supervisions in all stages of modelling development as well as during the writing up. Their supervisions are not only reflected on the quality but also on the readability of this thesis. I should also be very grateful to **Dr. Alasdair Crockett**, my external supervisor who passed away in 2006, for providing the data used for the modelling applications as well as for the suggestions in the early process of developing this thesis. **Robin** and **Alasdair** are also the ones who introduced me to the topic that they have done some research before.

I would like to express my highest gratitude to the University of Northampton which has awarded me a bursary for a three-year research scholarship. Without this, my study is simply impossible. I should also be very thankful to the school of Applied Sciences, University of Northampton for providing me with the academically excellent and socially enjoyable environment during my years in the University. My appreciations go to all staff members in the Graduate School for their help and kindness: **Charlotte Spokes, Dina Tailor, Sasha Finlay and David Watson.** They have been very hospitable and helpful.

Finally, my deep love and devotion are to my husband (**Dr. Guntur Sugiyarto**), who always helped and supported me during my course of study, and to my three children (**Gessa Werdhi Ayuningtyas, Garda Satyagraha,** and **Geulis Anggita Utami**) for their continues love, affection, support, and encouragement that enables me to go through all of the hard work and difficult times, especially during my illness from a virus infection that took away my balance and stopped me doing my research.

ABSTRACT

Unspecified counting practices used in a data collection may create rounding to certain 'based' number that can have serious consequences on data quality. Statistical methods for analysing missing data are commonly used to deal with the issue but it could actually aggravate the problem. Rounded data are not missing data, instead some observations were just systematically lumped to certain based numbers reflecting the rounding process or counting behaviour. A new method to analyse rounded data would therefore be academically valuable.

The neural network model developed in this study fills the gap and serves the purpose by complementing and enhancing the conventional statistical methods. The model detects, analyses, and quantifies the existence of periodic structures in a data set because of rounding.

The robustness of the model is examined using simulated data sets containing specific rounding numbers of different levels. The model is also subjected to theoretical and numerical tests to confirm its validity before being used on real applications. Overall, the model performs very well making it suitable for many applications.

The assessment results show the importance of using the right best fit in rounding detection. The detection power and cut-off point estimation also depend on data distribution and rounding based numbers. Detecting rounding of prime numbers is easier than non-prime numbers due to the unique characteristics of the former. The bigger the number, the easier is the detection. This is in a complete contrast with non-prime numbers, where the bigger the number, the more will be the "factor" numbers distracting rounding detection.

Using uniform best fit on uniform data produces the best result and lowest cut-off point. The consequence of using a wrong best fit on uniform data is however also the worst. The model performs best on data containing 10-40% rounding levels as less or more rounding levels produce unclear rounding pattern or distort the rounding detection, respectively. The modulo-test method also suffers the same problem.

Real data applications on religious census data confirms the modulo-test finding that the data contains rounding base 5, while applications on cigarettes smoked and alcohol consumed data show good detection results. The cigarettes data seem to contain rounding base 5, while alcohol consumption data indicate no rounding patterns that may be attributed to the ways the two data were collected.

The modelling applications can be extended to other areas in which rounding is common and can have significant consequences. The modeling development can be refined to include data-smoothing process and to make it user friendly as an online modelling tool. This will maximize the model's potential use.

ANALYSING ROUNDING DATA
USING RADIAL BASIS FUNCTION NEURAL NETWORKS MODEL

Table of Contents

CHAPTER III: NEURAL NETWORKS MODELLING

CHAPTER IV: DETECTING AND QUANTIFYING ROUNDING USING RADIAL BASIS FUNCTION NETWORKS

CHAPTER V: ASSESSING THE BEHAVIOUR OF THE NEURAL NETWORK MODEL IN DETECTING THE ROUNDING BASE

CHAPTER VI
ASSESSING THE ROBUSTNESS OF THE NEURAL NETWORK MODEL

CHAPTER VII
DETERMINING THE CUT-OFF POINT OF ROUNDING BASE DETECTION

CHAPTER VIII
APPLICATIONS OF THE MODEL ON REAL DATA SETS

CHAPTER IX

MAIN FINDINGS AND FURTHER RESEARCH

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1. Research Background

Data quality is an important factor in many application areas of data analysis. The application areas, in turn, also create an environment with stringent needs for reliable data. Most professions and decision makers rely on data for analysing information, that is, in the context of managing and decision making. In most cases, the results crucially depend on the quality of data used. Therefore, all data must have a level of quality appropriate for decisions for which they will be a part (Ballou and Tayi, 1999).

Implications of data quality have been the subject of various studies in different areas. A common case is data quality being sacrificed for a certain purpose such as convincing a wrong argument. Huff (1954) provides examples on how data quality can be manipulated for certain gains, as well as on statistical deceptions in general. Budd and Guinnane (1991) examines the difficulty encountered by pension authorities in Ireland in dealing with some discrepancies between the distribution of reported ages in the 1901 and 1911 censuses of Ireland, which show that the Irish exaggerated their age considerably.

Systematic inaccuracies in social data have recently attracted attention. Contemporary micro data in the United States (US) have shown some documented errors in reported job tenures. The errors have been shown to create biased estimates of the return to changing jobs (Brown and Light, 1989). Such errors can be more than just statistical nuisance as these can cause serious problems and policy implication.

Klesges *et al.* (1995) show a clear evidence for a digit preference in self-reporting of smoking in the 2nd National Health and Nutrition Examination Survey (NHANES) in the US. They compare the data of smoking from self-reporting and from an objective measure of smoking exposure (i.e., measured by carboxyhaemoglobin or COHb level in the blood). The distributions of cigarettes consumed per day from self-reporting and from the objective measure of smoking exposure were found significantly asymmetrical. Heavier smokers and those less educated were more likely to report a digit preference than lighter smokers and those more educated. The results suggest that self-reporting data may be biased toward a round number, particularly 20 cigarettes per day. There is also a strong digit preference (i.e., multiples of 10) in the number of cigarettes smoked. Almost 71% of all smokers report smoking in multiples of 10 cigarettes per day, with 20 cigarettes per day (typically one pack of cigarettes) being the most common response. In contrast to self-reporting data, COHb level data indicate a continuous distribution of exposure to smoking. There are no spikes for the number of smoking exposures such as multiples of 10. This suggests a pattern of digit preference in self-reporting data and the pattern depends on whether the respondents are heavy or light smokers, as well as their education.

Demographers have long noted the implications of age misreporting in the accuracy of demographic measures such as mortality and fertility indicators. Age misreporting, whether systematic or not, can make the census result undercount and produce biased demographic measures. Coale and Li (1991) analyse the effect of age misreporting on the calculation of mortality rates among the elderly in China. Both the Han Chinese (the majority ethnic group of China) and the other minority groups have the same perception about the importance of using an accurate calendar to record

the date of birth. The accurate reporting of date of birth is a characteristic of the Hans but, unfortunately, this is still not the case for some minority groups. Of the one billion persons listed in 1982, about 68 million were minorities. Some of them, such as the Mongols and the Koreans, share with the Hans the accurate knowledge about date of birth based on the Chinese calendar.[1] Other minority groups, however, do not have that same level of knowledge. In Xinjiang province, large minority groups have no precise knowledge about their respective dates of birth. As a result, mortality rate by age calculated from recorded deaths and enumerated population at higher ages (elderly) are wrong because of misstated ages. The average value of the deviation index at "decal ages"—from 40 to 90 for males—in this province is 1.647, compared to 1.018 for all of China. When the male population of Xinjiang is subtracted from the all-China population, the average deviation index for these ages is 0.999. They conclude that the age heaping in China is divisible by 10, which is a consequence of the very strong age rounding in Xinjiang province.

Misreporting and self-reporting to the nearest convenient number and prediction of digit preference are "estimated" data. Empirical results show that estimated data reduce data quality. The list of cases and their implications can be extended to include other areas of applications.

Most methods for analysing such kind of data have been based on statistical methods developed originally for analysing missing data (Heitjan and Rubin, 1990). Rounded or coarsened data, however, are not missing. They solely have shifts in count size in certain observations because of the rounding inherent in the estimation

[1] The Chinese calendar consists of a cycle of "animal years" that repeat every 60 years. There are 12 animals, and each animal has five different qualities. Each year is composed of either 12 or 13 lunar months. Since a lunar month has 29.5 days, there are about 12.4 such months in a year and a 12-month year would become increasingly not synchronised with the solar year of 365.25 days.

process. Thus, applications of missing data techniques, which seek to replace missing data according to probability distributions of sizes, are liable to distort rounded data by effectively duplicating the shifted observations at probabilistically determined "real" sizes.

Crockett and Crockett (2000) developed a modulo-test method to analyse estimated data. While the model is sufficient for its original purpose, it has constraints (e.g., its limitation in dealing with large data sets and its complexity in the modeling application that involves applying a series of statistical tests to identify the significant rounding base unit). Therefore, developing a new technique that can overcome these drawbacks would be very useful.

1.2. Main Purpose and Objective of the Study

The main purpose of this study is to develop a new methodology for analysing rounding data by making use of artificial intelligence (AI) technique. The new method will complement and enhance conventional statistical analysis based on missing data. More specifically, the main objectives of the modelling development are to detect and quantify the existence of rounding in a data set.

Therefore, this research aims to develop a new technique that can be used for:

- detecting the presence of periodic structures in data sets;
- analysing periodic structures present in data sets; and
- quantifying the effects of periodic structures present in data sets.

This research focuses on developing an innovative AI technique for detecting rounding in a data set and for quantifying the effects of such rounding. The new method provides an alternative technique to conventional statistical approaches based on missing data.

The AI technique used in this research is the radial basis function (RBF) neural network. This a further development of an earlier study by Turner *et al.* (2001), which has shown that neural networks can be used to detect periodic structures in a data set arising from rounding or estimation.

1.3. Modelling Development and Analysis

The modelling development is conducted by:

- First, considering the data sets to have "patterns" and employing pattern-recognition methods to detect periodic structures in a data set.
- Second, developing AI-based pattern-recognition techniques to recognise the characteristics of the patterns of periodic structures in a data set.
- Third, developing a model to quantify the effect/s of periodic structures.
- Fourth, comparing the results with the benchmark of previous research using modulo test.
- Fifth, analysing the results based on application in the model of the data concerned.

The neural network model developed in this study is then examined by using different kinds of simulated data sets. In addition, calculations of the cut-off point at which rounding to a certain base number in a data set can be detected are also conducted. The assessments are conducted using different rounding base numbers of different rounding levels in the theoretical data sets to also provide a sensitivity analysis. All are intended to ensure that the model is robust before implementation on real data sets.

1.4. Organisation of Writing

Chapter I establishes the thesis background, the objective of the study, and the methodology used in the modeling development and analysis. The background section puts this study in its relevant context, highlighting the rounding issue in a data set and modeling application, followed by the main purposes and objective of this study. The last section describes the methodological approach used in the analysis.

Chapter II reviews the main features of rounding data, conventional statistical analysis to handle such kind of data, and previous work on rounding data analysis. An overview about neural networks follows in the last section.

Chapter III highlights the more detailed aspects of neural networks by introducing pattern recognition, function approximation, and classification. To help explain the modeling design used in this study, issues of neural networks learning and architecture are discussed. This forms the basis for choosing radial basis function network as a modeling approach in this study.

Chapter IV describes neural network model simulations developed in this study, covering aspects such as underlying probability distribution of the data sets being investigated, setting training data of the model, developing radial basis Function network model, and testing the model.

Chapter V describes the analytical and numerical assessments of the model to further clarify the model behaviour in detecting rounding. It also includes the development of measurement indicators in the form of detection power and detection error. The assessments are based on small random and theoretical data sets.

Chapter VI assesses the robustness of the neural network model by using theoretical data sets containing rounding to bases 5, 7, 10, and 11 with each data set

containing different levels of rounding, i.e., from 10% to 50% with an interval of 10%. The data sets are uniformly and normally distributed, while the best-fit distributions used in the rounding detection are normal, lognormal, and uniform best fits.

Chapter VII discusses the calculation of the cut-off points in detecting rounding problems to certain base numbers by using the two approaches of direct investigation and regression model. Both methods are applied on two uniformly and normally distributed data sets with four different rounding bases of bases 5, 7, 10, and 11. The rounding detections use three best-fit distributions.

Chapter VIII examines the neural network model applications on real data sets. First, it is applied on data of cigarettes smoked by secondary schoolchildren (ages 11 to 15 years) in England, United Kingdom (UK) in 2001 as a result of the first survey on smoking, drinking, and drug use. The second application is on data of the amount of alcohol consumed by secondary schoolchildren in England in 2001 from the same survey.

Chapter IX summarizes the findings and conclusions as well as suggestions for further research.

CHAPTER II

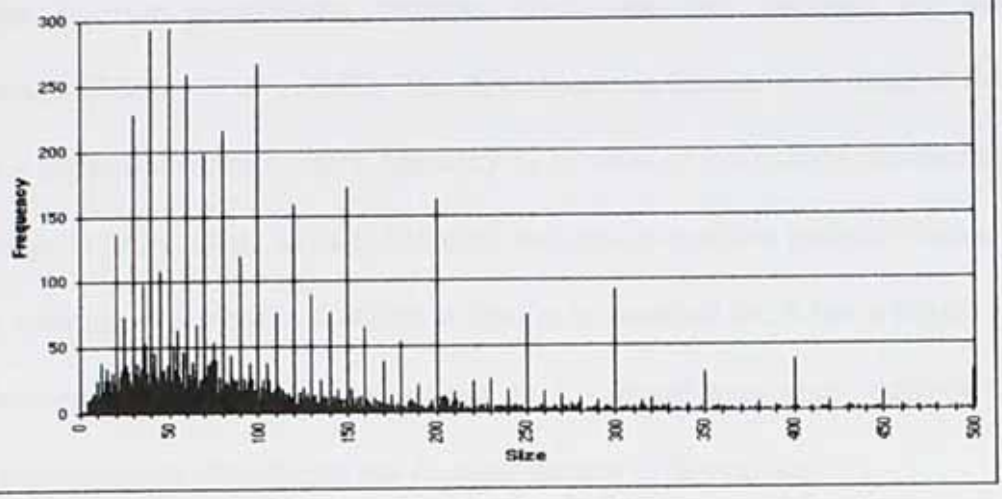
OVERVIEWS ON ROUNDED DATA

2.1. The Main Features of Rounded Data

Any data-collection activities (such as census, survey, and other enumeration exercise) may include a variety of unspecified counting practices. This includes a data-collection system using no clear concepts and definitions. As a result, data sets may contain some degree of estimation, such as rounding to the nearest convenient number or rounding up or down depending on the system's program. The rounding behaviour on the latter can be detected by examining the system's program in detail, but it is very difficult for the former since very limited information about their actual collection methods exist. Crockett and Crockett (1998) argue that, in general, the greater the age of the data source (i.e., longer recall time), the greater the probability that the recorded data represents "rounded" estimates as opposed to "exact" counts. Also, the greater the number and diversity of the persons involved in the data collection, the greater the probability of variation in the counting practices.

The type of data set investigated in this research is coarsened by rounding that is apparent from the frequency distribution of the data set as shown in Figure 2.1. Data used in this figure is from the Religious Census Data of 1851 for England and Wales. The frequency distribution shows that some rounding has been done as the graph does not indicate an expected reasonably smooth distribution. It also shows obvious excesses of observations at distinct, well-defined sizes that are periodic, with intervals—in this instance—of 5, 10, 20, 50, and 100

Figure 2.1 Frequency Distribution of Church Attendance in 1851 in England and Wales[2]



This particular type of periodic structures arises from estimation such that some enumerators did not count exactly but instead estimated the numbers of people attending the church congregation. In their estimation, the real underlying value is returned as a convenient multiple of a base unit, the closeness of the returned "round" number that depends on a variety of factors, including the observation size and the ability and diligence of individual enumerators. The base units are generally convenient multiples of the number system being used, such as 5 (counting "by fives") and 10 (counting "by tens") in decimal (base 10) number systems or 6 (half dozens) and 12 (dozens) in duodecimal (base 12) systems.

[2] Crockett and Crockett (1998)

2.2. Conventional Statistical Analysis

In a conventional statistical analysis, such kinds of data set (i.e., a single series of data) can be examined by using time-series analysis. The fluctuations can be attributed to four components, namely, trend, seasonal, cyclical, and random disturbances (Levine *et al.*, 2002). The disturbance is known as a trend if the data follows a general direction with a tendency of upward or downward movement such as a change in price, value, or rate. Seasonal movement contains periodic fluctuations, such as monthly or quarterly. Cyclical is similar to seasonal but it has a longer period of recurrence with a tendency of upward or downward movement through a long series, while random disturbance has no clear pattern of fluctuation.

The basic assumption in the time-series method is that any fluctuations influencing data pattern in the past and present will be similar in the future. The time-series method is intended to identify and isolate the fluctuation factors to get the actual values that can be used for forecasting and monitoring purposes. Time-series analysis will refine the original data set by using different kinds of smoothing methods based on the fluctuations found in the data set. De-trend, auto-regressive (AR), moving average (MA), auto-regressive moving average (ARMA), auto-regressive integrated moving average (ARIMA) are used in smoothing time-series data that will be discussed further below.

2. 2. 1. De-trend and Stationary Data Series

The purpose of de-trend is to get a stationary series by removing the long-term trend contained in the original data set. De-trend is to make the mean, variance, and auto-correlation structure of the data constant over time. The resulting data series would be a flat-looking series without a trend, with constant variance and auto-correlation structure over time, and no periodic fluctuations or seasonality. The

stationary process can be conducted by taking the difference of the original data that makes the number of observations in the difference data set will be less than the original data. For some cases, the difference needs to be calculated more than once to get a stationary data series. This will further reduce the number of observations.

For de-trending a data series containing a trend, the corresponding trend curve can be fitted to the data series and then the residuals from the fit can be treated as the real observations. For a data series with a nonconstant variance, the logarithm or square root of the original data series can be used to stabilise the variance (Makridakis *et. al.*, 1998). For a negative data series, adding a suitable constant to make all data positive can be used before applying any transformation. This constant is then subtracted from the new data series to obtain the predicted values. All the techniques described in this section are intended to generate a new series with constant location and scale.

2. 2. 2. Auto-regressive Model

A common approach for modelling univariate time series is to use the AR model, which can be defined in the following equation:

$$X_t = \delta + \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \ldots + \varphi_p X_{t-p} + A_t$$

where : X_t is the time series at time t; A_t is the white noise with 0 mean and a constant variance; δ is a fixed parameter to be estimated from the least-squares regression analysis; and $\phi_1, \phi_2, \ldots, \phi_p$ are auto-regressive parameters.

As can be seen from the equation, an auto-regressive model is basically a linear regression of the current value of the series against one or more prior values of the series. The value of p in the equation is called the order of the AR model, which refers to the size of the correlation between values in a time series that are p period apart. Application of AR models can use a number of methods—such as the first-order auto-regressive model that is similar to a simple linear regression model and the second-order to the p-order auto-regressive, which is similar in form to the multiple linear regression models.

2. 2. 3. Moving Average Model

Another common approach for modelling univariate time-series data is to use the MA model, which can be formed as follows:

$$X_t = \mu + A_t - \theta_1 A_{t-1} - \theta_2 A_{t-2} - \ldots - \theta_q A_{t-q}$$

where X_t is the time series at time t; μ is the mean of the series; A_{t-q} are white noise; $\theta_1, \ldots, \theta_q$ are the parameters of the model, and q is the order of the MA model.

The smoothing method of a time series using moving average depends on the length of the selected period (L) in calculating the average. In smoothing an annual time-series data, L should be an odd number of years (Levine *et al.*, 2002). There will be no MA in the first (L-1)/2 years and in the last (L-1)/2 years of the series. The higher the length of the period selected for constructing MA, the less observation results can be obtained. Therefore, there will be less number of observations than the original data. For example, if the length is five, there will be no MA obtained in the first and the last two in the time series; if the length is seven, there will be no MA obtained in the first and the last three in the time series; and so on.

2. 2. 4. Auto-regressive Moving Average Model

Box and Jenkins (1970) introduced an approach that combines the moving average and ARMA approaches. They refer to the method to respectively identify and estimate models by incorporating the two approaches the results in a powerful class of models. The model is a combination of AR and MA models, which can be formulated as follows:

$$X_t = \delta + \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \ldots + \varphi_p X_{t-p} + A_t - \theta_1 A_{t-1} - \theta_2 A_{t-2} - \ldots - \theta_q A_{t-q}$$

where the terms used in the equation have the same meaning and interpretation as described in the AR and MA models.

The model assumes that the time series must be stationary. A nonstationary series is differentiated once or more than once until the stationary data series is obtained. The new data series will have characteristics of the ARIMA model. Box-Jenkins models can be extended to include seasonal auto-regressive and seasonal moving average. The most general Box-Jenkins model includes the use of difference operators, auto-regressive model, moving average model, seasonal difference operators, seasonal auto-regressive model, and seasonal moving average model.

All methods of time-series analysis described above are meant to smooth the original time-series data for forecasting purposes. The data sets examined in this research are, however, characterised by systematic rounding. Therefore, the fluctuations observed in the data sets are not caused by trend, seasonal, cyclical, or random disturbances but by other systematic aspects, including the counting behaviour of enumerators following certain base numbers such as half dozens, dozens, and the use of decimal and duodecimal (base 12) systems, and machine

system default. Therefore, the time-series analysis methods cannot, in principle, be used in analysing coarsened or rounded data.

In searching for a new method to deal with such kind of data, Poli and Jones (1994) introduced a neural network approach to analyse nonlinear time-series data. They showed that nonlinearity in time series sometimes exhibits distinctive features that can well be described by existing parameterised classes of nonlinear models in the neural networks context. Moreover, James (1994) argued that neural networks can learn the patterns in complex time-series sequences and produce useful internalisations of the patterns.

2.3. Analysing Missing Data

Some research has been conducted on coarsened data, which is similar to rounded data. Coarsened data examined in the previous studies are, however, mostly randomly coarsened data with no systematic patterns of coarsening. Therefore, the problem in this kind of coarsened data can be regarded as a missing data problem. Little and Rubin (1987) introduced this concept by delineating a class of missing data mechanisms, which allow a statistical analysis to be done "as if the missing pattern had been fixed in advance, independently of the underlying complete data."

The missing data used in the previous studies are data sets with some unobserved values and therefore missing. For example, in a household income and expenditure survey, some respondents may refuse to report income or certain expenditures for specific reasons. In an industrial experiment, some results might be missing because of mechanical breakdowns, which are unrelated to the experimental

process. From the examples, it is quite clear that some data might be just missing for various reasons.

On the other hand, when there is no clear information about missing data and the data series shows a systemic pattern for some specific observations, it should not be regarded simply as a missing data problem. The missing observations for certain numbers might have been estimated already or counted to the nearest numbers. Accordingly, fixing the rounded numbers using the tools for missing data problem will cause a duplication in the new data set, as the incomplete observations in certain numbers are actually not missing but they have been rounded to the nearest convenient numbers. Therefore, the application of missing data problem in this case will distort the information of the real data set. The neural networks model developed in this study tries to deal with this kind of data, i.e., data coarsened by rounding and not by missing data.

2.4. Modulo Test Method

Crockett and Crockett (1998) introduced a modulo test method for analysing rounding numbers, which are reflected in spikes of the frequency data count. The data used in their study are the number of people who attended churches in 1851 in England and Wales as part of a religious census. The main purpose of the research was to determine whether the data recorded comprised solely exact counts or contained a proportion of estimates, which can be a result of rounding to the nearest convenient numbers. The method can also help in determining the proportion of counts when there were estimated numbers. The frequencies of various round numbers can be used to illuminate the methods of rounding used in compiling the data, i.e., whether the counting was predominantly carried out using decimal base,

duodecimal base, and so on. It follows that the technique can also be used to estimate over or underrepresentations of the data concerned compared to their "real" count values.

The study defined the count data into five categories, namely, exact count, rounding to the nearest base units, accurate approximate, honest estimates with varying degrees of accuracy, and dishonest estimates with false over-estimates (or under). The last two categories are generally a result of numbers with multiples of base units used in the estimation.

A crude version to estimate the probability distribution of exact count is simply to assume a constant probability distribution, while the more accurate one is to use a smooth curve with a variety of techniques available for smoothing. The better the smoothing process, the better the resulting data will conform to the real probability distribution.

2.4.1. Identifying Base Units Contained in the Rounded Data

At this stage, the goal is to separate the subset of multiples of a base unit which represents exact counts and the subset of multiples of a base unit which represent estimates. Value of 60, for example, could be:

i) Exact counts, which is an exact 60, or

ii) An estimated rounding to the nearest multiple of each of 5, 10, or 12 such as 60s estimated/rounded to 5 will be in the base-5 subsets, 60s estimated/rounded to 10 will be in the base-10 subsets, 60s estimated/rounded to 12 will be in the base-12 subsets.

2.4.1.1 Base-unit Test

First, **modulo test** is used to quantify the number of counts that are a multiple of a suspected base unit. This method involves dividing all counts by the suspected base unit and making use of integer division (modulo). If the result yields zero remainders, the counts are called exact multiples of the suspected base unit and, therefore, said to pass the modulo test. The number of counts that pass the modulo test can be compared to modulo test number which would be expected to pass the test according to the estimated "real" probability distribution. A binomial test is then used to establish the statistical difference between the modulo test passes and the expected passes, and to classify whether the base unit is a "potential" base unit or not.

The second stage is intended to test whether all potential base units are significant base units by examining the possible interactions among the potential base units. The simple way is to compare the excess frequencies at that number with the excesses at any lower-potential base units, which are factors of this number. If the excess at a higher number is greater than expected after comparing it with those at a lower base numbers, then this higher number can be considered a significant base unit. Otherwise, it can be discarded.

2 .4.1.2. Subset Model

To estimate the number of cases rounded to each base unit, a simultaneous equation method is used. The equation expresses the number passing modulo test in terms of the subset sizes, and the probabilities that members of the subsets are divisible by the potential base units.

For the study data shown in Figure 1.1, the subsets in base 1 is 52%; base 5 is 12%; base 10 is 20%; base 20 is 6%; base 50 is 7%; and base 100 is 3%. The model

results were then analysed statistically in relation to the expectation and for factor and multiple effects. This is conducted to determine the actual estimation base units, i.e., the base numbers whose multiples appear statistically significant more frequently in the data than would be expected.

The accuracy of the model is, however, dependent upon the accuracy in determining the probability distributions of the various estimation behaviours. The detail results are therefore data-set dependent. In general, the subset probability distributions for small base units—such as base 1 (exact counts), base 5 and base 10—follow closely their underlying frequency distributions. On the other hand, the probability distributions of larger base units show more variations. This is due to a variation in the counting or estimation behaviour for large observations. This study resulted in a new model of analysing rounding and similar estimation present in a data set.

While the model was sufficient for its original purpose, some constraints in its application exist (such as its limitation in dealing with large data set and its complexity in the modeling application) since it involves conducting a series of statistical tests independently to identify the significant base unit. These limitations call for the development of a new technique that can overcome these constraints. This study is intended to overcome these drawbacks by applying the neural networks technique.

CHAPTER III

NEURAL NETWORKS MODELLING

3.1. Introduction to Neural Networks

Neural networks are statistical models of real world systems built by tuning sets of parameters (Swingler, 1996). Neural networks excel at recognition and classification types of problems and have the capability of modelling complex nonlinear processes to arbitrary degrees of accuracy (Picton, 2000). The main feature of neural networks is their ability to classify patterns based on information learned from examples without introducing any programme for them to do it (Browne and Picton, 1999). Trained neural networks are able to decide on outputs for previously unseen inputs (Argos, 1999).

Training patterns or "examples" in neural networks are represented as vectors that can take different forms (such as images, speech signals, sensor data, robotic arm movements, financial data, and diagnosis information). Neural networks' "recogniser" may then be used as a modelling framework behind the process, whereby input vectors are mapped into output vectors. This makes neural networks effective for various applications.

Neural networks excel at solving problems involving patterns, such as pattern mapping, pattern completion, and pattern classification. They can translate images into keywords, financial data into financial predictions, and map visual images into robotic commands (Dayhoff, 1990). For example, neural networks trained to recall a complete pattern

can complete a noisy pattern with missing segments, as well as to produce a complete vehicle outline from an inputted outline of a partially obscured vehicle.

Neural networks learn by example. This represents a radically different approach to computing that involves developing computer programs. In a computer program, every step of computer execution is specified in advance by a programmer—a process that takes times and human resources. In contrast, neural networks begin with learning samples of inputs and outputs to provide the correct output for each input supplied without requiring human help in identifying features or developing algorithms and programs specific to the problem concerned. This suggests that neural networks applications can save valuable time and human resources.

The key element of neural networks process is the information processing system structure, which is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve a specific problem.

3.2. Neural Networks for Data Analysis

In the data analysis area, characteristics of neural networks have the potential to make powerful modelling tools to detect the presence of rounded data in a data set (Turner *et al.*, 2002). In this context, neural networks can be trained to recognise periodic "patterns" in the frequency distribution of a data set caused by rounding or estimation, as well as to approximate any functions within an arbitrary degree of accuracy since

neural networks are also good universal function approximators (Polhill and Weir, 2001).

Neural network methods have been used to generalise many classical statistical methods involving nonlinear data by modelling the interactions among them to perform nonlinear operations on inputs to produce the desired outputs.

Neural networks take a different approach to problem solving that makes them have no restriction in their problem-solving capability. This is in contrast with a computer program that can only solve problems that the programmers already understand and know how to solve.

With their remarkable ability to derive meaning from complicated or imprecise data, neural networks can extract patterns and detect trends that are too complex to be noticed by either humans or computer. They have the ability to learn to do the tasks based on information given during the training. Neural networks can create their own organisation or representation of information received when learning.

Neural networks are general-purpose programs that have numerous potential applications, including almost any problems of pattern recognition, function approximation, and classification.

3.2.1. Pattern Recognition

Pattern recognition is an important application of neural networks. In this application, the networks are trained to associate outputs with input patterns, i.e., identifying input pattern and producing its associated output. Neural networks can also identify an input pattern that has no output

associated with it by producing output that corresponds to a taught input pattern (i.e., it has the least difference from the given pattern). Pattern recognition aspects of neural networks have enhanced many important topics of data analysis.

Statistical pattern recognition offers much more direct and significant routes than other approaches. For example, the sum-and-threshold model of a neuron arises naturally as the optimal discriminant function, which can distinguish two classes of normal distributions with equal covariance matrices. Similarly, the logistic sigmoid is the function to allow network output to be interpreted as a probability, when the distribution of hidden unit activations is controlled by a member of an exponential family.

Pattern recognition is a research area intended to examine the operation and design of systems that recognise patterns in data. It includes subdisciplines of statistical and syntactical pattern recognition. The former covers discriminant analysis, feature extraction, error estimation, and cluster analysis, while the latter consists of grammatical inference and parsing. Important application areas include image analysis, character recognition, speech analysis, and diagnostics analysis.

From the perspective of pattern recognition, neural networks can be regarded as an extension of conventional techniques by using feed-forward network architectures such as the multilayer perceptron and the radial-basis function network. The latter approach is employed in this study to get the best result.

3.2.2. Function Approximation

Neural networks have been shown to have good function approximation capabilities which can ease modelling development effort. In many application areas, describing data often contains an analytical function. This includes function approximation of curve fitting and regression analysis to find a smooth-curve best fit to the data distribution without necessarily passing through all data points. The best fit is essentially minimising the sum-squared error given the data points and curve.

3.2.3. Classification

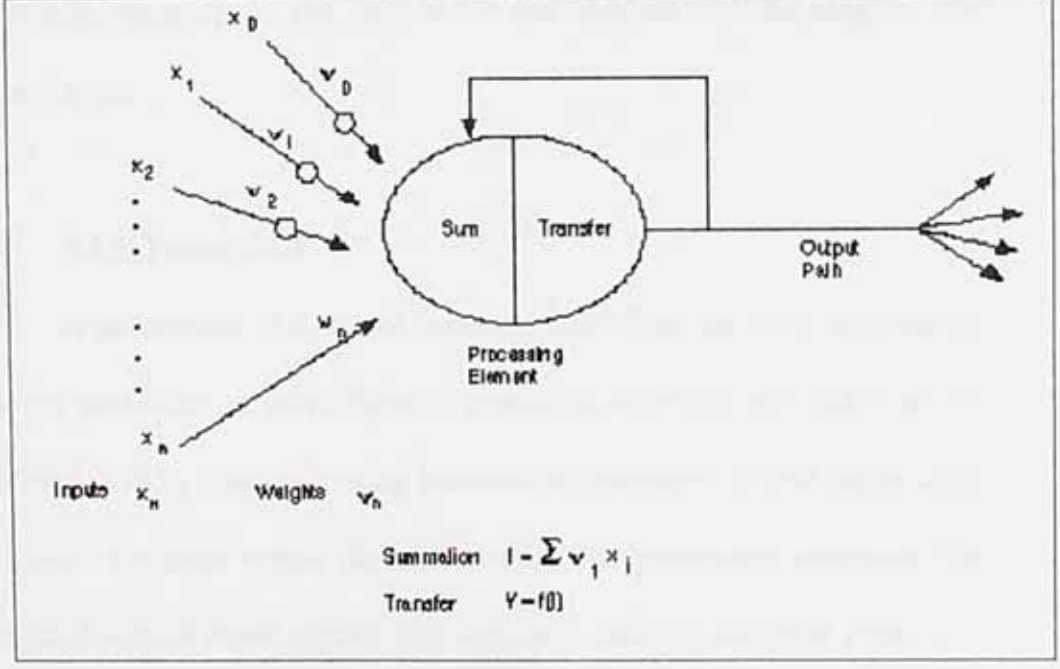
The classification concept involves learning about similarities and differences in patterns, which are abstractions of instances in a population of non-identical objects. This is different from identification, which is essentially recognizing an individual object as a unique singleton class. Classification is a process of grouping objects into classes according to their perceived similarities by learning from a set of population classes from a sample of patterns. Classification is a process conducted after pattern recognition is completed since it is impossible to classify without recognising the pattern. Classification method is applied in this research to classify data patterns associated with certain rounding base number present in a data set.

3.3. Artificial Neural Networks

Artificial neural network is a system loosely modelled after a human brain. This can be seen from the terms used, such as connectionism, parallel distributed processing, neuro-computing, natural intelligent systems, machine learning algorithms, and artificial neural networks. Artificial neural networks try to simulate the multiple layers of simple processing elements called neurons. Each neuron is linked to its neighbours with varying connectivity coefficients to represent the connection strengths. Learning is carried out by adjusting these strengths to ensure that the networks produce the intended results.

As mentioned, neural networks excel in problems of recognition and classification. The basic units of neural networks are artificial neurons, which simulate four basic functions of natural neurons. Figure 3.1 shows the schematic representation of an artificial neuron.

Figure 3.1. A Basic Artificial Neuron



As can be seen from the figure, various inputs to a network are represented by a mathematical symbol x_n. Each of these inputs is multiplied by a connection weight, which are represented by w_n. In the simplest case, the multiplication is simply a summation, fed through a transfer function to generate output.

3.4. Neural Network Learning Algorithm

There are many types of neural networks. Each has a highly processing capability arising from interconnected networks of simple computational elements. The networks differ in their architectures and learning or training methods. The architecture depends on the number and types of layers, while there are many types of learning that includes perceptron, adaline or madaline, hebbian learning, competitive learning, and back propagation. Back propagation is employed in this study given the characteristics of the data used and the main purpose of the study. Neural networks can use a combination of learning methods such as perceptron learning in the first layer and competitive learning in the second layer.

3.4.1. Perceptron

A perceptron is a neural network that offers the first algorithmic training procedure to solve linearly separable problems only (McCulloch and Pitts, 1943). The processing element of perceptron is still being used as a basis for most neural networks today. The processing computes the weighted sum of input signals and compares that net-weighted input to a

threshold value T. If the net input is greater than or equal to the threshold, the processing output is +1 and if not, the output is -1. Therefore, the perceptron uses a transfer function as follows:

$$I = \sum_{i=1}^{n} w_i x_i$$

where, y = +1, if $I \geq$ T and, y = -1, if $I <$ T.

In this case, I is the net weighted input to the processing, w_i and x_i are the weight vector and input vector respectively. The threshold value T is the minimum activity required for processing to generate a positive output.

McCulloch and Pitts (1943) defined a simple model of a neuron using Frank Rosenblatt's training procedure that has become the first trainable neural network and a standard today. Rosenblatt's perceptron training algorithm was introduced in 1958, providing the first procedure to allow a network to learn a task. The perceptron is used to separate pattern into two categories. It adapts the weight change proportional to the difference between the desired output and the actual output as follows:

$$w_{new} = w_{old} + \beta yx$$

where β = +1, if the perceptron answer is correct, and β = -1, if the perceptron answer is wrong. The perceptron is indistinguishable from the processing used, and since input patterns may change from time to time, it is useful to have a system that can adapt to the changing problem.

3.4.2. **Adaline** or **Madaline**

Adaline or madaline measures the network's overall performance. The system uses a minimum learning error for an adaptive linear element

(Widrow and Hoff, 1960). It uses a learning system that minimises an entire system's error. Since the system cannot always categorise each pattern correctly, some errors are associated with the system's overall performance. The idea is to find a system with the smallest error. A good measure of such error is the mean square error that shows how far, on the average, the system's response is from the correct answer.

Adaline has a simple bipolar output. It generates an output +1when its net weighted input is greater than 0, and -1 when its net weighted input is less than or equal to 0. Adaline computes its total input stimulus by taking a simple weighted sum as follows:

$$I = \sum_{i=1}^{n} w_i x_i$$

where, y = +1, if I > 0 and, y = -1, if I ≤ 0.

The outputs can be classified as +1 output, meaning that the input pattern belongs to the first category, and -1 output for the input pattern corresponding to the second category or class. These output categories are compared to the desired outputs to compute the adaline error as follows:

Error = (Desired Output) – (Actual Output)

Given the output constraints, the error can only have values of +2, -2, or 0. Once the error is computed, it can be used to adjust the weights in the input connections to the adaline. This is done by using a learning rule called *delta rule* that computes the changes in weights as follows:

$$w_{new} = w_{old} + \beta E x(|x|)^{-2}$$

where β is a learning constant between 0 and 1; E is the error computed above; x is the input vector and w is the weight vector.

The supervised training procedure here is more complex than perceptron training because of the way in which weight changes are performed. When the first pattern is applied to adaline, it either gives the correct or wrong response. Suppose it is a wrong response, the *delta rule* must then be applied repeatedly to the adaline's connections weights until the correct answer is given. The next pattern is then tried and if it gives the correct answer, no weights are changed and the third pattern is used. If the second pattern still gives the wrong answer, however, the *delta rule* is again applied until the correct answer is given.

In the process for adjusting pattern two, it is possible that pattern one may no longer be recognised. Thus, before continuing to the third pattern, the weights for pattern one need to be adjusted again. This constant process of making weight adjustments and then rechecking to confirm that the previous patterns are retained makes the process complex.

3.4.3. Hebbian Learning

Hebb (1949) described the changes that occurred in a cellular level of learning process known as the Hebb's Law. It says that when a neuron stimulates another neuron when the receiving cell is actively firing, the connection from the first cell to the second is strengthened. For example, if neuron A is stimulated repeatedly by neuron B when neuron A is active, then neuron A will become more sensitive to stimuli from neuron B.

The first attempt to model Hebbian learning was in 1950, involving synaptic input strengths adjustments leading to the

incorporation of adjustable synaptic weights on input lines to excite incoming signals.

The Hebbian learning principle becomes influential in the biological model of learning. An input vector $x = (x_1, x_2, x_3, \ldots, x_N)$, is linearly combined with the weight vector $w = (w_1, w_2, w_3, \ldots, w_N)$ through inner product to form the sum:

$$s = \sum_{n=1}^{N} w_N x_N$$

If the sum s is greater than the given threshold T, then the output is one; otherwise, it is zero. This threshold function is unipolar since the outputs are non-negative values of zero or one.

3.4.4. Competitive Learning

Most neural networks are trained using supervised learning. This means that most networks have to be told to get the desired response or they must be given feedback on their performance. Competitive learning uses unsupervised learning called self-organising that modifies connection strengths based only on the characteristics of input patterns presented to them. The simplest self-organising system is the Kohonen feature map.

A Kohonen feature map may exist as a layer within a larger network or two-layer networks in which the input layer is fully connected to the Kohonen layer. The input layer acts as a collection of fan-outs, distributing input pattern to each neuron in the Kohonen layer, which acts as output layer. In addition, this layer has a large number of connections linking neurons within a layer and to each other. This connection is a critical part of the feature map's self-organising property.

The connection within a layer is to create a competition between neurons in the layer to determine which neuron has the strongest response to input pattern. Once the Kohonen layer has stabilised and the wining neuron is determined, the output from the layer is a simple binary +1 response and no output from any other neuron.

Determining the winning neuron is the key to training the networks. Unlike most other neural networks, in a Kohonen network only the winning neuron and its neighbours modify the weights in their connections. The remaining neurons experience no learning. The training procedure used by the networks is:

$$\Delta w_i = \beta(x_i - w_i^{old})$$

where β is the learning constant or gain, and x_i is the input signal along the i-th weighted connection. The β is the range $0.0 \leq \beta \leq 1.0$ but mostly less than 0.2.

3.4.5. Back Propagation

Back propagation is a training process to adjust weight values to ensure that network behaviour matches the desired behaviour. The network learns by observing an ever-changing stream of data produced by some dynamic process, and producing a set of outputs. The difference between network outputs and the actual desired outputs is the network "error," and the back-propagation training is to minimize the error.

Suppose there is a pair of training data (x, d) consisting of input vector $x = (x_1, x_2, x_3, \ldots, x_N)$ and a desired output vector $d = (d_1, d_2, d_3, \ldots, d_M)$. For a given set w of weight values, the network produces the

output vector $y(w) = (y_1(w), \ldots, y_M(w))$. The error $e(w)$ is expressed as follows:

$$e(w) = \tfrac{1}{2} \| d - y(w) \|^2 = \tfrac{1}{2} \sum_{k=1}^{M} (d_k - y_k(w))^2$$

The goal is to find w that makes this error 0, but this is rarely achieved so that the actual goal is to minimise the value of w within some tolerance.

Back-propagation training is also called steepest descent algorithm, which is an iterative approach that moves downward or moves some amount in the negative derivative direction from a starting point. Given its desirable characteristics, back-propagation approach is used in this study.

3.5. Neural Network Architectures

A neural network is a nonlinear interpolator and extrapolator. It needs only adjusting the weights appropriately to train a neural network to map exemplar feature vectors into the desired identifier output vectors, or to approximate function.

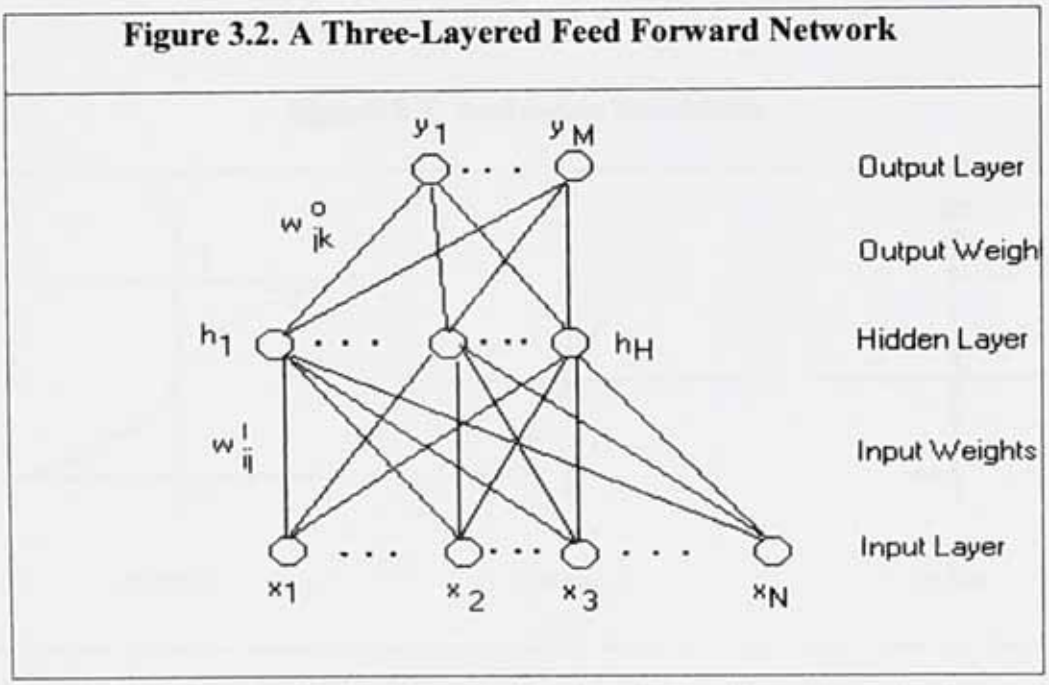
3.5.1. Feed Forward Network

The most commonly used neural networks are feed forward neural networks. The networks are layered networks representing a certain class of nonlinear regression or classification models relating a set of input variables (covariates, independent variable, and predictors) to one or more output variables (target variables, dependent variables, and response variables). The networks also include one or more layers of hidden units

that add flexibility to the model. Normally, outputs from a hidden unit are logistic (sigmoid) functions of linear combinations of inputs to that node, and the final outputs are linear combinations of outputs from the final layer of hidden units. This is known as multilayer perceptron and back-propagation network since the error back-propagation algorithm is used to train the network.

Feed forward neural networks include multiple layer perceptrons (MLPs) and radial basis function (RBF) which are powerful to perform nonlinear pattern discrimination. In this case, any network failures are not attributable to the neural network paradigm, but to inadequate training, inappropriate architecture for the problem at hand, or noise power that is non-separable to the data (Looney, 1997).

Figure 3.2. A Three-Layered Feed Forward Network



3.5.2. Multiple-Layered Perceptron

MLPs consist of input and output layers, with a number of hidden layers in between. Figure 3.2 shows a three-layered architecture including input, hidden, and output layers.

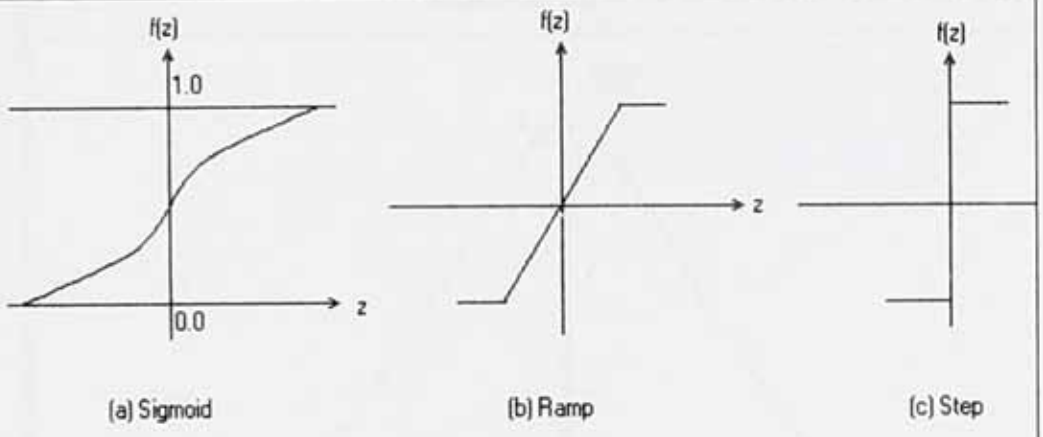
The input layer nodes correspond to input variables and the output layers are to output variables. Hidden neurons have no physical meanings and the neurons between adjacent layers are fully connected by branches. Transfer or activation functions describe each neuron in the hidden layer.

Some feed forward neural networks use a sigmoidal function as follows:

$$f(x) = 1/1+e^{-x}$$

$f(x)$ transfers an input x to neuron in the range of [0.0, 1.0] as shown in Figure 3.3 (a). Some other activation functions can also be used as shown in Figure 3.3 (b) and (c). However, for a specific feed forward neural networks structure, the neurons in the hidden and output layers are fixed.

Figure 3.3. Activation Functions



Each connection branch is described by weight, representing the connection strength between two linked nodes. The training process is the procedure to adjust the weights. A bias neuron supplying an invariant output is connected to each neuron in the hidden and output layers. The

bias provides a threshold to force an activation of neuron, and is essential to classify networks input patterns into various categories.

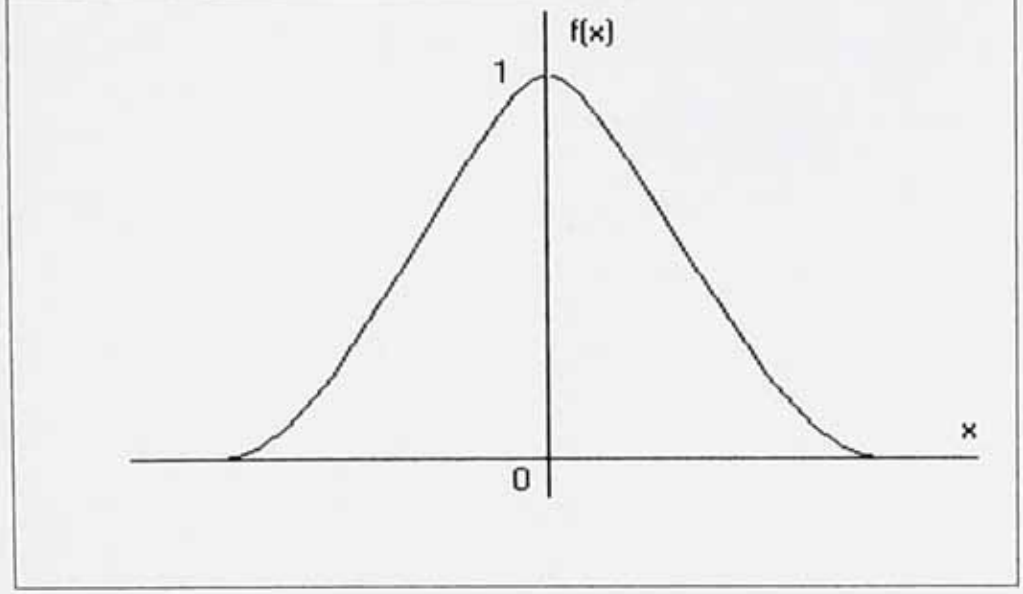
3.5.3. Radial Basis Function Network

The structure of radial basis function (RBF) network is the same as that of the MLPs. The main difference is the hidden layer transfer function used. In RBF, the hidden layer used a radial basis (Kernel) function as follows:

$$f(x) = exp(-x^2)$$

This function responds only when the input stimulus falls within a certain area. Figure 3.4 below shows a Gaussian Kernel function centred at x = 0.

Figure 3.4. A Gaussian Kernel Function Centred on x = 0



As the input approaches the function centre, the output approaches unity. This function is symmetric, and its centre is determined by input

weight vector to the node. Therefore, RBF can be used to approximate a function. The function adds neurons to the hidden layer of RBF network until it meets the specified mean squared error (MSE) goal.

RBF network creates a three-layered network with radial basis function in the hidden layer and product function of competitive learning algorithm in the output layer. An adaptive RBF network has been able to introduce confidence limits on output data (Leonard *et al.*, 1992). This information is unavailable with MLP. This is one of the main reasons RBF is used in this study.

CHAPTER IV

DETECTING AND QUANTIFYING ROUNDING USING RADIAL BASIS FUNCTION NETWORKS

4.1. Introduction

This chapter discusses radial basis function neural networks for detecting and quantifying rounding base numbers in a data set. The radial basis function (RBF) is a unique type of neural networks. The main areas of RBF application are function approximation and classification. Initially, RBF was used for sorting problems of multivariate interpolation. RBF is very effective as it is quick to train and accept full training algorithms (Looney, 1997).

RBF has a structure that can produce strong links with various areas of statistics. It has been motivated by a statistical pattern recognition theory, regression and regularisation, biological pattern formation, and analysis of "noisy" data. Therefore, RBF has a wide range of applications. The learning system is to find a surface in a multidimensional space that best fits the training data.

Based on the properties of RBF above, which is suitable to classify patterns of the case study data in this research and to enhance the conventional statistical method in dealing with nonlinear data, RBF was therefore selected and used for developing the model.

4.2. Radial Basis Function

The basic principles of RBF are discussed in this section, concentrating on the theory behind the use of RBF pattern recognition for classification problems and not for function approximation.

4.2.1. Theoretical Fundamentals

RBF was first introduced to the problems of multivariate interpolation to deal with irregularity of the data points' positions. According to Looney (1997), the general form of RBF is as follows:

$$x \rightarrow y_1 = f_1(x;v^{(1)}) \ldots x \rightarrow y_M = f_M(x;v^{(M)})$$

where,

$y_m = f_m(x;v^{(m)}) = exp[-||x-v^{(m)}||^2/(2\sigma_m^2)]$ and $m = 1, 2, \ldots, M$

$v^{(m)}$ is the centre of the function and y_m is maximum when $x = v^{(m)}$.

σ_m is the width or receptive field used to control the RBF spread so that its values decrease more slowly or more rapidly as **x** moves away from the centre vector $v^{(m)}$. The bias b_j at each output neuron (r_j) assures nonzero mean values of the sums

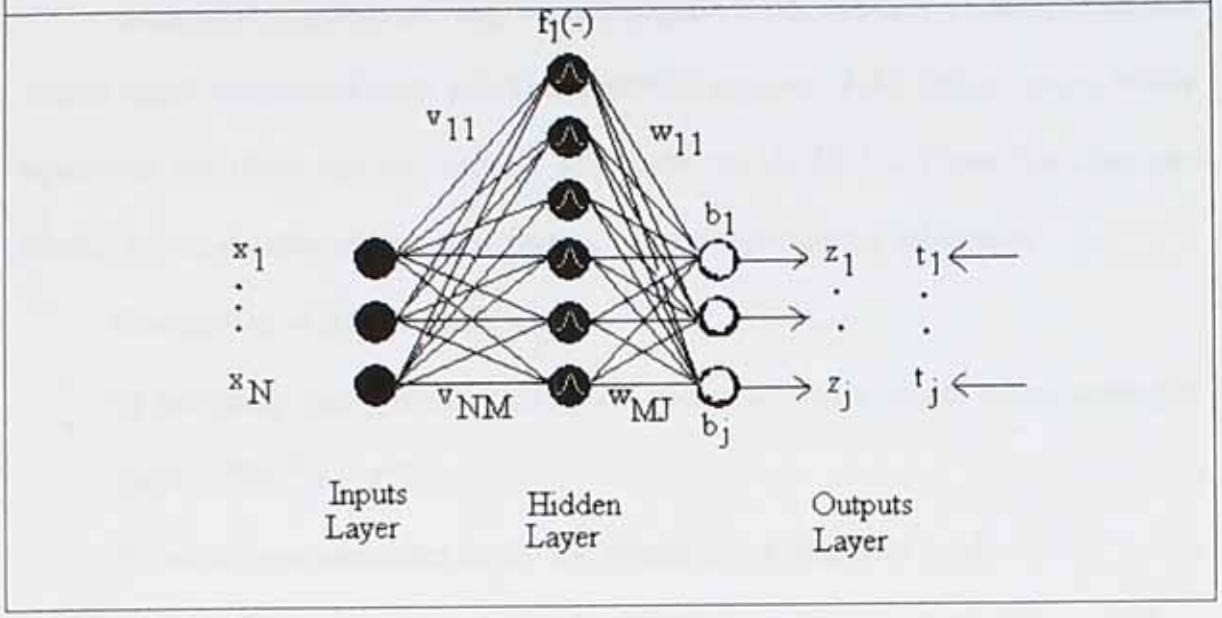
$$r_j = w_{1j}y_1 + w_{2j}y_2 + \ldots + w_{Mj}y_M + b_j.$$

4.2.2. RBF Structure

The RBF structure is relatively simple. It consists of three layers as shown in Figure 4.1. The first layer consists of a number of input neurons with the dimension of N of the input vector x.

The second or hidden layer consists of nonlinear neurons connected directly to all linear output neurons. The number of hidden neurons is equal to the number of sample M. The role of an input layer in RBF networks is to distribute all inputs to each hidden layer nodes. The weights linking the input and hidden layers are all set to unity and maintained during training. The biased term normally seen in feed-forward networks is not required in RBF networks.

Figure 4.1. A General Three-Layer RBF Network

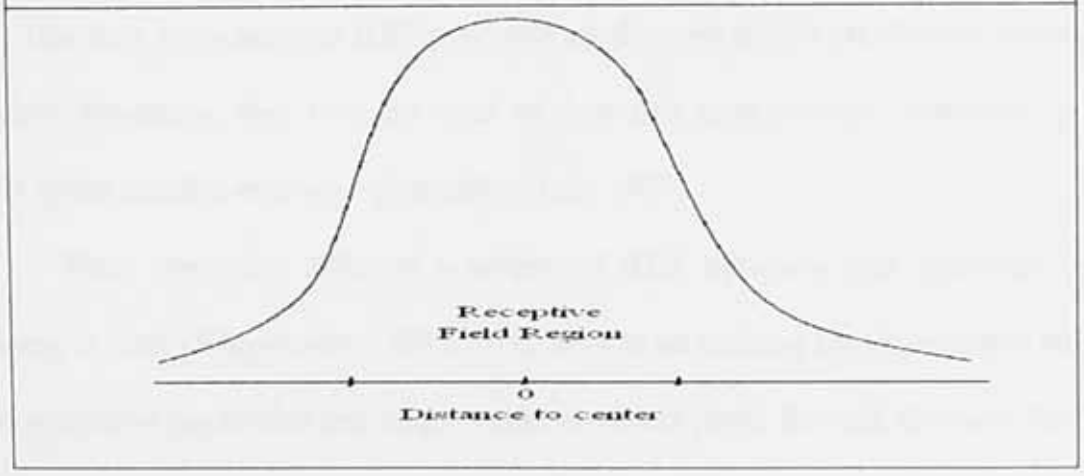


The activated values of y_m are summed to yield a network output z_j and determined by

$$z_j = (\sum_{m=1}^{M} w_m y_m)/(\sum_{m=1}^{M} y_m) \qquad \text{or} \qquad z_j = (1/M)(\sum_{m=1}^{M} w_m / y_m)$$

Figure 4.2 shows an RBF activation function at the m^{th} neuron for which the horizontal axis is the distance $\|x-v^{(m)}\|$ (see Looney, 1997). The region in the feature space where $f_m(x_i;v^{(m)})$ is high is called receptive field of that neuron (Wasserman, 1993).

Figure 4.2. An RBF Activation Function



4.2.3. RBF Initialisation and Learning

Each RBF is influential only on its receptive field, which is a small region of a feature space shown in Figure 4.2. The important regions of the feature space, where exemplars are clustered, are covered jointly by the M RBFs. These functions are centred on the clusters of exemplar feature vectors representing subclasses.

The training of RBF consists in:

(i) assigning each neuron parametric vector $\mathbf{v}^{(m)}$ as a centre of an exemplar vector $\mathbf{x}^{(q)}(\mathbf{v}^{(m)} \leftarrow \mathbf{x}^{(q)})$;

(ii) selecting a parameter σ_m for the spread of the receptive field;

(iii) drawing an initial weight set $\{w_{mj}^{(0)}\}$ for the output layer of neurons; and

(iv) performing supervised training of the weights $\{w_{mj}\}$ in the output layer to force the total sum square error (E) to decrease as much as possible, where

$$E = \sum_{q=1}^{Q} \sum_{j=1}^{J} (z_j^{(q)} - t_j^{(q)})^2.$$

This research adopts a supervised training. Therefore, the corresponding sample of exemplar pairs of input feature vectors and output target vectors of $\{x^{(q)}, t^{(q)}\}$ is given. Each RBF depends on its centre $v^{(m)}$, where it takes its maximum value and is activated by any input x near $v^{(m)}$. It has essentially no response when x is far from $v^{(m)}$. The main advantages of RBF networks are they are simple and trained extremely quickly. Moreover, they have no local minima and have reduced sensitivity to the order of the training exemplar (Bianchini *et al.*, 1995).

There are three different concepts of RBF networks that determine how training is done (Wasserman, 1993). First, there is no training for the simplest model. The weights at the hidden and output neurons remain fixed. Second, the more flexible

model conducts training only on the weights at the output neurons. Third, the most flexible model allows training of all weights at the hidden and output neurons.

RBF uses a method of steepest descent (derivative steepest descent) to adjust the initial weights at the neurons in the output layer. It is a quick training algorithm, which uses the gradient as follows:

$$-\Delta e = -(\partial e/\partial w_{11}, \ldots, \partial e/\partial w_{mj}).$$

Suppose the neuron's centre vectors or the neuron centre $\{\mathbf{v}^{(m)}\}$ is set equal to the exemplar $\{\mathbf{x}^{(n)}\}$. There are n neurons in the hidden layer as every exemplar input $\mathbf{x}^{(n)}$ has a hidden neuron. The neurons in the hidden layer may be too many for a large **N**. Looney (1997) defines that if $\mathbf{N > 200}$, then the network uses a smaller **M,** which is smaller than N, and then uses the full training algorithm.

4.3. Comparison RBF and Multilayer Perceptron

Comparison of RBF with MLP is inevitable since they are universal approximators and used for similar applications. This comparison leads to a better understanding of these two artificial neural network architectures. The differences between the two architectures are both structural (concerning the topology of the network) and functional (concerning the operation and use of the network).

RBFNs have a single hidden layer while MLPs can have more than one hidden layer. Hidden units in RBFNs are different from the output units. MLP hidden units are similar to the output units. The functional differences can be summarised as follows:

- RBF constructs local approximation to nonlinear input-output mappings, while MLPs construct global approximations.

- The output layer of an RBF is always linear, while the MLP output layer can be nonlinear depending on the application.
- RBF hidden units calculate the Euclidian norm between the input vector and their centre, while MLP hidden units compute their inner product of the input vector and their synaptic weight vector.
- MLPs exploit the logistic nonlinearity to create combinations of hyperplanes to dissect pattern space into separable regions. RBF dissect pattern space by modelling clusters of data directly and, therefore, are more concerned with data distribution.

4.4. Suitability of RBF for Detecting and Estimating Rounding Data

The use of RBF in the modelling development to detect rounding in this study is based on the properties of RBF mentioned above. As part of classification problems, this research goal is to recognise and classify the rounding pattern in a data set. RBF is used to find networks that can explain and observe the data set. The classifiers are trained using orthogonal least square algorithm in the Matlab Neural Network Toolbox (Chen *et al.*, 1991).

As mentioned, the general model of RBF is:

$$y_m = f_m(x;v^{(m)}) = exp[-||x-v^{(m)}||^2/(2\sigma_m^2)]$$

where, $m = 1, 2, ..., M$.

Here, the input layer $\boldsymbol{x}$ is the input training data, which is connected to $\mathbf{M}$ hidden neurons $\mathbf{v^{(m)}}$ in the centre of the function. The training data set is created according to the structure or pattern of the data set that contains rounding to a certain base number—i.e., discussed in more detail in the next section. It follows that $\mathbf{y_m}$ is the basis function output from the hidden layer, which is then summed up in the

output layer to yield a network output $\mathbf{z_j}$. This output is a probability density function (PDF), which represents the probability that corresponding to the desired output $\mathbf{y_m}$. PDF is a measure on how well the classifier accounts for the data.

4.5. Simulating the Model

Developing a neural network model for analysing rounding base numbers requires a complete understanding of the data set under investigation, including the pattern with which the data is most likely associated. As one of objectives in this research is to detect a periodic structure in a data set, a simulation on the training using specific data sets is therefore very important. This is to ensure that the neural network model developed subsequently can detect and recognize the different patterns in a data set due to different kinds of the rounding errors.

4.5.1. Underlying Probability Distribution of the Data Sets

Three most common best-fit probability distributions of the underlying data sets are used in this research, namely normal, uniform, and lognormal distributions. These probability distribution functions are intended to shape the training data sets to enable the model to conduct a classification.

The first approach is using normal distribution to examine a pattern in a data set. Based on the Central Limit Theorem[3] of statistics, a pattern of a data set with a large number of observations can always be approached with a normal distribution function.

---

[3] In its simplest form, the theorem states that the sum of a large number of independent observations from the same distribution has, under certain general conditions, an approximate of normal distribution. Moreover, the approximation steadily improves as the number of observations increases.

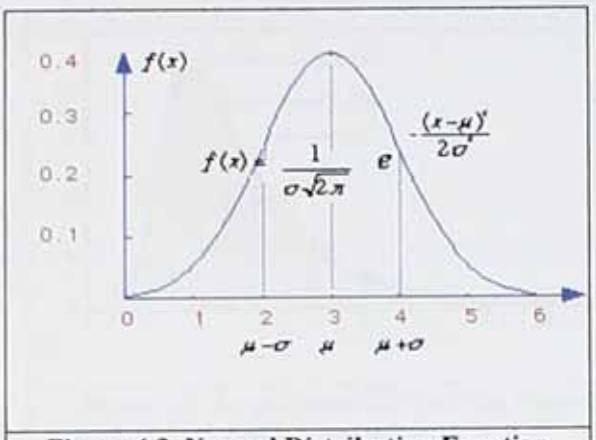


Figure 4.3. Normal Distribution Function

The second approach is to use a uniform distribution function to represent the case if there is no rounding pattern or if the data set is random data. In this case, data distribution is expected to have a "flat" uniform distribution with respect to the data's rounding pattern. In other words, the underlying probability distribution of random data or data set without rounding is expected to follow a uniform distribution.

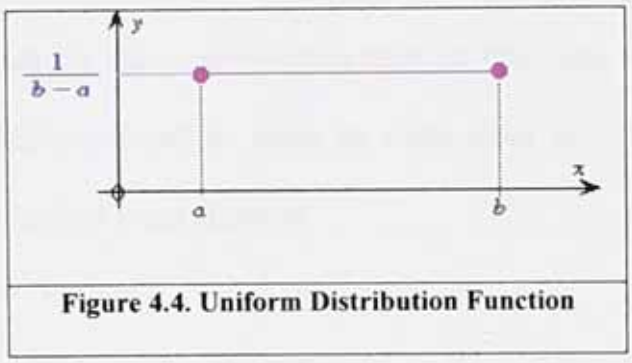


Figure 4.4. Uniform Distribution Function

The third approach is to use a lognormal distribution. This approach is for data sets with a very large range, containing high frequencies at the lower class and low frequencies at the upper class. The data scatter diagram fits a lognormal distribution function as can be seen from Figure 4.5.

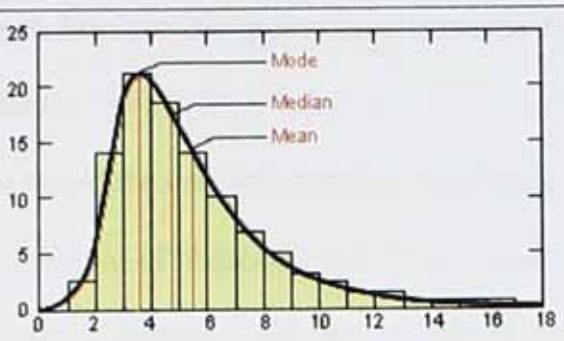


Figure 4.5. Lognormal Distribution Function

4.5.2. Developing Training Data

Training data is needed to develop a network that can subsequently generalise the tasks for which the network is trained. A successful training will enable the network to provide the correct answer for a new input, including a new input different from the training inputs. Therefore, to develop a successful network capable of producing a generalisation, the training data sets must include a variety of examples to prepare the network doing the generalisation task. In this case, constructing training data sets and training presentations must be maximised to ensure that the neural network can do an effective generalisation.

4.5.2.1. Training for Recognising Data Patterns

The initial assumption in this research is that the data set under examination contains rounding to a certain base number that is reflected in a certain pattern. Therefore, the training data set must be created according to the data-set structure or pattern that contains rounding to certain base numbers. This could be rounding to bases 2, 3, 4, 5, and so on. Accordingly, the data pattern will look like a coarsening pattern specific to those particular base numbers. The training of data sets should

therefore recognise each base-number pattern as a potential rounding base unit being investigated in this study.

It then follows that if the data set contains rounding to base 2, the frequencies of number 2 and the multiple of number 2 will have higher observations than other numbers. The frequencies of number 3 and the multiple of number 3 will have higher frequencies than other numbers when the data set contains rounding to base 3, and so on.

Following these rounding patterns, the training data sets are simulated using "dummy" frequency distributions, comprising appropriate patterns of 0s and 1s such as patterns of 0-1-0-1 for base 2, patterns of 0-0-1-0-0-1 for base 3, and patterns of 0 0-0-1-0-0-0-1 for base 4, and so on. Notice that the patterns are shaped according to the underlying probability distributions of the rounding patterns in the data sets and are scaled down to follow the data total frequencies. For example, a data set with 30% rounding to base 5, size of count 10 and total frequency 99 will have possible base numbers of {1, 5, 8, 5, 28, 11, 15, 7, 3, 16}. The probability distribution of base numbers are {0.0161, 0.0373, 0.0720, 0.1162, 0.1565, 0.1760, 0.1653, 0.1296, 0.0848, 0.0464}, respectively. Therefore, the training data set is as follows:

Table 4.1. Example of Training Data Set with 30% Rounding to Base 5, Size of Count 10 and Total Frequency 99

base1	base2	base3	base4	base5	base6	base7	base8	base9	base10
0.0161	0	0	0	0	0	0	0	0	0
0.0373	0.0373	0	0	0	0	0	0	0	0
0.072	0	0.072	0	0	0	0	0	0	0
0.1162	0.1162	0	0.1162	0	0	0	0	0	0
0.1565	0	0	0	0.1565	0	0	0	0	0
0.176	0.176	0.176	0	0	0.176	0	0	0	0
0.1653	0	0	0	0	0	0.1653	0	0	0
0.1296	0.1296	0	0.1296	0	0	0	0.1296	0	0
0.0848	0	0.0848	0	0	0	0	0	0.0848	0
0.0464	0.0464	0	0	0.1565	0	0	0	0	0.0464

4.5.3. An RBF Network Model for Diagnosing Coarsened Data

To examine how the probability value of a base number may be contained in a data set, a classification neural network to optimise this type of analysis is developed. A pair of training-data input and classification output is required for each base number, such as using a potential estimation of submultiple base units and a combination of possible base numbers.

The method used in this classification is ***Newpnn*** function of **Matlab**. This function uses an RBF suitable for a classification problem. ***Newpnn*** designs a probabilistic neural network and the process of designing network is very quick. The function takes two or three arguments: the first argument is a matrix of input vectors, the second is a matrix of target class vectors, and the third argument is a spread which, in statistics, is called a standard deviation. When the function uses only two arguments, it uses a default spread, which is 1.0, and will return a probabilistic neural network. If the spread is near zero, the network will act as the nearest neighbour classifier. As the spread becomes larger, the designed network will take several nearby design vectors into account.

The ***EnewpnnCP*** is a further development of the ***Newpnn*** function of Matlab. It creates a two-layer network. The first layer has radial basis function, calculating its weighted inputs to its distance. This layer outputs are used as inputs for the next layer. The second layer has a competitive function that calculates its weighted input. This network only has a bias in the first layer.

EnewpnnCP sets the first layer weights to a transpose of the training input data, and the first layer biases or spread are all set to 0.8326, resulting in radial basis functions that cross 0.5 at weighted inputs of around the spread (Demuth *et.al.*, 2006).

The second layer weights are set to the targets output class using a competitive function. The codes of ***EnewpnnCP*** are presented in Appendix 1.

The classification process is iterative. The iterations progressively identify potential estimation base units that can be summarised as follows:

(i) In the first iteration, each potential estimate of base unit has an individual training data set corresponding to it. A ranking of base numbers is then produced, i.e., the pattern associated with the top-ranked base number is recognised as being the most significant pattern in the frequency distribution.

(ii) In the second iteration, the highest ranked base number from the first iteration is considered in a combination with other base numbers. Thus, each training data set consists of a pair of training data sets from the first iteration combined. A refined ranking of base numbers is produced: the pattern associated with the top-ranked pair of base numbers is recognised as being the most significant pattern in the frequency distribution.

(iii) For the third iteration, the highest ranked pair of base numbers from the second iteration is considered in a combination with other base numbers. Thus, each training data set consists of triplets of training data sets from the first iteration combined. A more refined ranking of base numbers is produced, i.e., the pattern associated with the top-ranked triplets of base numbers is recognised as being the most significant pattern in the frequency distribution.

(iv) The process is then repeated, adding one base number per iteration, while the probability of the top-ranked combination continues to increase from

the previous iteration. The iteration is stopped when this probability ceases to increase or starts to decrease.

Functions of ***Enormal*** (presented in Appendix 2), ***Eln*** (Appendix 3), and ***Euniform*** (Appendix 4) are employed to do classification by developing ***Newpnn*** of Matlab. ***Enormal*** is a function to classify the data pattern using normal best-fit distribution. ***Eln*** is a function to classify the pattern using lognormal best-fit distribution, while ***Euniform*** is a function using uniform best-fit distribution.

The underlying assumption in this case is that in each iteration, the combination of base units best fit the real frequency distribution is selected and the fit is improving as the probability of the top-ranked combination continues to increase with each successive iteration. Once this probability value either stops increasing or starts to decrease, this means the goodness of the fit stops improving (Triastuti *et. al.*, 2000). Thus, the combination yielding the highest probability should contain all estimation base units plus, possibly, some base numbers which are factors and multiples or multiplicative combinations of the base units selected.

4.5.4. Data Preparation

4.5.4.1. Creating Training Data

A function of ***Etrainingdata*** is to create a training data set. This function takes three arguments of the data set under investigation, base_unit and ***p1***. Base unit is a vector of base number and ***p1*** is the probability of the best-fit distribution. The result of this function is a matrix of training data that are to be inputted into the model. The procedure below summarises the steps to create a training data:

- Define the dimension of the data set under investigation and the vector of a base number.

- Define the dimension of dummy data which should be the dimension of a data set under investigation by the dimension of the vector of a base number with zero values.
- Change the zero values of the dummy data to one if the position is in column ***j*** and the row indices are multiple ***base_unit(j)*** and ***base_unit(j)*** less than the maximum size of the data under investigation.
- Calculate the values of training data by multiplying the dummy data and the probability of the best-fit distribution of ***p1***.

The result is a matrix of training data with the dimension of the size of the data set under investigation by the size of the base number vector. The codes to generate training data are in Appendix 5.

4.5.4.2. Creating Underlying Probability Distributions of the Data Set

Functions of ***Enormfit***, ***Euniform,*** and ***Elognormfit*** are created to get best-fit distributions of the data set under investigation. ***Enormfit*** is used to create a normally best-fit distribution of the data set, ***Euniform*** is used to create uniformly best-fit distribution, and ***Elognormfit*** is used to create lognormal best-fit distribution. The functions take the argument of data set under investigation. For a training experiment, the data set under investigation is an artificial data set. An approach to create the uniform best-fit distribution is the simplest way in this case. The distribution values are the average of the data set under investigation, while the procedure below shows how to create the normally and lognormal best-fit distribution:

- Input a data set under investigation in the function and define the dimension of the data.

- If the number of data set columns is equal to one, then the maximum size of observations is the number of rows of the data set. Otherwise, define the dimension of the maximum size, which takes only the first column of the data set.
- Define or observe a vector of observation size and a vector of observation values or the frequencies of the data set.
- Count the mean, variance, and standard deviation of the data set. Count the logarithm's size for the lognormal distribution.
- Define the dimension of best-fit distribution, which is maximum size rows by three columns.
- Define the first column of best-fit distribution as a vector of base unit being investigated.
- Normalise the data set under investigation.
- Calculate percentages/proportions of the normalised data set under investigation.

The results of **Enormfit** function is **nl_dist** with dimension maximum size of data under investigation in the rows by three columns. The first column is a vector base unit, the second column is the normalised data, and the third column is the probability/proportion of the normalised data set under investigation. The codes to generate normal, lognormal, and uniform best fit distribution of the data sets are shown in Appendices 6, 7, and 8 respectively.

4.5.4.3. Creating Simulated Data Sets

The simulated data sets created include data sets to contain various rounding to various base numbers as well as various percentages level of rounding. These various percentages level of rounding are used to see the sensitivity of the model.

Function of ***Etestdata*** is created to get a data set called ***data_fd***. There are 2 functions, namely: *Etestdatanorm* and *Etestdatauni*. *Etestdatanorm* is to create normally distributed data set (Appendix 9) and *Etestdatauni* is to create uniformly distributed data set (Appendix 10). The function takes two arguments of ***r_frac*** and ***r_base***. The ***r_frac*** is a percentage of data set rounded to ***r_base,*** while the ***r_base*** is the base unit which contains rounding. The procedure below summarizes how to create the data sets:

- Define a value of ***r_frac*** that falls starting from 0 up to 100.
- Define a number of observations or total frequencies ***n***.
- Define a maximum of observation size ***m***.
- Calculate a maximum rounding which is a base unit with rounded values multiplied by a fix of the maximum observation size divided by the base unit with rounded values.
- Define the dimension of all variables used in creating the data set, mainly ***n_fd***, ***b_fd,*** and ***data_fd***. The ***n_fd*** is random exact data, ***b_fd*** is rounded data, and ***data_fd*** is the data set created. They are column vectors with size of maximum observation size.
- Calculate the number of base units that contain rounded values and number of base units with exact values. If the percentage of data set rounded to ***r_frac*** is greater than zero, then the data set contains some rounded values in some certain base units of ***b_fd*** and some exact values in other base units of

n_fd. If the percentage of data set rounded in ***r_frac*** is equal to zero, then all the values in the data are exact values of ***n_fd***.

- Finally, the data set created is the total of exact values of ***n_fd*** and rounded values of ***b_fd***.

4.5.5. Testing the Model

Various simulated data sets are applied to the model to test the robustness of the model. These simulated data used are data sets that contain rounding, including a various rounding of different base numbers as well as various levels of rounding. If a data set contains rounding to base 5, for example, the initial hypothesis is that the shared probability values of base number 5 or multiple base number 5 in the data set are higher than other base numbers.

Five data sets for every type of data sets above are created—i.e., to be tested by the model to see the goodness of the fit and the consistency of the model. The pattern can be summarised by testing with many types of data patterns to the model.

4.6. Data for Application of the Model

Some real data sets are investigated and applied to the model. The data sets are as follows:

1. Religious Census Data 1851. This data set is used to develop the model.
2. Cigarettes Consumption Data for 2001 that include the number of cigarettes smoked in a day and a week.
3. Alcohol Consumption Data in 2001 that include the amount of alcoholic drinks such as beer, shandy, wine, sherry, spirits and alco-pops consumed in the last 7 days.

CHAPTER V

ASSESSING THE BEHAVIOUR OF THE NEURAL NETWORK MODEL IN DETECTING THE ROUNDING BASE

5.1. Introduction

Before testing the neural network model (NNM) by using simulated data sets containing rounding of specific characteristics and different rounding levels in the next chapter, the behaviour of the model in detecting the rounding base is critically examined here. The examination concentrates on the relationship between the underlying distribution of the data being examined and the best-fit distribution used in the detection. In addition, the examination is also intended to come up with some measurements on the robustness of the detections that can be applied in each unique case or in comparison across the possible combinations of the detection. The measurements are developed from the NNM results.

First, two types of data sets are examined in this assessment. They are: distributed uniformly (DU) and distributed normally (DN). In detecting the presence of rounding bases in those data sets, three best-fit distributions are used, namely: best-fit distributions of uniform (BU), normal (BN), and lognormal (BL). Table 5.1 shows all possible cases of the model's applications in detecting the rounding pattern in data sets.

As can be seen from the table, there will be six possible outcomes because of combining two types of data sets and three possible best fits. Out of these six possible outcomes, the implementations of correct best fits for the underlying distributions of the data sets being examined i.e., BU-DU and BN-DN can provide a benchmark for the other

results of the same column. It then follows that a measurement of detection power and detection error can also be developed from the results produced by the model for each possible case summarised in Table 5.1.

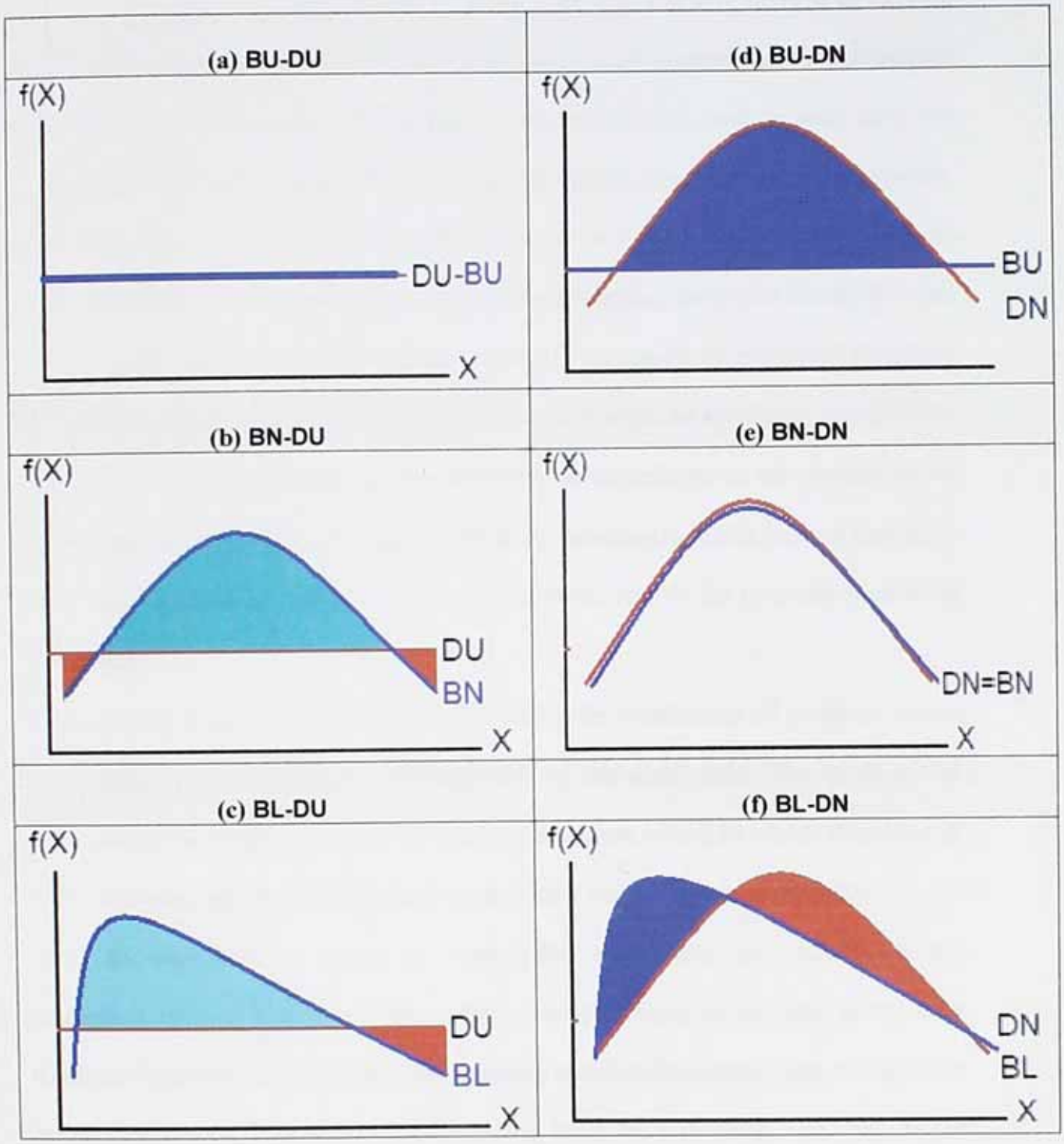
Table 5.1. Schematic Representation of All Possible Cases of Rounding Base Detections

Best-Fit Distribution Functions	Distribution of Data Sets	
	Distributed Uniformly (DU)	Distributed Normally (DN)
Uniform (BU)	BU-DU	BU-DN
Normal (BN)	BN-DU	BN-DN
Lognormal (BL)	BL-DU	BL-DN

5.2. Theoretical Analysis

Figure 5.1 illustrates the diagrammatic representation of the model's applications in detecting rounding to certain base numbers on the uniformly and normally distributed data sets using the three different best fits. The six panels from (a) to (f) show the relationship between the types of best fits used in the detection and the underlying distribution of the data sets being examined. The graphs highlight the robustness of the detection, which is shown by the overlapping areas, and the possibility of detections errors that are indicated by non-overlapping areas. The problematic areas, which arise because of using the wrong best-fit distribution, are shaded in the graphs.

Figure 5.1. Diagrammatic Representation of Detecting Rounding Base on Uniformly and Normally Distributed Data Sets Using Three Different Best-Fit Distributions



From the diagrams, three main outcomes can be summarised as follows:

(i) There should be no detection problem that comes from the use of wrong best fit in panels (a) BU-DU (which means that uniform best fit is used on data distributed uniformly) and (e) BN-DN (which means normal best fit used on data distributed normally). The reason is that the best fits used in both cases are consistent with the underlying distribution of the data sets under examination. Therefore, any problems in the detection results must be associated with something else, including the base number, rounding level, and distribution and size of data that would be discussed further in the sensitivity analysis of detection.

(ii) In panels (b) BN-DU and (d) BU-DN, there might be a tendency to overdetect or under detect and commit detection errors in the mid-points and two tails of the data sets. Panel (b) uses normal best fit for detecting rounding base on uniformly distributed data, while panel (d) uses uniform best fit for normally distributed data.

(iii) In panels (c) BL-DU and (f) BL-DN, the possibilities of problems in the detection are in both low and high ends of data distribution. This is due to the lognormal best fit used, which assumes that there would be a high frequency at the lower end and low frequency at the higher end of the data distribution.

To shed light on this issue, a numerical examination is conducted in the succeeding sections. First is by using random data containing no rounding to any base numbers. Second is by using data sets containing rounding to a certain base number of a certain level— i.e., base 5 of a 30% rounding level. In both cases, two data sets of uniformly and normally distributed in combination with the three best-fit distributions are

used. The main purpose is to show how the probability values of detections play an important role in identifying the rounding base in a data set as well as in quantifying the robustness of the detection. Therefore, some indicators of the robustness of the detection can be developed based on the probability values—i.e., the positive difference in the probability value. The indicators are valid in the context of a single detection or in a comparison across different best fits, base numbers, and rounding levels, because the total positive difference in the probability value of detection is comparable across different situations (see Table 5.2 and Table 5.3).

5.3. Numerical Assessment with Random Data

To conduct the numerical assessment, small data sets are set up so that the implementation of the NNM on the data sets can be traced until all the iterations have been completed. For this purpose, two random data sets are constructed where each has 10 counts and a total frequency of 100 with no rounding to a certain base number. One of the data sets is distributed uniformly, while the other is distributed normally.

To contrast the results and to see the effects of the detailed data distribution, the uniform data is set up as a perfect uniform. This means that each base number or size of count has the same 10 frequencies, while normal data is generated by the computer. The NNM developed in this study is then implemented on these two data sets to detect a rounding pattern to a certain base number, which does not exist in the data, by using the three best-fit distribution functions. Tables 5.2 and 5.3 summarise the detection results on the uniformly and normally distributed random data, respectively.

Before discussing the results, it is important to know how the summary tables describe the detection process. In detecting the rounding pattern, the model starts detecting with base number one as the potential rounding base and then moves to other base numbers following the data distribution. In each iteration, the model produces the probability value. If there is a sign of a spike in the frequency of a particular base number, the probability value increases. Otherwise, the probability value will decline. This iteration is carried out until the probability value decreases to zero. As can be seen in Table 5.2, the zero probability is achieved in the seventh iteration in the normal best fit and the eighth iterations in the other two best fits.

Looking at the column of difference in the pdf value, positive difference means that the probability of detection is still increasing. This indicates that the corresponding base number is more likely to be the potential rounding base that the model tries to identify. On the other hand, the negative difference means that the probability of detection is already decreasing, indicating that the corresponding base number is less likely to be the potential rounding base number and therefore must be dropped from the rounding base potential list. The total values of positive and negative difference must be equal to zero for it is the characteristic of a probability density function in indicating that all detections have been completed for all base numbers. It then follows that the total positive difference in the probability values shows the detection power or an indicator of the robustness in the detection. Accordingly, this indicator can be compared across different best fits or across different data sets.

Table 5.2 shows that using the uniform best fit on the uniformly distributed random data produces no rounding pattern at all even after completing eight iterations.

This is because the data is a perfect uniform data so that all the positive differences in the probability value are associated only to base one. The uniform results are also characterised by the maximum probability values of the detections in the first iteration, for in the remaining iterations the probability values decrease to zero.

If the normal best fit is used, two adverse impacts emerge from the results: (1) the total positive difference in the probability values decreases from 1 to 0.4472; and (ii) there is a positive difference in the probability values associated to base number 10 of 0.0834. As it is very clear from the data set that there is no rounding to base 10, this positive detection is clearly a detection error because of using normal best fit. Therefore, the consequences of using normal best fit instead of uniform best fit are decreasing the detection power to 44.7% and creating detection error of 8.3%, which is calculated from positive difference in the probability values associated to base number 10 divided by the total positive difference in the probability values of using uniform best fit, hence 0.0834 divided by 1.

On the other hand, if the lognormal best fit is used, the two adverse impacts highlighted before are even worse. First, the total positive difference in the probability values decreases from 1 to 0.2860; and second, there are positive differences in the probability values associated to bases number 10 and 9 of 0.0431 and 0.0481. As it is very clear from the data set that there is no rounding to bases 10 and 9, these positive detections are clearly detection errors because of using lognormal best fit. Therefore, the consequences of using lognormal best fit are decreasing the detection power to 28.6% and creating detection error of 12.0%. All percentages are calculated by using the uniform best-fit result as the benchmark.

Table 5.2: The Effects of Using Different Best Fits for Detecting Rounding to Certain Base Number on Uniformly Distributed Random Data with No Rounding

Rounding level=0% Size of Countn=10 Frequency=100 Uniformly Distributed Random Data	Base	Uniform data
	1	10
	2	10
	3	10
	4	10
	5	10
	6	10
	7	10
	8	10
	9	10
	10	10

Best Fit Distribution	Iteration	Base	Pdf	Difference in the pdf	Total Difference (+/-)	Detection Error
Uniform						0.0%
	1	1	1.0000	1.0000	1.0000	
	2	6	0.5000	-0.5000		
	3	7	0.2500	-0.2500		
	4	8	0.1250	-0.1250		
	5	9	0.0625	-0.0625		
	6	10	0.0313	-0.0312		
	7	4	0.0020	-0.0293		
	8	1	0.0000	-0.0020	-1.0000	
Total					0.0000	
Normal						8.3%
	1	1	0.3638	0.3638		
	2	10	0.4472	0.0834	0.4472	
	3	9	0.4137	-0.0335		
	4	8	0.1879	-0.2258		
	5	7	0.0316	-0.1563		
	6	6	0.0025	-0.0291		
	7	1	0.0000	-0.0025	-0.4472	
Total					0.0000	
Lognormal						12.0%
	1	1	0.1657	0.1657		
	2	10	0.2088	0.0431		
	3	9	0.2569	0.0481		
	4	8	0.2860	0.0291	0.2860	
	5	7	0.2478	-0.0382		
	6	6	0.1271	-0.1207		
	7	5	0.0270	-0.1001		
	8	1	0.0000	-0.0270	-0.2860	
Total					0.0000	

Table 5.3 summarises the results using normally distributed random data. The table shows that using normal best fit on the normally distributed random data produced no rounding pattern at all after the completed eight iterations. As the data set is not perfectly normal as in the case of uniform data discussed before, the total positive

difference in the pdf value is less than 1, but it is only 0.6250. The normal best-fit results are also characterised by highest probability values of the detections in the early

Table 5.3: The Effects of Using Different Best Fits for Detecting Rounding to Certain Base Number on Normally Distributed Random Data with No Rounding

Rounding level=0% Size of Countn=10 Frequency=100 Normally Distributed Random Data	Base	Normal data
	1	**1**
	2	6
	3	10
	4	10
	5	**19**
	6	18
	7	21
	8	9
	9	4
	10	**1**

Best Fit Distribution	Iteration	Base	Pdf	Difference in the pdf	Total diff (+/-)	Detection Error
Normal						0%
	1	1	0.6250	0.6250	0.6250	
	2	10	0.6115	-0.0135		
	3	9	0.5380	-0.0735		
	4	7	0.3018	-0.2362		
	5	8	0.1596	-0.1422		
	6	5	0.0069	-0.1527		
	7	6	0.0002	-0.0067		
	8	1	0.0000	-0.0002	-0.6250	
Total					0.0000	
Lognormal						29.2%
	1	1	0.1694	0.1694		
	2	**7**	0.3516	**0.1822**	0.3516	
	3	10	0.3080	-0.0436		
	4	8	0.2589	-0.0491		
	5	9	0.2156	-0.0433		
	6	6	0.0711	-0.1445		
	7	5	0.0030	-0.0681		
	8	1	0.0000	-0.0030	-0.3516	
Total					0.0000	
Uniform						14.3%
	1	1	0.0333	0.0333		
	2	**7**	0.0789	**0.0456**		
	3	**6**	0.1226	**0.0437**	0.1226	
	4	8	0.0540	-0.0686		
	5	5	0.0139	-0.0401		
	6	9	0.0030	-0.0109		
	7	4	0.0002	-0.0028		
	8	1	0.0000	-0.0002	-0.1226	
Total					0.0000	

iterations.

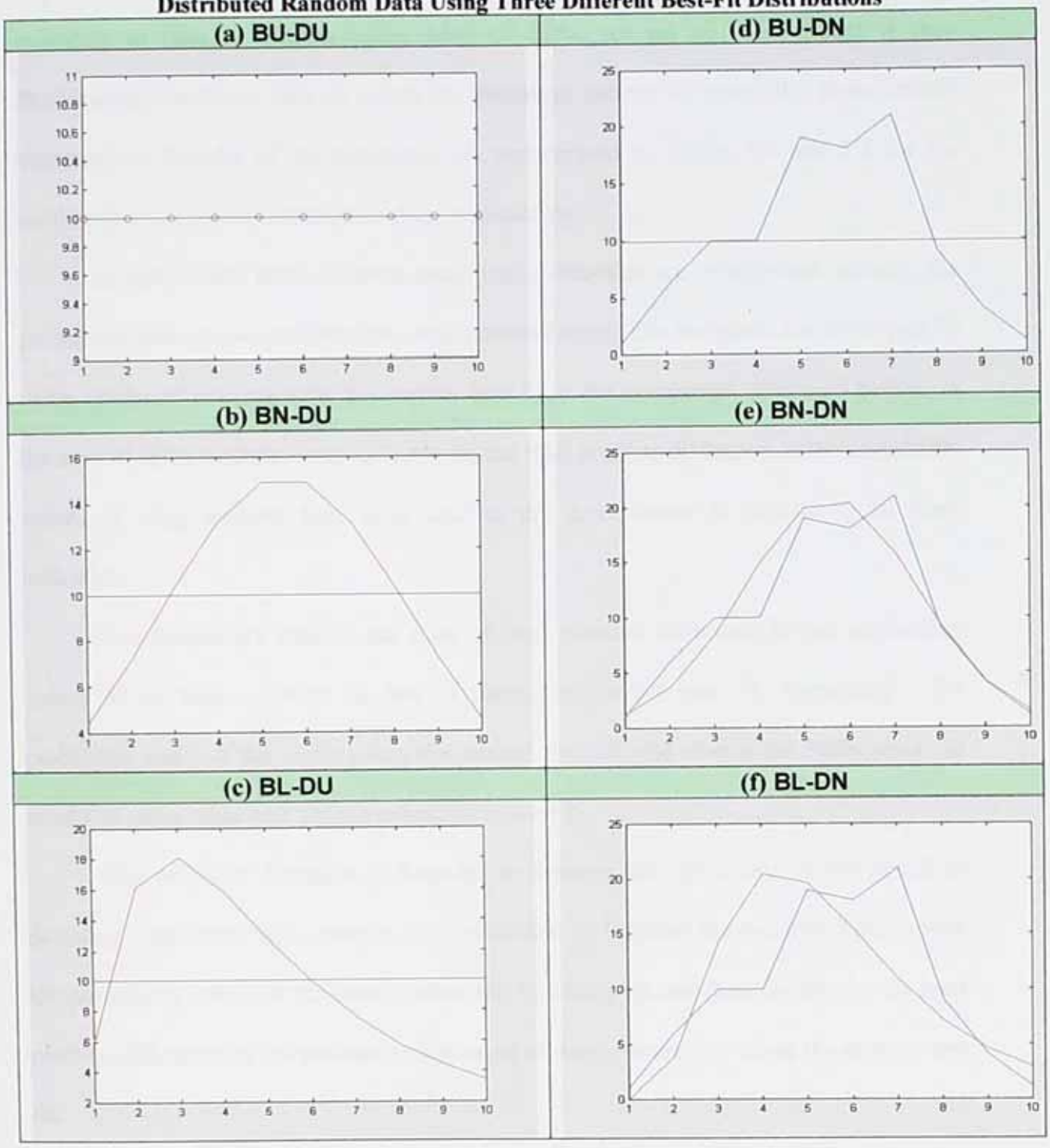
If the lognormal best fit is used, the two adverse impacts are: (i) the total positive difference in the probability values decreases from 0.6250 to 0.3516; and (ii) there is a positive difference in the probability values associated to base number seven of 0.1822. As the data set has no rounding to base seven, this positive detection is a detection error because of using lognormal best fit. Therefore, the consequences of using lognormal best fit instead of normal best fit are: decreasing the detection power to 56.3%—if the normal best fit result of 0.6250 is used as the benchmark of 100%—and creating detection error of 29.2% (ratio of the probability values associated to base number seven to the total positive difference of the normal best fit result).

On the other hand, if the uniform best fit is used, the two adverse impacts highlighted before become: (1) the total positive difference in the probability values decreases from 0.6250 to only 0.1226; and (ii) there are positive differences in the probability values associated to base numbers seven and six with the probability differences of 0.0456 and 0.0437. As the data set has no rounding to these base numbers, these positive detections are detection errors caused by using uniform best fit. Therefore, the consequences of using uniform best fit instead of normal best fit are decreasing the detection power to only 19.6% and creating detection error of 14.3%. All percentages are calculated by using the normal best-fit result as the benchmark discussed before.

Moreover, Figure 5.2 shows the diagrammatic representation of applying the three different best fits on the two random data sets. The figures are developed based on the actual results of the model, which show less smooth graphs than the theoretical graphs in Figure 5.1 From the six panels of Figure 5.2, panel (a) BU-DU is the best result, while

the worst is panel (c) BL-DU. The best detection is represented by the most overlapping areas of the two best fit and underlying data distribution graphs, and vice versa.

Figure 5.2.Diagrammatic Representation of Detecting Rounding Base on Uniformly and Normally Distributed Random Data Using Three Different Best-Fit Distributions



5.4. Numerical Assessment with Coarsened Data

To further conduct the numerical assessment on data sets containing rounding errors, two hypothetical data sets of uniformly and normally distributed containing rounding to base 5 with rounding level of 30%, are set up. The model is then implemented on these data to detect the rounding pattern by using the three best-fit distributions. Results of the detections are summarised in Tables 5.4 and 5.5 for the uniformly and normally distributed data, respectively.

In both tables, three different measures of detection are summarised, namely, the probability ratio, power of detection, and detection error. The measures are developed by using results of implementing the correct best fit as the benchmark discussed before. In the case of using uniform data, for instance, the total positive difference in the probability values of using uniform best fit is used as the denominator in calculating all three indicators.

The probability ratio is the ratio of total positive difference in the probability values of all base numbers to that of using the correct best fit. Accordingly, the probability ratio of the result using the correct best fit will always be 100% since all results of using other best fits are measured against it.

The power of detection is intended to measure the robustness of the model in detecting a particular base number. It is calculated by totalling the positive difference in the probability values of the base number and its multiples, and then divided by the total positive difference in the probability values of all base numbers of using the correct best fit.

Table 5.4: Results of Detecting Rounding Base-5 on Uniformly Distributed Data Sets Containing Rounding Base-5 of 30% by Using Three Different Best Fit Distribution Functions

Rounding Base =5 Magnitude of rounding=30% Size of Count=10 Total Frequency=100 Uniformly Distributed Data	Base Number	Frequency
	1	5
	2	7
	3	5
	4	7
	5	23
	6	5
	7	9
	8	5
	9	12
	10	22

Best Fit Distribution	Iteration	Base	Probability (Pdf)	Difference in the Pdf	Total difference (+/-)	Probability Ratio	Power of Detection	Detection Error
Uniform	1	1	0.0487	0.0487	**0.3896**	**100.0%**	**87.5%**	**0**
	2	**5**	0.3896	**0.3409**				
	3	9	0.257	-0.1326				
	4	10	0.1696	-0.0874				
	5	7	0.0738	-0.0958	**-0.3896**			
	6	6	0.0185	-0.0553				
	7	8	0.0046	-0.0139				
	8	4	0.0001	-0.0045				
	9	1	0	-0.0001				
Total					**0**			
Normal	1	1	0.0205	0.0205	**0.1561**	**40.1%**	**34.8%**	**0**
	2	**5**	0.0984	**0.0779**				
	3	**10**	0.1561	**0.0577**				
	4	9	0.1004	-0.0557	**-0.1561**			
	5	8	0.0069	-0.0935				
	6	7	0.0004	-0.0065				
	7	1	0	-0.0004				
Total					**0**			
Lognormal	1	1	0.0088	0.0088	**0.0598**	**15.3%**	**9.9%**	**3.2%**
	2	**5**	0.0243	**0.0155**				
	3	**10**	0.0474	**0.0231**				
	4	**9**	0.0598	**0.0124**				
	5	8	0.0276	-0.0322	**-0.0598**			
	6	7	0.0113	-0.0163				
	7	6	0.0009	-0.0104				
	8	1	0	-0.0009				
Total					**0**			

On the other hand, the detection error is geared to measure the error of detecting other base numbers. It is calculated by summing up all positive differences in the probability values of a base number that should not be detected by the model, and then

divided by the total positive difference in the probability values of all base numbers of using the correct best fit.

Table 5.4 shows that the probability ratios of detecting rounding base 5 on the uniformly distributed data containing 30% rounding level by using normal and lognormal best fits are 40.1% and 15.3%, respectively. This highlights the importance of using the correct best fit in detecting the rounding base. Having mentioned this, the power of detection using uniform best fit is not a 100%, but it is about 87.5%, which is calculated as the ratio of the positive probability value of detecting rounding base 5 to the total positive probability value in this best fit. Among others, the power of detection depends on the rounding base, level of error, and data distribution. If the normal and lognormal best fits are used, the power of detection will drop to 34.8% and 9.9%, respectively. Moreover, if the lognormal best fit is used, there will also be a detection error of 3.2% because of the model detecting base 9, which is not the multiple of base 5, with a positive difference in the probability value of 0.0214.

Results of the numerical assessment on the normal data summarised in Table 5.5 produce similar results but with less variation across different best-fit applications. Using the correct normal best fit, for instance, will generate detection power of 85.5%, while when using lognormal and uniform best fits, the detection power will decrease to 20.7% and 34.3%. The last two best fits produce probability ratios of 33.2% and 40.5% of the normal best-fit result. Moreover, if the uniform best fit is used, there will also be a detection error of 0.8 % because of the model detecting base 7, which is not the multiple of base 5, with a positive difference in the probability value of 0.0026.

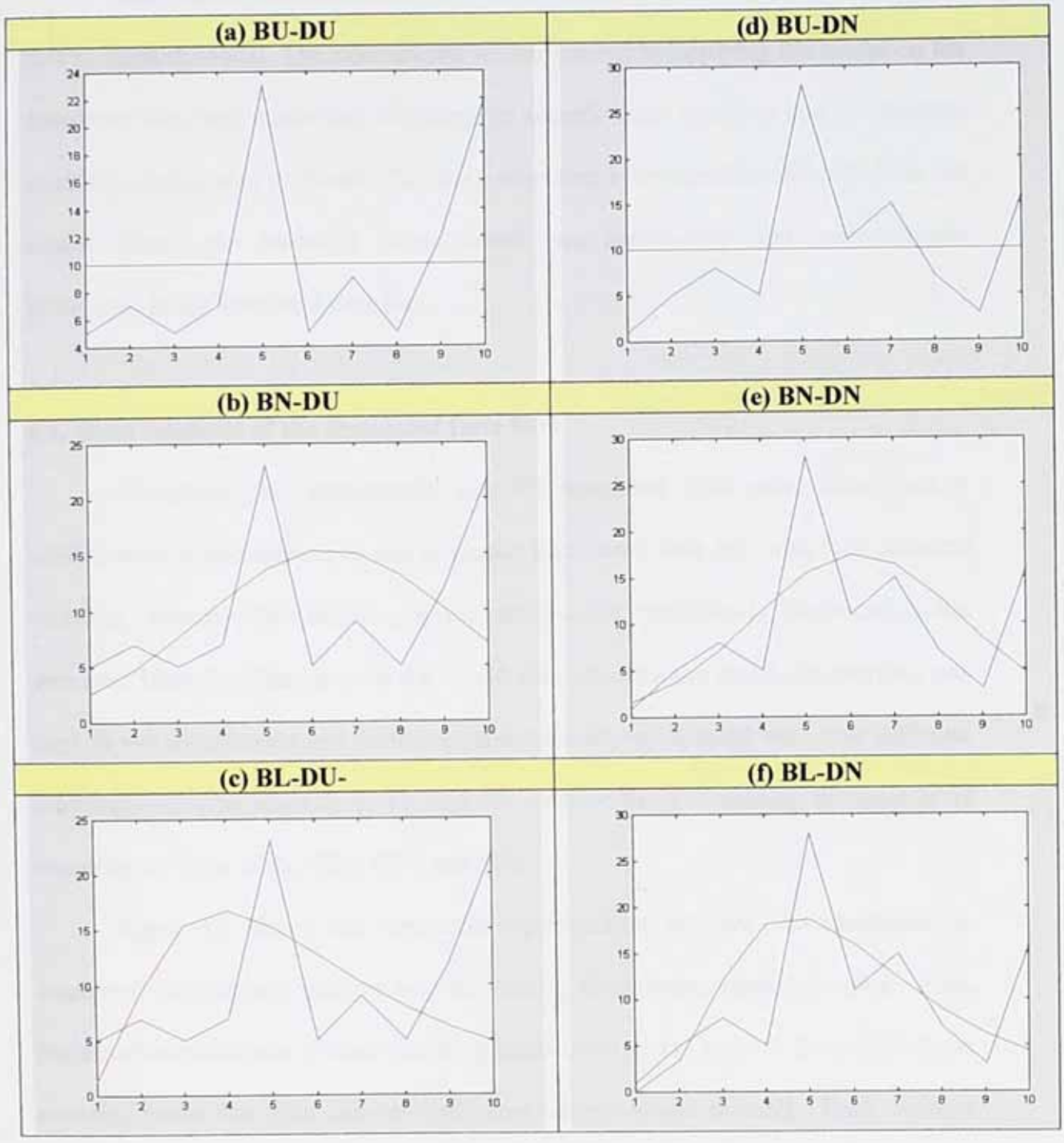
Moreover, Figure 5.3 shows the diagrammatic representation of applying the three different best fits on the two data sets containing rounding base 5 of 30% rounding level developed based on the actual results of the model. The figures show less smooth graphs than the theoretical graphs in Figure 5.1. From the six panels of Figure 5.3, panel (a) BU-DU seems to produce the best result, while the worst is for panel (c) BL-DU. In a graphical format, this can be seen from the overlapping areas of the two graphs of best fit used and the underlying data distribution.

Table 5.5: Results of Detecting Rounding Base-5 on Normally Distributed Data Sets Containing Rounding Base-5 of 30% by Using Three Different Best Fit Distribution Functions

Rounding Base =5 Magnitude of rounding=30% Size of Count =10 Total Frequency =100 Normally distributed data	Base Number	Frequency
	1	1
	2	5
	3	8
	4	5
	5	28
	6	11
	7	15
	8	7
	9	3
	10	16

Best Fit Distribution	Iteration	Base	Probability (Pdf)	Difference in the Pdf	Total difference (+/-)	Probability Ratio	Power of Detection	Detection Error
Normal	1	1	0.0452	0.0452	0.311	100.0%	85.5%	0%
	2	**5**	0.2318	**0.1866**				
	3	**10**	0.311	**0.0792**				
	4	9	0.0995	-0.2115	-0.311			
	5	7	0.0109	-0.0886				
	6	8	0.0012	-0.0097				
	7	6	0	-0.0012				
Total					0			
Lognormal	1	1	0.0389	0.0389	0.1033	33.2%	20.7%	0%
	2	**5**	0.0747	**0.0358**				
	3	**10**	0.1033	**0.0286**				
	4	9	0.0597	-0.0436	-0.1033			
	5	7	0.0341	-0.0256				
	6	8	0.0157	-0.0184				
	7	6	0.0009	-0.0148				
	8	1	0	-0.0009				
Total					0			
Uniform	1	1	0.0167	0.0167	0.1261	40.5%	34.3%	0.8%
	2	**5**	0.1235	**0.1068**				
	3	**7**	0.1261	**0.0026**				
	4	6	0.0736	-0.0525	-0.1261			
	5	8	0.0245	-0.0491				
	6	10	0.0072	-0.0173				
	7	9	0.0014	-0.0058				
	8	4	0	-0.0014				
Total					0			

Figure 5.3. Diagrammatic Representation of Detecting Rounding Base 5 on Uniform and Normal Data with Rounding Level of 30% Using Three Different Best Fit Distributions



CHAPTER VI

ASSESSING THE ROBUSTNESS OF THE NEURAL NETWORK MODEL

This chapter conducts additional assessments to examine the robustness of the neural network model. The assessments are carried out by applying the model on the simulated data sets containing rounding to specific base numbers and of different rounding magnitudes or levels. The main objective is to examine critically how the model detects the rounding bases, which are *deliberately* and *systematically* introduced in the simulated data sets.

6.1. Main Features of the Simulated Data Sets

Altogether, the assessments use 40 simulated data sets. These are a combination of two uniformly and normally distributed data sets with four different rounding bases and five rounding levels, which are systematically introduced in the data sets. Therefore, there are 2 x 4 x 5 = 40 data sets. In more detail, the two data sets used in the assessments are uniformly and normally distributed with four different roundings—i.e., to bases 5, 7, 11, and 10, each of them containing different level roundings of 10%, 20%, 30%, 40%, and 50%.

Figure 6.1 shows the schematic representation on how the assessment is conducted. It illustrates another way of looking at the assessments procedure. First, the simulated data sets contain rounding bases of 5, 7, 11, and 10. In each of these rounding bases, two data sets are distributed uniformly and normally. Each of these data has five different rounding levels introduced in the data sets, i.e., 10%, 20%, 30%, 40%, and 50%. Then to detect the rounding pattern on each, the model is applied by using the three best fits of uniform, normal, and lognormal.

The rounding bases of 5, 7, and 11, are chosen for they are prime numbers, which by definition can only be divided by one and the number itself. Therefore, the primary numbers have no factor numbers that can be associated. Given this unique characteristic, their existence in the form of rounding base in a data set can be detected clearly without any risk of attributing the rounding to other base numbers. In other words, using non-prime numbers in the assessment, for instance, will increase the risk of associating the rounding base number to its factors such as rounding base 6 with rounding bases 2 and 3.

It can be seen that there is one non-prime number (i.e., 10) introduced as a rounding base in the simulated data set. The rounding base 10 is included for mainly three reasons:

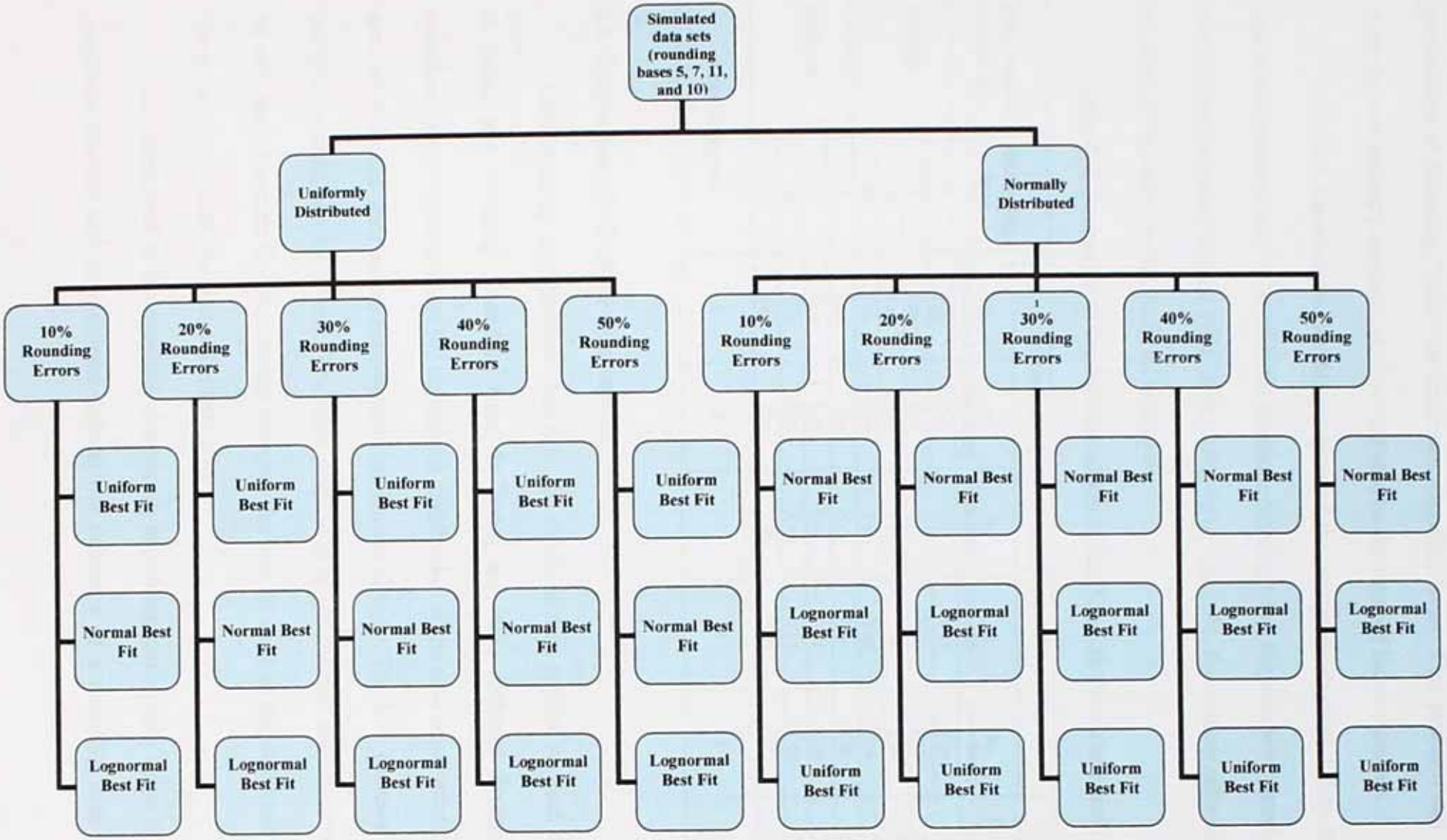
(i) It is arguably the most common rounding base, as counting in 10s is very common;

(ii) To examine the factor issue in a rounding base, as 10 can be seen as a multiple of five and two in the context of detecting a rounding base; and

(iii) Rounding base 5 was included based on the primary number criterion, to see also the effect of rounding base replication which, in this case, is the replication of rounding base 5.

Moreover, the different rounding levels of 10% to 50% with the interval 10% introduced in the simulated data are to examine further the model performance in detecting the rounding pattern, as well as for the sensitivity analysis in the context of increasing rounding magnitude in the data sets. The main idea is that as the rounding level increases, the model should be able to detect the existence of rounding bases more strongly. On the other hand, as the rounding becomes more dominant, the data sets can be distorted significantly that makes rounding detection more difficult with

Figure 6.1. Assessing the Goodness Fit of the Model Using Simulated Data Sets



possibilities of detecting "other" or irrelevant rounding bases. This problem can be worse for non-primary numbers as there will also be the issue of factor numbers.

Table 6.1 summarises the descriptions of the complete simulated data sets used in the assessments. As mentioned, for each rounding base, the data sets introduce five different rounding levels: 10%, 20%, 30%, 40%, and 50%. For further reference, the name of the data sets are A, B, C, D, and E.

Table 6.1. Summary of the Simulated Data Sets Used in the Assessment

Distribution of data sets	Rounding base	Percentage of rounding in the data sets (%)					Total Data Sets
		10 (Data Set A)	20 (Data Set B)	30 (Data Set C)	40 (Data Set D)	50 (Data Set E)	
Normal	5	Yes	Yes	Yes	Yes	Yes	5
	7	Yes	Yes	Yes	Yes	Yes	5
	10	Yes	Yes	Yes	Yes	Yes	5
	11	Yes	Yes	Yes	Yes	Yes	5
Uniform	5	Yes	Yes	Yes	Yes	Yes	5
	7	Yes	Yes	Yes	Yes	Yes	5
	10	Yes	Yes	Yes	Yes	Yes	5
	11	Yes	Yes	Yes	Yes	Yes	5
Total Data Sets		8	8	8	8	8	40

6.2. Conducting the Critical Assessments

Having set up the simulated data sets, the model is then applied on each data to detect the "existing" rounding base. In doing so, three different functional specifications or best fits about the underlying distribution of the data being examined are used although their actual distributions are known already. The best fits used are uniform, normal, and lognormal. The main purpose is to get the best detection result as well as to examine the relationship between the best fit used in the detection and the actual distribution function of the data sets.

As mentioned in the previous chapter, an increasing probability value of the particular number and its multiples reflects the existence of a rounding base to a

certain number in a data set. In a graphical format, this can be seen as spikes in otherwise a smooth graph.

Therefore, the increasing probability density function (pdf) of each rounding base can be used as an indicator of the rounding base present in a data set. It then follows that a higher increase in the pdf value of a particular base number shows the more likely that the number is the rounding base. Thus, a higher increase in pdf value of a potential rounding number shows a better detection of the rounding number.

The positive difference reflects the increasing pdf value of a base number. This means that the corresponding base number is more likely to be the potential rounding base that the model tries to identify. On the other hand, the negative difference in the pdf value means that the detection probability is already decreasing, indicating that the corresponding base number is less likely to be the potential rounding base number, and therefore, the base number must be dropped from the rounding base potential list.

Therefore, the probability values presented in the summary tables (i.e., Tables 6.2 to 6.9) in this chapter are only for the increasing probability values to indicate that their corresponding base numbers could be the rounding numbers that the model is trying to identify. This means that all decreasing probability values that leads to the zero probability value when the iterations completed (see Chapter 5 on numerical assessment), are excluded from the summary tables. In some cases, stagnant or stable probability values—not increasing but not decreasing either—are also included in the summary just to ensure that there is no potential rounding bases left in the detection process. All potential rounding bases are then scrutinised further to ensure that they are the actual rounding bases.

Moreover, to further clarify the comparisons of the robustness of the detection across different rounding levels, best fits used, and rounding base numbers some indicators of the robustness of the detection developed and discussed with some examples in the previous chapter are used in this chapter. They are the detection power and the detection error. These indicators are calculated from the positive difference in the probability value of a particular base number, divided by the total positive difference in the probability value of detecting the particular rounding base number by using the correct/consistent[4] best fit.

For instance, the detection power of rounding base 5 on the uniform data with 20% rounding level is calculated by the positive difference in the probability value of detecting base 5 (and its multiples) base numbers on this data set (uniform data with 20% rounding level) divided by the total positive difference in the probability value of detecting rounding base 5 using uniform best fit (the correct best fit). For the detection error, the numerator is the positive difference in the probability value of rounding base other than base 5 and its multiples (if any), while the denominator is the same—i.e., the total positive difference in the probability value of detecting rounding base 5 using a uniform best fit.

The indicators[5] are valid in the context of a single detection or in comparisons across different best fits, rounding levels, base numbers, and even data sets because the total positive difference in the probability value of detection is comparable across different situations.

It is important to note that the total positive difference in the probability values will decrease as the level of rounding contained in the data set increases. Recall from

---

[4] Using best-fit functional distribution, which is consistent with the underlying distribution of data being examined.

[5] Although the indicators are unit free, the actual number is more likely dependent on the characteristics of the data sets being examined such as the size, unit of measurement, degree of errors in data collection, and so on.

the random data experiment that in case of no rounding and the data is perfectly distributed, the pdf value will be equal to one and associated only to base 1. Therefore, the total positive difference in pdf value will also be equal to one when there is no rounding and the data is perfectly distributed. Hence, the more rounding contained in a data set, the lower would be the total positive difference in pdf value as the model starts to detect other base numbers. As can be seen from the table, the difference in the pdf values associated with the particular base number increases following the increase in the levels of rounding to the particular base number in the data sets. For instance, the difference in the pdf values of detecting rounding base 5 in data set C (at 30% rounding) using normal best fit is only 0.0029, while in the data set E (at 50% rounding) the difference is 0.9701. Therefore, more and more decreasing probability values are excluded from the total number as the level of rounding contained in the data sets increase from 10% to 50%.

Tables 6.2 to 6.9 summarise the assessment results of applying the model on the uniformly and normally distributed simulated data sets, each of which contains rounding to bases 5, 7, 11, and 10 of five different rounding levels. The rounding base detection is conducted by using uniform, normal, and lognormal best fits.

In discussing results, the best result of using the correct best fit is used as the benchmark so other results are compared to this. In addition, the Modulo test result is also used as an additional benchmark. Therefore, the tables also summarise the Modulo test results in data sets. In this context, any rounding base detected by NNM must also be confirmed by the same rounding base detection by the Modulo test method that is reflected in the increasing probability value of the particular base number.

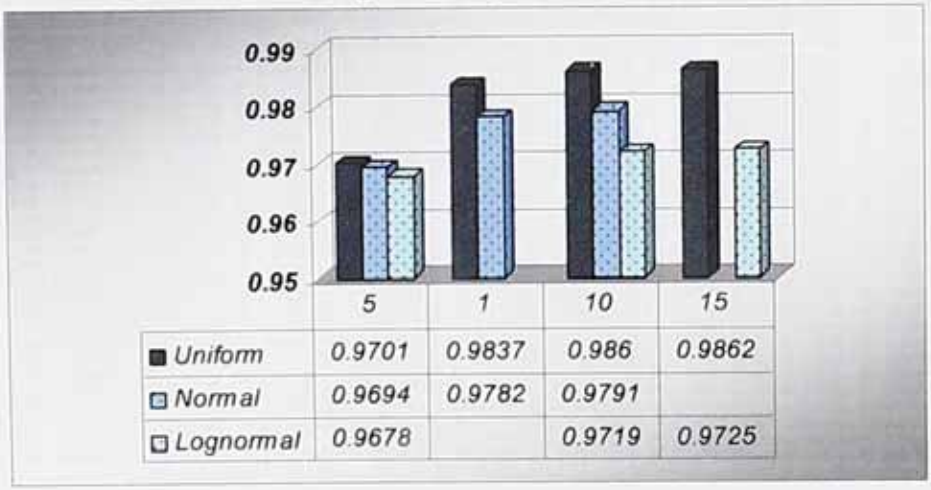
6.3. Assessments Using Uniformly Distributed Data Sets

The first assessment is to use the uniformly distributed data sets that contain four different rounding bases with five different rounding magnitudes as described in the beginning of this chapter. The results are discussed for each rounding base (i.e., rounding bases 5, 7, 11, and 10.

6.3.1. Detecting Rounding Base 5

Results from implementing NNM on the uniformly distributed simulated data sets containing rounding base 5 with different rounding magnitudes show that NNM can detect the existing rounding base very well. The rounding base 5 detection is getting stronger as the degree of rounding contained in the data set increases from 10% to 50% (i.e., in data sets A to E). The increasing number of rounding bases detected and/or the increasing total probability values of the detections show this. This result is consistent with the initial expectation on the overall results, hence confirming the goodness fit of NNM.

Figure 6.2. Probability Values of Detecting Rounding Base 5 in a Simulated Data Set Containing Rounding Base 5 at 50% Rounding



	5	1	10	15
Uniform	0.9701	0.9837	0.986	0.9862
Normal	0.9694	0.9782	0.9791	
Lognormal	0.9678		0.9719	0.9725

Moreover, although NNM can detect the rounding base regardless of the best fits used in the detection, the use of a correct best fit of the data set being examined

will guarantee best results. Figure 6.2, for instance, shows the probability values of detecting rounding base 5 on data containing rounding base 5 at 50% rounding level using the three best fits. The figure clearly shows the crucial role of using the right best fit, as using the wrong best fit will result in a lower probability of detection and/or non-detection of some rounding bases. All rounding bases detected by the uniform best fit have higher probability values than results of using normal and lognormal best fits. Additionally, using the last two best fits will not detect rounding bases 15 and 1, respectively.

Furthermore, Table 6.2 summarises the results of detecting rounding base 5 of different magnitudes contained in the uniform data by using the three best fits. The detailed detection results are discussed in turn as follows:

Table 6.2. Results of Detecting Rounding Base 5 Contained in the Uniformly Distributed Simulated Data Sets Using Three Different Best-Fit Distribution Functions

Best Fit Distribution	Iteration	Magnitude of Rounding Errors																			
		10%				20%				30%				40%				50%			
		Base	Pdf	Detection Power	Detection Error	Base	Pdf	Detection Power	Detection Error	Base	Pdf	Detection Power	Detection Error	Base	Pdf	Detection Power	Detection Error	Base	Pdf	Detection Power	Detection Error
Uniform	1	1	0.9921	0.00	0.00	1	**0.9891**	0.29	0.00	1	**0.9845**	0.66	0.00	1	**0.9781**	1.05	0.00	5	**0.9701**	97.26	0.00
	2					5	0.9920			5	**0.9910**			5	**0.9881**			1	**0.9837**		
	3													10	0.9886			10	0.9860		
	4																	15	0.9862		
Normal	1	1	0.9887	0.00	0.00	1	0.9857	0.09	0.00	1	0.9811	0.45	0.00	1	0.9747	0.81	0.00	5	**0.9694**	97.03	0.00
	2					5	0.9866			5	0.9856			5	**0.9827**			1	**0.9782**		
	3																	10	0.9791		
Lognormal	1	1	0.9830	0.00	0.00	1	0.9801	0.00	0.00	1	0.9755	0.09	0.00	5	0.9722	0.23	0.00	5	**0.9678**	97.25	0.00
	2									5	0.9764			10	0.9745			10	0.9719		
	3																	15	0.9725		
Modulo Test		1	1.0000			1	1.0000			1	1.0000			1	1.0000			1	1.0000		
		2	0.5018			2	0.5011			2	0.5032			2	0.5007			2	0.5006		
		3	0.3355			3	0.3356			3	0.3374			3	0.3398			3	0.3413		
		4	0.2470			4	0.2447			4	0.2460			4	0.2464			4	0.2445		
		5	**0.2828**			**5**	**0.3630**			**5**	**0.4431**			**5**	**0.5203**			**5**	**0.6008**		
		6	0.1675			6	0.1666			6	0.1711			6	0.1721			6	0.1751		
		7	0.1442			7	0.1448			7	0.1423			7	0.1419			7	0.1433		
		8	0.1253			8	0.1255			8	0.1243			8	0.1242			8	0.1223		
		9	0.1105			9	0.1093			9	0.1100			9	0.1106			9	0.1124		
		10	**0.1426**			**10**	**0.1807**			**10**	**0.2219**			**10**	**0.2605**			**10**	**0.3017**		
		11	0.0856			11	0.0856			11	0.0848			11	0.0858			11	0.0864		
		12	0.0838			12	0.0833			12	0.0860			12	0.0871			12	0.0874		
		13	0.0773			13	0.0778			13	0.0760			13	0.0768			13	0.0775		
		14	0.0746			14	0.0750			14	0.0741			14	0.0744			14	0.0747		
		15	**0.0942**			**15**	**0.1205**			**15**	**0.1486**			**15**	**0.1755**			**15**	**0.2042**		
		16	0.0620			16	0.0610			16	0.0592			16	0.0594			16	0.0583		
		17	0.0583			17	0.0594			17	0.0595			17	0.0587			17	0.0596		
		18	0.0545			18	0.0542			18	0.0566			18	0.0572			18	0.0591		
		19	0.0515			19	0.0504			19	0.0510			19	0.0504			19	0.0495		
		20	**0.0657**			**20**	**0.0836**			**20**	**0.1042**			**20**	**0.1242**			**20**	**0.1431**		
		21	0.0493			21	0.0488			21	0.0486			21	0.0501			21	0.0492		
		22	0.0439			22	0.0430			22	0.0417			22	0.0428			22	0.0426		
		23	0.0406			23	0.0402			23	0.0395			23	0.0409			23	0.0412		
		24	0.0425			24	0.0422			24	0.0434			24	0.0441			24	0.0442		
		25	**0.0879**			**25**	**0.0743**			**25**	**0.0916**			**25**	**0.1067**			**25**	**0.1239**		

Uniform Best-Fit Distribution Function

Detecting rounding base 5 on the uniform data using uniform best fit shows that the model detects rounding base 1, which is also an exact count base unit that includes base 5, as a potential rounding base in the first iteration on data sets A to D. However, in data set E, the model detects rounding base 5 in the first iteration and base 1 on the second iteration. In data set E, the model also detects the rounding base 15. In the second iteration, the model detects rounding base 5 in data sets B to D. The model then detects rounding base 10 in the third iteration in data sets D and E.

Normal Best-Fit Distribution

Detecting rounding base 5 in the uniform data using normal best fit produces similar results of uniform best fit. The differences are (i) the detection probability values are lower, (ii) there is no rounding base 10 detected in data set D, and (iii) there is also no rounding base 15 detected in data set E. Therefore, the normal best-fit results are less superior to uniform results.

Lognormal Best-Fit Distribution

Detection results using the lognormal best fit shows the least powerful. This can be seen n the facts that (i) there is no detection of rounding base 1 in the data sets D and E; and (ii) there is no detection of rounding base 5 in data set B. Therefore, the results of using uniform best fit are the best.

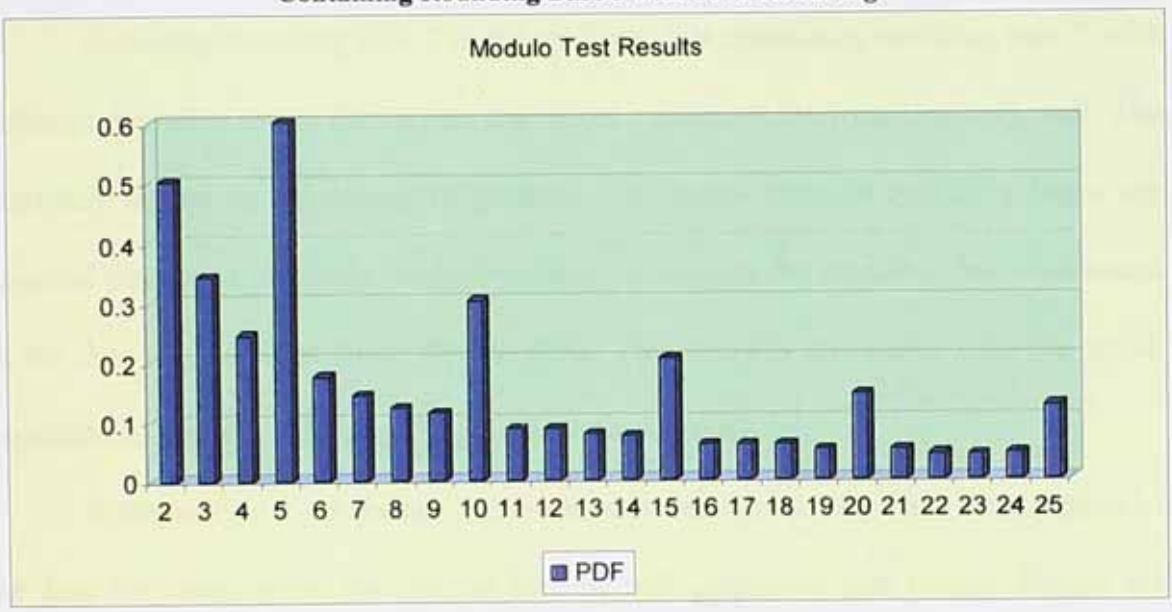
Modulo Test Method

The rounding base 5 detection in the data sets is further confirmed with the Modulo test results, which shows increasing probability values in the base numbers of five and its multiples, such as base numbers 10, 15, 20, and 25. Moreover, the probability values of each base number keep increasing as the rounding level increases from 10% to 50%. For instance, the rounding base 5 probability values in

data set A is 0.2828, while in data set E it increases to 0.6008. This increasing probability value is also observed in the base 5 multiples such as bases 10, 15, 20, and 25.

Figure 6.3 shows the rounding base 5 probability values in data containing rounding base 5 of 50% rounding level by using the Modulo test. The graph clearly shows the rounding base 5 presence, as indicated by the spikes in the probability values of base 5 and its multiples of bases 10, 15, 20, and 25. The base 5 probability values and its multiples keep decreasing from 0.6008 for the bases 5 to 0.1239 for base 25.

Figure 6.3. Modulo Test Results of Detecting Rounding Base 5 in a Simulated Data Set Containing Rounding Base 5 with 50% Rounding



Moreover, to compare the robustness of the model in detecting rounding base 5 across different rounding levels and best fits used, Table 6.2 also summarises the detection power and detection error for each case. The results can be summed up as follows:

(i) There is no detection of rounding base 5 in data set A in all best fits used and also in data set B using lognormal best fit.

(ii) Using the correct best fit will produce the highest detection power, followed by using normal and lognormal best fits. In data set E, however, lognormal best fit produces the second highest, while in data sets A to D, lognormal best fit produces very low detection power.

(iii) The detection power jumps significantly in data set E, from a maximum of 1% to more than 97%. This is because all best fits in data set E detect the rounding base 5 in their first iterations.

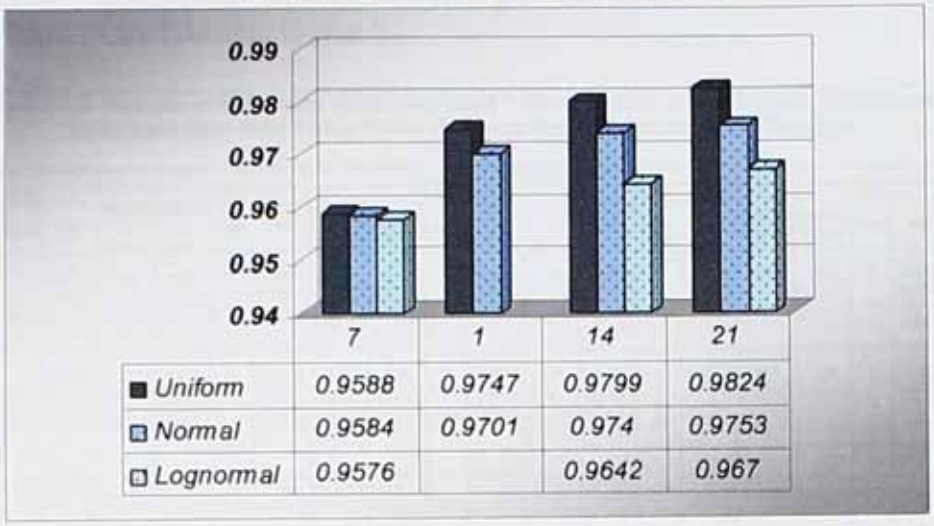
(iv) There is no detection error identified from the results.

6.3.2. Detecting Rounding Base 7

Detecting rounding base 7 in the uniform data containing rounding base 7 with different rounding levels shows that the model can detect the rounding very well. The detection shows an increasing magnitude—i.e., more relevant rounding bases are detected that result in higher total probability values—as the rounding level contained in the data set increases from 10% to 50%. This trend is consistent with the initial expectation, hence confirming the goodness fit of the model.

Furthermore, even though the model can detect rounding base 7 regardless of the best fits used, using the correct best fit will guarantee best results. Figure 6.4 shows the rounding base 7 probability values in data containing rounding base 7 of 50% by using the three best fits. The figure shows that using the wrong best fit will result in lower probability values and non-detection of some relevant rounding bases. For all rounding bases detected by the model, the uniform best fit produces the highest probability values.

Figure 6.4. Probability Values of Detecting Rounding Base 7 in a Simulated Data Set Containing Rounding Base 7 of 50% Rounding



	7	1	14	21
Uniform	0.9588	0.9747	0.9799	0.9824
Normal	0.9584	0.9701	0.974	0.9753
Lognormal	0.9576		0.9642	0.967

Table 6.3 summarises the results of detecting rounding base 7 in uniform data using the three best fits. The detailed detection results are discussed in turn as follows:

Uniform Best-Fit Distribution

Detecting rounding base 7 using uniform best fit in the uniform data containing rounding base 7 shows that the model detects the rounding base 7 in the second iteration in data sets A to D, and in the first iteration in data set E. The model starts to detect rounding base 14 in data sets C to E, and rounding base 21 in data sets D and E. These results are very good, which cannot be obtained by using other best fits. If other two best fits are used, rounding base 7 can only be detected in the second iteration in data sets B to D for normal while in data sets B and C for lognormal.

Normal Best-Fit Distribution Function

Results of detecting rounding base 7 using normal best fit have similar patterns with uniform results. The differences are: (i) normal best fit detects rounding base 7 and its multiples with lower probability values; (ii) it can only detect rounding base 7 in data sets B to E, while the uniform best-fit distribution can detect rounding base 7 in data sets A to E; (iii) it detects rounding base 21 in the third iteration and not

rounding base 14 as in the case of using uniform best-fit distribution; and (iv) detects rounding base 14 in data sets D and E.

Table 6.3. Results of Detecting Rounding Base 7 Contained in the Uniformly Distributed Simulated Data Sets Using Three Different Best-Fit Distribution Functions

Best Fit Distribution	Iteration	Magnitude of Rounding Errors: 10% Base	10% Pdf	10% Detection Power	10% Detection Error	20% Base	20% Pdf	20% Detection Power	20% Detection Error	30% Base	30% Pdf	30% Detection Power	30% Detection Error	40% Base	40% Pdf	40% Detection Power	40% Detection Error	50% Base	50% Pdf	50% Detection Power	50% Detection Error
Uniform	1	**1**	**0.9917**	0.07	0.00	**1**	**0.9875**	0.49	0.00	**1**	**0.9805**	0.98	0.00	**1**	**0.9710**	1.66	0.00	**7**	**0.9588**	98.38	0.00
	2	7	0.9924			7	0.9924			**7**	**0.9890**			7	**0.9832**			**1**	**0.9747**		
	3									14	0.9902			14	0.9863			**14**	**0.9799**		
	4													21	0.9874			21	0.9824		
Normal	1	1	0.9883	0.00	0.00	1	0.9842	0.34	0.00	1	0.9772	0.71	0.00	1	0.9678	1.30	0.00	7	0.9584	98.09	0.00
	2					7	0.9876			7	0.9841			7	0.9786			1	0.9701		
	3									21	0.9842			14	0.9804			14	0.9740		
	4													21	0.9804			21	0.9753		
Lognormal	1	1	0.9826	0.00	0.00	1	0.9790	0.12	0.00	1	0.9716	0.44	0.00	7	0.9661	97.87	0.00	7	0.9576	98.43	0.00
	2					7	0.9802			7	0.9759			1	0.9713			14	0.9642		
	3													14	0.9716			21	0.9670		
Modulo Test		1	1.0000			1	1.0000			1	1.0000			1	1.0000			1	1.0000		
		2	0.4988			2	0.4920			2	0.5058			2	0.5025			2	0.5064		
		3	0.3346			3	0.3310			3	0.3393			3	0.3413			3	0.3421		
		4	0.2455			4	0.2485			4	0.2495			4	0.2471			4	0.2482		
		5	0.2037			5	0.1980			5	0.2031			5	0.1992			5	0.2011		
		6	0.1669			6	0.1617			6	0.1734			6	0.1732			6	0.1767		
		7	**0.2297**			**7**	**0.3208**			**7**	**0.3999**			**7**	**0.4846**			**7**	**0.5720**		
		8	0.1261			8	0.1252			8	0.1296			8	0.1278			8	0.1270		
		9	0.1110			9	0.1067			9	0.1111			9	0.1106			9	0.1116		
		10	0.1026			10	0.0980			10	0.1040			10	0.1027			10	0.1051		
		11	0.0858			11	0.0912			11	0.0854			11	0.0858			11	0.0854		
		12	0.0822			12	0.0802			12	0.0861			12	0.0848			12	0.0847		
		13	0.0779			13	0.0767			13	0.0793			13	0.0792			13	0.0795		
		14	**0.1149**			**14**	**0.1610**			**14**	**0.2046**			**14**	**0.2456**			**14**	**0.2925**		
		15	0.0674			15	0.0671			15	0.0680			15	0.0667			15	0.0664		
		16	0.0613			16	0.0651			16	0.0627			16	0.0613			16	0.0618		
		17	0.0580			17	0.0622			17	0.0593			17	0.0586			17	0.0580		
		18	0.0545			18	0.0526			18	0.0559			18	0.0557			18	0.0571		
		19	0.0505			19	0.0512			19	0.0509			19	0.0513			19	0.0506		
		20	0.0452			20	0.0493			20	0.0472			20	0.0458			20	0.0474		
		21	**0.0775**			**21**	**0.1072**			**21**	**0.1373**			**21**	**0.1668**			**21**	**0.1967**		
		22	0.0432			22	0.0464			22	0.0439			22	0.0439			22	0.0437		
		23	0.0419			23	0.0463			23	0.0415			23	0.0425			23	0.0417		
		24	0.0420			24	0.0369			24	0.0437			24	0.0429			24	0.0428		
		25	0.0408			25	0.0392			25	0.0413			25	0.0406			25	0.0408		

Lognormal Best-Fit Distribution

Lognormal best fit even performs worse, as indicated by its inability to detect rounding base 14 in data set C, and rounding base 21 in data set D. The order of rounding base detection is also different with uniform and normal best fits results. This is true especially in data sets D and E. In data set D, uniform best fit detects rounding base 1 in the first iteration, followed by rounding bases 7, 14, and 21 in the next iterations, while lognormal detects base 7, followed by bases 1 and 14.

In data set E, uniform best fit detects rounding base 7 in the first iteration, followed by bases 14 and 21 in the following iterations, whereas the lognormal detection order is base 7, followed by bases 1, 14, and 21. Therefore, using incorrect

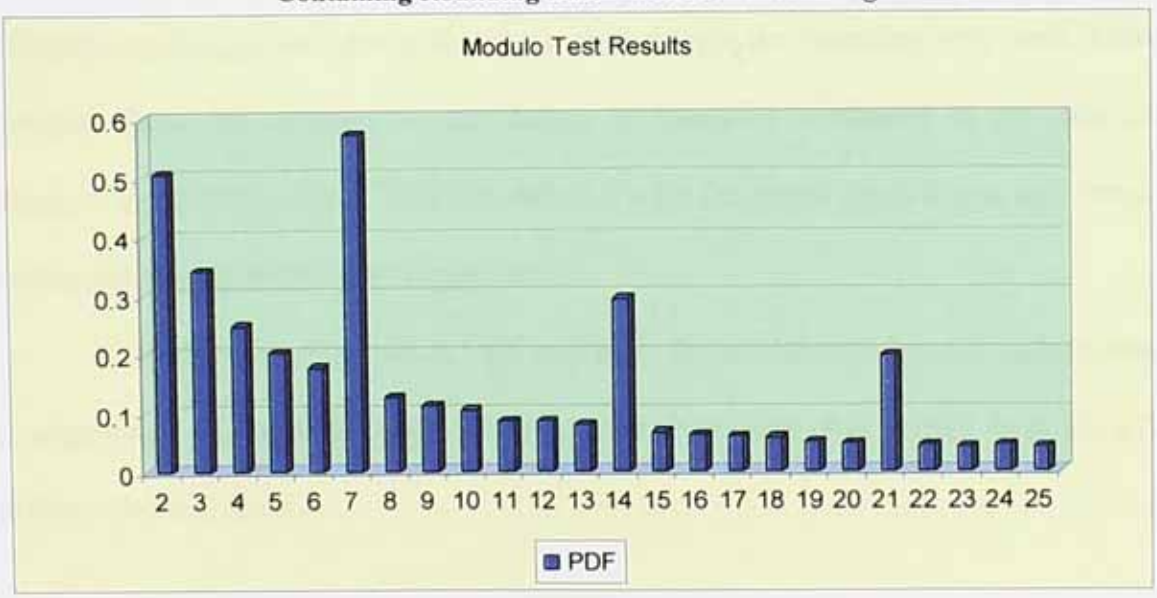
best fit can also produce different detection order and non-detection of some rounding bases. This is in addition to the problems of lower probability values mentioned before.

Modulo Test Method

The existence of rounding base 7 in the data sets is further confirmed by the Modulo test results, which show increasing probability values on the rounding base 7 and its multiples of bases 14 and 21. The probability values of these rounding bases keep increasing as the rounding level increases from 10% to 50%. For instance, the rounding base 7 probability values in data set A is 0.2297, while it increases to 0.572 in data set E. This increasing probability values are also observed in the multiples of base 7, such as bases 14 and 21.

Figure 6.5 shows the probability values of detecting rounding base 7 using the Modulo test on a data set containing rounding base 7 with 50% rounding. The graph shows the presence of rounding base 7 as indicated by the spikes in the probability values of rounding base 7 and its multiples. The probability values of rounding base 7 is higher than its multiples, i.e., 0.572 for rounding bases 7 to 0.1967 for base 21.

Figure 6.5. Modulo Test Results of Detecting Rounding Base 7 on a Simulated Data Set Containing Rounding Base 7 with 50% Rounding



Moreover, to compare the model's robustness in detecting rounding base 7 across different rounding levels and best fits used, Table 6.3 also summarises the detection power and detection error for all cases. The results can be summed up as follows:

(i) There is no detection of rounding base 7 in data set A using normal and lognormal best fits;

(ii) In data sets A to C, uniform best fit produces the highest detection power, followed by normal and lognormal best fits. For data sets D and E, however, lognormal best fit produces the highest detection power.

(iii) The detection power jumps significantly in data set D using lognormal best fit and in data set E using all the three best fits—i.e., from a maximum of 2% to more than 97%. This is because the model detects rounding base 7 in their first iteration on all those data sets.

(iv) There is no detection error identified from the results.

6.3.3. Detecting Rounding Base 11

Detecting rounding base 11 in uniform data containing rounding base 11 with different rounding levels shows that the model detects the rounding very well. More rounding bases are detected as the degree of rounding contained in the data set increases from 10% to 50%. This is consistent with the initial expectation, and hence, confirming the goodness fit of the model.

As in the case of previous base numbers, the model can detect rounding base 11 regardless of the best-fit distribution used, but using the correct best fit will produce the best result.

Uniform best fit detects rounding base 11 in the second iteration in data sets A to E. Normal best fit will produce similar results but with less probability values, while the lognormal best fit will detect rounding base 11 only in data sets B to E.

Moreover, Figure 6.6 shows the probability values of detecting rounding base 11 in the uniform data containing 30% rounding using the three best fits. Uniform best fit produces the highest probability values in all rounding bases detected, followed by normal and lognormal best-fit results. The probability value of rounding base 11 using uniform best fit is 0.9733, while the probability value of using lognormal best fit is 0.9644.

Figure 6.6. Probability Values of Detecting Rounding Base 11 in a Simulated Data Set Containing Rounding Base 11 of 30% Rounding

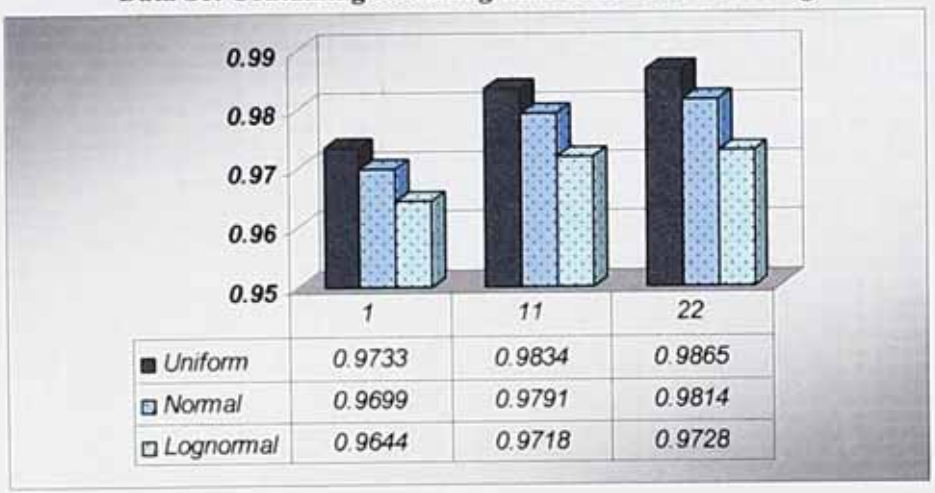


Table 6.4 further summarises the rounding base 11 detection results in the uniform data containing rounding base 11 of different magnitudes by using the three best fits.

Uniform Best-Fit Distribution

Detecting rounding base 11 in the uniform data using uniform best fit shows that the model detects rounding base 11 in the second iteration in data sets A to E. This very good performance cannot be obtained by using other best-fit distributions.

Using normal best fit will produce similar results but with less probability values, while using lognormal best fit will only detect rounding base 11 in data sets B to E.

Normal Best-Fit Distribution

Detecting rounding base 11 using normal best fits on data sets A to D, will produce results similar to using uniform best fit—only with lower probability values. In data set E, normal best fit detects rounding base 11 in the first iteration and not in the second iteration as in the uniform best fit, but still with lower probability value. Bases 1 and 22 then follow the detection.

Table 6.4. Results of Detecting Rounding Base 11 Contained in the Uniformly Distributed Simulated Data Sets Using Three Different Best-Fit Distribution Functions

Best Fit Distribution	Iteration	Magnitude of Rounding Errors: 10% Base	10% Pdf	10% Detection Power	10% Detection Error	20% Base	20% Pdf	20% Detection Power	20% Detection Error	30% Base	30% Pdf	30% Detection Power	30% Detection Error	40% Base	40% Pdf	40% Detection Power	40% Detection Error	50% Base	50% Pdf	50% Detection Power	50% Detection Error
Uniform	1	1	0.9911	0.19	0.00	1	0.9844	0.72	0.00	1	0.9733	1.34	0.00	1	0.9573	1.95	0.00	1	0.9375	2.57	0.00
	2	11	0.993			11	0.9904			11	0.9834			11	0.9713			11	0.9553		
	3					22	0.9916			22	0.9865			22	0.9763			22	0.9622		
Normal	1	1	0.9877	0.09	0.00	1	0.981	0.55	0.00	1	0.9699	1.17	0.00	1	0.954	1.77	0.00	11	0.937	98.03	0.00
	2	11	0.9886			11	0.9861			11	0.9791			11	0.9671			1	0.9511		
	3					22	0.9865			22	0.9814			22	0.9713			22	0.9573		
Lognormal	1	1	0.982	0.00	0.00	1	0.9754	0.35	0.00	1	0.9644	0.85	0.00	11	0.952	97.81	0.00	11	0.9363	97.82	0.00
	2					11	0.9789			11	0.9718			1	0.9599			1	0.9441		
	3									22	0.9729			22	0.9628			22	0.949		
Modulo Test		1	1			1	1			1	1			1	1			1	1		
		2	0.5018			2	0.5052			2	0.5039			2	0.4996			2	0.4997		
		3	0.3323			3	0.333			3	0.3344			3	0.3383			3	0.3384		
		4	0.2498			4	0.2526			4	0.251			4	0.2483			4	0.2471		
		5	0.2018			5	0.1994			5	0.1995			5	0.1962			5	0.1988		
		6	0.1654			6	0.1665			6	0.1686			6	0.1696			6	0.1713		
		7	0.1444			7	0.1465			7	0.1427			7	0.1422			7	0.1444		
		8	0.1264			8	0.1274			8	0.1259			8	0.1242			8	0.1214		
		9	0.1102			9	0.1099			9	0.1106			9	0.112			9	0.1121		
		10	0.101			10	0.0997			10	0.0984			10	0.0964			10	0.0983		
		11	**0.1772**			**11**	**0.2587**			**11**	**0.3597**			**11**	**0.4514**			**11**	**0.5422**		
		12	0.0827			12	0.0827			12	0.0837			12	0.0836			12	0.0836		
		13	0.0781			13	0.0786			13	0.0775			13	0.078			13	0.0785		
		14	0.0737			14	0.0759			14	0.0748			14	0.0732			14	0.0741		
		15	0.065			15	0.0655			15	0.0658			15	0.0652			15	0.066		
		16	0.0626			16	0.0625			16	0.061			16	0.0603			16	0.0588		
		17	0.0589			17	0.0596			17	0.0595			17	0.059			17	0.0594		
		18	0.0549			18	0.0552			18	0.0563			18	0.0582			18	0.0588		
		19	0.0523			19	0.0519			19	0.0537			19	0.0522			19	0.0518		
		20	0.0459			20	0.0456			20	0.0442			20	0.0429			20	0.0429		
		21	0.0489			21	0.048			21	0.0462			21	0.0469			21	0.0474		
		22	**0.0906**			**22**	**0.1381**			**22**	**0.1611**			**22**	**0.2253**			**22**	**0.2701**		
		23	0.0405			23	0.0413			23	0.0425			23	0.0441			23	0.044		
		24	0.0414			24	0.041			24	0.0413			24	0.0414			24	0.0408		
		25	0.0399			25	0.0392			25	0.0401			25	0.039			25	0.04		

Lognormal Best-Fit Distribution

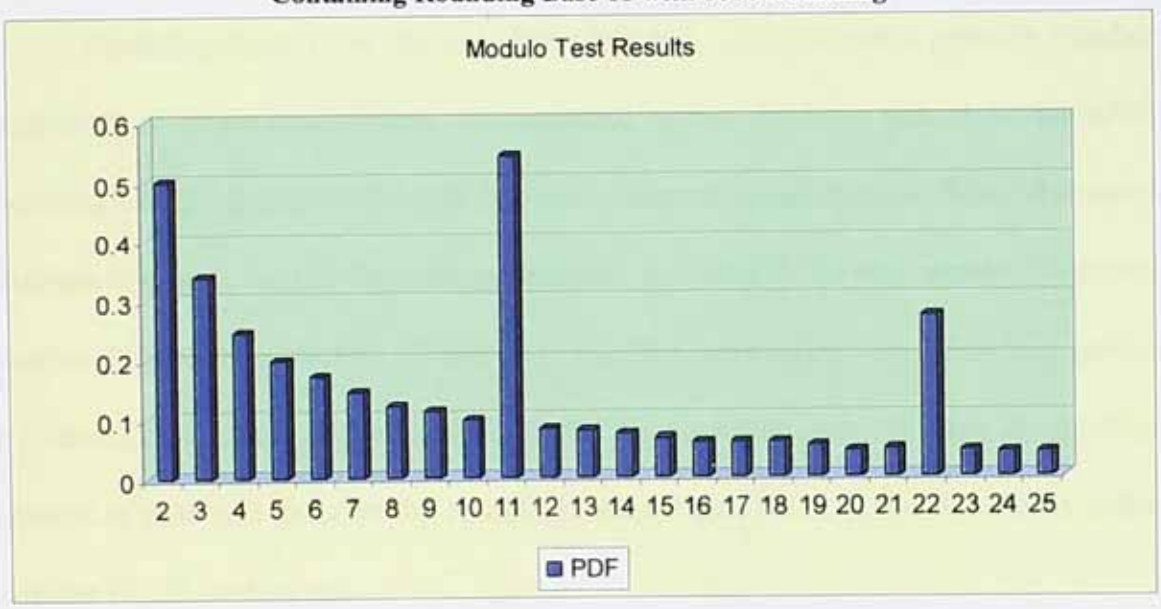
Lognormal best fit detects rounding base 11 in data sets B to E. However, no rounding base 22 is detected in data set B. These results are different with uniform and normal best-fit results. In data set D, for instance, the lognormal best fit detects rounding base 11 in the first iteration despite a lower probability value, and not in the

second iterations like in the uniform and normal best-fit results. While in data set E, the order of rounding base detection using lognormal is the same with normal best fit

Modulo Test

The Modulo test results also show increasing probabilities in the rounding base 11 and its multiple of base 22 that is consistent with the model results. This shows that the model can detect rounding base 11. Figure 6.7 shows the strikingly higher probability values of rounding bases 11 and its multiple of rounding base 22. The probability value of rounding base 11 is 0.5422 while the probability value of rounding base 22 is 0.2701.

Figure 6.7. Modulo Test Results of Detecting Rounding Base 11 in a Simulated Data Set Containing Rounding Base 11 with 50% Rounding



Furthermore, to compare the robustness in detecting rounding base 7 across different rounding levels and best fits used, Table 6.4 also presents the detection power and detection error for all cases. The results can be summarised as follows:

(i) There is no detection of rounding base 11 in data set A using lognormal best fit.

(ii) Uniform best fit produces the highest detection power, followed by normal and lognormal in data sets A to C.

(iii) The detection power in data sets A to D using uniform and normal best fits increases gradually following a smooth trend while in the model using lognormal best fit only happened in the data sets A to C. In data set E, the model detects rounding base 11 in the first iteration for all the three best fits, which makes the detection power jump very high. This happened to data set D using lognormal best fit too.

(iv) No detection error is identified from the results.

6.3.4. Detecting Rounding Base 10

Rounding base 10 is the only base number, which is not a primary number, experimented in the assessments. As discussed before, the main reason for including rounding base 10 is that it could be the most common rounding base. Base 10 is also a multiple of base 5, one of the primary numbers included in the assessment. Therefore, number 10 has two factors, 2 and 5. This fact may affect the rounding pattern detection since a data set severely distorted by rounding base 10 may also exhibit patterns of data sets distorted by rounding bases 5 and 2 due to the significant spikes in bases 10, 20, and so on.

Results from implementing the model on uniform data containing rounding base 10 with different rounding magnitudes show that the model can detect the rounding very well. The rounding base 10 detection shows an increasing intensity, which is shown by the detection of rounding base 10 and other relevant rounding bases as the degree of rounding the data set increases from 10% to 50%. This result is

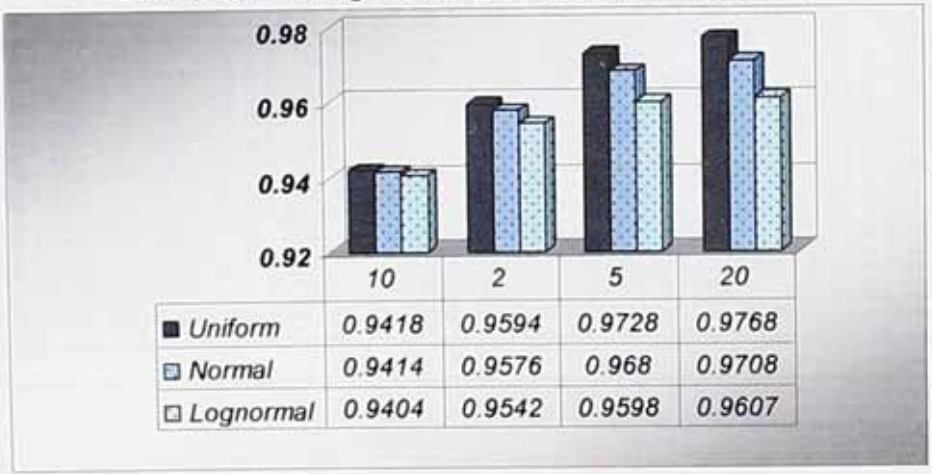
consistent with the overall results and model behaviour, confirming the goodness fit of the model.

Furthermore, although the model can detect the existence of rounding base 10 regardless of the best-fit distributions used, the use of a correct best fit for the data set being examined will guarantee the best result.

The existence of rounding base 10 is detected in the second iteration in data sets A to E if the uniform or normal best fits are used, while if the lognormal best fit is used, the model can only detect the rounding base 10 in data sets B to E.

Moreover, Figure 6.8 shows the probability values of detecting rounding base 10 in the uniform data containing 50% rounding using the three best fits. As can be seen from the graph, the use of uniform best fit produces the highest probability values for all rounding bases detected, followed by normal and lognormal best-fit results. The probability value of detecting rounding base 10 using uniform best fit on data set E is 0.9418 while if using lognormal best fit, the probability value is 0.9404. For rounding base 20, the probability values are 0.9678 and 0.9607, respectively.

Figure 6.8. Probability Values of Detecting Rounding Base 10 in a Simulated Data Set Containing Rounding Base 10 at 50% Rounding



	10	2	5	20
Uniform	0.9418	0.9594	0.9728	0.9768
Normal	0.9414	0.9576	0.968	0.9708
Lognormal	0.9404	0.9542	0.9598	0.9607

Table 6.5 summarises the results of detecting rounding base 10 on the uniform data by using the three best fits. The detailed detection results are discussed in turn as follows:

Uniform Best-Fit Distribution

Detecting rounding base 10 in the uniform data containing rounding base 10 by using uniform best fit, shows that the model detects rounding base 10 in the second iteration in data sets A to D and in the first iteration in data set E. The model always detects rounding base 1 in the first iteration in data sets A to D. The model also detects rounding base 5 in data sets D and E, and rounding base 2 in data set E.

Table 6.5. Results of Detecting Rounding Base 10 Contained in the Uniformly Distributed Simulated Data Sets Using Three Different Best-Fit Distribution Functions

Best Fit Distribution	Iteration	Magnitude of Rounding Errors: 10% Base	10% Pdf	10% Detection Power	10% Detection Error	20% Base	20% Pdf	20% Detection Power	20% Detection Error	30% Base	30% Pdf	30% Detection Power	30% Detection Error	40% Base	40% Pdf	40% Detection Power	40% Detection Error	50% Base	50% Pdf	50% Detection Power	50% Detection Error
Uniform	1	**1**	**0.991**	0.19	0.00	**1**	**0.9846**	0.84	0.00	**1**	**0.9742**	1.26	0.00	**1**	**0.9599**	1.63	0.55	**10**	**0.9418**	96.83	3.17
	2	10	0.9929			**10**	**0.9906**			**10**	**0.9842**			**10**	**0.9738**			**2**	**0.9594**		
	3					20	0.991			20	0.9866			**5**	**0.9792**			**5**	**0.9728**		
	4													20	0.9813			20	0.9768		
Normal	1	**1**	**0.9876**	0.09	0.00	**1**	**0.9812**	0.46	0.00	**1**	**0.9708**	1.07	0.00	**1**	**0.9566**	1.67	0.02	**10**	**0.9414**	96.66	2.72
	2	10	0.9885			**10**	**0.9863**			**10**	**0.9799**			**10**	**0.9695**			**2**	**0.9576**		
	3					20	0.9858			20	0.9814			20	0.973			**5**	**0.966**		
	4													15	0.9732			20	0.9708		
Lognormal	1	**1**	**0.9819**	0.00	0.00	**1**	**0.9756**	0.14	0.00	**2**	**0.9653**	0.10	97.84	**10**	**0.9546**	97.28	1.40	**10**	**0.9404**	96.37	1.99
	2					**10**	**0.9786**			**5**	**0.9723**			**5**	**0.965**			**2**	**0.9542**		
	3					20	0.977			10	0.9733			2	0.9685			**5**	**0.9598**		
	4																	20	0.9607		
Modulo Test		1	1			1	1			1	1			1	1			1	1		
		2	0.5459			2	0.6022			2	0.6512			2	0.6957			2	0.7517		
		3	0.3314			3	0.3341			3	0.3286			3	0.3367			3	0.3428		
		4	0.271			4	0.2935			4	0.3244			**4**	**0.3441**			**4**	**0.3649**		
		5	**0.2527**			**5**	**0.363**			**5**	**0.4392**			**5**	**0.5293**			**5**	**0.6008**		
		6	0.1778			6	0.1981			6	0.2126			6	0.2356			6	0.2589		
		7	0.1394			7	0.1456			7	0.1482			7	0.1412			7	0.1454		
		8	0.137			8	0.1505			8	**0.1653**			**8**	**0.1734**			**8**	**0.1817**		
		9	0.1089			9	0.1109			9	0.1099			9	0.112			9	0.1143		
		10	**0.1928**			10	0.2818			10	0.3693			10	0.4668			10	0.5628		
		11	0.0863			11	0.0842			11	0.0847			11	0.0898			11	0.0827		
		12	0.0683			**12**	**0.0687**			12	**0.1068**			**12**	**0.1165**			**12**	**0.1263**		
		13	0.0742			13	0.0764			13	0.077			13	0.0788			13	0.076		
		14	0.0745			**14**	**0.0892**			**14**	**0.0977**			**14**	**0.0985**			**14**	**0.1112**		
		15	**0.0917**			**15**	**0.119**			**15**	**0.1477**			**15**	**0.1753**			**15**	**0.2057**		
		16	0.0677			16	0.073			16	0.0815			16	0.0866			16	0.0859		
		17	0.0588			17	0.0611			17	0.0577			17	0.0579			17	0.0611		
		18	0.0583			18	0.0659			**18**	**0.0714**			**18**	**0.0794**			**18**	**0.0866**		
		19	0.0545			19	0.0508			19	0.0502			19	0.0526			19	0.0499		
		20	**0.0926**			**20**	**0.1324**			**20**	**0.1823**			**20**	**0.2314**			**20**	**0.2835**		
		21	0.0425			21	0.0494			21	0.0486			21	0.0476			21	0.0512		
		22	0.0471			22	0.0506			22	0.06			**22**	**0.0631**			**22**	**0.0626**		
		23	0.0437			23	0.0408			23	0.0404			23	0.0404			23	0.0424		
		24	0.0453			**24**	**0.0495**			24	0.0563			**24**	**0.0605**			**24**	**0.0636**		
		25	**0.0584**			**25**	**0.0711**			**25**	**0.0903**			**25**	**0.1072**			**25**	**0.1204**		

Normal Best-Fit Distribution

Using normal best fit produces similar results to using uniform best fit. The difference is only in data set D, in which rounding base 5 is not detected and the

model detects rounding base 15 instead. In the remaining data sets, the results are the same with the uniform best-fit results, only with lower probability values.

Lognormal Best-Fit Distribution

Results of using lognormal best fit are more different. Different such that (i) rounding base 10 was not detected in data sets C to E; (ii) no rounding base 20 was detected in data set A; (iii) rounding base 2 was detected in data sets C to E, while rounding base 2 was detected only in data set E in other best-fit results.

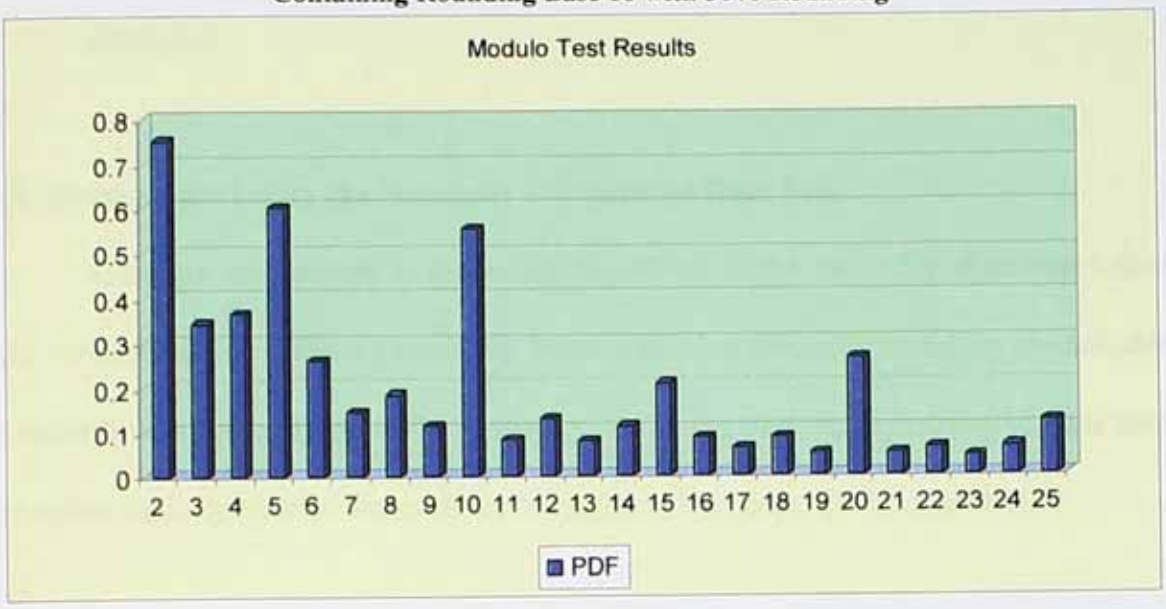
Notice that the pattern of rounding base detection in the data set containing rounding base 10 is not as clear as in the data sets containing primary base numbers such as bases 5, 7, and 11. As can be seen in the results, as the rounding magnitude of base 10 in the data sets intensifies, the model starts to mistakenly detect the presence of rounding bases 5 and 2. All three best fits used in this study suffer from this problem, including the Modulo test method that will be discussed next. The problem of mis-identification of rounding base 10 with its factor numbers will worsen if it is combined by using the wrong best fits.

Modulo Test

The Modulo test result also shows increasing probabilities in the rounding base 10 and its multiples of base 20 that is consistent with the model results. This shows the model's consistency in detecting rounding base 10.

Figure 6.9 shows the higher probability values, especially in the rounding bases 10 and 20. Note also the high probability values of bases 2 and 5, as well as of the multiples of base 5 (such as bases 15 and 25). This shows the distorting power of rounding, which becomes obvious if the rounding contained in the data sets is more than 30%.

Figure 6.9.Modulo Test Results of Detecting Rounding Base 10 on a Simulated Data Set Containing Rounding Base 10 with 50% Rounding



To compare the robustness of detecting rounding base 10 across different rounding levels and best fits used, Table 6.5 also details the detection power and detection error for all cases. The results can be summarised as follows:

(i) There is no detection of rounding base 10 on data set A using lognormal best fit.

(ii) Uniform best fit produces the highest detection power, followed by normal and lognormal. This finding applies in all data sets A to E;

(iii) The detection power in data sets A to D increases gradually following a smooth trend, except that it jumps in data set D using lognormal best fit.

(iv) Detection errors are identified from the results in data set C using lognormal best fit, and data sets D and E using all best fits. The detection error in data set C using lognormal best fit is very significant at 97.8% due to the detection of rounding bases 2 and 5 in the first and second iterations. In the other cases, the detection errors range from 0.02% to 3.17%. These are also because of detection of rounding bases 2 and 5 (the factor numbers of base 10). In data set D, lognormal best fit produces the highest detection error at

1.40%, while uniform best fit produces the highest detection error of 3.17% in data set E.

6.4. Assessments Using the Normally Distributed Data Sets

The next assessment is implementing NNM in the normally distributed data sets containing four different rounding bases with five different rounding magnitudes as described in the early part of this chapter. As in the uniformly distributed data sets, the detection is conducted for each rounding base, i.e., bases 5, 7, 11, and 10.

6.4.1. Detecting Rounding Base 5

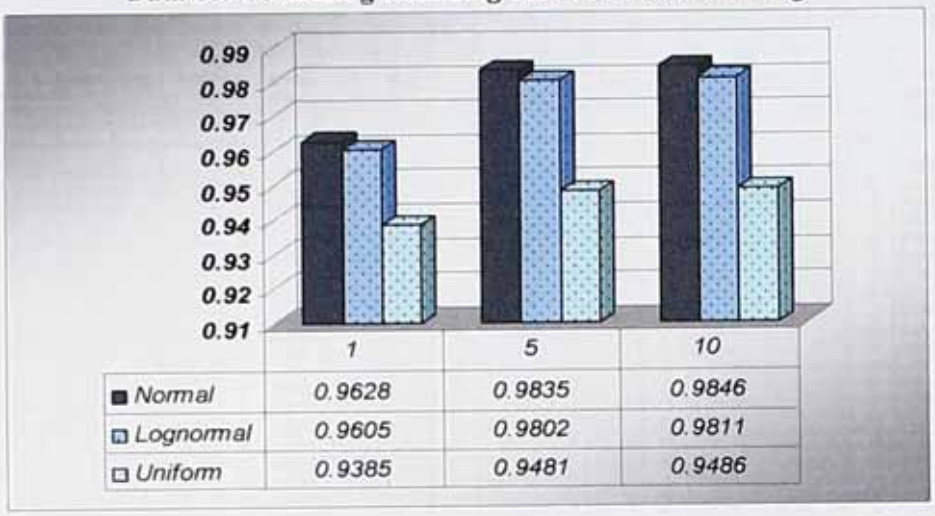
Implementing NNM on normal data containing rounding base 5 with different rounding magnitudes shows that the model detects rounding base 5 very well. The detection probability increases as the rounding level increases from 10% to 50%. This is consistent with the initial expectation on the overall results, hence confirming the goodness fit of NNM.

The model detects the rounding bases in normal data very well, regardless of the best fits used in the detection. However, using the correct best fit of the data set being examined will guarantee the best result. This is shown by most rounding bases detection with the highest probability values.

Figure 6.10, for instance, shows the probability values of detecting rounding base 5 in normally distributed simulated data set containing rounding base 5 at 40% rounding level by using the three best fits. The figure shows that using the correct best fit will detect the rounding bases with the highest probability and using the wrong best fit will result in lower probability. In other words, the normal best fit detects the rounding bases with the highest probability values. The probability value of detecting

rounding base 5 using normal best fit is 0.9835, while the probability value becomes 0.9481 if uniform best fit is used. For rounding base 10, the probability values are 0.9846 and 0.9486, respectively.

Figure 6.10. Probability Values of Detecting Rounding Base 5 in a Simulated Data Set Containing Rounding base 5 at 40% Rounding



	1	5	10
Normal	0.9628	0.9835	0.9846
Lognormal	0.9605	0.9802	0.9811
Uniform	0.9385	0.9481	0.9486

Furthermore, Table 6.6 summarises the results of detecting rounding base 5 contained in the normal data by using the three best fits. The detailed detection results are discussed in turn as follows:

Normal Best-Fit Distribution

Detecting rounding base 5 in the normally distributed data set using normal best-fit distribution shows that the model detects rounding base 1 in the first iteration in data sets A to D, rounding base 5 in the second iteration in data sets B to D, and in the first iteration in data set E. Rounding base 10 is detected in the third iteration in data sets D and E. In data set E, after detecting rounding base 5 in the first iteration, the model detects rounding bases 1, 10, and 15 in the second, third, and fourth iterations.

Table 6.6. Results of Detecting Rounding Base 5 Contained in the Normally Distributed Simulated Data Sets Using Three Different Best-Fit Distribution Functions

Best Fit Distribution	Iteration	Magnitude of Rounding Errors																	
		10%				20%				30%				40%				50%			
		Base	Pdf	Detection Power	Detection Error	Base	Pdf	Detection Power	Detection Error	Base	Pdf	Detection Power	Detection Error	Base	Pdf	Detection Power	Detection Error	Base	Pdf	Detection Power	Detection Error
Normal	1	1	0.9915	0.00	0.00	1	0.9858	0.57	0.00	1	0.9758	1.34	0.00	1	0.9628	2.18	0.00	5	0.9459	95.12	0.00
	2					5	0.9915			5	0.9891			5	0.9835			1	0.9738		
	3													10	0.9846			10	0.9790		
	4																	15	0.9791		
Lognormal	1	1	0.9884	0.00	0.00	1	0.9833	0.56	0.00	1	0.9735	1.27	0.00	1	0.9605	2.06	0.00	1	0.9429	3.20	0.00
	2					5	0.9889			5	0.9861			5	0.9802			5	0.9694		
	3													10	0.9811			10	0.9745		
	4																	15	0.9749		
Uniform	1	1	0.9672	0.00	0.00	1	0.9604	0.26	0.00	1	0.9509	0.62	0.00	1	0.9385	1.01	0.00	1	0.9220	1.50	0.00
	2					5	0.9630			5	0.9570			5	0.9481			5	0.9348		
	3													10	0.9486			10	0.9370		
Modulo Test		1	1.0000			1	1.0000			1	1.0000			1	1.0000			1	1.0000		
		2	0.4945			2	0.5060			2	0.4983			2	0.4921			2	0.5006		
		3	0.3315			3	0.3277			3	0.3297			3	0.3366			3	0.3325		
		4	0.2449			4	0.2523			4	0.2482			4	0.2436			4	0.2518		
		5	0.2827			5	0.3604			5	0.4392			5	0.5293			5	0.5996		
		6	0.1608			6	0.1648			6	0.1615			6	0.1679			6	0.1695		
		7	0.1386			7	0.1426			7	0.1476			7	0.1469			7	0.1484		
		8	0.1243			8	0.1227			8	0.1279			8	0.1213			8	0.1243		
		9	0.1084			9	0.1092			9	0.1106			9	0.1126			9	0.1086		
		10	0.1415			10	0.1812			10	0.2165			10	0.2633			10	0.3004		
		11	0.0862			11	0.0921			11	0.0905			11	0.0912			11	0.0947		
		12	0.0795			12	0.0816			12	0.0817			12	0.0819			12	0.0800		
		13	0.0744			13	0.0780			13	0.0770			13	0.0815			13	0.0753		
		14	0.0671			14	0.0716			14	0.0764			14	0.0710			14	0.0732		
		15	0.0918			15	0.1167			15	0.1488			15	0.1752			15	0.1984		
		16	0.0611			16	0.0587			16	0.0627			16	0.0611			16	0.0634		
		17	0.0583			17	0.0590			17	0.0576			17	0.0584			17	0.0605		
		18	0.0531			18	0.0555			18	0.0554			18	0.0578			18	0.0578		
		19	0.0539			19	0.0525			19	0.0509			19	0.0521			19	0.0520		
		20	0.0664			20	0.0910			20	0.1060			20	0.1308			20	0.1511		
		21	0.0421			21	0.0477			21	0.0472			21	0.0498			21	0.0502		
		22	0.0421			22	0.0474			22	0.0453			22	0.0457			22	0.0477		
		23	0.0438			23	0.0404			23	0.0430			23	0.0414			23	0.0427		
		24	0.0408			24	0.0393			24	0.0439			24	0.0415			24	0.0373		
		25	0.0583			25	0.0681			25	0.0884			25	0.1051			25	0.1174		

Lognormal Best-Fit Distribution

Results using lognormal best fit are similar to the ones using normal distribution. The differences are: (i) lower probability values and (ii) rounding base 5 is detected in the second iteration and not in the first iteration in data set E. On the first point, using normal best fit in data set C, for instance, the model detects rounding base 5 with the probability value of 0.9891, while the probability value will be decreased to 0.9861 if the lognormal best fit is used.

Uniform Best-Fit Distribution

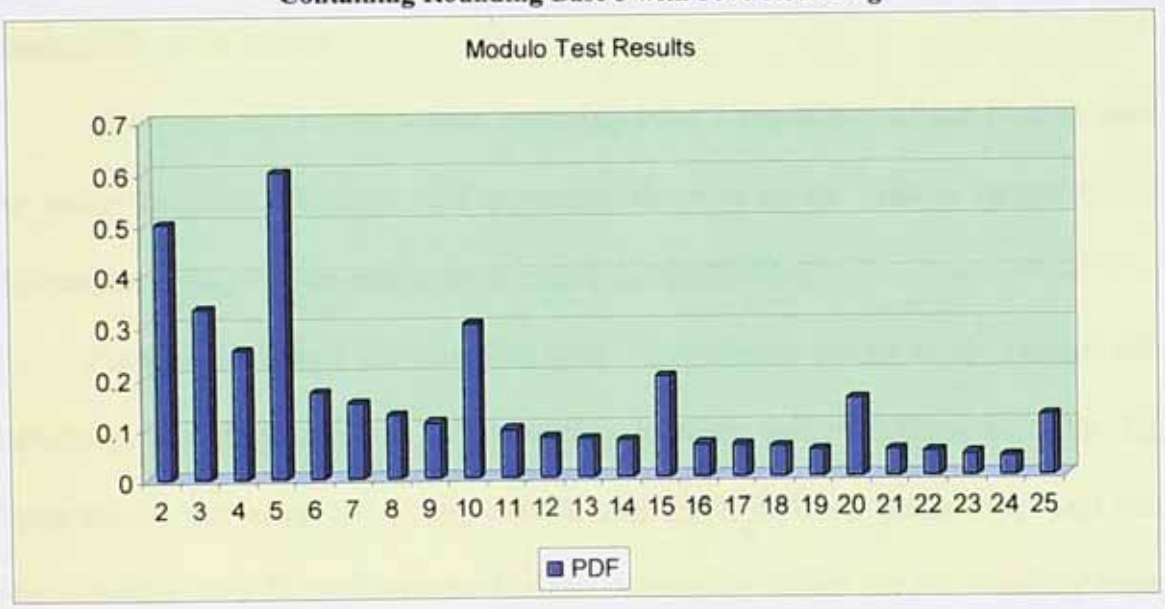
If the uniform best fit were used, the results would be the same in data sets A to D, but with lower probability values. In data set E, there are two different results.

First, rounding base 5 is detected in the second iteration. Second, no rounding base 15 is detected in the fourth iteration.

Modulo Test

The existence of rounding base 5 is also shown in the Modulo test results. Figure 6.11 shows increasing probabilities in the rounding base 5 and its multiples such as bases 10, 15, 20, and 25. This is consistent with the model results, confirming the model's consistency in detecting rounding base 5. Moreover, the rounding base 5 probability value is always higher than those of its multiples. In data see E, for instance, the rounding base 5 probability value is 0.5996, while the probability value in base 25 is only 0.1174.

Figure 6.11. Modulo Test Results of Detecting Rounding Base 5 on a Simulated Data Set Containing Rounding Base 5 with 50% Rounding



To compare the robustness of detecting rounding base 5 across different rounding levels and best fits used, Table 6.6 also presents the detection power and detection error for all cases. The results can be summarised as follows:

(i) There is no detection of rounding base 5 in data set A using all best fits.

(ii) Normal best fit produces the highest detection power, followed by lognormal and uniform. This finding applies in all data sets A to E;

(iii) The detection power in data sets A to E increase gradually following a smooth trend, except in data set E using normal best fit in which the detection power jumps from 2.18% in data set D to 95.12% in data set E. This is because of the detection of rounding base 5 in the first iteration.

(iv) No detection error is identified from the results.

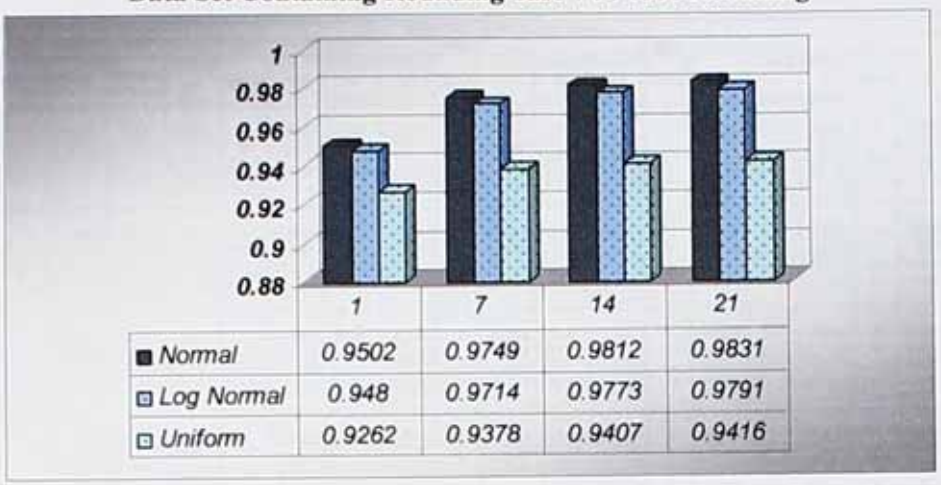
6.4.2. Detecting Rounding Base 7

Implementation in the normal data containing rounding base 7 with different rounding levels shows that the model can detect the rounding very well. The power of detection increases as the rounding level contained in the data set increases from 10% to 50%. This result is consistent with the initial expectation, hence confirming the goodness fit of the model.

Moreover, the model detects rounding base 7 regardless of the best fit used, but using the correct best fit will guarantee the best result. This is shown by the highest probability values and more rounding bases detected.

Figure 6.12 shows the rounding base 7 probability values in the normal data containing rounding base 7 at 50% rounding level by using the three best fits. The figure shows that using the wrong best fit will result in lower probability, and vice versa. Normal best fit produces the highest probability values for all rounding bases detected. The normal best-fit probability value of rounding base 7 is 0.9749 while the probability value of uniform result is 0.9378. For rounding base 14, the probability values are 0.9812 and 0.9407, respectively.

Figure 6.12. Probability Values of Detecting Rounding Base 7 in a Simulated Data Set Containing Rounding base 7 at 40% Rounding



	1	7	14	21
Normal	0.9502	0.9749	0.9812	0.9831
Log Normal	0.948	0.9714	0.9773	0.9791
Uniform	0.9262	0.9378	0.9407	0.9416

Moreover, Table 6.7 summarises the results of detecting rounding base 7 contained in the normally distributed simulated data sets by using three different best fits. The detailed detection results are summarised as follows:

Normal Best-Fit Distribution

Detecting rounding base 7 using normal best fit in the normal data shows that the model detects rounding base 1 in the first iteration in data sets A to D, rounding base 7 in the second iteration in data sets B to D, and in the first iteration in data set E. This very good performance cannot be obtained by using other best-fit distributions. Other best-fit distributions, for instance, always detect rounding base 7 in the second iteration.

Lognormal Best-Fit Distribution

If the lognormal best fit is used, the model detects rounding base 7 in data sets A to E, rounding base 14 in data sets C to E, and rounding base 21 in data sets D and E.

Table 6.7. Results of Detecting Rounding Base 7 Contained in the Normally Distributed Simulated Data Sets Using Three Different Best-Fit Distribution Functions

Best Fit Distribution	Iteration	Magnitude of Rounding Errors: 10% Base	10% Pdf	10% Detection Power	10% Detection Error	20% Base	20% Pdf	20% Detection Power	20% Detection Error	30% Base	30% Pdf	30% Detection Power	30% Detection Error	40% Base	40% Pdf	40% Detection Power	40% Detection Error	50% Base	50% Pdf	50% Detection Power	50% Detection Error
Normal	1	1	0.9911	0.00	0.00	1	0.9817	0.98	0.00	1	0.9671	2.08	0.00	1	0.9502	3.35	0.00	7	0.9219	96.66	0.00
	2					7	0.9914			7	0.9849			7	0.9749			1	0.9543		
	3									14	0.9876			14	0.9812			14	0.9640		
	4													21	0.9831			21	0.9688		
Lognormal	1	1	0.9882	0.02	0.00	1	0.9791	0.95	0.00	1	0.9648	1.95	0.00	1	0.9480	3.16	0.00	1	0.9239	4.44	0.00
	2	7	0.9884			7	0.9885			7	0.9816			7	0.9714			7	0.9540		
	3									14	0.9841			14	0.9773			14	0.9625		
	4													21	0.9791			21	0.9669		
Uniform	1	1	0.9665	0.05	0.00	1	0.9563	0.47	0.00	1	0.9424	0.97	0.00	1	0.9262	1.57	0.00	1	0.8985	2.20	0.00
	2	7	0.9673			7	0.9610			7	0.9508			7	0.9378			7	0.9134		
	3									14	0.9520			14	0.9407			14	0.9177		
	4													21	0.9416			21	0.9198		
Modulo Test		1	1.0000			1	1.0000			1	1.0000			1	1.0000			1	1.0000		
		2	0.4965			2	0.5021			2	0.5036			2	0.5051			2	0.4961		
		3	0.3334			3	0.3250			3	0.3273			3	0.3392			3	0.3334		
		4	0.2437			4	0.2479			4	0.2515			4	0.2519			4	0.2476		
		5	0.2026			5	0.1998			5	0.2003			5	0.2006			5	0.2049		
		6	0.1629			6	0.1607			6	0.1634			6	0.1699			6	0.1640		
		7	0.2250			7	0.3195			7	0.4047			7	0.4826			7	0.5709		
		8	0.1228			8	0.1262			8	0.1269			8	0.1297			8	0.1224		
		9	0.1094			9	0.1055			9	0.1058			9	0.1162			9	0.1075		
		10	0.1029			10	0.1016			10	0.1004			10	0.1024			10	0.1035		
		11	0.0872			11	0.0881			11	0.0911			11	0.0909			11	0.0972		
		12	0.0801			12	0.0804			12	0.0825			12	0.0839			12	0.0779		
		13	0.0737			13	0.0774			13	0.0788			13	0.0775			13	0.0771		
		14	0.1107			14	0.1633			14	0.2065			14	0.2461			14	0.2796		
		15	0.0663			15	0.0671			15	0.0680			15	0.0701			15	0.0672		
		16	0.0610			16	0.0624			16	0.0618			16	0.0628			16	0.0627		
		17	0.0593			17	0.0582			17	0.0589			17	0.0616			17	0.0590		
		18	0.0538			18	0.0513			18	0.0524			18	0.0593			18	0.0546		
		19	0.0544			19	0.0518			19	0.0525			19	0.0527			19	0.0547		
		20	0.0472			20	0.0506			20	0.0508			20	0.0518			20	0.0516		
		21	0.0718			21	0.1031			21	0.1320			21	0.1622			21	0.1918		
		22	0.0427			22	0.0432			22	0.0442			22	0.0446			22	0.0471		
		23	0.0430			23	0.0393			23	0.0382			23	0.0368			23	0.0436		
		24	0.0404			24	0.0425			24	0.0441			24	0.0433			24	0.0362		
		25	0.0412			25	0.0398			25	0.0403			25	0.0407			25	0.0384		

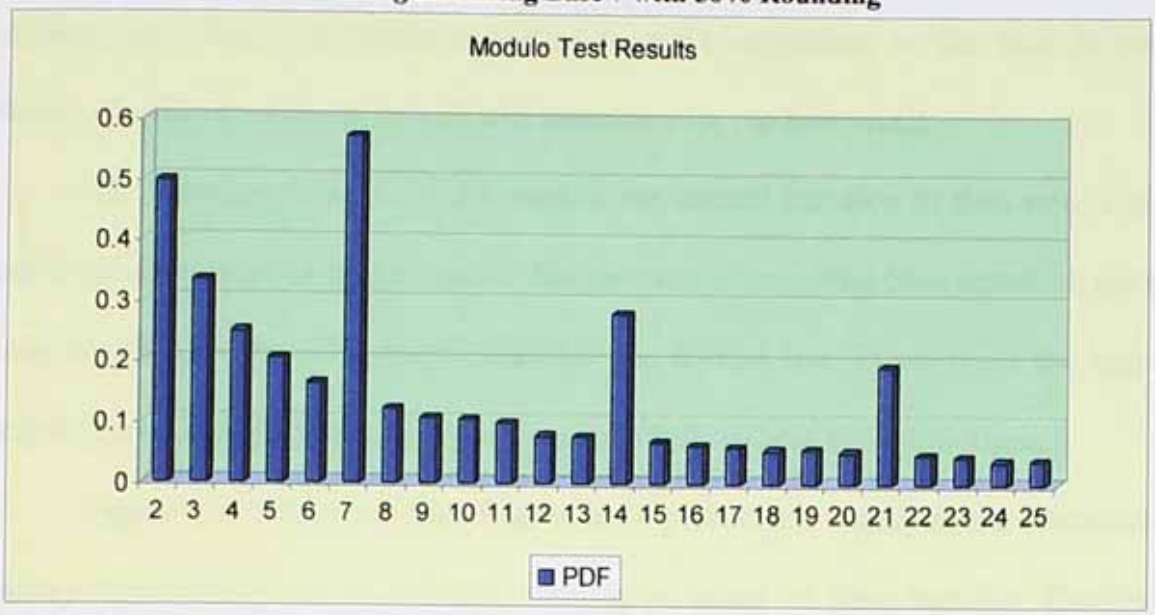
Uniform Best-Fit Distribution

Using uniform best fit will produce similar results to using lognormal best fit, but with lower probability values.

Modulo Test

Applying Modulo test in data sets also shows increasing probabilities in rounding base 7 and its multiples of rounding bases 14 and 21. This is consistent with the NNM results, further confirming that NNM can detect rounding base 7. Moreover, the rounding base 7 probability value is always higher than those of its multiples. In data set E, for instance, the rounding base 7 probability is 0.5709, while the probability value for base 25 is only 0.1918 (Figure 6.13).

Figure 6.13. Modulo Test Results of Detecting Rounding Base 7 in a Simulated Data Set Containing Rounding Base 7 with 50% Rounding



Moreover, to compare the robustness of detecting rounding base 7 across different rounding levels and best fits used, Table 6.7 also summarises the detection power and detection error for all cases. The results can be summed up as follows:

(i) There is no detection of rounding base 7 in data set A using normal best fit.

(ii) Normal best fit produces the highest detection power, followed by lognormal and uniform. The exception is in data set A where normal best fit cannot detect rounding base 7.

(iii) The detection power in data sets A to E increase gradually following a smooth trend.

(iv) No detection error is identified from the results.

6.4.3. Detecting Rounding Base 11

Results from detecting rounding base 11 in the normal data containing rounding base 11 with different rounding levels, show that the model detects the rounding very well. The detection shows an increasing magnitude as the degree of rounding contained in the data set increases from 10% to 50%. This result is

consistent with the initial expectation, confirming the goodness fit of the NNM. In addition, the NNM can detect rounding base 11 regardless of the best fit used. However, using the correct best fit will guarantee for the best result.

The rounding base 11 is detected in the second iteration in data sets A to D and in the first iteration in data set E. The patterns of rounding base detection are the same for all three best fits distribution, but the normal best fit produces the highest probability values, followed by lognormal and uniform best-fit distributions.

Figure 6.14 shows the rounding base 11 probability values on the normal data containing rounding base 11 of 40% rounding by using the three best fits. The figure shows that the right best fit produces the highest probability values, and vice versa. The rounding base 11 probability of normal best fit is 0.9488, while the probability value of uniform best fit is 0.9105. For rounding base 22, the probability values are 0.9591 and 0.9153, respectively.

Figure 6.14. Probability Values of Detecting Rounding Base 11 in a Simulated Data Set Containing Rounding Base 11 at 40% Rounding

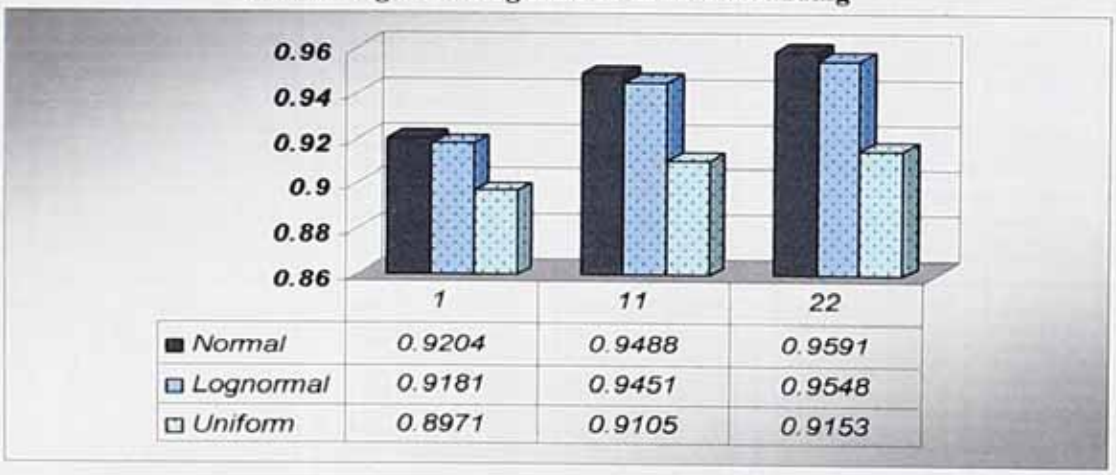


Table 6.8 shows the results of detecting rounding base 11 on the normal data using the three best fits. The table also summarises the Modulo test results. The detailed detection results are discussed in turn as follows:

Normal Best-Fit Distribution

Detecting rounding base 11 using normal best fit shows that the model detects rounding base 11 in the second iteration in data sets A to D, and in the first iteration in data sets E. The model also detects rounding base 22 in the third iteration in data sets B to E. This is a very good performance that cannot be matched by using other best fits.

Lognormal Best-Fit Distribution

The lognormal best-fit results are the same with the normal best-fit results but only with lower detection probability. In data set E, for instance, the rounding base 11 probability of lognormal is 0.8738, which is lower than using normal best fit of 0.8765.

Table 6.8. Results of Detecting Rounding Base 11 Contained in the Normally Distributed Simulated Data Sets Using Three Different Best-Fit Distribution Functions

Best Fit Distribution	Iteration	Magnitude of Rounding Errors: 10% Base	10% Pdf	10% Detection Power	10% Detection Error	20% Base	20% Pdf	20% Detection Power	20% Detection Error	30% Base	30% Pdf	30% Detection Power	30% Detection Error	40% Base	40% Pdf	40% Detection Power	40% Detection Error	50% Base	50% Pdf	50% Detection Power	50% Detection Error
Normal	1	1	0.9899	0.28	0.00	1	0.9742	1.51	0.00	1	0.9497	2.86	0.00	1	0.9204	4.04	0.00	11	0.8765	96.18	0.00
	2	11	0.9927			11	0.9872			11	0.9712			11	0.9488			1	0.9119		
	3					22	0.9893			22	0.9777			22	0.9591			22	0.9268		
Lognormal	1	1	0.9868	0.29	0.00	1	0.9718	1.42	0.00	1	0.9475	2.72	0.00	1	0.9181	3.83	0.00	11	0.8738	95.78	0.00
	2	11	0.9897			11	0.9840			11	0.9678			11	0.9451			1	0.9073		
	3					22	0.9860			22	0.9741			22	0.9548			22	0.9212		
Uniform	1	1	0.9656	0.18	0.00	1	0.9491	0.71	0.00	1	0.9255	1.33	0.00	1	0.8971	1.90	0.00	11	0.8542	92.88	0.00
	2	11	0.9674			11	0.9551			11	0.9354			11	0.9105			1	0.8705		
	3					22	0.9562			22	0.9385			22	0.9153			22	0.8771		
Modulo Test		1	1.0000			1	1.0000			1	1.0000			1	1.0000			1	1.0000		
		2	0.4956			2	0.5023			2	0.5040			2	0.5023			2	0.5047		
		3	0.3335			3	0.3235			3	0.3258			3	0.3235			3	0.3393		
		4	0.2456			4	0.2512			4	0.2544			4	0.2512			4	0.2530		
		5	0.2031			5	0.1963			5	0.1966			5	0.1963			5	0.2029		
		6	0.1618			6	0.1602			6	0.1616			6	0.1602			6	0.1712		
		7	0.1395			7	0.1440			7	0.1438			7	0.1440			7	0.1452		
		8	0.1249			8	0.1279			8	0.1291			8	0.1279			8	0.1216		
		9	0.1100			9	0.1073			9	0.1078			9	0.1073			9	0.1104		
		10	0.1029			10	0.0991			10	0.1002			10	0.0991			10	0.1019		
		11	0.1768			11	0.2732			11	0.3665			11	0.2732			11	0.5476		
		12	0.0808			12	0.0811			12	0.0828			12	0.0811			12	0.0838		
		13	0.0738			13	0.0763			13	0.0781			13	0.0763			13	0.0746		
		14	0.0667			14	0.0745			14	0.0748			14	0.0745			14	0.0712		
		15	0.0660			15	0.0658			15	0.0670			15	0.0658			15	0.0695		
		16	0.0613			16	0.0640			16	0.0626			16	0.0640			16	0.0594		
		17	0.0602			17	0.0605			17	0.0593			17	0.0605			17	0.0621		
		18	0.0532			18	0.0523			18	0.0529			18	0.0523			18	0.0560		
		19	0.0546			19	0.0494			19	0.0515			19	0.0494			19	0.0516		
		20	0.0483			20	0.0498			20	0.0509			20	0.0498			20	0.0492		
		21	0.0428			21	0.0460			21	0.0460			21	0.0460			21	0.0504		
		22	0.0877			22	0.1376			22	0.1846			22	0.1376			22	0.2772		
		23	0.0433			23	0.0396			23	0.0395			23	0.0396			23	0.0421		
		24	0.0412			24	0.0429			24	0.0442			24	0.0429			24	0.0395		
		25	0.0409			25	0.0397			25	0.0404			25	0.0397			25	0.0414		

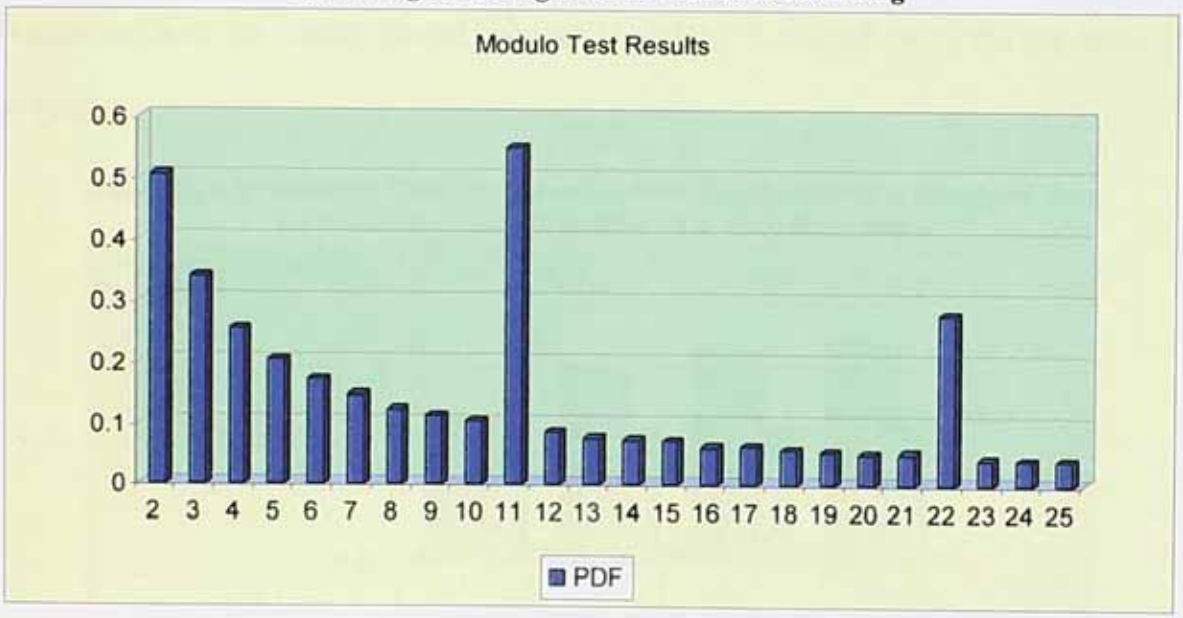
Uniform Best-Fit Distribution

Using uniform best-fit distribution will produce similar results but with much lower probability values.

Modulo Test

The Modulo test results also show increasing probabilities in rounding base 11 and its multiple of rounding base 22. This result is consistent with the NNM finding, which shows that the model can detect rounding base 11. Moreover, rounding base 11 probability value is always higher than those of its multiples. In data set E, for instance, rounding base 11 probability is 0.5476, while the probability for base 22 is only 0.2772 (Figure 6.15).

Figure 6.15. Modulo Test Results of Detecting Rounding Base 11 on a Simulated Data Set Containing Rounding Base 11 with 50% Rounding



Moreover, to compare the robustness of detecting rounding base 11 across different rounding levels and best fits used, Table 6.8 also summarises the detection power and detection error for all cases. The results can be summed up as follows:

(i) There is always a detection of rounding base 11 in data sets A to E using all best fits.

(ii) Normal best fit produces the highest detection power, followed by lognormal and uniform.

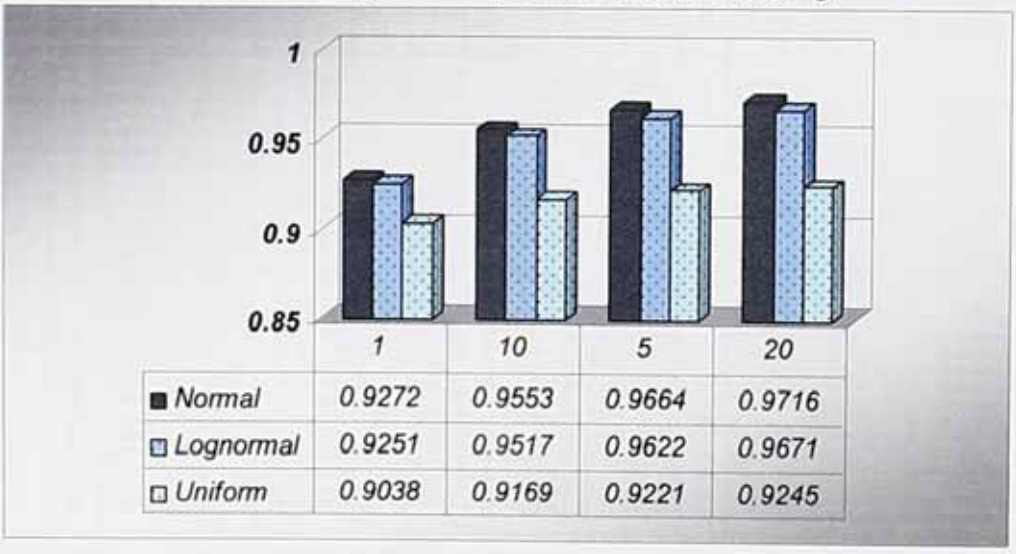
(iii) The detection power in data sets A to E increase gradually following a smooth trend.

(iv) No detection error is identified from the results.

6.4.4. Detecting Rounding Base 10

Implementation in the normal data containing rounding base 10 with different rounding magnitudes shows that the model detects the rounding pattern very well. The detection shows an increasing number of bases as the rounding level increases from 10% to 50%. This result is consistent with the initial expectation on the overall results and how the model should behave. Therefore, it is confirming the goodness fit of NNM.

Figure 6.16. Probability Values of Detecting Rounding Base 10 in a Simulated Data Set Containing Rounding Base 10 at 40% Rounding



	1	10	5	20
Normal	0.9272	0.9553	0.9664	0.9716
Lognormal	0.9251	0.9517	0.9622	0.9671
Uniform	0.9038	0.9169	0.9221	0.9245

Even though the NNM can detect rounding base 10 regardless of the best fit used, the use of a correct functional distribution for the data set being examined will guarantee the best result. Figure 6.16 shows that normal best fit detects rounding base

10 in data set D with a probability value of 0.9553, while uniform best fit probability value is 0.9169. The model detects rounding base 10 in the second iteration in data sets containing rounding as low as 10% regardless of the best-fit distributions used.

Table 6.9 summarises the results of detecting rounding base 10 in the normally distributed data sets by using three different best-fit distributions of normal, lognormal, and uniform. The detailed detection results are discussed as follows:

Table 6.9. Results of Detecting Rounding Base 10 Contained in the Normally Distributed Simulated Data Sets Using Three Different Best-Fit Distribution Functions

Best Fit Distribution	Iteration	Magnitude of Rounding Errors																			
		10%				20%				30%				40%				50%			
		Base	Pdf	Detection Power	Detection Error	Base	Pdf	Detection Power	Detection Error	Base	Pdf	Detection Power	Detection Error	Base	Pdf	Detection Power	Detection Error	Base	Pdf	Detection Power	Detection Error
Normal	1	1	**0.9895**	**0.28**	**0.00**	1	**0.9766**	**1.36**	**0.00**	1	**0.9554**	**2.64**	**0.02**	1	**0.9272**	**3.43**	**1.14**	2	**0.8877**	**4.69**	**3.09**
	2	10	0.9923			10	**0.9889**			10	**0.9759**			10	**0.9553**			10	**0.9234**		
	3					20	0.9902			20	**0.9813**			5	**0.9664**			5	**0.9511**		
	4									25	0.9815			20	**0.9716**			20	**0.9605**		
	5																	15	0.9625		
	6																	25	0.9625		
Lognormal	1	1	**0.9864**	**0.29**	**0.00**	1	**0.9741**	**1.30**	**0.00**	1	**0.9532**	**2.50**	**0.01**	1	**0.9251**	**3.24**	**1.08**	2	**0.8849**	**4.39**	**2.90**
	2	10	0.9893			10	**0.9858**			10	**0.9726**			10	**0.9517**			10	**0.9182**		
	3					20	0.9871			20	**0.9777**			5	**0.9622**			5	**0.9440**		
	4									25	0.9778			20	0.9671			20	**0.9530**		
	5																	15	0.9551		
	6																	25	0.9551		
Uniform	1	1	**0.9652**	**0.19**	**0.00**	1	**0.9514**	**0.63**	**0.00**	1	**0.9310**	**1.22**	**0.00**	1	**0.9038**	**1.60**	**0.54**	2	**0.8549**	**2.12**	**1.36**
	2	10	0.9671			10	**0.9571**			10	**0.9405**			10	**0.9169**			10	**0.8811**		
	3					20	0.9577			20	0.9430			5	**0.9221**			5	**0.8934**		
	4									25	0.9430			20	0.9245			20	0.8976		
	5																	15	0.8984		
Modulo Test		1	**1.0000**			1	**1.0000**			1	**1.0000**			1	**1.0000**			1	**1.0000**		
		2	0.5459			2	0.6004			2	0.6512			2	0.6957			2	0.7520		
		3	0.3314			3	0.3252			3	0.3288			3	0.3367			3	0.3373		
		4	0.2710			4	0.2985			4	0.3244			4	**0.3441**			4	**0.3765**		
		5	**0.2827**			5	**0.3590**			5	**0.4392**			5	**0.5293**			5	**0.5996**		
		6	0.1778			6	0.1935			6	0.2126			6	0.2356			6	0.2555		
		7	0.1394			7	0.1493			7	0.1482			7	0.1412			7	0.1427		
		8	0.1370			8	0.1516			8	**0.1653**			8	**0.1734**			8	**0.1844**		
		9	0.1089			9	0.1087			9	0.1099			9	0.1120			9	0.1128		
		10	**0.1928**			10	**0.2796**			10	**0.3693**			10	**0.4668**			10	**0.5518**		
		11	0.0883			11	0.0911			11	0.0947			11	0.0898			11	0.0924		
		12	0.0883			12	**0.0969**			12	**0.1068**			12	**0.1165**			12	**0.1232**		
		13	0.0742			13	0.0777			13	0.0770			13	0.0788			13	0.0738		
		14	0.0745			14	**0.0915**			14	0.0977			14	**0.0985**			14	**0.1073**		
		15	**0.0917**			15	**0.1198**			15	**0.1477**			15	**0.1753**			15	**0.2032**		
		16	0.0677			16	0.0756			16	0.0815			16	0.0866			16	0.0944		
		17	0.0568			17	0.0583			17	0.0577			17	0.0579			17	0.0599		
		18	0.0563			18	**0.0641**			18	**0.0714**			18	**0.0794**			18	**0.0872**		
		19	0.0545			19	0.0496			19	0.0502			19	0.0526			19	0.0493		
		20	**0.0926**			20	**0.1387**			20	**0.1823**			20	**0.2314**			20	**0.2758**		
		21	0.0425			21	0.0478			21	0.0486			21	0.0476			21	0.0493		
		22	0.0471			22	**0.0537**			22	**0.0600**			22	**0.0631**			22	**0.0694**		
		23	0.0437			23	0.0409			23	0.0404			23	0.0404			23	0.0431		
		24	0.0453			24	**0.0497**			24	**0.0563**			24	**0.0605**			24	**0.0573**		
		25	**0.0584**			25	**0.0712**			25	**0.0903**			25	**0.1072**			25	**0.1212**		

Normal Best-Fit Distribution

Application of normal best fit to detect rounding base 10 in the normally distributed data containing rounding base 10 shows that the model detects rounding

base 10 in the second iteration, followed by other bases, such as bases 20 and 25 in data set C, and other bases in other data sets. The model detects rounding base 5 in data set D and rounding bases 2 and 25 in data set E. It always detects rounding base 1 at the first iteration in data sets A to D, but the model in data set E detects rounding base 2 first, followed by other rounding bases of 10, 5, 20, 15, and 25.

Lognormal Best-Fit Distribution

Detecting rounding base 10 using lognormal best fit produces similar results to using normal best fit but with lower probability density function.

Uniform Best-Fit Distribution

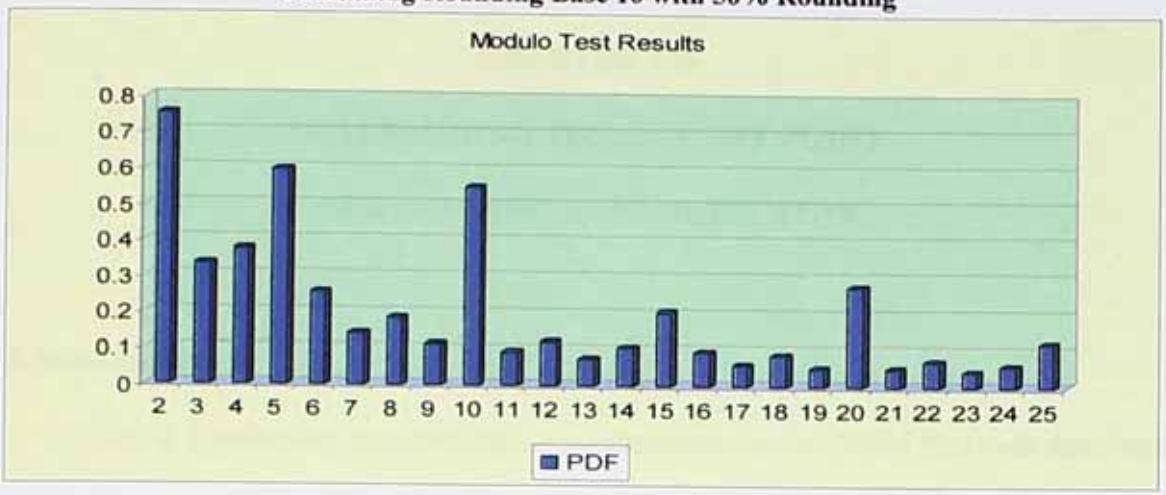
Using uniform best fit in data sets A to D produces similar results of using normal and lognormal best fit, only with lower probability values. In data set E, however, the results are slightly different. No rounding base 25 is detected when using uniform best fit. In fact, there is no significant difference in the detection rounding base 25 as compared to the detection of rounding base 15 using normal and lognormal best-fit distributions.

Modulo Test

The Modulo test results also show the increasing probability values of rounding base 10 and its multiple of rounding base 20. This is consistent with the NNM finding which shows that NNM can detect rounding base 10.

Moreover, Figure 6.17 shows the higher probability values in the rounding bases 10 and 20. However, note also the high probability values of bases 2 and 5, as well as of the multiples of base 5 such as bases 15 and 25. This shows the distorting power of rounding, which becomes obvious if the rounding level is more than 30%.

Figure 6.17. Modulo Test Results of Detecting Rounding Base 10 on a Simulated Data Set Containing Rounding Base 10 with 50% Rounding



Moreover, to compare the robustness of detecting rounding base 10 across different rounding levels and best fits used, Table 6.9 also summarises the detection power and detection error for all cases. The results can be summed up as follows:

(i) There is always a detection of rounding base 10 in data sets A to E using all best fits.

(ii) Normal best fit produces the highest detection power, followed by lognormal and uniform.

(iii) The detection power in data sets A to E increase gradually following a smooth trend.

(iv) Detection errors are identified from the results in data set C using normal and lognormal best fits and in data sets D and E using all best fits. In case of detection error, normal best fit also always produces the highest detection error as a result of the model detecting rounding bases 2 and 5, the factor numbers of base 10.

CHAPTER VII

DETERMINING THE CUT-OFF POINT OF ROUNDING BASE DETECTION

7.1. Introduction

Having conducting comprehensive assessments on the NNM that was developed in this study by using simulated data sets containing rounding to specific base numbers and different rounding magnitudes, one crucial question remains, i.e., how to determine the cut-off point of the rounding level at which the model can no longer detect the existent rounding base in the data set. The earlier assessments using five different levels of rounding from 10% to 50% with a 10% interval in between cannot provide the answer on the cut-off point of the rounding base detection.

To shed light on this issue, this chapter develops two different methods for determining the cut-off point. The first approach is by using direct testing by applying NNM on different kinds of data sets with different rounding levels until the model can no longer detect the existence of a rounding base. The second approach is by maximizing the existence detection results to come up with a regression model that can predict the cut-off point.

From the results of the previous two chapters, it is very clear that the cut-off point cannot be determined for one nor for all, but it will depend on three important factors. These are the data distribution, best-fit functional distribution used, and base numbers.

Table 7.1 provides the schematic representation of all combinations used in determining the cut-off points of rounding base detection. The cut-off points depend on

the underlying data distributions, best fits used, and base numbers. Accordingly, this chapter would identify 2 x 3 x 4 = 24 cut-off points.

On the other hand, Table 7.1b shows the comparison of results of calculating the cut-off points using big and small data sets. Note some differences in the cut-off points' results using the two sizes of data sets. The cut-off points calculated from the smaller data sets tend to be lower. This implies that detection rounding base in bigger data sets will take longer than on the smaller one.

Table 7.1. Schematic Representation of Determining the Cut-off Points for Different Data Distributions, Best Fits, and Base Numbers

Data Distributions	Best Fits	Rounding Base Number (%)				Total Number COP
		Base 5	Base 7	Base 11	Base 10	
Uniform (U)	Uniform (BU)	UBU-5	UBU-7	UBU-11	UBU-10	4
	Normal (BN)	UBN-5	UBN-7	UBN-11	UBN-10	4
	Lognormal (BL)	UBL-5	UBL-7	UBL-11	UBL-10	4
Normal (N)	Uniform (BU)	NBU-5	NBU-7	NBU-11	NBU-10	4
	Normal (BN)	NBN-5	NBN-7	NBN-11	NBN-10	4
	Lognormal (BL)	NBL-5	NBL-7	NBL-11	NBL-10	4
Total Number COP		6	6	6	6	24

7.2. Determination of the Cut-off Point Using Direct Testing on the Uniform Data Set

7.2.1. Determining the Cut-off Point for Rounding Base-5

From the results of detecting rounding base 5 on the simulated uniform data sets in the previous chapter, one will discover that the starting point for determining the cut-off point of a rounding level on which the model can just detect the rounding base 5 should be more than 10%. The reason is that there is no detection of rounding base 5 in data set A (containing 10% rounding). Table 7.2 shows that at an 12% rounding level, there is no detection of rounding base 5. Accordingly, the search moves upward to rounding level 13%. At this rounding level, the uniform best fit starts to detect the existence of rounding base 5, while normal and lognormal best fits still cannot detect the existence of rounding base 5. Moving forward to higher rounding levels, the normal and lognormal best fits start to detect the existence of rounding base 5 in the data sets containing 19% and 29% of rounding, respectively. Therefore, the cut-off points for the three best fits of uniform, normal, and lognormal are 12%, 18%, and 28%.

Table 7.2. Determination of Cut-off Point of Rounding Base 5 on the Uniformly Distributed Data Set

Best Fit	Iteration	12%		13%		18%		19%		28%		29%	
		base	pdf	base	pdf	base	pdf	base	pdf	base	pdf	base	pdf
Uniform	1	1	0.9917	1	0.9914	1	0.9898	1	0.9895	1	0.9856	1	0.9851
	2			**5**	**0.9917**	5	0.9920	5	0.9920	5	0.9914	5	0.9912
Normal	1	1	0.9883	1	0.9880	1	0.9864	1	0.9861	1	0.9822	1	0.9817
	2					10	0.9866	**5**	**0.9866**	5	0.9860	5	0.9859
Lognormal	1	1	0.9826	1	0.9823	1	0.9807	1	0.9804	1	0.9765	1	0.9760
	2									25	0.9768	**5**	**0.9766**

7.2.2. Determining the Cut-off Point for Rounding Base 7

Results of rounding base 7 detection in the uniform data in the previous chapter shows the detection of rounding base 7 in the data set containing 10% rounding (data set A) using uniform best fit. Yet, no rounding base 7 is detected in data set A using normal and lognormal best fits. Accordingly, the cut-off points for the uniform best fit should be less than 10%, while other best fits should be more than 10% but less than 20% since rounding base 7 is detected in other data sets using all three best fits.

Initial detection in the data set containing 8% of rounding still results in no detection of rounding base 7. Moving forward to the data set containing rounding base 7 with 9% rounding level, the uniform best fit starts to detect the existence of rounding base 7. Furthermore, observing data sets containing more than 10% rounding level, the normal and lognormal best fits detect the existence of rounding base 7 in the data sets containing 13% and 20% rounding levels, respectively. Therefore, the cut-off points for detecting rounding base 7 in the uniform data sets with the three best fits of uniform, normal, and lognormal are 8%, 12%, and 19%. The complete detection results for rounding base 7, together with their probability values, are summarised in Table 7.3.

Table 7.3. Determination of Cut-off Point of Rounding Base 7 on the Uniformly Distributed Data Set

Best Fit	Iteration	8%		9%		12%		13%		19%		20%	
		base	pdf	base	pdf	base	pdf	base	pdf	base	pdf	base	pdf
Uniform	1	1	0.9925	1	0.9919	1	0.9911	1	0.9907	1	0.9880	1	0.9875
	2			**7**	**0.9923**	7	0.9926	7	0.9926	7	0.9923	7	0.9922
Normal	1	1	0.9891	1	0.9885	1	0.9877	1	0.9873	1	0.9846	1	0.9842
	2					14	0.9878	**7**	**0.9877**	7	0.9875	7	0.9874
Lognormal	1	1	0.9834	1	0.9828	1	0.9820	1	0.9816	1	0.9790	1	0.9785
	2									14	0.9795	**7**	**0.9793**

7.2.3. Determining the Cut-off Point for Rounding Base-11

Results of detecting rounding base 11 in the uniform data in the previous chapter shows no detection of rounding base 11 in the data set containing 10% rounding using lognormal best fit. This means that the cut-off point for rounding base 11 using lognormal best fit will be more than 10% (but less than 20%); while the cut-off points should be less than 10% for other best fits. Initial detection in the data set containing 5% of rounding base 11 still results in no detection of rounding base 11 that leads to using data sets with a higher rounding level. In the data set containing 6% of rounding base 11, the model starts to detect the existence of rounding base 11. Therefore, this is the cut-off point for using the uniform best fit in this data set.

Moving to the data set containing rounding base 11 with an 9% rounding level, the normal best fit starts to detect the existence rounding base 11. Furthermore, observing data sets containing more than 10% rounding level, the lognormal best fit detects the existence of rounding base 7 in the data sets containing 13%. Therefore, the cut-off points for detecting rounding base 11 in the uniform data sets with the three best fits of uniform,

normal, and lognormal are 5%, 8%, and 12%. The complete detection results for rounding base 11, together with their probability values are summarised in Table 7.4.

Table 7.4. Determination of Cut-off Point of Rounding Base 11 on the Uniformly Distributed Data Set

Best Fit	Iteration	5%		6%		8%		9%		12%		13%	
		base	pdf	base	pdf	base	pdf	base	pdf	base	pdf	base	pdf
Uniform	1	1	0.9926	1	0.9924	1	0.9919	1	0.9915	1	0.9901	1	0.9895
	2			**11**	**0.9926**	11	0.9929	11	0.993	11	0.9929	11	0.9927
Normal	1	1	0.9892	1	0.989	1	0.9885	1	0.9881	1	0.9867	1	0.9861
	2					22	0.9886	**11**	**0.9887**	11	0.9885	11	0.9883
Lognormal	1	1	0.9835	1	0.9833	1	0.9828	1	0.9825	1	0.9811	1	0.9805
	2									22	0.9812	**11**	**0.981**

7.2.4. Determining the Cut-off Point for Rounding Base-10

Results of detecting rounding base 10 in the uniform data in the previous chapter shows no detection of rounding base 10 in the data set containing 10% rounding using lognormal best fit. This means that the cut-off point for rounding base 10 using lognormal best fit will be more than 10% (but less than 20%); while the cut-off points should be less than 10% for other best fits.

Initial detection in the data set containing 5% of rounding 10 still produces no detection of rounding base 10. This leads to the use of data sets with a higher rounding level. The model starts to detect the existence of rounding base 10 in the data set with 6% rounding level. Therefore, this is the cut-off point for using the uniform best fit in this data set. Moving forward to the normal best fit, the model starts to detect the existence of rounding base 10 in data sets containing a 10% rounding level.

Table 7.5. Determination of Cut-off Point of Rounding Base 10 on the Uniformly Distributed Data Set

Best Fit	Iteration	5%		6%		7%		8%		12%		13%	
		base	pdf	base	pdf	base	pdf	base	pdf	base	pdf	base	pdf
Uniform	1	1	0.9925	1	0.9923	1	0.9921	1	0.9918	1	0.9901	1	0.9895
	2			**10**	**0.9925**	10	0.9927	10	0.9928	10	0.9928	10	0.9927
Normal	1	1	0.9891	1	0.9889	1	0.9887	1	0.9884	1	0.9867	1	0.9861
	2							**10**	**0.9885**	10	0.9884	10	0.9883
Lognormal	1	1	0.9834	1	0.9832	1	0.983	1	0.9827	1	0.981	1	0.9805
	2											**10**	**0.9808**

Moreover, observing data sets containing more than a 10% rounding level, the lognormal detects the existence of rounding base 10 in the data sets containing a 13% rounding level. Therefore, the cut-off points for detecting rounding base 10 in the uniform data sets with the three best fits of uniform, normal, and lognormal are 5%, 9%, and 12%. Table 7.5 summarises the complete detection results for rounding base 10, complete with their probability values.

7.3. Determination of the Cut-off Points Using Direct Testing on the Normal Data Sets

7.3.1. Determining the Cut-off Point for Rounding Base-5

Results of detecting rounding base 5 in the normal data in the previous chapter shows no detection of rounding base 5 in the data set containing 10% rounding using all three best fits. This means that the cut-off point for rounding base 5 in the normal data must be more than 10% and less than 20%.

Table 7.6. Determination of Cut-off Point for Rounding Base 5 on the Normally Distributed Data Set

Best Fit	Iteration	13%		14%		15%	
		base	pdf	base	pdf	base	pdf
Normal	1	1	0.9901	1	0.9900	1	0.9889
	2	10	0.9904	10	0.9904	**5**	**0.9906**
Lognormal	1	1	0.9873	1	0.9874	1	0.9865
	2	10	0.9877	**5**	**0.9879**	5	0.9883
Uniform	1	1	0.9657	1	0.9658	1	0.9633
	2	10	0.966	**5**	**0.9664**	5	0.9642

Initial detection in the data set containing 13% of rounding still produces no detection of rounding base 5. This leads to using data sets with a higher rounding level. In the data set with a 14% rounding level, the lognormal and uniform best fits start to detect rounding base 5. Moving forward to the normal best fit, the model starts to detect the existence of rounding base 5 in data sets containing a 15% rounding level. Therefore the cut-off points for detecting rounding base 5 in normal data sets with the three best fits of normal, lognormal, and uniform, are 14% and 13%, respectively. Table 7.6 summarises the complete detection results for rounding base 5, complete with their probability values.

7.3.2. Determining the Cut-off Point for Rounding Base-7

Results of detecting rounding base 7 in the normal data discussed in the previous chapter show that no rounding base 7 is detected in the data set containing 10% rounding using normal best fit. This means that the cut-off point for this best fit must be more than 10% but less than 20%. Initial detection in the data set containing 8% of rounding still produces no detection of rounding base 7. But at a rounding level of 9%, the uniform best

fit starts to detect the existence of rounding base 7. The lognormal best fit starts to detect the existence of rounding base 7 in the data set with 10% rounding level.

Table 7.7. Determination of Cut-off Point for Rounding Base 7 on the Normally Distributed Data Set

Best Fit	Iteration	8%		9%		10%		11%	
		base	pdf	base	pdf	base	pdf	base	pdf
Normal	1	1	0.9914	1	0.9913	1	0.9911	1	0.9901
	2							**7**	**0.9913**
Lognormal	1	1	0.9887	1	0.9885	1	0.9882	1	0.9875
	2					**7**	**0.9884**	7	0.9888
Uniform	1	1	0.9668	1	0.9668	1	0.9668	1	0.9654
	2	14	0.9672	**7**	**0.9671**	7	0.9673	7	0.9663

Moving forward to the data sets containing more rounding bases, the normal best fit actually starts to detect the existence of rounding base 7 in the data set containing 11% rounding level. Accordingly, the cut-off points for detecting rounding base 7 in the normal data sets with the three best fits of normal, lognormal, and uniform are 10%, 9%, and 8%. The complete detection results for rounding base 7, together with their probability values, are summarised in Table 7.7.

7.3.3. Determining the Cut-off Point for Rounding Base-11

Results of detecting rounding base 11 in the normal data discussed in the previous chapter show that there is always detection of rounding base 11 in all data sets using all three best fits. This means that the cut-off point for rounding base 11 must be less than 10%.

Initial detection in the data set containing 4% of rounding still produces no detection of rounding base 11, but at rounding level 5%, the uniform best fit starts to detect the existence of rounding base 11. The normal and lognormal best fits start to detect the existence of rounding base 11 in the data set with 6% rounding level.

Table 7.8. Determination of Cut-off Point for Rounding Base 11 on the Normally Distributed Data Set

Best Fit	Iteration	4%		5%		6%	
		base	pdf	base	pdf	base	pdf
Normal	1	1	0.992	1	0.9924	1	0.9914
	2					**11**	**0.9915**
Lognormal	1	1	0.9893	1	0.9896	1	0.9886
	2					**11**	**0.9889**
Uniform	1	1	0.9666	1	0.9672	1	0.9664
	2			**11**	**9672**	11	0.9668

Accordingly, the cut-off points for detecting rounding base 11 in the normal data sets with the three best fits of normal, lognormal, and uniform, are 5%, 5%, and 4%. The complete detection results for rounding base 11, together with their probability values, are summarised in Table 7.8.

7.3.4. Determining the Cut-off Point for Rounding Base-10

Results of detecting rounding base 10 in the normal data discussed in the previous chapter show that there is always detection of rounding base 10 in all cases using all three best fits. This means that the cut-off points for rounding base 10 for the three best fits must be less than 10%.

Initial detection in the data set containing 5% of rounding still produces no detection of rounding base 10, but at rounding level 6%, the uniform best fit starts to detect the existence of rounding base 10. The normal and lognormal best fits start to detect the existence of rounding base 10 in the data set with 7% rounding level.

Table 7.9. Determination of Cut-off Point for Rounding Base 10 on the Normally Distributed Data Set

Best Fit	Iteration	5%		6%		7%	
		base	pdf	base	pdf	base	pdf
Normal	1	1	0.9925	1	0.9915	1	0.9908
	2			20	0.9917	**10**	**0.9914**
Lognormal	1	1	0.9898	1	0.9887	1	0.9882
	2			20	0.9889	**10**	**0.9889**
Uniform	1	1	0.9673	1	0.9665	1	0.966
	2			**10**	**0.9668**	10	0.9667

Accordingly, the cut-off points for detecting rounding base 10 on the normal data sets with the three best fits of normal, lognormal, and uniform are 7%, 7%, and 6%. The complete detection results for rounding base 10, together with their probability values, are summarised in Table 7.9.

7.3.5. Summary and Conclusion

From the discussions in this chapter, some conclusions emerge that can be summarised according to two main issues related to base numbers and best fits used in the detection. In addition, Table 7.10 summarises the results of determining cut-off points for different data sets, best fits used, and base number.

Base Number

(i) In detecting the cut-off points for different rounding bases of prime numbers, it seems that the higher the number, the easier would be for the model to detect the rounding base number. Rounding base 11 is easier to be detected than rounding base 7, while rounding base 7 is easier to be detected than rounding base 5, and so on.

(ii) For nonprime numbers, the opposite may be the case. The bigger the number, the more factors would be associated with the number. Earlier results discussed in the previous chapter show that the existing factor numbers could distract the rounding base detection, especially in the data sets containing high rounding levels.

Table 7.10. Cut-off Points for Detecting Rounding Base for Different Data Distributions, Best Fits, and Base Numbers Using Direct Testing

Data Distributions	Best Fits	Rounding Base Number (%)			
		Base 5	Base 7	Base 11	Base 10
Uniform (U)	Uniform (BU)	12	7	5	5
	Normal (BN)	18	12	8	7
	Lognormal (BL)	28	19	12	12
Normal (N)	Uniform (BU)	13	8	4	5
	Normal (BN)	14	10	5	6
	Lognormal (BL)	13	9	5	6

Best Fits Used

(i) Comparing across different best fits, the use of uniform best fit in the uniform data produces the lowest cut-off point—indicating it to be the most effective in detecting the rounding base. In fact, the use of uniform best fit in the normal data sets also produce second best result, only less superior in the case of using uniform best fit to detect rounding base 5 in normal data.

(ii) The use of uniform best fit in normal data is even more effective in determining the cut-off point than using normal best fit in normal data.

(iii) The adverse impact of using wrong best fit in the cut-off point is more severe in uniform data than in normal data.

These findings bring us to an important conclusion—i.e., once the distribution of data is known to be uniform, then uniform best fit must be used. In case the underlying distribution is not known, the safest approach seems to be to use uniform best fit in detecting the rounding bases.

7.4. Determination of the Cut-off Points Using a Regression Model

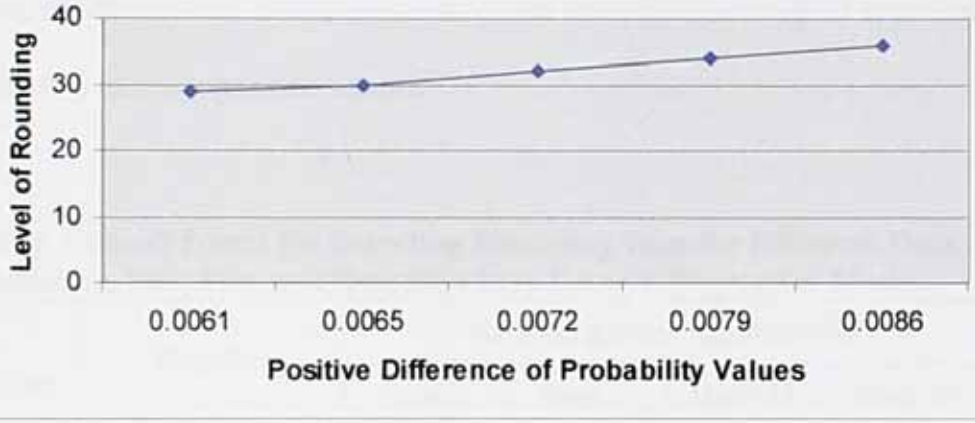
The second method to determine the cut-off points for a combination of different data distribution, best fits, and base numbers is by developing regression models of the relationship between total positive differences in the probability values of detection with the level of rounding in the data sets. This relationship captured in the regression must be linear in nature since it is a process of one to one mapping of the level of rounding exists

in the data set and its probability values (see figure 7.1. for example). The following model shows the relationship of the two:

$$Y_t = \alpha + \beta X_t + \varepsilon_t \quad (1)$$

Where Y_t is the level of rounding contained in the data set, X_t is the positive difference in the probability values of detection, and ε_t, is the error term.

Figure 7.1. Linear Plot of Positive Difference of PDF and Level of Rounding to Base 5 Uniformly Distributed Data to Estimate Cut-off Point



The model is developed for each rounding base to be detected and for best-fit distribution to be used. Therefore, following the schematic representation presented in Table 7.1, there will be 24 regression modelling estimations. They are for two data sets of uniform and normal, three best fits of uniform, normal, and lognormal, and four rounding bases of 5, 7, 11, and 10.

In the context of determining the cut-off points, the application of the regression model above is purely for forecasting, i.e., estimating what would be the value of Y, which is the rounding level contained in the data set when the value of X that is the positive difference in the probability value of detection, is equal to zero. The positive

difference in the probability value of detection equals to zero reflects that the model can no longer detect the rounding bases. Table 7.11 summarises the estimation results using the 24 regression models described in equation (1).

Compared to Table 7.10, there is not much difference between the results of using direct investigation and regression models, especially in terms of comparing the results across different base numbers and best fits. The conclusion arrived at in the previous section is still valid with regard to the regression results. On the actual value of the cut-off point, however, there are some differences but the maximum difference is less then 5%. Therefore, all discussions in the previous direct investigation method are, more or less, also applicable to the regression results.

Table 7.11. Cut-off Points for Detecting Rounding Base for Different Data Distribution, Best Fits, and Base Numbers Using a Regression Model

Data Distributions	Best Fits	Rounding Base Number (%)			
		Base 5	Base 7	Base 11	Base 10
Uniform (U)	Uniform (BU)	11.75	7.26	5.00	5.21
	Normal (BN)	17.53	11.38	7.54	10.06
	Lognormal (BL)	29.50	17.71	11.71	13.91
Normal (N)	Uniform (BU)	12.36	8.69	5.05	5.70
	Normal (BN)	12.63	9.75	6.10	6.88
	Lognormal (BL)	12.35	9.41	5.89	6.73

CHAPTER VIII

APPLICATIONS OF THE MODEL ON REAL DATA SETS

Having assessed the model's performance using simulated data sets in the previous chapter, the model is then implemented on real data sets. The first real data set is from the religious census conducted in England and Wales in 1851 that referred to the number of congregation attending the Church. The main purpose of this application is to emulate what Crockett and Crockett (1998) found in detecting rounding base on the data set. By doing so, the application proves that the model developed in this study works well as it can emulate what modulo test has done before but now in the neural network context. They found that the underlying distribution function of the data was log normal and that the data contained rounding base 5.

Table 8.1a summarises the results of applying the model on the religious data. A summary of applying the modulo-test on the data set is also included at the end of the table for easy reference. The modulo test results clearly show that the data contain rounding base 5 as can be seen from the significantly increase probability values on rounding base 5 and its multiples from 10 to 25.

The results of applying neural network show that, using the lognormal best fit in detecting the rounding will produce the best results, hence the underlying distribution function of the religious census data is lognormal. More particular, probability values of detecting rounding base 5, 10, 20 and 25 in the log normal case are the highest among the normal and uniform results. Therefore both findings from the previous research were confirmed by the neural network modelling results.

Table 8.1a. Results of Detecting Rounding Patterns in the Number of People Attending Churches in Census Data 1851

Best Fit Distribution	Iteration	Base	Pdf
Lognormal	1	1	0.5581
	2	10	0.6857
	3	5	0.774
	4	20	0.8121
	5	25	0.8231
Normal	1	1	0.5014
	2	10	0.582
	3	5	0.6375
	4	20	0.6673
	5	25	0.6831
Uniform	1	1	0.4128
	2	10	0.4253
	3	5	0.4326
	4	20	0.4364
	5	25	0.4389
Modulo Test		1	1
		2	0.7018
		3	0.3452
		4	0.3868
		5	0.5833
		6	0.2427
		7	0.1284
		8	0.1934
		9	0.0993
		10	0.4734
		11	0.0588
		12	0.1269
		13	0.0584
		14	0.0786
		15	0.2022
		16	0.0767
		17	0.0366
		18	0.0572
		19	0.036
		20	0.271
		21	0.0324
		22	0.0339
		23	0.0256
		24	0.0471
		25	0.1905

The second sets of real data applications are applying the model on the data of the **Number of cigarettes smoked** and **Amount of alcohol consumed** by secondary school pupils (aged 11–15) in England, United Kingdom, in 2001 as a result of the first *Survey of Smoking, Drinking and Drug Use* among. The model applications aim at examining the rounding patterns that might be present in the two data sets.

To examine the rounding patterns in the two data sets, the three best-fit distribution functions of normal, lognormal, and uniform are used in the detection. The main purpose of using all three best fits is twofold: (i) to have the best results of detecting the rounding pattern, and (ii) to examine what would be the underlying distribution of the data from concluding that the correct best fit distribution function used in the detection will result in the highest probability of detection. For this reason, the alternative way of examining first the distribution of the data set being examined and then detecting the rounding pattern using the appropriate best-fit distribution function based on the examination result, is not conducted in this study for it is also inefficient.

8.1. Data Characteristics

Before discussing the application results on the two real data sets, it is important to know the characteristics of the data under investigation especially those that could affect the data quality.

In collecting data on the number of cigarettes smoked, three different time references are used in the survey: **everyday**, **weekdays,** and **weekends.** A close examination on the data reveals that the three data are not independently collected such that the three data will always be consistent among them. In this case, there is a

kind of built-in cross-check on the number of cigarettes smoked during the three different time periods that make them always consistent.

Moreover, given the three survey reference periods used in the data collection, data for **everyday** consumption is expected to have the best quality in terms of smoothness because the number of cigarettes smoked everyday must be taken as the average number of the cigarettes smoked during the weekdays and over the weekends. Therefore, there is already a built-in control check on the number of cigarettes smoked every day as it can be derived from the weekdays and week-end consumptions.

The data quality, including the rounding patterns, would be very different if the number of cigarettes smoked, for instance, was asked only for a specific period such as everyday without any control on the number of cigarettes smoked during the weekdays and weekends. Moreover, if a different survey's reference period such as last month were used, it would also have produced different data with different quality. In general, the longer the survey's reference period, the worse will be the quality of the data set. *Ceteris paribus*, this is because of the errors in the respondent's memory recall.

On the other hand, data on the amount of alcohol consumed are grouped into 14 different categories, based on the six types of alcoholic drinks covered in the survey: from beer to alcopops, and the different units of measurements or packaging used: pints, half pints, large can, small can, bottle, and glass. Table 8.2 summarizes the 14 different classifications of alcoholic drinks used in the survey and examined in this chapter.

Table 8.1b: Classifications of the Alcoholic Drinks Consumed by Pupils Aged 11–15 in England, United Kingdom in 2001

X1: Beer in pint X2: Beer in half pint X3: Beer in large can X4: Beer in small can X5: Beer in bottle X6: Shandy in pint X7: Shandy in half pint	X8: Shandy in large can X9: Shandy in small can X10: Wine in glass X11: Sherry in glass X12: Spirits in glass X13: Alcopops in can X14: Alcopops in bottle

Unlike in the number of cigarettes smoked, the reference period used in the survey for alcoholic drink consumption is **in the last 7 days** only. No questions were asked on the amount of alcoholic drink consumption for **everyday**, **weekdays,** and **weekends** as in cigarette consumption**.** Therefore, by ignoring other factors, the two data sets are collected differently which may affect their qualities, especially with regard to their rounding patterns that may contain in the data sets.

Moreover, the use of different units of measurement or packaging in the survey on alcoholic drinks provides an opportunity to examine the relationship between packaging and consumption patterns, especially in their relation with rounding patterns. In this context, the result can show that many factors such as gender and age can influence some preferences on size and packaging. This issue could be useful for marketing and production management since it has a strategic value.

Recall that the respondents of the survey are pupils aged 11–15; the survey on cigarette smoking, for instance, found that prevalence of smoking (defined here as smoking at least one cigarette per day) was strongly related to age. Only 1% of 11-year-olds were regular smokers compared with 22% of 15-year-olds. As with cigarette smoking, the consumption of alcoholic drinks was also strongly related to age. Only

6% of 11-year-olds had drunk alcohol in the last week compared with 52% of 15-year-olds (Boreham and Shaw 2002).

Moreover, the use of detailed types of alcoholic drinks with different kinds of measurement or package for each alcoholic drink included in the survey can have a positive impact on data quality, including on reducing the possibility that the data may contain a rounding error to a certain base number. This is because the six different types of alcoholic drink and the different type of packages (see the list in Table 8.1b for details) serve as **an additional probing method** in the data collection. By setting the questionnaire in this "multi-layer" way, the respondents will less likely make up their answers since there will be a cross-consistency check in data collection and processing. For instances, a respondent claiming to drink a lot of bottled **alcopops** tends to drink **alcopops** in can, too, despite the preference of packaging discussed before. According to the latest information available, the types of alcohol drink have also changed over time. In 2001, beer, lager, and cider were still the most common drink (drunk by 70% of drinkers in the last week), but prevalence of alcopops had increased in recent years to reach 68% of drinkers in 2001. The proportion of those who had drunk spirits in the last week had increased from 35% in 1990 to 57% in 2001, whereas prevalence of drinking shandy, wine, or fortified wine in the last week have decreased in recent years (Boreham and Shaw 2002).

On the other hand, a respondent claiming to drink a lot of **wine in glass** is unlikely to drink a lot of other alcoholic drinks as well since there is a limit for a pupil can drink alcoholic drink in the last 7 days without affecting his health or school attendance. Moreover, wine and beer, for instance, are usually not considered as a substitute for each other despite both containing alcohol. The same also applies to other types of alcoholic drinks.

This **"built-in"** consistency check in the questionnaire used in the data collection will increase the accuracy and, therefore, quality of the data[6]. In other words, the quality of the data will be very different or less accurate if the question is only "how many pint/bottle/glass/can of alcoholic drink did you drink during last week?" without detailing the types of the alcoholic drink and packages. Given this fact and comparing with the way the number of cigarettes smoked data is collected, the quality of data on the amount of alcoholic drink consumed is expected to be more accurate than the number of cigarettes smoked, since there are no detailed information asked on the kinds of cigarettes and types of package of cigarettes smoked by the pupils.

8.2. Applications of the model on Cigarettes Smoked Data

Table 8.2 summarises the results of applying the model on the number of **cigarettes smoked data**, which are grouped into three different time periods, namely, for **everyday, weekdays, and weekends**.

The overall results show that:

(i) Detecting rounding base using a normal best-fit distribution function produces the highest probability values compared to other results of using other best-fit distribution functions. For instance, detecting rounding base on the everyday cigarettes consumption using log normal best-fit distribution

[6] A classic example on the effects of applying the right probing method can also be found in the labour force survey, in which a question of "Are you working" or "Did you work last week" will always produce lower employment level and therefore higher unemployment rates than their actual ones. These questions will more likely be excluding casual and other low or unpaid jobs since people tend to consider that "working" in the questions is only for a good paying job. In this context, the use of additional probing method—such as asking **what kind of activities conducted by respondent during the last week** and then determining whether the respondent is actually working or not based on the list of complete activities during the last week—will produce much more accurate employment and unemployment statistics, i.e., not understating and overstating employment and unemployment rates.

function will not detect the presence of rounding base 5, while the probability of detecting rounding base 5 using normal best-fit distribution function will be 0.3206. If uniform best-fit distribution function is used, the probability value will be 0.0311. For rounding base 10 of the same data set using the same three best fit of normal, lognormal, and uniform distribution functions will result in the probability values of 0.4535, 0.1808, and 0.0343, respectively.

The earlier finding of implementing the model using a simulated data set (Chapter 5) concludes that the use of the most appropriate best-fit distribution will guarantee best results, which are reflected in the most number of rounding bases detected with the highest probability values. This finding therefore suggests that the data on the number of cigarettes smoked by pupils age 11–15 in England tends to be distributed **normally**.

(ii) Rounding base 5 and its multiplication numbers seems very dominant as the rounding bases detected by the model. The normal best-fit distribution function detects rounding base 1 in the first iteration, followed by rounding bases 5, 10, and 20 in the second to four iterations. For the log normal best fit, the order of rounding base detected in four iterations is bases 1, 10, 20, and 15. The use of uniform best fit detects rounding bases 1, 5, and 10 in the first, second, and third iterations. This finding is further confirmed by the results of applying the **modulo test**, which shows increasing probability values on rounding base 5 and its multiple numbers.

(iii) Comparing the results across columns, consumption of cigarette for everyday, weekdays, and weekends and across the three best-fit distributions

of lognormal, normal, and uniform shows that the cigarette consumption for everyday demonstrates the clearest patterns of rounding base as shown by the highest probability values of rounding base 5 than the other two data sets. In the everyday consumption data set, the rounding base 1 is detected in the first iteration with the significant probability value of 0.1223, rounding base 5 in the second iteration with the significant probability value of 0.3206, rounding base 10 in the third iteration with the significant probability value of 0.4535, and rounding base 20 in the fourth iteration with the significant probability value of 0.4785. This performance cannot be matched by using other best-fit distribution functions and on other two data sets of weekdays and weekend consumption.

(iv) The uniform best-fit distribution performs very weakly in detecting the rounding pattern. The probability detections on some base numbers are very small, especially on the weekends and weekdays data. On the everyday consumption data, uniform best fit detects the least number of rounding bases, i.e. with only rounding bases 1; 5; and 10 in the first, second, and the third iterations and with the lowest probability values. This finding may be attributed to the characteristics of the data set, which is not uniformly distributed.

Table 8.2. Results of Detecting Rounding Patterns in the Number of Cigarettes Smoked Data by Using Different Best Fit Distributions

Best Fit Distribution	Iteration	Cigarettes Smoked					
		Everyday		Weekdays		Weekend	
		Base	Pdf	Base	Pdf	Base	Pdf
Normal	1	1	0.1223	5	0.0065	5	0.0043
	2	5	0.3206	10	0.0346	10	0.0245
	3	10	0.4535	1	0.1008	1	0.0631
	4	20	0.4785	20	0.1377	20	0.1044
	5			15	0.1592	15	0.1261
Lognormal	1	1	0.0907	5	0.0042	5	0.003
	2	10	0.1808	10	0.0175	10	0.0131
	3	20	0.2293	2	0.0433	2	0.0279
	4	15	0.27	15	0.0659	20	0.0441
	5	17	0.2738	20	0.09	15	0.0691
	6	25	0.2749	25	0.0913	25	0.0718
	7	23	0.2758				
Uniform	1	1	0.0229	5	0.0007	5	0.0005
	2	5	0.0311	10	0.0011	10	0.001
	3	10	0.0343	1	0.0016	1	0.0013
				20	0.0017	20	0.0016
						15	0.0016
Modulo Test	1	1	1	0.9999	1	1.0005	
	2	0.5595	2	0.6232	2	0.6485	
	3	0.2833	3	0.2901	3	0.2878	
	4	0.2888	4	0.2904	4	0.3273	
	5	0.4771	5	0.7049	5	0.7682	
	6	0.1377	6	0.1024	6	0.1167	
	7	0.0596	7	0.0408	7	0.0295	
	8	0.057	8	0.0404	8	0.0454	
	9	0.0415	9	0.0159	9	0.0146	
	10	0.3054	10	0.4527	10	0.5099	
	11	0.0286	11	0.0041	11	0.0063	
	12	0.068	12	0.0344	12	0.0313	
	13	0.0514	13	0.0083	13	0.0084	
	14	0.0137	14	0.0061	14	0.0066	
	15	0.1091	15	0.1756	15	0.2004	
	16	0.0101	16	0.0038	16	0.0047	
	17	0.0433	17	0.0087	17	0.0084	
	18	0.0188	18	0.0087	18	0.0087	
	19	0.0094	19	0.0004	19	0.0004	
	20	0.1486	20	0.2027	20	0.2546	
	21	0.0022	21	0	21	0.0004	
	22	0.0127	22	0.0015	22	0.0011	
	23	0.0228	23	0.0015	23	0.0022	
	24	0.0014	24	0.0004	24	0.0004	
	25	0.0202	25	0.0279	25	0.0437	

8.3. Applications of the model on Alcohol Consumption Data

Tables 8.3 to 8.5 summarise the results of applying the model on **alcohol consumption data**, which are grouped into 14 classifications based on the types of alcoholic drinks and the different kinds of packages or measurements.

The overall results show that:

(i) The lognormal best-fit distribution detects rounding bases with highest probability values, suggesting that the data sets tend to be distributed as lognormal. This distribution is different with the cigarettes consumption data which is distributed normally.

(ii) The uniform best-fit distribution performs very weak in detecting the rounding pattern. The probability detections on some base numbers are very small, even close to zero values. This can be observed in all kinds of alcoholic drink data of X1 to X14.

(iii) The patterns of rounding bases detected by the model for all types of drinks using different best fit distribution functions seem very unclear with the probability values mostly very small. If the probability values of detection are relatively high, the combinations of the base numbers identified are hard to believe to represent any rounding pattern. The applications of lognormal best fit on the variables X1, X3, X4, X5, X6, X10, X11, X12, X13, and X14, for instance, produce relatively high probability values, i.e. more than 50%, but the combinations of the base numbers detected are very inconsistent, making it very hard to conclude that there is a rounding pattern in the data sets.

(iv) Further examination using **modulo test** also shows the unclear pattern of rounding bases in all types of alcoholic drinks examined in this study. Only alcoholic drinks of X3, X5, X10, and X12 show a sign of rounding base

detection but again the probability values of the detections are very small and the base numbers detected are inconsistent, making these hard to be classified as a rounding pattern. For example, it is simply impossible for a data set to contain rounding errors to a consecutive series of base numbers such as from 20 to 25.

The results of detecting rounding base in the alcoholic consumption are, therefore, very different with the finding on the cigarettes consumption data, which shows a clear and consistent rounding base pattern. The overall finding seems to suggest that no rounding pattern that can be identified in the data sets of the alcohol consumption.

To some extent, this finding may be attributed to the way the data are collected, which is discussed before in Section 8.1. Basically, this is a positive impact of using detailed probing method in the data collection that significantly improves the quality of the data by primarily avoiding any kinds of "estimation" from the respondents' side. In addition to the detailed probing method, alcoholic consumption was only asked in one reference period of "the last 7 days," unlike in the cigarettes smoked everyday, during weekdays, and weekends. Therefore, there must be a kind of smoothing process from the respondents' side, i.e., in the form of "averaging" alcoholic drink consumption during the last 7 days. If the question were asked only for a specific time such as daily or last weekend or to follow the cigarettes smoked everyday, during weekdays and weekends, for instance, the quality of the data and therefore the rounding pattern would be very different.

In addition to the general finding above, there is also an interesting finding from the applications of the model on alcohol consumption data. Comparing rounding detection results of variables X1 and X2, variable X3 and X4, variables X6 and X7,

and variables X8 and X9 show that the measures of packages of the alcoholic drink have no important role or cannot be examined as mentioned and intended before. The different package such as **a pint** and **half pint**, and **large can** and **small can** for the same products, produce the same results of no rounding patterns. Therefore, whether respondents may prefer one particular package compared to the others is not clearly indicated. In addition to respondents' income and spending power, respondents' preference to one particular size, which is usually related to price, might be related to the respondents' characteristics such as age and buying place, etc. As commonly observed, young people tend to drink less expensive and sophisticated drink such as wine but they drink more popular drinks such as Alcopops and Beer.

Table 8.3. Results of Detecting Rounding Patterns on Variables X1 to X5 of Alcohol Consumption Data by Using Different Best Fit Distributions

Best Fit Distribution	Iteration	Alcohol Consumed									
		x1		x2		x3		x4		x5	
		Base	Pdf	Base	Pdf	Base	Pdf	Base	Pdf	Base	Pdf
Lognormal	1	1	0.839	1	0.4246	1	0.6293	1	0.6427	1	0.9451
	2	10	0.8426	6	0.4262	8	0.646	8	0.6433	10	0.952
	3	12	0.8451	8	0.4273	10	0.649	9	0.6437	12	0.955
	4	15	0.8455	7	0.4282	18	0.6492	10	0.6439	15	0.9559
	5	17	0.8456	11	0.4282	21	0.6492	12	0.644	20	0.956
	6	14	0.8456	10	0.4282	16	0.6493	11	0.644	16	0.9562
	7					18	0.6493			22	0.9562
	8									18	0.9562
	9									21	0.9562
Normal	1	1	0.0134	1	0.0002	1	0.0161	1	0.001	1	0.0485
	2	10	0.0135	6	0.0002	14	0.0161	8	0.001	12	0.0486
	3	12	0.0135	7	0.0002	15	0.0161	9	0.001	15	0.0486
	4	13	0.0135	8	0.0002	16	0.0161	10	0.001	14	0.0486
	5	14	0.0135	11	0.0002	18	0.0161	12	0.001	13	0.0486
	6	15	0.0135	10	0.0002	21	0.0161	11	0.001	16	0.0486
	7	17	0.0135	9	0.0002	24	0.0161	7	0.001	18	0.0486
	8	16	0.0135	5	0.0002	25	0.0161	6	0.001	20	0.0486
	9	11	0.0135	4	0.0002	23	0.0161			22	0.0486
	10					22	0.0161			21	0.0486
Uniform	1	1	0	1	0	1	0	1	0	1	0.0001
	2	10	0	6	0	21	0	8	0	12	0.0001
	3	12	0	8	0	24	0	9	0	15	0.0001
	4	9	0	7	0	22	0	10	0		
	5	15	0	1	0	23	0	12	0		
	6	13	0			25	0	1	0		
	7	14	0			14	0				
	8	1	0			16	0				
	9					18	0				
	10					15	0				
Modulo Test	1	1	1	1	1	1	1	1	1	1	1
	2	0.4668	2	0.3109	2	0.5336	2	0.3205	2	0.4882	
	3	0.1919	3	0.1154	3	0.1766	3	0.141	3	0.25	
	4	0.128	4	0.0705	4	0.2015	4	0.0641	4	0.1401	
	5	0.0711	5	0.0224	5	0.0672	5	0.0256	5	0.1047	
	6	0.045	6	0.016	6	0.0557	6	0.0064	6	0.0602	
	7	0.0237	7	0.0096	7	0.025	7	0	7	0.0288	
	8	0.0166	8	0.0128	8	0.0461	8	0.0064	8	0.017	
	9	0.0095	9	0	9	0.0038	9	0.0064	9	0.0092	
	10	0.0142	10	0	10	0.0192	10	0.0064	10	0.0236	
	11	0	11	0.0032	11	0.0038	11	0	11	0.0052	
	12	0.0118			12	0.0077	12	0.0064	12	0.0118	
	13	0.0024			13	0			13	0.0013	
	14	0.0024			14	0.0038			14	0.0013	
	15	0.0047			15	0.0019			15	0.0065	
	16	0			16	0.0038			16	0.0026	
	17	0.0024			17	0			17	0	
					18	0.0038			18	0.0013	
					19	0			19	0	
					20	0.0019			20	0.0039	
					21	0.0019			21	0	
					22	0			22	0.0013	
					23	0					
					24	0.0019					
					25	0					

Table 8.4. Results of Detecting Rounding Patterns on Variables X6 to X10 of Alcohol Consumption Data by Using Different Best Fit Distributions

Best Fit Distribution	Iteration	Alcohol Consumed									
		x6		x7		x8		x9		x10	
		Base	Pdf	Base	Pdf	Base	Pdf	Base	Pdf	Base	Pdf
Lognormal	1	1	0.7398	1	0.2484	1	0.18	1	0.4768	1	0.6287
	2	4	0.7442	7	0.2508	4	0.2035	6	0.4931	10	0.9295
	3	7	0.7478	6	0.2528	5	0.2066	8	0.4975	15	0.9297
	4	8	0.7499	8	0.2532	6	0.2076	11	0.498	20	0.9297
	5	10	0.7509			7	0.2084			14	0.9297
	6									25	0.9297
	7									24	0.9297
	8									23	0.9297
	9									22	0.9297
	10									21	0.9297
Normal	1	1	0.0031	1	0	1	0	1	0.0029	1	0.007
	2	7	0.0031	6	0	6	0	8	0.0029	13	0.007
	3	8	0.0031	7	0	7	0	9	0.0029	12	0.007
	4	10	0.0031	8	0	5	0	11	0.0029	14	0.007
	5	9	0.0031	5	0	4	0	10	0.0029	15	0.007
	6			4	0	3	0			20	0.007
	7			3	0	2	0			21	0.007
	8			2	0					22	0.007
	9									23	0.007
	10									24	0.007
Uniform	1	1	0	1	0	1	0	1	0	1	0
	2	7	0	7	0	4	0	6	0	20	0
	3	8	0	5	0	5	0	8	0	16	0
	4	10	0	6	0	1	0	7	0	17	0
	5	1	0	1	0			9	0	18	0
	6							1	0	19	0
	7									21	0
	8									22	0
	9									23	0
	10									24	0
Modulo Test	1	1	1	1	1	1	1	1	1	1	1
	2	4203	2	0.2925	2	0.2868	2	0.4189	2	0.5076	
	3	0.1014	3	0.0566	3	0.0662	3	0.1486	3	0.2085	
	4	0.1014	4	0.0566	4	0.0809	4	0.1216	4	0.0952	
	5	0.0154	5	0.0189	5	0.0294	5	0.0473	5	0.0634	
	6	0	6	0.0189	6	0.0147	6	0.0473	6	0.0453	
	7	0.0145	7	0.0283	7	0.0147	7	0.0135	7	0.0287	
	8	0.0145	8	0.0094			8	0.0203	8	0.0151	
	9	0					9	0.0068	9	0.0045	
	10	0.0145					10	0	10	0.0151	
							11	0.0068	11	0.0015	
									12	0.0015	
									13	0.0015	
									14	0.003	
									15	0.0045	
									16	0	
									17	0	
									18	0	
									19	0	
									20	0.0045	
									21	0	
									22	0	

Table 8.5. Results of Detecting Rounding Patterns on Variables X11 to X14 of Alcohol Consumption Data by Using Different Best Fit Distributions

Best Fit Distribution	Iteration	Alcohol Consumed							
		x11		x12		x13		x14	
		Base	Pdf	Base	Pdf	Base	Pdf	Base	Pdf
Lognormal	1	1	0.7087	1	0.9249	1	0.6423	1	0.9342
	2	10	0.7093	10	0.9332	10	0.6453	12	0.9373
	3			12	0.9342	9	0.6474	10	0.939
	4			15	0.9346	8	0.6484	15	0.94
	5			20	0.9346	12	0.6487	20	0.9401
	6			25	0.9346			24	0.9401
	7			23	0.9346			23	0.9401
	8			24	0.9346			22	0.9401
	9			22	0.9346				
	10			21	0.9346				
Normal	1	**1**	**0.0399**	**1**	**0.0592**	**1**	**0.0065**	**1**	**0.0672**
	2	6	0.0401	12	0.0593	8	0.0065	12	0.0674
	3	10	0.0401	13	0.0593	9	0.0065	15	0.0674
	4	9	0.0401	15	0.0593	10	0.0065	14	0.0674
	5	8	0.0401	14	0.0593	12	0.0065	16	0.0674
	6	7	0.0401	20	0.0593	11	0.0065	17	0.0674
	7			22	0.0593			18	0.0674
	8			23	0.0593			20	0.0674
	9			24	0.0593			24	0.0674
	10			25	0.0593			21	0.0674
Uniform	1	1	0	1	0	1	0	1	0.0001
	2	6	0	21	0	10	0	15	0.0001
	3	10	0	22	0	8	0	20	0.0001
	4	7	0	23	0	9	0	13	0.0001
	5	1	0	24	0	12	0		
	6			25	0	1	0		
	7			15	0				
	8			20	0				
	9			14	0				
	10			16	0				
Modulo Test	**1**	1	**1**	1	**1**	1	**1**	1
	2	0.4122	2	0.5155	2	0.497	2	0.495
	3	0.1757	3	0.2547	3	0.1361	3	0.254
	4	0.1081	4	0.1584	4	0.1183	4	0.1538
	5	0.0541	5	0.1046	5	0.0533	5	0.1079
	6	0.0203	6	0.0569	6	0.0237	6	0.0581
	7	0	7	0.0186	7	0	7	0.0337
	8	0	8	0.0217			8	0.0268
	9	0	9	0.0083			9	0.0107
	10	**0.0068**	**10**	**0.0238**			**10**	0.0191
			11	0.0062			11	0.0038
			12	0.0083			**12**	0.013
			13	0.0041			13	0.0008
			14	0.001			14	0.0008
			15	0.0052			**15**	0.0069
			16	0			16	0.0008
			17	0			17	0.0008
			18	0			18	0.0008
			19	0			19	0
			20	0.0021			20	0.0031
			21	0			21	0
			22	0			22	0
			23	0			23	0
			24	0			24	0.0008
			25	0				

CHAPTER IX

MAIN FINDINGS AND FURTHER RESEARCH

9.1. Summary and Main Findings

- **Rounding Issues**

Unspecified counting practices creating rounding to the nearest "convenient" or "based" number and "prediction" of digit preference, are basically estimated data that can have serious consequences on data quality. This is an important issue that attracts research interests.

Statistical methods for analysing missing-data are commonly used for dealing with rounded data, but they are not missing data. Rounded data are shifted or lumped to certain based number(s) reflected by rounding process or enumerators' counting behaviour. The missing-data technique application to replace may therefore further distort the rounded data.

Accordingly, a new methodology to deal with rounded data is needed. Crockett and Crockett (1998) introduced the modulo test model to detect and analyse rounding contained in a data set. Whilst the model was sufficient for its original purpose, it has constraints in dealing with large data sets. Its implementation is also complex involving a series of statistical tests. Therefore, a new technique that could overcome the drawbacks would be useful. The model developed in this study fills this gap and serves the purpose.

- **Modelling Development**

The study develops a neural network model to detect, analyse, and quantify the periodic structure present in a data set because of rounding. The model is developed

using Artificial Intelligent (AI) technique of Radial Basis Function (RBF) neural network. The modelling development and applications can be summarised as follows:

(i) considering a data set as having rounding following certain patterns, the main concern here is how to employ pattern-recognition method to analyse the data set;

(ii) developing AI based pattern-recognition techniques to recognize and classify rounding as a spike in pattern characteristics;

(iii) developing neural network to detect the existence of rounding patterns or periodic structure in a data set;

(iv) assessing the model's goodness of fit by applying on simulated data sets containing rounding to certain base numbers of different rounding levels, as well as comparing with modulo test results; and

(v) applying the model on real data sets of the religious census data and the data on cigarettes smoked and alcohol consumed. In all cases, the modeling results are also compared with the benchmark of modulo test results;

- **Modelling Assessments**

Theoretical and numerical assessments using specific and simulated data sets are carried out to clarify the model's behaviour in detecting rounding. These include developing robustness indicators comparable across different scenarios based on a combination of data distribution, best fits used in the detection, and rounding bases numbers.

The first assessment using specific data set is to check the model's "logical framework" or "brain" by emulating the modulo test results. The second assessment is for a "positive verification" by detecting different kinds of rounding base numbers systematically introduced in **uniformly** and **normally** distributed simulated data sets.

In all the assessments, the model performs very well, making it suitable for implementations on real data sets. The model detects the existing rounding bases and the overall results are consistent with initial expectations, hence confirming the model's goodness fit.

- **Lessons Learned**

The assessment results using simulated data sets show the importance of using the right best-fit in detecting rounding. Using the wrong best-fit could result in:

(i) Detecting the correct rounding bases but with lower probability values of detections;

(ii) Detecting the correct rounding bases but with a different order of detections in the iterations;

(ii) Detecting the wrong rounding base number such as the factor number. Instead of detecting the rounding base 10, the model detects rounding bases 2 and 5;

(iii) Failing to detect any rounding base number actually present in a data set; and

(iv) Different combinations of (i) to (iii) above.

Given the importance of using the right best-fit, the suggested procedure for real applications is therefore:

(i) Explore the data set to find its underlying distribution function.

(ii) Use the right best-fit consistent with the data exploration results.

(iii) Apply the model to detect rounding base number.

(iv) Correct the data distribution by taking the findings in (iii) into account.

The suggested procedure is however assuming that only a single detection is allowed. If one can afford to use various best-fits in the detections, the underlying distribution of the data set under investigation will be reflected in the best results.

Therefore, the exploration to find its underlying distribution function of the data set (i.e. the first step) is not really necessary.

Moreover, modeling application on data significantly distorted by rounding (i.e. more than 40%) should be carried out very carefully since the results may contain inconsistent detection patterns such as detecting not only the actual rounding base number but also its factor numbers. On the other hand, a low level of rounding in a data set may also create unclear detection patterns. Therefore, determining the cut-off points on which the model can no longer detect the existing rounding in a data set is important. This study estimates 24 cut-off points of each possible case using direct investigation and regression methods. The estimated cut-off points range from 4% to 28% of rounding levels that depend not only on the data size and distribution but also on the best fit used and rounding base numbers.

On the base number, detecting prime base numbers seems easier than non-prime base numbers because of unique characteristics of the prime numbers. The higher the number, the easier is the detection, i.e. detecting rounding base 11 is easier than base 7, and so on. For the non-prime number, the opposite is the case, as the bigger the number, the more would be the factors associated with the number. Detecting rounding base 10, for instance, may also detect its factor numbers of 5 and 2, especially if the data sets contain significant rounding.

Comparing results across different best fits, applying uniform best fit on uniform data produce the best result and the lowest cut-off point, indicating the most effective detection. Using uniform best fit on normal data still produces the second best result since it is more effective in determining the cut-off point than using normal best fit on normal data. Moreover, the consequences of using wrong best fit are more severe on uniform data than on normal data. This suggests that for uniform data the

uniform best fit must be used. The uniform best fit can also safely be used in case the data distribution is not known.

As a rule of thumb, the model seems to perform best on data sets containing 10% to 40% rounding. On the latter, the model can detect the rounding well but the pattern might be distorted by its factor numbers. The modulo test method also suffers from the same problem.

The summary below provides a result comparison of modelling applications on simulated uniform and normal data containing certain rounding bases of different levels using three different best fits of uniform, normal, and lognormal.

Data Distribution	Base 5	Base 7	Base 11	Base 10
Uniform	(i) Detecting no rounding base 5 on data sets A and B with lognormal best fit; (ii) Correct best fit produces the highest detection power, followed by using normal and lognormal best fits; (iii) Detection power increases significantly on data set E; and (iv) No detection error	(i) Detecting no rounding base 7 on data set A using normal and lognormal best fits; (ii) Uniform best fit produces the highest detection power, followed by normal and lognormal best fits. (iii) Detection power increases significantly on data set E; and (iv) No detection error	(i) Detecting no rounding base 11 on data set A using lognormal best fit; (ii) Uniform best fit produces the highest detection power, followed by normal and lognormal; (iii) Detection power on data sets A to E increases gradually following a smooth trend; and (iv) No detection error	(i) Detecting no rounding base 10 on data set A using lognormal best fit; (ii) Uniform best fit produces the best results, followed by normal and lognormal; (iii) Detection power on data sets A to E increases gradually; and (iv) Detection errors on data set C using lognormal best fit and on data sets D and E using all best fits.
Normal	(i) Detecting no rounding base 5 on data set A using all best fits; (ii) Normal best fit produces the highest detection power, followed by lognormal and uniform; (iii) Detection power on data sets A to E increases gradually; and (iv) No detection error.	(i) Detecting no rounding base 7 on data set A using normal best fit; (ii) Normal best fit produces the highest detection power, followed by lognormal and uniform; (iii) Detection power on data sets A to E increases gradually; and (iv) No detection error	(i) Detecting rounding base 11 on all cases; (ii) Normal best fit produces the highest detection power, followed by lognormal and uniform; (iii) Detection power on data sets A to E increases gradually; and (iv) No detection error.	(i) Detecting rounding base 10 in all cases; (ii) Normal best fit produces the highest detection power, followed by lognormal and uniform; (iii) Detection power on data sets A to E increases gradually; and (iv) Detection errors on data set C using normal and lognormal best fits and data sets D and E in all cases.

- **Modelling Applications on the Real Data Sets**

Applying the model on the religious census data confirms the earlier finding that the data contain rounding base 5. Other applications on data of cigarettes smoked and alcohol consumed by secondary school pupils (aged 11–15) in England (result of the first Survey of Smoking, Drinking and Drug use among secondary school pupils conducted in England, UK in 2001) show good detection results. The number of cigarettes smoked data seems to contain rounding to base 5, while the amount of alcohol consumed data contain no rounding.

Given the respondents and enumerators of the two data sets are the same, the different results can be attributed to the different ways of collecting the two data sets. The number of cigarettes smoked was asked during weekend, weekdays, and everyday without any reference to the cigarettes types and their different packages. On the other hand, the alcohol consumption was asked for different types of alcoholic drinks and different packages such as a pint and half pint, large and small cans, glass, and bottle. Therefore, the data quality of the cigarettes smoked is inferior than alcohol consumption due to the questionnaire designs.

Using different packages of the same products in a data collection brings to important issues of packaging role and that makes some people prefer one particular package than others. If people are indifferent about different packages, the rounding detection between the two different packages of the same product would not be much significant. Unfortunately, no rounding is detected in the data so that this issue cannot be examined further.

9.2. Further Research

The model developed in this study is useful for analysing rounding patterns in a data set. Its applications on real data shows the importance of using the correct best-fit and a built-in consistency check in a data collection to increase the data quality.

From a practical point of view, the modelling development and applications are still relatively cumbersome. Further refinements would therefore be desirable, especially in integrating them in an interactive way of a user-friendly modelling tool. The integration could also include data exploratory analysis and smoothing process so that the integrated system can examine the rounding patterns and come up with a refined data free from rounding. The modeling applications can also be implemented in different areas where rounding is common and can have significant effects.

BIBLIOGRAPHY

1. Araujo, F., B. Ribeiro, and L. Rodrigues. 2001. **A Neural Network for Shortest Path Computation.** *IEEE Transactions on Neural Networks,* Volume 12, No. 5, September.
2. Argos, J.A. 1999. **The Developing of A Neural Network in Order to Improve the Extraction of Formants from Demysillables**. University of Nottingham.
3. Ballou, D., and G. Tayi. 1999. **Enhancing data quality in data warehouse environments.** Communications of the ACM. 42: 73–78.
4. Bianchini, M., P. Frasconi, and M. Gori. 1995. **Learning Without Local Minima in Radial Basis Function Networks**. *IEEE Transactions on Neural Networks*, Volume 6, No. 3, pp. 749–756. May.
5. Bishop, C.M., and G.E. Hinton. 1995. **Neural Network for Pattern Recognition.** Clarendon Press. ISBN: 0198538642.
6. Boreham, R., and A. Shaw A. 2002. **Drug Use, Smoking and Drinking Among Young People in England in 2001**. The Stationary Office: London.
7. Box, G.E.P., and G.M. Jenkins. 1970. **Time Series Analysis: Forecasting and Control**, Holden-Day. San Francisco: CA.
8. Box, G.E.P., G.M. Jenkins, and G.C. Reinsel. 1994. **Time Series Analysis: Forecasting and Control**. 3rd ed. Prentice Hall, Englewood Clifs: NJ.
9. Brown, J., and A. Light. 1989. **Interpreting Panel Data on Job Tenure**. Working Paper, Department of Economics: State University of New York at Stony Brook.

10. Browne, A., and P.D. Picton. 1999. **Two Analysis Techniques for Feedforward Networks, Behaviormetrika: Special Issue on Analysis of Knowledge Representations by Neural Network Models**. 26, 1, pp. 75–87.

11. Budd, J.W., and T. Guinnane. 1991. **Intentional Age–Misreporting, Age-Heaping, and the 1908 Old Age Pensions Act in Ireland**. *JSTOR, Population Studies*, Volume 45, Issue 3, pp. 497–518. November.

12. Chen, S., C.F.N. Cowan, and P.M. Grant. 1991. **Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks**. *IEEE Transactions on Neural Networks*, Volume 2, No. 2, pp. 302–309. March.

13. Chui, C.K., X. Li, and H.N. Mhaskar. 1994. **Neural Networks for Localised Approximation**. *JSTOR, Mathematics of Computation*, Volume 63, No. 208, pp.607–623. October.

14. Coale, A.J., and S. Li. 1991. **The Effect of Age Misreporting in China on the Calculation of Mortality Rates at Very High Ages**. *JSTOR, Demography*, Volume 28, Issue 2, pp. 293–301. May.

15. Cox, L.H. 1987. **A Constructive Procedure for Unbiased Controlled Rounding**. *Journal of the American Statistical Association*, Volume 82, Issue 398, pp. 520–524. June.

16. Crockett, R.G.M., and A.C. Crockett. 1998 **Historical Sources: How People Counted. A Method for Estimating the Rounding of Numbers**. *History and Computing*. 9, 1/2/3, pp. 43–57.

17. Crockett, R.G.M., and A.C. Crockett. 2000. **People and Counting: How People Count in Enumeration Exercises**. *3rd European Social Science and History Conference*. Vrije Universiteit: Amsterdam. April.

18. Dayhoff, J.E. 1990. **Neural Network Architectures, An Introduction**. Van Nostrand Reinhold: New York.

19. Dempster, A.P., and D.P. Rubin. 1983. **Rounding Error in Regression: The Appropriateness of Sheppard's Corrections**. *Journal of the Royal Statistical Society. Series B (Methodological)*, Volume 45, Issue 1, pp. 51–59.

20. Demuth, H., and M. Beale. 1998. **Neural Network Toolbox, For Use with Matlab.** User's Guide Version 3. The MathWorks Inc.

21. Demuth, H., M. Beale, and M. Hagan. 2006. **Neural Network Toolbox, For Use with Matlab**. User's Guide Version 5. The MathWorks Inc.

22. Faraway, J., and C. Chatfield. 1998. **Time Series Forecasting with Neural Networks: A Comparative Study Using Airlines Data**. *JSTOR, Applied Statistics*, Volume 47, No. 2 pp. 231–250.

23. Hansen, J.V., J.B. McDonald, and R.D. Nelson. 1999. **Time Series Prediction with Genetic-Algorithm Designed Neural Networks: An Empirical Comparison with Modern Statistical Models**. *Journal of Computational Intelligence*, Volume 15, Number 3.

24. Hebb, D.O. 1949. **The Organisation of Behaviour: A Neuropshychological Theory**. John Wiley & Sons: New York.

25. Heitjan, D.F., and D.B. Rubin. 1990. **Inference from Coarse Data via Multiple Imputation with Application to Age Heaping**. *Journal of the American Statistical Association*, 85, 410, pp. 304–314.

26. Heitjan, D.F., and D.B. Rubin. 1991. **Ignorability and Coarse Data**, *Annals of Statistics*, 19, 4, pp. 105–117.

27. Huff, D. 1954. **How to Lie with Statistics**. W.W. Norton & Company: New York.

28. James, H. 1994. **Software for Studying and Developing Applications of Artificial Neural Networks**. *JSTOR, The Economic Journal*, Volume 104, No. 422, pp. 181–196. January.

29. Kay, J.W., and D.M. Titterington. 1999. **Statistics and Neural Network**. Oxford University Press, Inc.: New York.

30. Kay, J.W., and D.M. Titterington. 2000. **Statistics and Neural Network: Advance and Interface**. Oxford University Press, Inc.: New York.

31. Klesges, R.C., M. Debon, and J.W. Ray. 1995. **Are Self-Reports of Smoking Rate Biased? Evidence from the Second National Health and Nutrition Examination Survey**. *PERGAMON. J Clin Epidemiol,* Volume 48, No. 10, pp. 1225–1233. Elsevier Science Ltd.

32. Leonard, J.A., M.A. Kramer, and L.H. Ungar. 1992. **Using RBFs to Approximate A Function and Its Error Bounds**. *IEEE Transaction on Neural Networks*. July.

33. Levine, D.M., D. Stephan, T.C. Krehbiel, and M.L. Berenson. 2002. **Statistics for Managers Using Microsoft Excel**. Third Edition, Prentice Hall International Inc. ISBN 0-13-097082-4.

34. Little, R.J.A., and D.B. Rubin. 1987. **Statistical Analysis with Missing Data**. John Wiley & Sons: New York. ISBN 0-471-80254-9.

35. Looney, C.G. 1997. **Pattern Recognition Using Neural Networks. Theory and Algorithm for Engineers and Scientists**. Oxford University Press.

36. Lowe, D. 2000. **Radial Basis Function Networks and Statistics: Advance and Interface**. Neural Computing Research Group. Aston University, Aston Triangle: Birmingham, UK.

37. Makridakis, S., S.C. Wheelwright, and R.J. Hyndman. 1998. **Forecasting:**

Methods and Applications. John Willey & Sons: New York.

38. McCulloch, W., and W. Pitts. 1943. **A Logical Calculus of Ideas Immanent in Nervous Activity**. *Bulletin of Mathematical Biophysics,* 5:115–133.

39. Navia-Vazquez, A., F. Perez-Cruz, A. Artes-Rodriguez, and A.R. Figueiras-Vidal. 2001. **Weighted Least Squares Training of Support Vector Classifiers to Compact and Adaptive Schemes**. *IEEE Transactions on Neural Networks*, Volume 12, No. 5, September.

40. Orr, M.J. 1998. **Optimising the Widths of Radial Basis Functions**. Centre for Cognitive Science. Edinburgh University, Edinburgh: Scotland, UK.

41. Picton, P. 2000. **Neural Networks**. 2nd Edition. Palgrave: New York.

42. Polhill J.G., and M.K. Weir. 2001. **An Approach to Guaranteeing Generalisation On Neural Networks**. *Pergamon, Neural Networks,* 14 pp. 1035–1048.

43. Poli, I., and R.D. Jones. 1994. **A Neural Net Model for Prediction**. *JSTOR, Journal of The American Statistical Association*, Volume 89, No. 425, pp. 117–121. March.

44. Rosenblatt, F. 1958. **The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain**. *Psychological Review,* 65, pp. 386–408.

45. Swingler, K. 1996. **Applying Neural Networks: A Practical Guide**. Academic Press. Harcourt Brace & Company Publisher: London.

46. Triastuti, E., E.G.M. Crockett, P.D. Picton, and A.C. Crockett. 2002. **Neural Network Analysis of Estimated Data**. *Proc. Eunite 2002*, pp. 161-166. Albufeira, Portugal. 19–21 September .

47. Triastuti, E., R.G.M. Crockett, P.D. Picton, and A.C. Crockett. 2002. Neural **Network Analysis of Estimation of Data**. *Proc. 4th International Conference on Recent Advance in Soft Computing (RASC).* Nottingham. 12–13 December.

48. Tricker, A.R. 1984. **Effects of Rounding on the Moments of a Probability Distributions**. *Annals of Statistics*, Volume 33, Issue 4.

49. Turner, S.J., R.G.M. Crockett, P.D. Picton, E. Triastuti. 2001. **Genetic Algorithms for Simulating Counting Behaviour**. *Proc. 19th Biennial Conference on Numerical Analysis*. University of Dundee. 26–29 June 2001.

50. Turner, S.J., E. Triastuti, R.G.M. Crockett, P.D. Picton, and A.C. Crockett. 2002. **Intelligent Techniques for Detecting Estimated and Falsified Data**. *Proc. Sixth Multi-Conference on Systemics, Cybernetics and Informatics*. pp. 445-450. Orlando, Florida: USA. 14–18 July.

51. Warner, B., and M. Misra. 1996. **Understanding Neural Network as Statistical Tools**. *JSTOR, The American Statistician*, Volume 50, No. 4, pp. 284–293. November.

52. Wasserman, P.D. 1993. **Advanced Methods in Neural Computing**. New York:Van Nostrand Reinhold.

53. Welstead, S.T. 1994. **Neural Network and Fuzzy Logic Applications in C/C++**. John Wiley & Sons, Inc.

54. Widrow, B., and M.E. Hoff. 1960. **Adaptive Switching Circuit**. Wescon Convention Record, Part IV, pp. 96–104.

55. Wozniakowski, H. 1974. **Rounding Error Analysis for the Evaluation of a Polynomial and Some of its Derivatives**. *SIAM Journal of American*

Analysis, Volume 11, Issue 4, pp. 780–787. June.

56. Yiu, K.F.C., S. Wang, K.L. Teo, and A.C. Tsoi. 2001. Nonlinear System Modeling via Knot-Optimizing B-Spline Networks. *IEEE Transactions on Neural Networks*, Volume 12, No. 5. September.

APPENDICES

Appendix 1:

```
function net=EnewpnnCP(p,t,spread)

if nargin < 2, error('Not enough input arguments'), end

% Defaults
if nargin < 3, spread = 0.1; end

% Dimensions
[R,Q] = size(p);
[S,Q] = size(t);

% Architecture
net = network(1,2,[1;0],[1;0],[0 0;1 0],[0 1]);

% Simulation
net.inputs{1}.size = R;
net.inputWeights{1,1}.weightFcn = 'dist';
net.layers{1}.netInputFcn = 'netprod';
net.layers{1}.transferFcn = 'radbas';
net.layers{1}.size = Q;
net.layers{2}.size = S;
net.layers{2}.transferFcn = 'ECompProb';

% Weight and Bias Values
net.b{1} = zeros(Q,1)+sqrt(-log(.5))/spread;
net.iw{1,1} = p';
net.lw{2,1} = t;
```

Appendix 1a:

```
function a = ECompProb(n,b)

if nargin < 1,error('Not enough arguments.'); end

% FUNCTION INFO
if isstr(n)
 switch (n)
  case 'deriv',
   a = '';
  case 'name',
   a = 'Competitive';
  case 'output',
   a = [0 maxn];
  case 'active',
   a = [-inf inf];
  case 'type',
   a = 2;

   % **[ NNT2 Support ]**
  case 'delta',
   a = 'none';
    nntobsu('compet','Use COMPET("deriv") instead of COMPET("delta").')
  case 'init',
   a = 'midpoint';
    nntobsu('compet','Use network propreties to obtain initialization info.')

  otherwise
   error('Unrecognized code.')
 end
 return
end

% CALCULATION

% **[ NNT2 Support ]**
if nargin == 2
 nntobsu('compet','Use COMPET(NETSUM(Z,B)) instead of COMPET(Z,B).')
 n = n + b(:,ones(1,size(n,2)));
end

[S,Q] = size(n);
[maxn,indn] = max(n);
b = sparse(indn,1:Q,maxn,S,Q);          %outputs of maximum values
%a=n;                                   %outputs of all values
a = [[vec2ind(b)] [maxn]];              %base number and probability outputs
```

Appendix 2:

```
function Y1 = Enormal1(DUI,base_unit)

%Arguments
%  DUI : a column vector of data under investigation, the values are [0 1]
%  p1  : a column vector of best fit log normal distribution of the DUI
%  base_unit : a row vector of investigated base unit

if nargin < 2, error('Not enough input arguments'),end
ln_dist = Enormfit(DUI);
p1=ln_dist(:,3);

P=Etrainingdata(DUI,base_unit,p1);
[r,c]=size(P);
s=P;
Tc=(1:1:c)';
T=ind2vec(Tc);
net=EnewpnnCP(s,T);
Y1=sim(net,DUI);

%Dimension
%b=size(base_unit,2);    %number of base unit
%n=size(DUI,1);          %number of data under investigation
%d=zeros(n,b);           %assingning dummy variable

%Creating dummy variable
%for j=1:b
%   for i=base_unit(j):base_unit(j):n
%      d(i,j)=1;
%   end
%end

%d = Edummydata(base_unit,DUI);

%Creating training data
%for i=1:j
%   P(:,i)=p1.*d(:,i);
%end
```

Appendix 3:

```
function Y1 = Eln1(DUI,base_unit)

%Arguments
%  DUI : a column vector of data under investigation, the values are [0 1]
%  p1  : a column vector of best fit log normal distribution of the DUI
%  base_unit : a row vector of investigated base unit

if nargin < 2, error('Not enough input arguments'),end
ln_dist = Elognormfit(DUI);
p1=ln_dist(:,3);

P=Etrainingdata(DUI,base_unit,p1);
[r,c]=size(P);
s=P;
Tc=(1:1:c)';
T=ind2vec(Tc);
net=EnewpnnCP(s,T);
Y1=sim(net,DUI);
```

Appendix 4:

```
function Y1 = Euniform1(DUI,base_unit)

%Arguments
%  DUI : a column vector of data under investigation, the values are [0 1]
%  p1  : a column vector of best fit log normal distribution of the DUI
%  base_unit : a row vector of investigated base unit

if nargin < 2, error('Not enough input arguments'),end
uniform_dist = Euniformfit(DUI);
p1=uniform_dist(:,3);

P=Etrainingdata(DUI,base_unit,p1);
[r,c]=size(P);
s=P;
Tc=(1:1:c)';
T=ind2vec(Tc);
net=EnewpnnCP(s,T);
Y1=sim(net,DUI);
```

Appendix 5:

```
function P = Etrainingdata(DUI,base_unit,p1)

%if nargin < 2, error('Not enough input arguments'),end

%Dimension
b=size(base_unit,2);    %a vector which contains base unit elements
n=size(DUI,1);
d=zeros(n,b);

for j=1:b
  for i=base_unit(j):base_unit(j):n
    d(i,j)=1;
    P(:,j)=p1.*d(:,j);
  end
end
```

Appendix 6:

```
function nl_dist = Enormfit(DUI)

% Generates best-fit normal distribution.
% Prints log_mean, log_stdev & correlation coeff (as goodness-of-fit).
%    ...
% Description: normally-random real arrays.
% Call:       nl_dist = norm_fit (fd_data)
% Parameters:  nl_dist is the returned probability vector.
%              - col 1 - size(x-axis)
%              - col 2 - best-fit normal prob dist (normalised)
%              - col 3 - scaled col-2 (same total frequency as fd_data))
%             fd_data is the input (observed freq dist)
%              - if one col, assumes frequencies at unit-size steps
%              - if two col, assumes size-freq x-y pairs
%                in order of increasing size
%
% obtain two col-vectors; sizes 1:max_size, frequencies
% Author:  Robin Crockett (robin.crockett@northampton.ac.uk)


[n_r,n_c] = size(DUI);
if(n_c==1)
 max_size = n_r;
 o_size = (1:n_r)';                % size (x-data)
 o_freq = DUI;                     % frequency of size
else
 max_size = DUI(n_r,1);
 o_size = (1:max_size)';
 o_freq = zeros(max_size,1);
 f_k = 1;
 for s_i = 1:max_size
  for f_i = f_k:n_r
   if(s_i==DUI(f_i,1));
    o_freq(s_i) = DUI(f_i,2);     %results are the same as fd_data
    f_k = f_i;
    continue;
   end
  end
 end
end

% normal stuff

f_tot = sum(o_freq);

o_mean = sum(o_freq.*o_size)/f_tot;      % log_mean
o_szmn = o_size - o_mean;
o_varc = sum(o_freq.*o_szmn.^2)/f_tot;   % log_variance
```

```
o_sdev = sqrt(o_varc);                % log_stdev

% op stuff

tps = o_sdev*sqrt(2*pi);

nl_dist = zeros(max_size,3);
nl_dist(:,1) = (1:max_size)';         % all sizes 1:max_size
for s_i = 1:max_size
  nl_dist(s_i,2) = exp(-(((s_i-o_mean)/o_sdev)^2)/2)/tps;
end
nl_dist(:,2) = nl_dist(:,2)/sum(nl_dist(:,2));
nl_dist(:,3) = nl_dist(:,2)*f_tot;
```

Appendix 7:

```
function ln_dist = Elognormfit(DUI)

% Generates best-fit lognormal distribution.
% Prints log_mean, log_stdev & correlation coeff (as goodness-of-fit).
%    ...
% Description: Lognormally-random real arrays.
% Call:       ln_dist = lognorm_fit (fd_data)
% Parameters:  ln_dist is the returned probability vector.
%              - col 1 - size(x-axis)
%              - col 2 - best-fit lognormal prob dist (normalised)
%              - col 3 - scaled col-2 (same total frequency as fd_data))
%             fd_data is the input (observed freq dist)
%              - if one col, assumes frequencies at unit-size steps
%              - if two col, assumes size-freq x-y pairs
%                in order of increasing size
% obtain two col-vectors; sizes 1:max_size, frequencies
% Author:  Robin Crockett (robin.crockett@northampton.ac.uk)


[n_r,n_c] = size(DUI);
if(n_c==1)
 max_size = n_r;
 o_size = (1:n_r)';                % size (x-data)
 o_freq = DUI;                     % frequency of size
else
 max_size = DUI(n_r,1);
 o_size = (1:max_size)';
 o_freq = zeros(max_size,1);
 f_k = 1;
 for s_i = 1:max_size
  for f_i = f_k:n_r
   if(s_i==DUI(f_i,1));
    o_freq(s_i) = DUI(f_i,2);
    f_k = f_i;
    continue;
   end
  end
 end
end

 % lognormal stuff

 lg_size = log(o_size);
 f_tot = sum(o_freq);

 lg_mean = sum(o_freq.*lg_size)/f_tot;     % log_mean
 lg_szmn = lg_size - lg_mean;
 lg_varc = sum(o_freq.*lg_szmn.^2)/f_tot;  % log_variance
```

```
lg_sdev = sqrt(lg_varc);                % log_stdev

% op stuff

tps = lg_sdev*sqrt(2*pi);

ln_dist = zeros(max_size,3);
ln_dist(:,1) = (1:max_size)';           % all sizes 1:max_size
for s_i = 1:max_size
  ln_dist(s_i,2) = exp(-(((log(s_i)-lg_mean)/lg_sdev)^2)/2)/(s_i*tps);
end
ln_dist(:,2) = ln_dist(:,2)/sum(ln_dist(:,2));
ln_dist(:,3) = ln_dist(:,2)*f_tot;

% correlation

%d_mn = mean(o_freq);
%f_mn = mean(ln_dist(:,3));

% next four lines do the corr. coeff. if 'corrcoef' not implemented in Matlab
%c_cdd = sum(o_freq.^2)/max_size - d_mn^2;
%c_cff = sum(ln_dist(:,3).^2)/max_size - f_mn^2;
%c_cdf = sum(o_freq.*ln_dist(:,3))/max_size - d_mn*f_mn;
%c_c = c_cdf/sqrt(c_cdd*c_cff);

%c_c = corrcoef(o_freq,ln_dist(:,3));

%fprintf('log_mean = %f\tlog_sdev = %f\tcorrel = %f\n', lg_mean,lg_sdev, c_c);
```

Appendix 8:

```
function uniform_dist = Euniformfit(DUI)

% Generates best-fit uniform distribution.
% Prints uniform distribution & correlation coeff (as goodness-of-fit).
%    ...
% Description: uniform arrays.
% Call:       uniform_dist = uniform_fit (fd_data)
% Parameters:  uniform_dist is the returned probability vector.
%              - col 1 - size(x-axis)
%              - col 2 - best-fit uniform prob dist
%              - col 3 - scaled col-2 (same total frequency as fd_data))
%             fd_data is the input (observed freq dist)
%              - if one col, assumes frequencies at unit-size steps
%              - if two col, assumes size-freq x-y pairs
%                in order of increasing size
%
% obtain two col-vectors; sizes 1:max_size, frequencies
% Author:  Robin Crockett (robin.crockett@northampton.ac.uk)


[n_r,n_c] = size(DUI);
if(n_c==1)
 max_size = n_r;
 o_size = (1:n_r)';                  % size (x-data)
 o_freq = DUI;                       % frequency of size
else
 max_size = DUI(n_r,1);
 o_size = (1:max_size)';
 o_freq = zeros(max_size,1);
 f_k = 1;
 for s_i = 1:max_size
  for f_i = f_k:n_r
   if(s_i==DUI(f_i,1));
    o_freq(s_i) = DUI(f_i,2);      %results are the same as fd_data
    f_k = f_i;
    continue;
   end
  end
 end
end

% uniform stuff

f_tot = sum(o_freq);

uniform_dist = zeros(max_size,3);
uniform_dist(:,1) = (1:max_size)';          % all sizes 1:max_size
for s_i = 1:max_size
```

```
  uniform_dist(s_i,2) = f_tot/max_size;
end
uniform_dist(:,2) = uniform_dist(:,2)/sum(uniform_dist(:,2));
uniform_dist(:,3) = uniform_dist(:,2)*f_tot;
```

Appendix 9:

```
function data_fd = Etestdatanorm1(r_frac,r_base)

%## Makes dummy data sets, uniformly distributed
%## 10000 integer observations in size range 1-3000.
%## Rounding base and proportion user-defined.
%##
%## Usage:      data_fd = normaldata(r_frac,r_base)
%## Arguments:  r_frac - proportion of data-set rounded (as %, 0-100)
%##             r_base - rounding base

%## Author: Robin Crockett (robin.crockett@northampton.ac.uk)

%if((r_frac<0)||(r_frac>100))
%  usage("r_frac must be a number between 0 & 100");
%end

n_obs = 500;              %## preset number of observations, total frequency
max_size = 150;           %## preset max. observation size
max_round = r_base * fix(max_size/r_base);   %## max rounded value

mu = fix(max_size/2);             %## set mean to mid-point of fd

r_fd = zeros(max_size,1);
n_fd = zeros(max_size,1);
b_fd = zeros(max_size,1);
data_fd = zeros(max_size,1);

%Counting number of exact values and number of rounding values
if(r_frac>0)
  n_round = round((r_frac/100)*n_obs);
  n_exact = n_obs - n_round;
else
  n_exact = n_obs;
end

%Working with exact values
n_data = randn(n_exact,1);          %## generate unrounded data
n_mean = mean(n_data);              %mean of unrounded data
n_data = n_data - n_mean;           %## centre freq dist
s_data = round(max_size*n_data);    %?
d_max = abs(max(s_data));           %## scaling factor
d_min = abs(min(s_data));
if(d_max>d_min)
  w_fac = max_size*mu/d_max;
else
  w_fac = max_size*mu/d_min;
end
n_data = w_fac * n_data + mu;
```

```
n_data = ceil(n_data);
for f_i = 1:max_size                %## do frequency dist.
 n_fd(f_i) = sum(n_data(:)==f_i);
end

%Working with rounding values
if(r_frac>0)                        %## if rounding present
 r_data = randn(n_round,1);             %## generate rounded data
 r_mean = mean(r_data);
 r_data = r_data - r_mean;          %## centre freq dist
 s_data = round(max_round*r_data);
 r_max = abs(max(s_data));          %## scaling factor
 r_min = abs(min(s_data));
    if(r_max>r_min) %new editing
        rw_fac = max_round*mu/r_max; %new editing
    else
        rw_fac = max_round*mu/r_min; %new editing
    end
    r_data = rw_fac * r_data + mu; %new editing
    r_data = ceil(r_data);
    for f_i = 1:max_round               %## do frequency dist.
    r_fd(f_i) = sum(r_data(:)==f_i);
    end
    r_i_p = 1;
    for r_i = r_base:r_base:max_round       %## do rounding in freq. dist.
   b_fd(r_i) = sum(r_fd(r_i_p:r_i));
   r_i_p = r_i_p+r_base;
    end
end

data_fd = n_fd + b_fd;
```

Appendix 10:

```
function data_fd = Etestdatauni(r_frac,r_base)


%Makes dummy data sets, uniformly distributed
%10000 integer observations in size range 1-3000.
% Rounding base and proportion user-defined.
%
% Usage:      data_fd = testdatauni(r_frac,r_base)
% Arguments:  r_frac - proportion of data-set rounded (as %, 0-100)
%             r_base - rounding base

%Author: Robin Crockett (robin.crockett@northampton.ac.uk)


%if((r_frac<0)||(r_frac>100))
%  usage("r_frac must be a number between 0 & 100");
%end

n_obs = 10000;              % preset number of observations, total frequency
max_size = 3000;            %preset max. observation size
max_round = r_base * fix(max_size/r_base);

r_fd = zeros(max_size,1);
n_fd = zeros(max_size,1);
b_fd = zeros(max_size,1);
data_fd = zeros(max_size,1);

    if(r_frac>0)
        n_round = round((r_frac/100)*n_obs);
        n_exact = n_obs - n_round;
    else
        n_exact = n_obs;
    end

n_data = rand(n_exact,1);
n_data = max_size * n_data;
n_data = ceil(n_data);
    for f_i = 1:max_size
        n_fd(f_i) = sum(n_data(:)==f_i);
    end

    if(r_frac>0)
        r_data = rand(n_round,1);
        r_data = max_round * r_data;
      r_data = ceil(r_data);
        for f_i = 1:max_round
        r_fd(f_i) = sum(r_data(:)==f_i);
        end
```

```
        r_i_p = 1;
        for r_i = r_base:r_base:max_round
        b_fd(r_i) = sum(r_fd(r_i_p:r_i));
        r_i_p = r_i+1;
    end
  end

data_fd = n_fd + b_fd;
```

NEURAL NETWORK ANALYSIS OF ESTIMATED DATA

Endang Triastuti, Robin Crockett, Phil Picton and Alasdair Crockett*
School of Technology & Design
University College Northampton, St. George's Avenue, Northampton, NN2 6JD, UK
Phone: + 44 (0) 1604 735500, Fax: + 44 (0) 1604 717813
email: {endang.triastuti, robin.crockett, phil.picton}@northampton.ac.uk
*UK Data Archive
University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, UK
Phone: + 44 (0) 1206 872875, Fax: + 44 (0) 1206 872003
email: crockett@essex.ac.uk

ABSTRACT: A set of data may be 'coarsened' as a result of enumerators' or compilers' efforts to estimate (or falsify) observations. This type of coarsening typically results in excesses of 'convenient' numbers (such as multiples of 5 or 10 in decimal number systems) in the data sets, apparent as patterns of periodic unit-width spikes in the frequency distribution.

We report on the development of novel radial basis function neural-network techniques for detecting this type of coarsening and quantifying the estimation which generates it. The objective is to provide an alternative to conventional statistical approaches based on missing-data techniques: data coarsened thus are not actually missing, solely shifted in size. The results show that the neural networks can successfully detect and classify the coarsening in data-sets and, hence, yield insights into the ways in which people count when performing enumeration or other numerical data-compilation exercises.

KEYWORDS: Data quality, data coarsening, missing data, neural networks, radial basis functions

INTRODUCTION

This research is being carried out at the University College Northampton in collaboration with the University of Essex. It is a development of an ongoing research programme originally initiated in response to problems encountered in the analysis of data from the Religious Census of 1851 for England and Wales [1], [2]. The initial work resulted in a new model for analysing rounding and estimation present in the data-sets but, whilst sufficient for the original purpose, the constraints of this model indicated that new techniques would be advantageous.

The general nature of a data-set coarsened by rounding is apparent from the frequency distribution (of congregation-sizes in this instance) shown in Figure 1. Instead of the expected reasonably smooth curve, indicating a 'predictable' distribution of observations, this frequency distribution shows excesses of observations at distinct, well-defined sizes which are periodic with intervals (in this instance) of 5, 10, 20, 50 and 100. The periodic nature of the coarsening can be confirmed by autocorrelation techniques.

This particular type of periodic structure arises from estimation: certain of the original enumerators did not count exactly but estimated the numbers of people attending church. When estimating thus, the real underlying value is returned as a convenient multiple of a base-unit, the closeness of the returned 'round' number to the real underlying value being dependent on a variety of factors including the observation size, as well as the ability and diligence of individual enumerators. The base-units are generally convenient numbers within the number system being used, such as 5 (counting 'by fives') and 10 (counting 'by tens') in decimal (base 10) number systems or 6 (half-dozens) and 12 (dozens) in duodecimal (base-12) systems.

Previously, analysis of such coarsening has been based on statistical methods developed originally for the analysis of missing-data problems [3], [4]. Data coarsened by estimation are not missing, solely shifted in size as a result of the rounding inherent in the estimation process. Thus, application of missing-data techniques, which seek to replace missing data according to probability distributions of sizes, are liable to distort coarsened data by

Presented at EUNITE 2002 (The European Network on Intelligent Technology), Albufeira-Portugal, 19-21 August 2002

effectively duplicating the shifted observations at probabilistically determined 'real' sizes. Therefore, techniques which analyse the shifts in size, such as those under development by the authors, potentially avoid introducing the distortions to which missing-data based techniques can be susceptible, as well as providing information regarding the counting behaviour.

Neural networks have been shown to be universal function approximators in that they can approximate any function to within an arbitrary degree of accuracy; thus, in theory at least, should be capable of fulfilling the user's hope on finding the underlying periodic structure(s) in the data [5]. This property of neural networks, particularly Radial Basis Function (RBF) networks potentially makes them unbiased modelling tools capable of detecting the presence of rounded data [6], [7]. In essence, the neural networks recognise the periodic 'patterns' created in the frequency distribution by the presence of coarsening due to estimation.

THE CURRENT MODEL

Without other information, it is not possible to distinguish an exact 'round' observation from an estimated one: for example, an observation (count) of size 65 could be exact or could represent underlying observations of size 63 or 68 counted 'by fives' or one of size 70 could be exact or could represent underlying observations of size 68 or 73 counted 'by tens' or 'by fives'. However, it is possible to estimate the excesses of 'round' observations and the corresponding deficits of 'non-round' observations resulting from estimation processes and, within this, determine the relative importance of, for example, base-5 and base-10 estimation processes.

Our current model of such data-sets assumes that the enumerators can be divided into groups according to the base-unit used for estimation: to continue with the foregoing example, the base-5 subset comprises all observations counted 'by five' and the base-10 subset comprises all observations counted 'by ten'. Thus, for the data summarised in Figure 1, there are base-5, base-10, base-20, base-50 and base-100 estimation subsets and an exact-count subset (also labelled base-1, noting that 1 is the effective base-unit of exact counts). Whilst it is not possible to distinguish individual observations within the whole data-set as being exact or rounded/estimated to a particular base-n, we can determine probabilistically how many observations are contained within each subset [1]. Figure 2 shows the subset structure of the study data summarised in Figure 1.

The accuracy of the model is dependent upon the accuracy to which the probability distributions of the various estimation behaviours can be determined. The details are data-set dependent but, in general, the subset probability distributions for the small base-units, such as base-1 (exact counts), base-5 and base-10, closely follow the underlying frequency distribution. The probability distributions of larger base-units, however, show more variation due to variation of counting/estimation behaviour with size of observation.

RADIAL BASIS FUNCTION NEURAL NETWORK MODELS

A RBF neural network consists of an input layer, an output layer with, usually, one or more hidden layers between them. Each of these layers contains nodes which are connected to nodes at adjacent layers. Each node in neural network is a processing unit that contains a weight and a summation function. In this case, the input layer uses a radial basis transfer function and the (single) hidden layer uses a linear transfer function. For function approximation, both layers have biases whereas for classification only the first layer has bias.

A RBF network is a special case of regularisation network. The method solves an interpolation problem by constructing a set of linear equations of the basis function. This method constructs a linear function space which depends on the positions of the known data points according to an arbitrary distance, d:

$$d = \sum_{i=1}^{n} |x_i - w_{ji}|b$$

where, d is the distance between input nodes and weights.
x_i are the input nodes.
w_{ji} are the weights in layer j, transpose of x_i in this case.
b is the bias, equal to 1 in unbiased layers.

The method uses the Gaussian basis function (normal distribution,) [8]:

$$f(d) = e^{-d^2/2\sigma^2}$$

where, $f(d)$ is the (radial) basis function.
d is the distance between input nodes and weights.
σ is a parameter specifying the width of the basis function.

For function approximation, the hidden layer computes a linear transfer function:

$$y_j = \sum_{i=1}^{n} x_i w_{ji}$$

where; y_j are the output nodes in layer j (hidden layer).
x_i , w_{ji} are as above.

For classification, the hidden layer computes the same linear transfer function but feed into a competitive function which selects the largest y_j-value to yield the classification (the most significant base-number).

NEURAL NETWORK MODELS TO IDENTIFY ROUNDING BASE-UNITS

In these early stages of the research, the neural network modelling has been performed using Matlab and the Neural Network Toolbox. Currently, both function approximation and classification are used in the models. As the work proceeds and the models are developed, increasing development of custom neural networks, optimised for this type of analysis, is being implemented.

For both function approximation and classification, a training data-set for each base-number (i.e. a potential estimation subset base-unit) being investigated is required. These training data-sets are 'dummy' frequency distributions comprising appropriate patterns of 0s and 1s (e.g. 0-1-0-1... for base-2, 0-0-1-0-0-1... for base-3, 0-0-0-1-0-0-0-1... for base-4, etc.) shaped according to the underlying probability distribution of the data-under-investigation (DUI) and scaled according to the total frequency of the DUI.

FUNCTION APPROXIMATION

The '*newrbe*' function of the Neural Network Toolbox is currently used to approximate data patterns. The function approximation process entails iterative investigation of a set of base-numbers. For each base-number:

- the neural network is trained using the base-*n* training data-set with a training target comprising the underlying probability distribution of the data-under-investigation (DUI) scaled according to the total frequency of the DUI;
- the DUI is then input to the trained network; and
- the degree of recognition for base-*n* is assessed by evaluating the sum-squared error (SSE) between the output from the DUI and the base-*n* training output.

The base-numbers which are most recognisable at this stage, i.e. those which yield the smallest SSEs, will include the subset base-units. Further analysis is then required to determine which of the base-numbers thus identified are 'real' subset base-units and which are merely (sub-) multiples of them: resolving these (sub-) multiple effects is one of the objectives of the ongoing research.

CLASSIFICATION

The '*newpnn*' function of the Neural Network Toolbox is currently used to classify data patterns. The classification process can be iterated, eliminating the most significant class/base-number at each iteration, producing a ranking of base-numbers from the most significant (which will include the subset base-units) to the least significant. The procedure entails:

- training the new '*newpnn*' neural network function using *all* the training data-sets as inputs and a vector of base-numbers as a target output;
- obtaining the base-*n* classification (i.e. the most significant of the base-numbers) by inputting the DUI to the trained neural network;

- repeating the above processes, having removed both the previously identified base-*n* (from the vector of base-numbers) and the base-*n* training data-set, until all the base-numbers under investigation have been ranked.

Alternatively, the competitive function ('*compet*') of the Neural Network Toolbox can be accessed to obtain the probability contribution of every base-number at the first iteration above, thus producing a ranking of base-numbers without the need for subsequent iterations. Note that the iterative and non-iterative rankings are identical.

RESULTS

Typical results obtained by applying the models to data-sets are shown in Tables I and II. Note that we have focused on base-numbers 1-20, 50, 100 on the basis of previous research in regard of the specific data-set summarised in Figure 1: the selection of base-numbers may differ with different data-under-investigation.

Table I shows the results from the study-data summarised in Figure 1. Table II shows the results from ten random data-sets having the same total frequency and underlying frequency distribution as the study-data: The left-hand columns of Table II show the neural-network classification results; the right-hand columns show the modulo-test results. In both tables, the modulo-test results serve as a *de facto* reference: a base-*n* modulo-test directly measures integer (base-*n*) divisibility, i.e. the proportion of the observations in the data-set which are exactly divisible by base-*n*. Thus, the modulo-test ranking indicates the relative significance of base-numbers in the data-set

It is evident from Table I that the ranking produced by function approximation is the same as that produced by classification (the iterative and non-iterative classification rankings are identical), and that both of these show some differences compared to the ranking produced by the modulo-test procedure. However, it is clear that pattern recognition by the neural-network models produces rankings which are similar to the *de facto* reference produced by modulo-tests, indicating that the neural networks can usefully 'see' the patterns resulting from coarsening. The differences between the neural-network rankings and the modulo-test rankings are attributable to the (sub-) multiple effects amongst the base-numbers and how these interact to produce the patterns which are actually being 'seen' by the neural network: the detail of this is currently under investigation.

It is evident from Table II that the neural networks are not deceived: when there are no patterns, i.e. no periodic structure, neural networks produce the rankings which are very similar to those expected. It should be noted that for ideal 'random' data, ignoring the effects of specific probability distributions, one half of the observations will be divisible by base-2, one third by base-3, one quarter by base-4, etc.; so that the expected ranking is the reverse of that of the base-numbers. The random data-sets in this study are not ideal: they are lognormally distributed (as are the real data) as well as having random deviations from the ideal. The modulo-test rankings show that all ten show some deviation from the ideal at base-numbers greater than base-10, i.e. when the proportion of multiples of a base-number falls below ca. 10% of the whole.

It is further evident from Table II that the neural-networks become relatively unreliable in comparison to the modulo-test at base-numbers greater than base-7, i.e. when the proportion of multiples of a base-number falls below ca. 14% of the whole. This sensitivity of the neural network is currently under investigation.

CONCLUSIONS

The techniques under development by the authors seek to complement conventional statistical approaches by detecting, modelling and quantifying the levels of estimation (and/or falsification) present in numerical data-sets. The results obtained so far, of which typical examples have been presented, show that suitably configured RBF neural networks can be used to detect the presence of estimation in data-sets via the coarsening patterns it produces. The results also show that the neural networks can rank base-numbers, i.e. potential subset base-units, according to their significance. Development of the neural networks to distinguish the (sub-) multiple effects amongst the base-numbers and, thus, determine the subset base-units, is ongoing. The ongoing work reported herein is an important step towards being able to use neural networks to identify and model estimation (and/or falsification) behaviour present in the compilation of numerical data-sets.

REFERENCES

[1] Crockett, R. G. M. & Crockett, A. C., "Historical Sources: How People Counted. A Method for Estimating the Rounding of Numbers", History and Computing, 9, 1/2/3, 1999, pp. 43-57.

[2] Crockett, R. G. M. & Crockett, A. C., "People and counting: how people count in enumeration exercises", 3rd European Social Science and History Conference, Vrije Universiteit, Amsterdam, April 2000.

[3] Heitjan, D. F. & Rubin, D. B., "Ignorability and Coarse Data" , Annals of Statistics, 19, 4, 1991, pp. 105-117.

[4] Heitjan, D. F. & Rubin, D. B., "Inference from Coarse Data via Multiple Imputation with Application to Age Heaping", Journal of the American Statistical Association, 85, 410, 1990, pp. 304-314.

[5] Polhill, J. G. & Weir, M. K., "An Approach to Guaranteeing Generalisation On Neural Networks" , Pergamon, Neural Networks 14 (2001), pp. 1035-1048.

[6] Turner, S. J., Crockett, R. G. M., Picton, P. D., Triastuti E., "Genetic Algorithms for Simulating Counting Behaviour", Proc. 19th Biennial Conference on Numerical Analysis, University of Dundee, 26-29 June 2001, p. 38.

[7] Turner, S. J., Triastuti E., Crockett, R. G. M., Picton, P. D., Crockett, A. C., "Intelligent Techniques for Detecting Estimated and Falsified Data" Sixth Multi-Conference on Systemics, Cybernetics and Informatics. Orlando, Florida, USA. 14-18 July 2002.

[8] Chen, S., Cowan, C. F. N. & Grant, P. M., "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks" , IEEE Transactions on Neural Networks, vol. 2, no. 2, March 1991, pp. 302-309.

Figure 1: Study Data Frequency Distribution (detail).

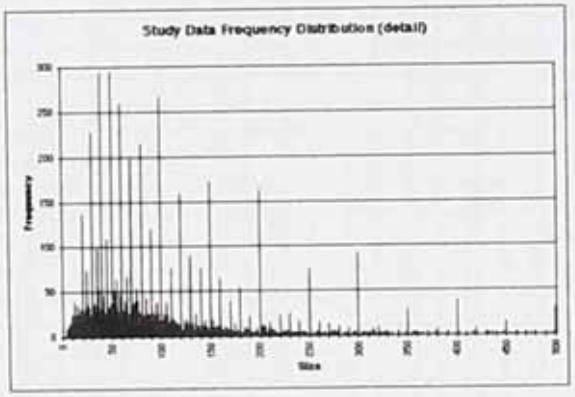


Figure 2: Study Data Subset Structure.

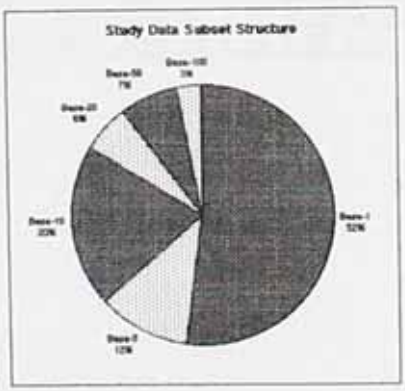


Table I:. Pattern Recognition Results of Study Data.

Rank	Base-Numbers		
	Funct. App'n.	Classification	Modulo Test
1	1	1	1
2	5	5	2
3	2	2	5
4	10	10	10
5	4	4	4
6	20	20	3
7	3	3	20
8	15	15	6
9	6	6	15
10	8	8	8
11	50	50	50
12	12	12	7
13	7	7	12
14	9	9	9
15	14	14	100
16	100	100	14
17	16	16	16
18	18	18	11
19	13	13	13
20	11	11	18
21	19	19	17
22	17	17	19

Table II:. Number-Base Rankings from Random Data.

Rank	Base-Numbers by RBF Classification										Base-Numbers by Modulo-Test									
	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	8	8	9	8	9	8	9	8	8	9	8	8	8	8	8	8	8	8	8	8
9	9	9	8	9	10	10	8	9	10	8	9	9	9	9	9	9	9	9	9	9
10	11	10	10	10	8	9	10	10	9	12	10	10	10	10	10	10	10	10	10	10
11	10	11	11	11	11	11	12	11	11	10	11	11	11	11	11	11	11	11	11	12
12	12	12	13	12	13	14	11	12	12	14	12	12	12	12	12	12	12	12	12	11
13	13	15	12	13	14	12	13	13	15	11	13	13	13	13	13	13	13	13	13	14
14	16	13	14	15	15	13	14	15	13	13	14	15	14	14	14	14	14	14	14	13
15	19	14	15	14	12	15	16	14	20	18	15	14	15	15	15	15	15	15	15	15
16	15	16	18	18	19	17	18	16	19	19	16	16	16	16	16	17	16	16	16	18
17	14	20	19	17	17	16	15	17	16	15	17	18	18	18	17	16	17	17	20	19
18	17	18	16	16	16	18	17	20	14	16	19	20	19	17	19	18	18	18	19	17
19	18	19	17	20	18	20	19	18	17	17	18	19	17	20	18	19	19	20	17	16
20	20	17	20	19	20	19	20	19	18	20	20	17	20	19	20	20	20	19	18	20

Presented at EUNITE 2002 (The European Network on Intelligent Technology), Albufeira-Portugal, 19-21 August 2002

Appendix 12

NEURAL NETWORK ANALYSIS OF ESTIMATION OF DATA

Endang Triastuti, Robin Crockett, Phil Picton, *Alasdair Crockett.

School of Technology & Design, University College Northampton, Northampton, NN2 6JD, UK
*UK Data Archive, University of Essex, Colchester, CO4 3SQ, UK

Abstract

A set of data may be 'coarsened' as a result of enumerators' or compilers' efforts to estimate (or falsify) observations. This type of coarsening typically results in excesses of 'convenient' numbers in the data sets, such as multiples of 5 or 10 in decimal number systems, apparent as patterns of periodic unit-width spikes in the frequency distributions.

We report on the development of novel Radial Basis Function neural-network techniques for detecting numerical data coarsened by rounding/estimation (or falsification) and quantifying that rounding/estimation. The objective is to provide an alternative to conventional statistical approaches based on missing-data techniques: data coarsened thus are not actually missing, solely shifted in size. The results show that the neural networks can successfully detect and classify the coarsening in data-sets and, hence, yield insights into the ways in which people count when performing enumeration or other numerical data-compilation exercises.

Keywords

Data quality, data coarsening, missing data, neural networks, radial basis functions.

Introduction

This research is being carried out at the University College Northampton in collaboration with the University of Essex. It is a development of an ongoing research programme initiated in response to problems encountered in the analysis of data from the Religious Census of 1851 for England and Wales [1,2]. The initial

Presented at the 4th International Conference on RASC 2002 (Recent Advance in Soft Computing), Nottingham-UK, 12 & 13 December 2002.

work resulted in a new model for analysing rounding and estimation present in the data-sets but, whilst sufficient for the original purpose, the constraints of this analysis indicated that new techniques would be advantageous.

The general nature of a data-set coarsened by rounding is apparent from the frequency distribution (of congregation-sizes in this instance) shown in Figure 1. Instead of the expected reasonably smooth curve, indicating a 'predictable' distribution of observations, this frequency distribution shows excesses of observations at distinct, well-defined sizes which are periodic with intervals (in this instance) of 5, 10, 20, 50 and 100. The periodic nature of the coarsening can be confirmed by autocorrelation techniques.

This particular type of periodic structure arises from estimation: certain of the original enumerators did not count exactly but estimated the congregation-sizes. When estimating thus, the returned value is a 'round' number, i.e. a convenient multiple of a base-unit, with the closeness of the returned value to real observed value being dependent on a variety of factors including the observation size, as well as the ability and diligence of individual enumerators. The base-units are generally convenient numbers within the number system being used, such as 5 (counting 'by fives') and 10 (counting 'by tens') in decimal number systems, or 6 (half-dozens) and 12 (dozens) in duodecimal systems.

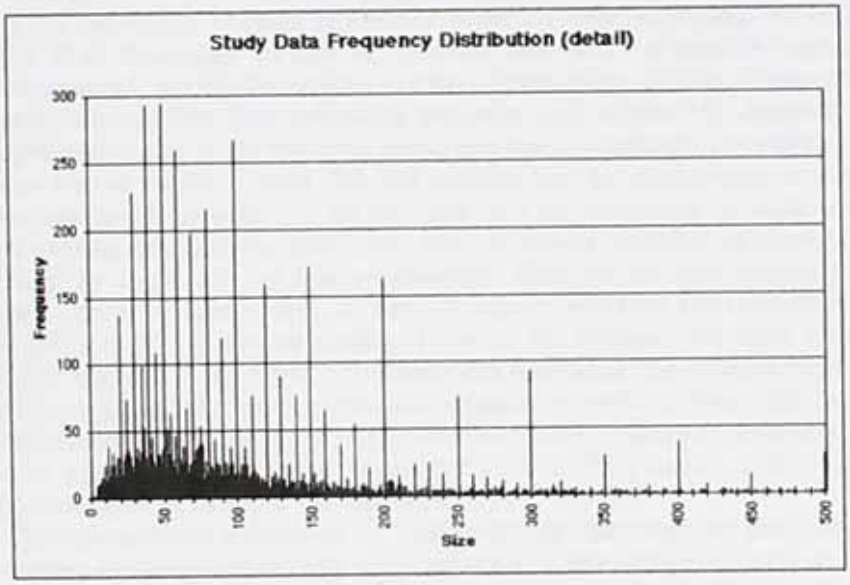


Fig. 1. Study Data Frequency Distribution

Conventionally, analysis of such coarsening utilises statistical methods developed originally for missing-data problems [3,4]. Data coarsened by estimation are not missing, solely shifted in size as a result of the rounding inherent in the estimation process. Thus, application of missing-data techniques, which seek to replace missing data according to probability distributions of sizes, are liable to distort coarsened data by effectively duplicating the shifted observations at

probabilistically determined 'real' sizes. Therefore, techniques which analyse the shifts in size, such as those under development by the authors, potentially avoid introducing the distortions to which missing-data oriented techniques can be susceptible, as well as providing information regarding the counting behaviour.

Neural networks have been shown to be universal function approximators in that they can approximate any function to an arbitrary degree of accuracy; thus, in theory at least, should be capable of fulfilling the user's hope on finding the underlying periodic structure(s) in the data [5]. This property of neural networks, particularly Radial Basis Function (RBF) networks potentially makes them unbiased modelling tools capable of detecting the presence of rounded data [6,7]. In essence, the neural networks recognise the periodic 'patterns' which such coarsening creates in the frequency distributions.

The Current Model: The Benchmark

Without other information, it is not possible to distinguish an exact 'round' observation from an estimated one: for example, an observation (count) of size 65 could be exact or could represent underlying observations of size 63 or 68 counted 'by fives'; or one of size 70 could be exact or could represent underlying observations of size 68 or 73 counted 'by tens' or 'by fives'. However, it is possible to estimate the excesses of 'round' observations and the corresponding deficits of 'non-round' observations resulting from estimation processes and, within this, determine the relative importance of, for example, base-5 and base-10 estimation processes.

Our current model of such data-sets assumes that the enumerators can be divided into groups according to the base-unit used for estimation: to continue with the foregoing example, the base-5 and base-10 subsets comprise all observations counted 'by fives' and 'by tens' respectively. Thus, for the data summarised in Figure 1, there are base-5, base-10, base-20, base-50 and base-100 estimation subsets and a base-1 exact-count subset (1 being the effective base-unit of exact counts). Whilst it is not possible to distinguish individual observations within the whole data-set as being exact or rounded/estimated to particular base-units, we can determine probabilistically how many observations are contained within each subset [1]. For the study data, the subsets are: base-1, 52%; base-5, 12%; base-10, 20%; base-20, 6%; base-50, 7%; base-100, 3%.

The base-units are determined by 'modulo-testing' the frequency distribution, a procedure entailing investigation of the numbers of observations exactly divisible by given base-numbers (integers). The base-number results for the data summarised in Figure 1 are shown in Table I. These results are then analysed statistically both in relation to the expectation and for (sub-) multiple effects in order to determine the actual estimation base-units: these being the base-numbers whose multiples appear statistically significantly more frequently in the data than expected, independently of (sub-) multiple effects.

The accuracy of the model is dependent upon the accuracy to which the probability distributions of the various estimation behaviours can be determined. The

details are data-set dependent but, in general, the subset probability distributions for the small base-units, such as base-1 (exact counts), base-5 and base-10, closely follow the underlying frequency distribution. The probability distributions of larger base-units, however, show more variation due to variation of counting/estimation behaviour with size of observation.

Radial Basis Function Neural Network Model

A RBF neural network consists of input, hidden and output layers, each layer containing nodes which are connected to nodes in adjacent layers and each node being a processing unit containing weight and summation functions. The input and hidden layers use radial basis and linear transfer functions respectively.

A RBF network is a special case of regularisation network. The method solves an interpolation problem by constructing a set of linear equations of the basis function. This method constructs a linear function space which depends on the positions of the known data points according to an arbitrary distance, d:

$$d = \sum_{i=1}^{n} |x_i - w_{ji}| b \tag{1}$$

where, d is the distance between input nodes and weights.
x_i are the input nodes.
w_{ji} are the weights in layer j, transpose of x_i in this case.
b is the bias, equal to 1 in unbiased layers.

The method uses the Gaussian basis function (normal distribution) [8]:

$$f(d) = e^{-d^2/2\sigma^2} \tag{2}$$

where, $f(d)$ is the (radial) basis function.
σ is a parameter specifying the width of the basis function.

The hidden layer computes a linear transfer function and then feed into a competitive function which selects the largest y_j-value to yield the classification (the most significant base-number).

$$y_j = \sum_{i=1}^{n} x_i w_{ji} \tag{3}$$

where; y_j are the output nodes in layer j (hidden layer).

RBF Neural Network Diagnosis of Coarsening

In these early stages of the research, the neural network modelling has been performed using Matlab and the Neural Network Toolbox. Although both function approximation and classification have been used in the research [9], the model now solely uses classification. As the work proceeds and the model is refined, a custom neural network, optimised for this type of analysis is being progressively developed.

A pair of training-data input and classification output is required for each base-number (i.e. potential estimation subset base-unit) or combination of base-numbers being investigated. The basic training data-sets are 'dummy' frequency distributions comprising appropriate patterns of 0s and 1s (e.g. 1-1-1-1... for base-1, 0-1-0-1... for base-2, 0-0-1-0-0-1... for base-3, etc.) shaped and scaled according to the underlying probability distribution of the data-under-investigation. The target outputs are classes corresponding to the base-numbers under investigation.

The classification process is iterative, the iterations progressively identifying (potential) estimation base-units.

- First iteration. Each potential estimation base-unit has an individual training data-set corresponding to it. A ranking of base-numbers is produced: the pattern associated with the top-ranked base-number is being recognised as being the most significant pattern in the frequency distribution at this stage.
- Second iteration. The highest ranked base-number (from the previous iteration) is considered in combination with each of the other base-numbers: each training data-set comprises a combination of two training data-sets from the first iteration. A refined ranking of base-numbers is produced: the pattern associated with the top-ranked pair of base-numbers is being recognised as being the most significant pattern in the frequency distribution at this stage.
- Third iteration. The highest ranked pair of base-numbers (from the previous iteration) is considered in combination with each of the other base-numbers: each training data-set comprises a combination of three training data-sets from the first iteration. A further refined ranking of base-numbers is produced: the pattern associated with the top-ranked triplet of base-numbers is being recognised as being the most significant pattern in the frequency distribution at this stage.
- Fourth and higher iterations. The process is then repeated, adding one base-number to the combination per iteration, until the probability of the top-ranked combination ceases to increase from one iteration to the next.

The hypothesis is that at each iteration, the combination of base-units which best fits the real frequency distribution is selected and that, whilst the probability of the top-ranked combination continues to increase with each successive iteration, the fit is improving. Once this probability stops increasing, the goodness of fit stops improving. Thus, the combination which yields the highest probability should contain all the estimation base-units plus, possibly, some base-numbers which are (sub-) multiples or multiplicative combinations of them. Further analysis is then required to determine which of the base-numbers thus identified are

'real' subset base-units and which are merely (sub-) multiples of them: resolving these (sub-) multiple effects is one of the objectives of the ongoing research.

Results

The probability assigned to each base-number or combination thereof by the neural network — and by which these are classified and ranked — is the probability that the pattern associated with that base-number (or combination) would be selected as being the frequency distribution of the target data. Thus, this probability is a measure of significance of a base-number (or combination) within the data. In the first iteration (with individual base-numbers) all base-numbers from 1 to 100 are used but for subsequent iterations it has proved sufficient to use those base-numbers ranking higher than (the higher of) base-11 or base-13 in the first iteration. This criterion is empirical: unless there is other information or it is clearly apparent from the first iteration to the contrary, it is highly improbable that numbers as 'inconvenient' as base-11 or base-13, the first two prime numbers greater than ten, would form the basis of estimation processes (baker's dozens notwithstanding) and so it is even less probable base-numbers ranking below these would form the basis of estimation processes.

The results, for the data summarised in Figure 1, are presented in Tables 1 and 2. Table 1 shows the results of the first iteration and also shows the results yielded by the modulo test (the basis of the current model) as a *de facto* benchmark. The results are restricted to the highest ranked twenty-six in accordance with the criterion described above; base-13 ranks 27 and base-11 ranks 34 in this case. There are some differences between the ranking produced by neural network that produced by the modulo test. However, it is clear that the neural network produces rankings similar to that of the modulo test, indicating that the neural network can both 'see' and identify the patterns which result from coarsening. As described previously [9], when there are no patterns (i.e. no periodic structure) in the data set, as is the case with random data, the neural network produces the ranking which would be expected, e.g. more multiples of base-1 than of base-2, more multiples of base-2 than of base-3, etc.

Table 2 shows the top-ranked combinations of base-numbers and their probabilities. The increasing trend in probability breaks after the ninth iteration, yielding the best-fit combination (in descending order) of base-1, base-10, base-5, base-20, base-50, base-100, base-30, base-40 and base-70. Up to the sixth iteration (base-100), the neural network recognises the estimation base-units determined by modulo-testing with no spurious results. The probability, however, continues to increase for a further three iterations, additionally indicating base-30, base-40 and base-70, results which differ from the modulo-test benchmark. These reasons for this are not fully understood at present but are due, at least in part, to interactions between base-numbers and how these affect the patterns actually being 'seen' by the neural network.

numbers, i.e. potential subset base-units, according to their significance producing results very similar to the modulo-test benchmark.

The neural network, however, indicates further base-units in comparison to the benchmark and work is ongoing to determine the reasons for this and, hence, possible modifications to the method to eliminate these 'false positives'. It should be noted, however, that if the subset model is applied to all nine base-units indicated by the neural network, then the three spurious base-units (base-30, base-40 and base-70) would yield subsets of zero size and so their elimination would be possible at this stage.

References

[1] Crockett RGM, Crockett AC (1999) Historical Sources: How People Counted. A Method for Estimating the Rounding of Numbers, *History and Computing*. 9, 1/2/3, 1999, pp. 43-57.

[2] Crockett RGM, Crockett AC (2000) People and counting: how people count in enumeration exercises, *3rd European Social Science and History Conference*, Vrije Universiteit, Amsterdam, April 2000.

[3] Heitjan DF, Rubin DB (1991) Ignorability and Coarse Data, *Annals of Statistics*, 19, 4, 1991, pp. 105-117.

[4] Heitjan DF, Rubin DB (1990) Inference from Coarse Data via Multiple Imputation with Application to Age Heaping, *Journal of the American Statistical Association*, 85, 410, 1990, pp. 304-314.

[5] Polhill JG, Weir MK (2001) An Approach to Guaranteeing Generalisation On Neural Networks, *Pergamon, Neural Networks* 14 (2001), pp. 1035-1048.

[6] Turner SJ, Crockett RGM, Picton PD, Triastuti E (2001) Genetic Algorithms for Simulating Counting Behaviour, *Proc. 19th Biennial Conference on Numerical Analysis*, University of Dundee, 26-29 June 2001.

[7] Turner SJ, Triastuti E, Crockett RGM, Picton PD, Crockett AC (2002) Intelligent Techniques for Detecting Estimated and Falsified Data, *Proc. Sixth Multi-Conference on Systemics, Cybernetics and Informatics*. Orlando, Florida, USA. 14-18 July 2002, pp. 445-450.

[8] Chen S, Cowan CFN, Grant PM (1991) Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks, *IEEE Transactions on Neural Networks*, vol. 2, no. 2, March 1991, pp. 302-309.

[9] Triastuti E, Crockett RGM, Picton PD, Crockett AC (2002) Neural Network Analysis of Estimated Data, *Proc. Eunite 2002*, Albufeira, Portugal. 19-21 September 2002; pp. 161-166.