Reservoir computing in financial forecasting with committee methods

Konrad Stanek

Kongens Lyngby 2011
IMM-MSC-2011-64

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk IMM-MSC: ISSN 0909-3192

Summary

Reservoir Computing (RC) methods are an active area of research in the field of machine learning and intelligent processing. In particular, reservoir networks (echo-state networks, ESN) have been successfully applied to many engineering problems such as chaotic time series forecasting, primarily due to their efficiency, speed of training, and avoidance of many common shortcomings of typical recurrent neural networks. The initial concept of echo state networks became soon extended with such techniques as supervised/unsupervised reservoir adaptation, weights pruning and feature selection, improved training algorithms. Simultaneously, other research efforts concentrated on combining individual networks into hierarchical structures or voting collectives. In this work we follow this concept and evaluate various types of ESN committees. Furthermore, we investigate different member ranking algorithms and show circumstances in which they constitute promising alternative to simple output averaging. The results of our comparative studies suggest several design principles concerning committee models.

Secondly, we shall apply the reservoir committee models to non-trivial engineering task of financial forecasting. The global markets constitute one of the most complex, non-linear systems created by modern society. For decades it was a goal of many research endeavors to understand and foresee the essential mechanisms of markets dynamics. While for many contributors the ability to forecast the chaotic financial time series is the purpose in itself, for others, like banks, investment funds, or governmental entities, application of steadily better models is the integral part of the investment strategy and decision taking processes. Multitude of various approaches are intensively investigated in light of their applicability to financial forecasting, however it still remains uncertain

if any of the proposed models can clearly outperform the others in this task. In the scope of this thesis we employ the ESN committee models to forecast the probable market movements. We shall consider a range of optimization schemes and training configurations. Important part of the thesis will relate to domain analysis in order to facilitate selection and preprocessing of the input data, so that optimal amount of information is provided to the system.

Preface

This thesis was prepared at the Department of Informatics and Mathematical Modelling (IMM), at the Technical University of Denmark (DTU), under supervision of Ole Winther, Associate Professor, IMM, DTU. The project was carried out from November 2010 to August 2011 in fulfillment of the requirements for acquiring the M.Sc. degree in Engineering.

Acknowledgements

First and foremost, I would like to express special gratitude to my supervisor, Ole Winther, Associate Professor, IMM, DTU, for all his support and inspiration throughout the entire project. His objective evaluation, accurate hints and inspiring ideas were truly motivating and always kept the research on the correct path.

I am immensely grateful to my wonderful friends for their support and reassurance. And to the fellow students, doctorants and postdocs from Technical University of Denmark, for all the constructive discussions and exchange of opinions.

Contents

Chapter 1

Introduction

1.1 Purpose

In the recent decades there has been a growing demand for intelligent systems for forecasting dynamics of financial markets and future directions of global economy. Various proposed algorithms concentrate on both technical and fundamental analysis of macroeconomical factors, in attempt to predict future market dynamics, price tendencies, and thus enhance investment decisions. Due to emergence of on-line investment platforms supporting meta-trading scripting languages, it became possible to create automated algorithmic trading systems that operate without human interaction. This makes it possible to eliminate human weaknesses such as emotional, irrational decisions, stress factor, decision delay – and hence fully rely on the strength of the investment algorithm. The key issue remains how to design an algorithm capable to produce reliable predictions in such immensely complex and apparently chaotic environment as global financial markets. The classic algorithms often rely on the assumption, that market dynamics are governed by rationality and statistical regularity. They often base on classic fundamental theories and simple linear models combining several variables in a determined way. However, the observations and analysis of market behavior lead to conclusion that one of the main factors to consider is group psychology of millions of private and institutional investors, striving to maximize their profits and reduce losses. Constant interaction of rational

decisions with human factors such as fear, greed, and stress, makes the global financial markets one of the most complex nonlinear systems created in the modern society. The classic algorithms recognize only limited number of major factors influencing markets, and rarely can quantify that influence. It seems therefore, that much wider context is necessary in order to capture market dynamics and increase the efficiency of predictions.

The financial domain constitute a promising environment for application of systems based on recurrent neural networks (RNN). In particular, the state-of-art echo state networks (ESN) will be investigated in this project, which were shown to offer many advantages over classic RNNs in terms of performance and training efficiency. Generally speaking, the potential of neural network based systems lies in the fact that the algorithm is self-created in long process of learning, instead of being explicitly predefined by designer. Through analysis of large multivariate data sets of correlated financial data and macro-economical indicators, the system can theoretically capture those patterns and relations in market dynamics, that are not recognized by classic theories and expert systems. Ability to detect such patterns will have immediate impact on quality of prediction of future market movements. Moreover, with currently available computational power, it is possible to train in relatively short time large populations of networks, varied by structure and trained on different subsets of input time series, optimize their architecture, and combine their expertise by connecting them into larger structures - voting committees or mixtures-of-experts. The motivation of this work comes from assumption, that carefully trained collectives of echo state networks will have potential to outperform classic algorithms and human reasoning in the task of market prediction. Furthermore, due to their robustness and flexibility, such collectives can be easily adapted to other forecasting and classification tasks.

In the scope of this project we will concentrate on aspects of design, training and evaluation of the reservoir committees. Although our ultimate goal is financial forecasting, the major part of the project is centered around general principles and design issues of the system, from the machine learning perspective. Common design issues will be addressed, such as stability, regularization, bias-variance tradeoff, overfitting, optimization. Finally, the model will be adapted to economic applications, in particular to predictions of nonlinear dynamics of global financial markets, on example of American S&P500 index, German DAX, and EUR/USD currency exchange rate. We shall present how the committee model can be used as automated trading agent or investment decision support, by means of predicting next-day market directions.

1.2 Predictive model

Recurrent neural networks (RNN) are still one of the most commonly used models in the task of time series forecasting. However, their structure, training methods and optimization algorithms have evolved significantly since their origins. Variety of different research approaches resulted in significant improvement of the forecasting accuracy of RNN predictors. Furthermore, computational power available now allows more extensive optimization, evaluation and thorough empirical studies of large-scale network models.

Particularly prominent, state-of-art architecture is Echo State Network (ESN) [1, 2], class of Reservoir Computing methods. ESN differs significantly from commonly used RNNs, in terms of structure, training and optimization methods. From the design and training perspective, it can be considered as a bridge between connectionist and stochastic methods. Structurally, it displays similarity to biological networks. The essence lies in the complex dynamics of randomly generated "neural reservoir" – a cloud of sparsely connected neurons, having distinct temporal characteristics due to recurrent connections and nonlinear activation functions of neurons. In the contrary to classic RNNs, only the readout layer needs to be trained, while internal reservoir connectivity remains constant. The readout training aims at selection of desired nonlinear transformations from the large reservoir container, what can be accomplished with well-known linear regression methods. ESNs avoid many shortcomings of common RNNs, such as local minima convergence and slow, computationally demanding training. Moreover, ESNs were shown to perform surprisingly well in variety of forecasting tasks. Therefore we shall adopt echo-state approach as the basic approach in this thesis. Furthermore, we will combine populations of such base models into generalized committees to enhance predictive accuracy and robustness of the resulting system.

Detailed specification of ESN architecture as well as design principles are subject of Chapter 3, while Chapter 4 advances the concepts to committee level. Chapter 5 elaborates on engineering applications of the model in financial domain.

1.3 Financial domain

The predictor model that will be the subject of this project, can be adapted to virtually any type of forecasting, classification or control task. However, selection of global financial markets as an the experimental field is not accidental.

The domain offers several characteristics, that will be beneficial for our purposes:

- The global financial markets constitute a complex non-linear system, that presents non-random chaotic dynamics. The behavior is conditioned by wide range of macroeconomical, social and technical factors. Those factors compose a dynamic network of relations and dependencies, where change within one variable will influence (directly or indirectly) the others. However, the strength and range of that influence is not always possible to detect and quantify. High dimensional, spatio-temporal patterns need to be found between those variables in order to improve prediction accuracy. This is not feasible for classic algorithms, but constitutes a rich and challenging training playground and research environment for ESN-based system.
- Data availability. Complete sets of historical data sets of market dynamics, macroeconomic variables, sentiment indicators can be downloaded from online sources, for periods as long as recent 60 years. Such extensive data supplies will be beneficial for teaching and testing networks. Since different economic indicators are strongly related, the system will base its forecasts on high dimensional multivariate input, comprising range of correlated financial time series.
- The demand for novel solutions for intelligent investment decision support is ever increasing. The number of automatized trading platforms constantly grows, and in the time of writing this paper more than half of the transactions are initiated by algorithms rather than humans. The markets became a testing ground, where competing intelligent algorithms try to outsmart the others. Therefore an intelligent system that would show potential to outperform the other solutions, may be of interest for external institutions willing to contribute to further research (e.g. banks, investment funds, government entities).

More insight into domain aspects will be presented in Chapter 2. We will discuss the main factors that drive markets dynamic and make them non-trivial to forecast. We will preselect data sets of economic time series to work with, list the data sources, and finally, in Chapter 5 , we shall focus on important aspects of data transformations and preprocessing, which are essential for efficient forecasting.

Chapter 2

Domain analysis

In this chapter we discuss the basic concepts related to the global financial markets, relevant with respect to the project purposes. The thorough investigation of the underlying markets mechanisms is beyond the scope of this thesis, hence we recommend a selection of literature committed to the subject [32, 33, 34, 35, 36, 37]. We shall attempt, however, to point out several concepts and factors that determine nonlinear chaotic market dynamics, and hence make forecasting a non-trivial task. Furthermore, we will select, out of large variety of available data, those time series that constitute "good candidates" for input and output of the system. Several databases and online sources will be investigated that offer economic and financial data sets.

2.1 Financial markets as complex nonlinear system

The global financial markets constitute a complex network of correlated factors, where change of one will propagate, directly or indirectly, to the others. It is difficult to forecast given economic variable or financial index without insight into overall market situation. There are several concepts and factors that need to be considered in economic forecasting.

2.1.1 Stock markets and indices

By issuing stocks (called also shares) on a stock market, companies can raise funds from external investors. Current stock prices are shaped by relation between supply and demand, reflecting not the real value of the companies, but rather the expectation of investors about its future value. Promising prospects will increase demand on the company's stocks, what will elevate their price. Some investors buy the stocks with long-term investment horizon, counting for positive development of the company net value and for other shareholder's benefits (voting right, dividends, etc.). Others purchase the stocks in purely speculative manner, hoping to benefit from the volatility of share price by selling higher.

Stock exchanges are the physical locations bringing companies and investors together. However nowadays the majority of trading activities are carried through electronic networks rather than physically at the facilities of stock exchange. Most of free-market countries have one or more national stock exchanges, each quoting a number of companies, usually between tens and thousands. The national stocks are accessible for foreign investors, in some cases with certain limitations. Furthermore, stock exchanges can offer derivates, which are more complex financial instruments based on the stocks, indices, and currencies, and can be traded in similar manner as stocks.

Based on the stock prices, the market indices are defined, being an average value of certain groups of stocks. National stock indices group the largest companies quoted on given stock exchange, thus reflecting well the condition of national economy. An example is American Standard & Poor's 500 Composite Index (S&P500), which averages through 500 largest corporations quoted on NYSE stock exchange. Other indices may represent companies belonging to particular sectors, for instance financial, telecommunications or transportation sectors. Yet another indices measure performances of selected global shares (e.g. S&P 100 Global), or entire global markets (e.g. MSCI Emerging Markets Index).

The important is that apart from being used as indicators of condition of given market section, the indices can be themselves the subject of trade. Trading the indices can be for instance done by means of future contracts (contracts on future prices), where investors can open 'short position' or 'long position', counting for market growth or fall respectively. In a way, buy/sell transactions on futures markets are symmetric - short position means that the sell operation precedes the buy operation.

More information on stock markets can be found in [33, 34]. In the scope of this work, we shall mostly concentrate on predictions of the leading national indices,

rather than individual stocks.

2.1.2 Currency market

Currency market (foreign exchange market) is in fact the largest and the most intensively traded global market. Every large event, whether political, social or environmental, will be immediately reflected by exchange rates. Currency market is unregulated and in contrary to stock exchanges it has no physical location. Instead, the transactions are made world-wide by banks, investment funds and even governments.

Currency exchange rates on FOREX market are the essential factor shaping the international trade and import/export prosperity. They constitute important uncertainty parameter considered by banks and institutions in determining investment strategies, and by private investors purchasing foreign stocks or commodities. Moreover, the currency exchange rates not only serve to value foreign assets in national currency, but also are subject of the speculative trading [36], by means of direct transactions, future contracts and options on currency pairs.

The relations between three important currencies will be of our interest - Euro (EUR), US dollar(USD) and Japanese Yen (JPY). Currency markets are treated in detail in [36, 34]

2.1.3 Commodity market

Considering global economy it is important to emphasize significance of commodities and natural resource markets. For instance, the price levels of crude oil will directly influence production and transportation costs, and indirectly nearly every aspect of modern economies, so much dependent on combustive fuels. Oil prices, consequently, are very sensitive to international politics, stability and relations between developed and emerging economies. Other commodities, like e.g. agricultural products or metals, will influence prosperity of the corresponding industrial sectors, and hence the related stock prices. Trends of gold prices in turn often reflect the uncertainty level on the markets. Since gold is considered as a safe investment, its price will be elevated in times of uncertainty, since it is the commodity where investors allocate the capital withdrawn from other, more risky securities. More information about specifics of commodity markets can be found in [37]

2.1.4 Macroeconomic factors

There are several important macroeconomic indicators and variables worth to be considered in forecasting tasks. The main of them is gross domestic product (GDP), which reflects value of all the final goods and services that given economy produced in certain period, and thus it is considered to be the main indicator of the economy health. GDP is often expressed in terms of its annual growth, that is GDP growth (or simply: output growth). Another important macroeconomic variable is the unemployment rate, which reflects a ratio of the unemployed citizens to the number of citizens in the labor force. The unemployment rate has large social and economic impact, and influences other variables such as consumer spending, consumer confidence, output growth, and others. The third essential variable is inflation (or: consumer price index, CPI), which corresponds to the growth of general price levels. Too large inflation affects unequally income distribution, increases uncertainty about future, and usually discourages investment decisions.

In fact all those variables are closely correlated. High GDP growth is usually coupled with decrease in the unemployment, and vice-versa (Okun's law). Relation between CPI inflation and unemployment is not always obvious, but usually very low unemployment will be accompanied by increase of inflation (Philips curve). The key task of governments, or more generally macroeconomic policy-makers, is to maintain economic growth (measured by GDP) simultaneously with reduction of unemployment rate and maintaining stable inflation rate. The positive trends within those values will result in optimistic long-term economy prospects and willingness of citizens to invest capital in stocks and other securities, what results in elevating the valuation of the assets. Apart from governmental activities, the monetary policy of central banks (or: money supply) needs to be considered. Higher money supply will reduce the interest rate, which is the cost of borrowing the money. This in turn will stimulate the output growth, however increases the risk of high inflation. The optimal equilibrium is not trivial to determine, nor to maintain.

Of course there are other macroeconomic factors that influence GDP in short-, medium- and long-term. They will not be further elaborated in this thesis, instead the reader is referred to the literature covering the aspects of macroeconomy [34, 35]. We conclude saying that macroeconomic variables have both short-term and long-term implications on financial markets. Periodic release of the updated values generates certain reactions among investors, reflected in immediate price changes. Sometimes the impact on the markets can be significant. For this reason, macroeconomic variables can be a beneficial part of the predictor's multivariate input.

2.1.5 Group behaviors

To appreciate complexity of the system, we need to have a closer look at the diversity of the actors responsible for market dynamics. The most influential are large financial institutions like central banks, investment and pension funds and large-cap international corporations. Their decisions may have substantial impact of the market movements. In the contrary, individual investors have no sufficient resources to influence the markets, instead they attempt to exploit the trends and regularities. Another powerful group involved in international cash flow are governments. It is important to note that purely free-market economies, where entire system is regulated exclusively by consumers and producers (demand and supply), in fact do not exist. In reality, the free-market economies are always a mixture between central control and market determination [35]. It means that government can impose financial law regulations as well as intervene according to the needs on the domestic markets and currency markets in order to secure the interests of the citizens.

Classic theories often assume, that all the parties (whether individuals, corporations, or institutions) act in a rational manner to maximize their profits and cut down the losses. However the reality shows that the system is far more complex, and similarly like other large-scale social systems, the financial markets are often driven by group-psychology effects. This often results in irrational behaviors, such as panic-driven sell-off of stock and other securities in the time of crisis, or so called "speculation bulbs" elevating the prices of certain equities far above their objective values.

2.1.6 Automated trading

Another aspect, that made market forecasting yet more challenging in the recent years, is rapidly growing proportion of automatized trading in the overall number of transactions, especially in highly-developed economies. For instance, according to research&consulting company Aite Group, the companies involved in automatized, high-frequency trading are responsible for approximately 73% of the entire US equity trading volume, as for 2009 [38]. The high-frequency-trading (HFT) algorithms are designed to generate rapid investment decisions in attempt to capture trading opportunities that appear for as short as fractions of seconds. They often benefit from marginal gains from thousands or tens of thousands of transactions initiated per day.

The automatized trading is no longer limited to the largest market participants. Many brokers already started to provide the online investment platforms for

individual investors, accepting meta-trading scripting languages to define algorithmic trading agents. An example is MetaQuotes Language 4 (MQL4) [39], supporting design and implementation of own trading strategies and expert advisors. The growing popularity of algorithmic trading changes the dynamics of the markets making them more non-stationary than ever before. A lot of innovation-oriented companies emerged, that specialize in development of constantly smarter trading algorithms, having primary task to detect and exploit the imperfections of other methods.

2.1.7 Theories and approaches

Thinking about economic variables and financial data, one can be tempted to assume, that after deep analysis of all relations and dependencies between them, it should be possible to construct a deterministic, mathematical model to simulate precisely a development of future market trends in the global economy. However, there are at least three arguments why such model is not feasible to be ever designed. First of all, the complexity of such model would be immense. Most of the classic economic models concentrate just on small subgroup of interacting values, and they are bounded by severe constraints and simplifications. Secondly, there are many random events that may occur, which cannot be predicted regardless of the model complexity - these include: climatic anomalies, natural catastrophes, terrorist attacks, financial law violations including inside trading, and others. None of the models can predict such events, although in theory smart solutions should be able to quickly adjust their dynamics short after such events had occurred. Thirdly, the last link in the chain of macroeconomic relations, market dependencies and international trading is the human taking investment decisions. Human factors like emotions, fear, greed and irrational group behavior make the markets dynamics particularly complex.

Popular approach in finances is known as Efficient-Market Hypothesis [40]. The week form of EMH assumes that all information is already included in asset price, and no excess returns (higher than average market returns) can be achieved in long run by sole analysis of historical data. EMH assumes that no patterns exist in price movements, or in other words - asset prices follow a random walk. Stronger form of EMH implies furthermore that no excess returns can be earned by trading on newly released public information, since it becomes immediately reflected in the asset prices.

Another approach, called the technical analysis (TA) [41], is based on three principal assumptions: market action discounts all available information, prices move in trends and historical patterns tend to repeat themselves. Technical analysts believe that observations of historical charts (prices and transaction

volumes) can help to determine the repeatable patterns, that account for both fundamental facts and irrational market emotions. Technical analysis can not fully predict the future market directions, but solely the fact that many market participants are aware of TA and interpret certain patterns in a common way can actually imply certain behaviors.

Fundamental analysis, in contrary, focuses on overall state of economy, macroeconomic variables, and specific information related to given market or security. The fundamental analysis assumes that every stock (or index) has its "correct", fundamentally explicable value, that will be eventually reached, even if it is under-estimated or over-estimated by current market value.

The attitude standing behind this work is somewhat similar to technical analysis, in a way that it is based on the same principal assumptions. On the other hand, in the contrary to AT we do not impose any interpretations on the price patterns, but instead allow the reservoir network to learn to interpret the historical data and generalize it onto future data. Furthermore we presume that far more information about price dynamics can be extracted if the patterns are searched in high dimensional multivariate input space. Such patterns could be difficult to identify with classic TA charting methods.

2.1.8 Summary

The financial markets are highly nonlinear system, due to large number of interacting parties and complex relations between price levels, currency rates and macroeconomic policies. Hence, optimal selection of the variables for the predictor's input is not a trivial task. The selection will certainly depend on the target signal chosen to be forecasted - whether it is a large-cap index, particular stocks, currency exchange rate or maybe economic variable. In fact, the selection of input data can be considered an important parameter to optimize, in order to obtain satisfactory prediction accuracy. We present exemplary set of candidate variables in the following section.

2.2 Preliminary data selection - market indices and economic indicators

After minor adaptation and tuning, our predictor model can be trained to work with arbitrary time series. However, for practical reasons, major global indices will be primarily in our focus. In particular - leading US index (*S&P500*), which

reflects capitalization of world's largest markets – *NYSE* and *Nasdaq*, and thus have immense impact on global economy. The S&P500 index is highly traded, relatively stable, and closely related to other global economies, in particular to that of the Eurozone. Secondly, the largest European market - German DAX - will be considered, for similar reasons as above. The index is interesting to work with, because in the contrary to S&P500 and DAX it displays the signs of recession in period April 2010 til august 2011. Finally, we shall consider *EUR/USD* exchange rate as another forecasting target.

Having target time series chosen, a selection of relevant input data becomes one of the essential problems. Proper input data is perhaps more important for accurate forecasting than the model design itself. The main idea is to include not only historical values of forecasted indices (univariate input), but also other types of data that have impact on the market movements (multivariate input) - primarily foreign market indices, currency exchange rates, transaction volume information. Other variables, such as commodity prices, macroeconomic factors and investors sentiment indicators can be considered to fine-tune the prediction. How those factors are correlated, and how they influence financial markets, was shortly discussed in the previous section and is treated in detail in [32, 33, 34, 35, 36, 37]. Such multivariate input will increase probability of finding regular spatio-temporal patterns, which in turn can boost prediction accuracy. The exact selection of inputs will depend on particular prediction task, and can be a subject of further optimization. This can be accomplished either by common techniques of feature selection or by resorting to prior domain knowledge. In fact, those two approaches are often combined. Below we shall suggest several good candidates to be considered as a part of the system input. Some of them will be used in the empirical studies in Chapter 5, while the others are presented for completeness but will not be used in the project scope.

Major global indices (These indices reflect the national economic condition, by averaging stock prices of large-cap corporations)

- S&P500 (US Standard&Poor's leading index of 500 large-cap American corporations)
- DJIA (US Dow Jones Industrial Average index of 30 American blue-chip stocks)
- DAX (Germany, Eurozone's engine)
- FTSE 100 (Great Britain)
- Shanghai Composite (China, second world's largest economy)
- Nikkei 225 (Japan)
- Global Dow (150 leading global stocks, reflects well condition of global economy)

Currency rates (Direct influence on international trade, export/import prosperity, and foreign policy. Currency rates have strong impact on all free market economies with no exception)

- EUR/USD (EURO / US dollar)
- USD/JPY (US dollar / Japanese Yen)
- USD/CNY (US dollar / Chinese Yuan)

Commodities (Fundamentals that drive global economy, constitute important link in the financial markets)

- CRUDE OIL (essential resource influencing every aspect of contemporary civilization)
- COPPER (influence on heavy industry)
- GOLD (often referred to as "investors safe-heaven", commodity to allocate financial resources in high-risk market periods)

Macroeconomic factors (Fundamental indicators of economy health, often used as variables in classic economic models)

- GNP (Gross National Product)
- Unemployment Rate
- Consumer Price Index (inflation rate)
- Interest Rates

Social factors and sentiment indicators (Represent indirect forces driving the markets)

- Consumer Confidence Index (Conference Board)
- Consumer Sentiment Index (Univ. of Michigan)
- ISEE Sentiment Index (bullish-bearish market direction indicator)

Depending on the experimental results and desired complexity of the system, the suggested range of the input data might need to be constrained in the scope of the project. We will mostly concentrate on market indices and currency exchange rates. On the other hand, if the system is to be employed to other economic prediction tasks in further research, the range of input time series might need to be extended accordingly.

2.3 Data sources

Before choosing the global financial markets as the project domain, it was essential to verify whether the relevant data is freely available, what resolution of time series can be obtained, and whether reliable data providers can be found. As a result we found numerous sources of data, which can be useful in further research. Below we list several of them, that will provide us sufficient data to evaluate accuracy of our prediction models. Depending on the needs, the list can be extended by other sources, if more specific data is required (for instance local stocks prices or indicators related to particular national markets).

The listed providers offer in most cases raw time series but sometimes also preprocessed statistics. In theory, data sets can be independently obtained from different sources and then compared in order to increase their reliability.

Database of Federal Reserve, central bank of America (FED) - offers wide choice of essential macroeconomic indicators released periodically by FED. Data can be downloaded in several formats, and for arbitrary period. The most important indicators here include: *Industrial Production (IP), Interest Rates, Consumer Credit, Foreign Exchange Rates* (in relation to USD). Website: *https://www.federalreserve.gov/datadownload*

US Department of Labor, Bureau of Labor Statistics - convenient access to crucial data having large impact on markets, including: *Consumer Price Index (CPI), Unemployment Rate, Average Earnings.* Website: *http://www.bls.gov/data*

World Federation of Exchanges (WFE) The service committed to collect, combine and distribute comparative data of global markets characteristics and dynamics. Although time resolution of data is lower (month intervals) the statistics found here will be of great help for domain analysis and preselection of data. Website: *http://www.world-exchanges.org/statistics*

Online Financial Services - main source of historical time series – daily closing values of world's major market indexes, natural resources, commodities, stocks, indicators can be freely browsed and exported from the services listed below:

- **Yahoo Finance** (*http://finance.yahoo.com*)
- **Google Finance** (*http://www.google.com/finance*)
- **Stooq** (*http://stooq.com*) – in contrary to many other sources, this service does not limit range of downloadable data to recent time period, and offers e.g. DJIA index daily data series since 1896, gold prices since 1969, etc.

National Stock Exchange databases - official stock exchange databases can provide any historical data, even quite specific type of information, and high-resolution real-time data. A country specific leading economic indicators and local stocks prices can be found here as well. In some cases, depending on the requested details and data size, this service may be charged with fee. A lot of data is freely available though. Examples:

- **New York Stock Exchange (NYSE)** - US stock exchange, world's largest market in terms of capitalization *http://www.nyse.com* and *http://www.nyxdata.com*
- **Tokyo Stock Exchange (Nikkei)** - Japanese stock exchange, *http://e.nikkei.com/e/fr/marketdatatable.aspx*
- **Shanghai Stock Exchange** - Chinese stock exchange *http://static.sse.com.cn*
- **Frankfurt Stock Exchange** - German stock exchange *http://deutsche-boerse.com*
- **London Stock Exchange** - UK stock exchange *http://www.londonstockexchange.com*
- **Copenhagen Stock Exchange (CSE)** - Danish stock exchange, part of NASDAQ OMX Nordic Group *http://www.nasdaqomxnordic.com*
- **Warsaw Stock Exchange (WSE)** - Polish stock exchange *http://gpw.pl*

Independent data sources - there are many freely accessible, independent databases, clustering diverse data from numerous sources. Few examples include:

- **US Polling Report** (*http://pollingreport.com/consumer.htm*) – large source of independent data illustrating well consumer sentiment and public opinion
- **Economagic** (*www.economagic.com/popular.htm*) – list of essential data series
- **Econdat** (*www.econdata.net*) – rich collection of links to variety of online data sources. This website is a good entry point for further data mining, if needed.

2.4 Preprocessing overview

In most of the cases, the economic and financial data in raw form can not be directly applied to the system input. Several preprocessing steps need to be

undertaken first. We shall discuss those issues in detail in Chapter 5, while in this section we only highlight the main preprocessing steps. It is important to note that data selection and preprocessing is the integral part of solving any financial forecasting problems. Failure in this step will lead to poor performance, regardless how efficient the model itself is.

In the beginning, appropriate data sets need to be downloaded and converted to desired format. Financial time series usually consist of five values for each date - day-open, day-max, day-min, day-close prices, and transaction volume. The first preprocessing step aims at identification and elimination of trends in the time series, so as to obtain stationary data sets characterized by stable mean and variance. Secondly, the detrended data needs to be properly scaled to match predictor's preferred input ranges. In case of multivariate input, what is usually the case in financial tasks, the special considerations needs to be given to synchronization of the time series, that accounts for different calendars, time zones, trading hours. Finally, linear transformations of the data can be optionally applied, to enhance feature extraction and provide statistical information about the time series. Technical analysis indicators can be used for this purpose.

2.5 Summary

After this brief introduction to the domain related basic concepts, data acquisition and preprocessing, we shall now leave the the financial domain and focus on the model design and analysis (Chapters 3 and 4). In the Chapter 5 of the thesis we shall revisit the financial concepts and combine them with the predictive models.

Chapter 3

Reservoir Computing and Echo State Networks

In this chapter we analyze static and dynamic properties of echo state networks, that will constitute base model for our collective predictor. We start with introducting basic idea of reservoir computing and review of the current research, with emphasis on echo state networks. Following this, formal specification of ESN will be given, including design principles and training methods. Finally, a selection of experiments is presented to show certain properties of model, its forecasting ability, and optimization methods. Benchmarking environment is introduced that will be used in this and subsequent chapter, in particular performance metrics and artificial chaotic time series.

3.1 Survey of literature and publications

Reservoir Computing (RC) is a relatively new concept in the field of neural networks and machine learning. In the contrary to the classic recurrent neural networks (RNN), where all connections are adapted in training process, RC systems are conceptually splitted into two distinct parts: a large reservoir of sparsely connected neurons, that remains unchanged, and a readout layer that is the only subject of adaptation. A function of the reservoir is to expand input

signal into high-dimensional, nonlinear, state-space representation. Assuming that the reservoir contains sufficient variety of nonlinearities, the readout is then computed with well-known regression techniques to reconstruct the target signal while minimizing the error function.

The two most common approaches in Reservoir Computing are known as Echo State Network (ESN) proposed first by Herbert Jaeger [1, 2] and asynchronous Liquid State Machine (LSM) introduced by Wolfgang Maas [3]. The former of them, being relatively easy to tune and fast to train, has been applied to various engineering problems, often outperforming other solutions in prediction accuracy [4, 5, 6, 7, 8]. ESNs are therefore essential component of the ranked committees elaborated in this paper. The latter approach, based on biologically realistic, synaptic models of spiking neural networks, has become more popular in computational neuroscience field and less widespread in engineering applications. In fact, RC model can essentially have any reservoir of either mathematical, physical or biological nature, that provides measurable responses to given inputs [9].

It is important to emphasize that ESN design, structure and training methods evolved significantly since they were first introduced. A lot of remarkable research was done to optimize performance and broaden their applicability. Efficiency of reservoir networks was boosted with such techniques as supervised/unsupervised reservoir optimization [11, 12, 13], imposing topological structure [14, 15], decoupling [16], pruning and feature selection[17, 18], leaky-neurons[19], varying training algorithms and adapting evolutionary optimization methods [20, 21]. Simultaneously, lot of the research efforts concentrates on combining multiple networks into larger scale structures. Some of the examples include corrective cascades [22], multi-reservoir structure [16], mixture-of-experts with gating ESN [23]. Very common approach is a simple averaging committee, which trains k independent ESN members on the same task, and combines their outputs to produce final committee response [6, 19]. For comprehensive review of currently ongoing RC research and challenges we refer the reader to excellent work of Lukosevicius and Jaeger [9] and Verstraten at al. [10].

3.2 ESN specification

Echo state network (ESN) is composed of three main layers - an input, a reservoir, and an output. The input layer is responsible for receiving input signals, possibly scaling and/or shifting them, and distributing them to internal reservoir neurons. The reservoir consists of relatively large number of sparsely connected

neurons. Its main task is to transform input signal into high-dimensional, non-linear, state-space representation. The output layer, or readout, is the only trainable part of the ESN. It linearly combines reservoir neurons activations so as to provide possibly accurate reconstruction of desired target signal. Fig.3.1 illustrates basic structure of ESN. Dotted lines denote trainable connections.

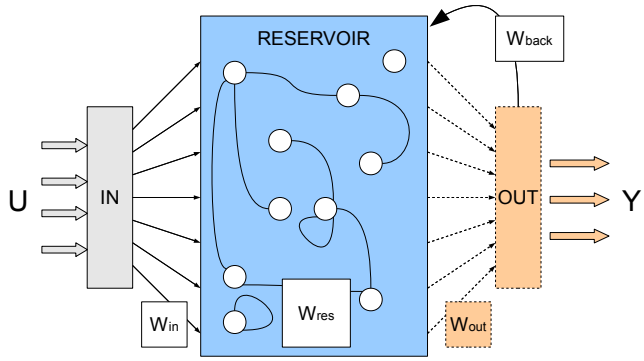

Figure 3.1: Echo State Network architecture.

We will now discuss the essential steps necessary to create ESN. The first step is to determine number of inputs, outputs and reservoir size. Given desired input dimension K, reservoir size N, output dimension L, we define ESN by specifying:

1. Input weights matrix W_{in} of the size $N \times K$
2. Reservoir connectivity matrix W_{res} of the size $N \times N$
3. Output weights matrix W_{out} of the size $L \times (N + K)$
4. Feedback weights matrix $W_{back}$of the size $N \times L$ (optional)
5. Activation function of reservoir neurons f_{res}
6. Activation function of output neurons f_{out}
7. Initial state vector S_o of the size $N \times 1$

Although there no strict constraints on how to initiate those parameters, the common practice is to set them as follows: draw W_{in} and W_{back} randomly from normal distribution $[-1, 1]$ with zero mean, leave arbitrary W_{out}[1], select sigmoid $tanh()$ function as reservoir neuron activation and identity function $\cdot()$ as output neuron activation, and set initial state S_0 to zero.

[1] W_{out}will be anyway replaced in the training process.

The essential part of constructing ESN is a design of its reservoir (W_{res} matrix), since it will affect learning ability, memory capacity and stability of the model. Three parameters are used in this process: reservoir size N, connectivity density c, and spectral radius p. Reservoir is characterized by sparse connectivity, usually in the range 1-20%. Size will range between hundred and few thousands neurons. After being randomly initiated, the weights of W_{res} are scaled down to reach desired spectral radius p. The stability requirement will hold if $p < 1$ [1].

$$W_{res} = p \cdot \frac{W_{res}}{eig_{max}\left(W_{res}\right)} \tag{3.1}$$

where $eig_{max}(W_{res})$ is the maximum eigenvalue of the reservoir matrix, or in other words - spectral radius of W_{res} before scaling.

Having all the above parameters initiated, the network is ready to receive inputs and produce outputs, although the output layer it is not trained yet. To compute subsequent state s_{t+1} and output y_{t+1}, following equations are used:

$$s_{t+1}^{lin} = W_{in} \cdot u_{t+1} + W_{res} \cdot s_t + W_{back} \cdot y_t + \upsilon_{res} \tag{3.2}$$

$$s_{t+1} = f_{res}\left(s_{t+1}^{lin}\right) \tag{3.3}$$

$$y_{t+1} = f_{out}\left(W_{out} \cdot \begin{bmatrix} s_{t+1} \\ u_{t+1} \end{bmatrix}\right) \tag{3.4}$$

where u_t, y_t, s_t are input, output, state vectors correspondingly in time step t, υ_{res} indicates normally distributed noise of relatively low amplitude $max\left(\upsilon_{res}\right) \ll max\left(s_t\right)$.

In the training process, only W_{out} matrix is adapted, while W_{in}, W_{res} and W_{back} remain unchanged[2]. The training process starts from feeding the network with subsequent training samples $U_{train} = [u_1, u_2, ..., u_r]$ and storing corresponding states in state collecting matrix $S = [s_1, s_2, ...s_r]$ and desired target outputs in matrix $D = [d_1, d_2, ..., d_r]$. Once matrices S and D are complete, we compute the output weights with pseudo-inverse matrix calculation:

[2]However, as we mentioned in the introduction section, a lot of research has been done to facilitate adaptation and optimization of reservoirs before actual training. Range of supervised and unsupervised methods were proposed, such as intrinsic plasticity, imposing topological structure, enhancement of separation property.

$$W_{out} = \left(S^T S\right)^{-1} S^T D \tag{3.5}$$

This is the original method proposed by Jaeger [1], but essentially any other regression method can be applied. The pseudo-inverse method brings up a risk of overfitting the model, if the number of parameters is too large in relation to available training samples. This would require adjustment of model complexity to length of available data. If however it is desirable to maintain large reservoir (e.g. high model capacity is needed due to complexity of the task in hand), we may need to employ regularization methods. In such case pseudo-inverse regression is often replaced by other techniques, like ridge regression [24]. The method incorporates regularization component λI, that penalizes large weights that do not contribute to error reduction. Regularization tends to reduce output variance at the cost of increasing the bias, what is commonly known as bias-variance trade-off. Finding optimal proportion will minimize mean squared error on testing data, or in other words - enhance generalization ability of the network. Output weights are computed with ridge regression as follows:

$$W_{out} = \left(S^T S + \lambda I\right)^{-1} S^T D \tag{3.6}$$

where I is a unity matrix and scalar λ is a free regularization parameter that should be carefully optimized to given task.

It is important to emphasize that formula (5) or (6) may be repeatedly used to connect any number of additional readouts to the reservoir, without affecting already existing ones. In this way the same reservoir can be reused for multiple prediction tasks. In particular, several independent readouts can be trained to forecast directly entire trajectory of target signal $Y_{traj} = \{y_{t+1}, y_{t+2}, ..., y_{t+k}\}$, where each prediction horizon y_{t+i} corresponds to the output of $i'th$ readout.

More details on ESN preparation, optimization and training can be found in comprehensive publications dedicated to the the subject, some of which are suggested in section3.1.

3.3 Extensions of basic model

3.3.1 Topological SHESN

As we have mentioned earlier, there is a lot of reseach committed to unsupervised optimization of reservoir. An interesting approach is based on imposing topolog-

ical structure on reservoirs, rather than using random sparse connectivity. The topology (usually 2-dimensional) is determined by means of preferential connectivity rules, which results in power law outdegree distribution and creation of multiple domains of clustered neurons. Such reservoirs are referred to as complex ESN (CESN) or scale-free highly-clustered ESN (SHESN). The networks display interesting properties, making them similar to real biological or social networks (e.g. topology of the Internet). Topological reservoirs were repeatedly reported in literature [15, 14] to offer interesting static and dynamic properties, and often to outperform classic ESNs in certain tasks. Due to different distribution of eigenvalues, spectral radius can be lifted to higher values without distorting the stability. Furthermore, clustering neurons into distinct synergies reduces coupling of neural activations, which can boost feature extraction and enhance predictor performance on complex tasks.

Note that since only readout layer is trainable, all the algorithms and routines characteristic for ESNs remain unchanged. The only additional effort is construction and optimization of reservoir. In the contrary to ESN reservoirs, which require only three parameters - size, connectivity, and spectral radius, SHESNs are governed by significantly more generic parameters. The construction of reservoir consists of the following steps:

- Generation of backbone neuron (BN) framework. Number of BNs usually do not exceed 0.5–5% of all the neurons. BNs are randomly allocated on topology grid, while minimum distance between two neurons must be maintained.
- Stochastic selection of sparse connectivity between BNs. Includes feedback connections.
- Individual allocation of local neurons (LN) on the grid. Firstly, one of the BNs is selected, with equal probability. Secondly, the new LN is placed in BN's proximity in the distance governed by bounded Pareto distribution. Minimum distance must be maintained. The LN is added to the nearest BN's domain, though physical link does not need to exist.
- Determination of connectivity for each LN. The important aspect is that LN can only be connected to neurons from the same domain (including feedback connection to itself and/or connection with backbone neuron). The preferential connectivity mechanism is used, so that probability of connection with given neuron (from the same domain) depends linearly on current outdegree of target neuron (clustering, highly connected nodes will attract yet more connections) and exponentially on euclidean distance between new neuron and target neuron, euclidean distance between target neuron and domain BN (tendency to extend towards domain center).

Moreover, the expected degree of new LN is dependant on it's proximity to BN, which favors centrally located LNs above peripheral ones.

The described algorithm results in generation of topological reservoirs (see Fig.3.2) with the following characteristics:

- Multiple domains connected only by means of backbone neurons. Each of them contains complex diverse networks of local neurons. Such a hierarchical, sophisticated structure is probable to offer wider and richer set of nonlinear dynamics, on which the output readout can be trained, comparing to classic stochastic reservoirs.

- Reservoir is a scale-free network – the neural outdegree distribution (connectivity distribution) follows power law, as in case of biological and social networks. Different neurons vary significantly in terms of their degree and localization, and thus can perform different subtasks in the overall prediction task. It enriches the set of the reservoirs nonlinear dynamics.

- Total connectivity is typically one magnitude sparser that normal ESN. This makes even large reservoirs relatively economic in terms of computational resources.

- Network is stable even with significantly larger spectral radius, than it is possible in case of typical ESN. This is due to different spectral distribution of eigenvalues of connectivity matrix. The tolerance to higher spectral radius enhances echo property, and hence can benefit memory capability.

SHESN reservoirs are an interesting alternative for modelling of complex nonlinear systems. Similarly like decoupled reservoirs [16] they can be used to construct mixture-of-experts type of models. Here, however, the experts (domain neurons) can communicate by means of the sparse backbone connectivity to generate final response. In further sections of the thesis, we constrain our considerations only to classic ESN models. Our goal is to concentrate on committee approach, and SHESNs would introduce additional parameters, making our reasoning less transparent. We decided however to commit to them this short section of the thesis, because their interesting characteristics were investigated during the project work and constitute promising alternative for the future research.

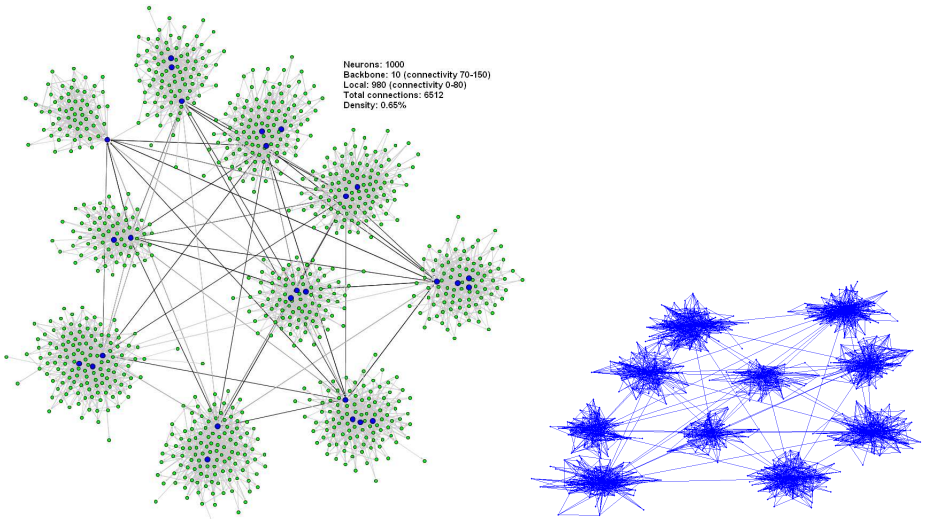

Figure 3.2: Topological visualization of exemplary SHESN - by Matlab (right) and Guess (left). Level of shading indicates connection strengths, blue circles denote backbone neurons.

3.3.2 Committee approach

Interesting alternative to using single network is a committee approach that takes advantage of entire population of similar models. The concept of committee model is general and does not constrain to echo state networks. It can comprise various models, either in homogeneous or heterogeneous setting. The most general committee is described by the following equation:

$$y(u;\mathcal{D}) = \sum_i \omega_i(u,\mathcal{D}) y_i(u;\mathcal{D}) \tag{3.7}$$

where y_i is the output and ω_i input-dependent weight of $i'th$ model. The weights ω_i are often designed to be input-invariant, and are estimated in cross-validation process. The most common approach is to set $\omega_i = \frac{1}{M}$, where M denotes number of models in the ensemble. In this way we obtain simple averaging committee.

Committees of reservoir networks, both averaging and generalized, constitute essential part of this thesis, and will be treated in detail in Chapter 4.

3.4 Experimental time series

The ultimate goal for the system is to forecast the financial time series. However, financial time series display highly nonlinear, chaotic behavior, and display large noise. Therefore in this chapter we resort to simpler, artificially generated time series, that will facilitate analysis of ESN dynamics and optimization. In the following experiments we mostly utilize Mackey-Glass time series as well as non-trivial harmonic time series. Experiments with financial time series will be the main subject of Chapter 5.

Mackey-Glass timeseries Mackey-Glass (MG) time series are very commonly used in the publications committed to time series analysis and forecasting. They are often considered as a benchmark of predictors accuracy. To generate the series of any arbitrary length we will use Mackey-Glass nonlinear time-delay differential equation of the form:

$$\frac{dx}{dt} = \beta \frac{x_{t-\tau}}{1 + x_{t-\tau}^n} - \gamma x \tag{3.8}$$

where $x_{t-\tau}$ is a value of x at time $t - \tau$, and other parameters are set as follows: $\beta = 0.2$, $\gamma = 0.1$, $n = 10$. The variable x displays increasingly chaotic behavior as the time lag parameter τ is incremented above 17. Fig. 3.3 displays Mackey-Glass time series with several different time lags. Note the increasing complexity.

Complex periodic-derived timeseries Another time series that we will utilize for testing and comparison are derived from periodic functions. In particular the following three functions will be of our interest:

$$f(x) = 0.4sin(x + 2) + 0.2sin(5x) + 0.1sin(11 * (x + 1)), \tag{3.9}$$

$$f(x) = sin(x + sin(x^2)), \tag{3.10}$$

$$f(x) = sin(\frac{x}{2} + sin((\frac{x}{2})^2)), \tag{3.11}$$

defined on discrete domain $x \in \{1, 2, 3, ..., n\}$. The time series are presented on fig. 3.4.

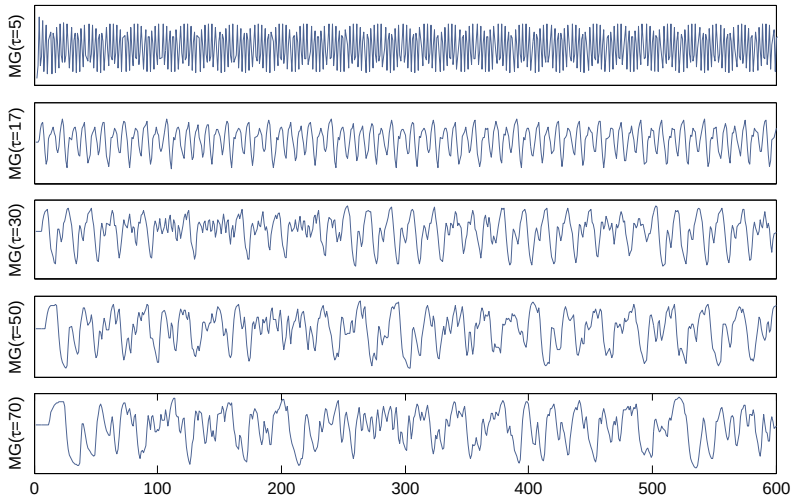

Figure 3.3: Mackey-Glass time series with varying time lag τ.

3.5 Performance metrics

In further experiments we will need objective measures of performance for trained ESN predictors. In most cases *Mean Square Error (MSE)* will be preferred. However, depending on the experiment purpose or task requirements, we might want to compute *Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Signed Error (MSE), Mean Absolute Percentage Error (MAPE), Mean Percentage Error (MPE).* Other measures, that are more specific to financial domain and investment support simulation, will be introduced in section 5.2.

MSE and RMSE *Mean Square Error (MSE)* is one of the most commonly used measures to estimate error of predictor $\hat{\theta}$ on dataset θ. The *MSE* is the expected value of squares of differences between real values and predicted values, for each accounted sample. Value of zero signifies perfect prediction. *MSE* strongly penalizes the predictor for any forecast that is highly diverging from the desired value (outliers). Depending on the experimental context this can be advantageous or not. Alternatively *Root Mean Square Error (RMSE)* can be used alternatively, which equals to root square of *MSE*. *RMSE* can be understood as a Cartesian distance between vectors of desired and predicted outputs.

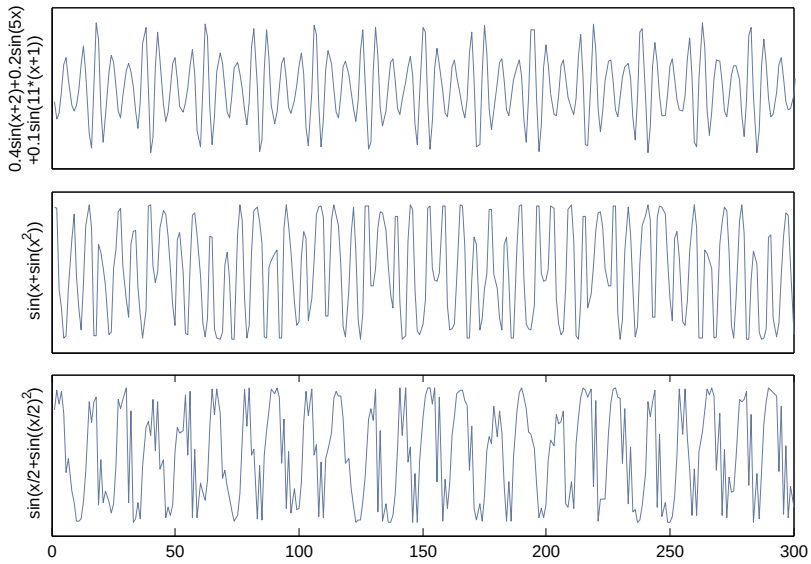

Figure 3.4: Periodic-derived time series of varying complexity.

MSE and *RMSE* are computed according to:

$$MSE(\hat{\theta}) = E[(\hat{\theta} - \theta)^2] = E[(D - Y)^2] = \frac{1}{n}\sum_{i=1}^{n}(d_i - y_i)^2$$

$$RMSE(\hat{\theta}) = \sqrt{MSE(\hat{\theta})} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i - y_i)^2}$$

where n is the number of samples, $D = \{d_1, ..., d_n\}$corresponds to desired values and $Y = \{y_1, ..., y_n\}$ to the predictor outputs.

MAE and MSE Other commonly used error measures are Mean Absolute Error (MAE) and Mean Signed Error (MSE). These error measures reflect expected value of difference between desired and predicted values, while the former accounts for absolute differences and the latter for signed differences. MSE can be helpful to determine whether predictor $\hat{\theta}$ has biased or unbiased output.

$$MAE(\hat{\theta}) = E[\|\hat{\theta} - \theta\|] = E[\|D - Y\|] = \frac{1}{n}\sum_{i=1}^{n}\|(d_i - y_i)\|$$

$$MSE(\hat{\theta}) = E[\hat{\theta} - \theta] = E[D - Y] = \frac{1}{n}\sum_{i=1}^{n}(d_i - y_i)$$

where n is the number of samples, $D = \{d_1, ..., d_n\}$corresponds to desired values and $Y = \{y_1, ..., y_n\}$ to the predictor outputs.

MAPE and MPE In many contexts it will be useful to measure predictor error as a relative value to the desired value, rather than as an absolute value. That provides a performance measure independent of input/output signal magnitude. Furthermore, percentage error estimation might be practical in certain financial domain applications. Hence we will frequently refer to *Mean Absolute Percentage Error (MAPE)* and *Mean Percentage Error (MPE)*, that are relative error measures corresponding to *MAE* and *MSE,* respectively.

$$MAPE(\hat{\theta}) = E[\|\frac{\hat{\theta} - \theta}{\theta}\|] = E[\|\frac{D - Y}{Y}\|] = \frac{1}{n}\sum_{i=1}^{n}\|\frac{(d_i - y_i)}{y_i}\|$$

$$MPE(\hat{\theta}) = E[\frac{\hat{\theta} - \theta}{\theta}] = E[\frac{D - Y}{Y}] = \frac{1}{n}\sum_{i=1}^{n}\frac{(d_i - y_i)}{y_i}$$

where n is the number of samples, $D = \{d_1, ..., d_n\}$corresponds to desired values and $Y = \{y_1, ..., y_n\}$ to the predictor outputs.

3.6 Model analysis and experiments

In the following subsections we shall perform different experiments to show dynamic characteristics of ESNs. Several important aspects of the networks training and exploitation will be considered, such as stability issues, overfitting, trajectory projection. Parametric optimization will be discussed. However we should note here, that the main goal of the project is combining models into higher hierarchical committee structures. Hence the optimizations in this chapter do not exhaust the subject, but rather are supposed to give better understanding of dynamics of our base model. This seems reasonable, since many aspects of stability and optimization discussed here generalize to committee level. The committee approach is a subject of Chapter 4.

3.6.1 Reservoir dynamics and stability

In this section we shall look closer at several essential aspects of reservoir dynamics. At this stage we do not attempt to train the readout yet, but instead concentrate on the most important parameters influencing the reservoir network and the individual neurons. Such experiments and analysis constitute an important part of the initial stage of working with echo state networks, since they give a good overall understanding of the complex dynamics of reservoirs. In the further sections of the thesis, this understanding will often influence design decisions.

Reservoir size Usually the first design step is a selection of reservoir size N. The value is essential for at least two reasons. Firstly, N corresponds to the effective number of parameters of the model, because there is exactly one trainable output weight associated with each reservoir neuron. Therefore higher N increases the capability of network to model more complex systems. Difficult tasks tend to require larger reservoirs. Secondly, N constrains the maximum memory capacity of the network (whereas the dynamic memory effects are governed by spectral radius as we shall see later).

It could indicate that larger reservoirs are always beneficial. Adding additional, randomly connected neurons enriches the bucket of nonlinear transformations of input signals, that can be used to construct the output signal. As we shall see in further sections, this is true provided that regularization is employed in the training process. Otherwise, excessive number of parameters may bring up the problem of overfitting. The optimal value for N is a function of task complexity and the size of the available training data. Usually a good initial guess is to set N to approximately 20-50% of the training data size.

Besides, another factor that in certain cases may influence the choice of reservoir size is computational constraints. In this aspect we notice a significant advantage of ESN networks. The training complexity is only linearly dependent on the number of neurons, while such dependence is quadratic in case of classic recurrent neural networks, trained with gradient decent methods.

Connectivity ratio and spectral radius The most desired characteristics of a "good" reservoir is stable behavior and richness of nonlinear transformations of the input signals. Considering specification of reservoir construction (section 3.2), stability of the system is primarily a matter of proper adjustment of spectral radius ρ, which corresponds to the highest eigenvalue of connectivity matrix W_{res}. The sufficient condition is that $\rho < 1$. However the condition is

not necessary, and reservoirs with higher spectral radii may, but do not have to, be stable as well. Another parameter, connectivity ratio c of the reservoir, has the secondary importance considering stability, because it is always followed by scaling of W_{res} matrix, so that ρ remains on the desired level. As a result, for a given value of ρ the network will be either densely connected with low connection weights, or more sparse with higher connection weights. The stability will be maintained in either case, however other characteristics of the network will change, e.g. excessive connectivity ratio will lead to stronger coupling of neural internal states and reduce reservoir diversity. It is common to hold the connectivity ratio on a constant, low level (usually $0.01-0.2$), while the spectral radius is optimized to given task (usually $0.5-1.0$). In that way the richness of the internal nonlinear states is ensured by sparse connectivity, while the optimal memory effect is determined by finding the proper spectral radius.

Fig.3.5 shows typical stable behavior of arbitrary neuron, after feeding network with low frequency square signal. The reservoir behaves like excitable medium and presents dampening behavior - initial oscillations after the input impulse are gradually suppressed and stable state is finally reached. The oscillations can be also interpreted as "echo states", or reflection of the input and state history. Spectral radius in this case was fixed at $\rho = 0.9$. Fig.3.6 illustrates the significance of the spectral radius for system stability. The same square signal is placed on the input, and we observe internal states of four arbitrary reservoir neurons. For moderate value ($\rho = 0.8$, left column) the reservoir is input driven, and transition to stable state is almost immediate. When $\rho = 1.0$ oscillations need long time to converge to constant level and system is working close to the edge of stability. Further increasing of ρ leads to more autistic behavior of the reservoir, since it amplifies and maintains bounded oscillations even though the input is hold constant. The reservoir is driven primarily by its previous states. The final column shows unstable dynamics for $\rho = 1.5$. The neurons oscillate widely between the extreme values of sigmoid activation function. Amount of information that can be encoded in this setting is limited. Further increasing of ρ would prevent the reservoir from stabilizing even if input was removed, due to amplifying echo states.

Input scaling Another aspect that has strong influence on reservoir dynamics is the scaling factor of input and feedback signals. The idea behind is that due to sigmoid activation function of the neurons, the scale of input signal will determine whether the system works in linear mode (input will use only linear region of sigmoid function), binary mode (large input will drive the neural outputs to extreme values of sigmoid function $\{-1, 1\}$), or nonlinear mode (optimally scaled input uses entire curvature of sigmoid activation function). The last mode is generally desired when modelling chaotic systems. Fig.3.7 illustrates the ex-

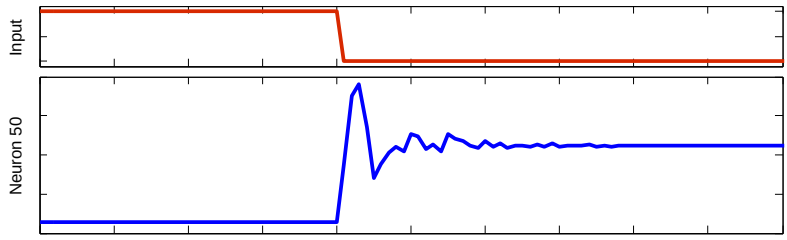

Figure 3.5: Neural state oscillations and convergence to stable state, after square wave excitation of the reservoir (spectral radius $p = 0.9$).

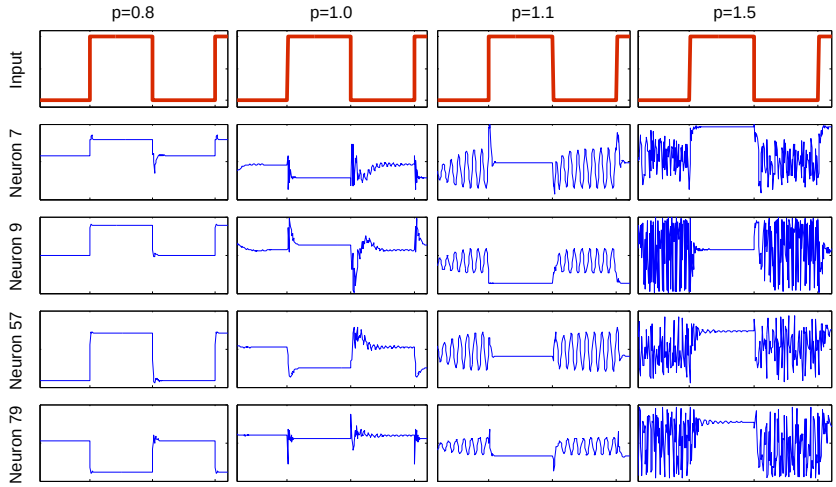

Figure 3.6: Stability of neural states as the spectral radius increases.

emplary input and output of the network, and activations of several arbitrary neurons working in linear, binary and nonlinear mode. Note that output is not trained yet at this stage.

To be precise, not only the global input scaling is important, but also the local scaling, which is regulated by input weights vector and feedback weights vector, i.e. W_{in} and W_{back}. Since those vectors are initiated randomly from the range $[-1, 1]$, the proper global scaling should ensure that reservoir contains rich combinations of neurons ranging from linear, through nonlinear to the binary ones. It is the common practice to set global input scaled to the ranges between $\langle -5, 5\rangle$ or $\langle -1, 1\rangle$, depending on how much nonlinearity is needed. However, to achieve the best performance, scaling can be a subject of further optimiza-

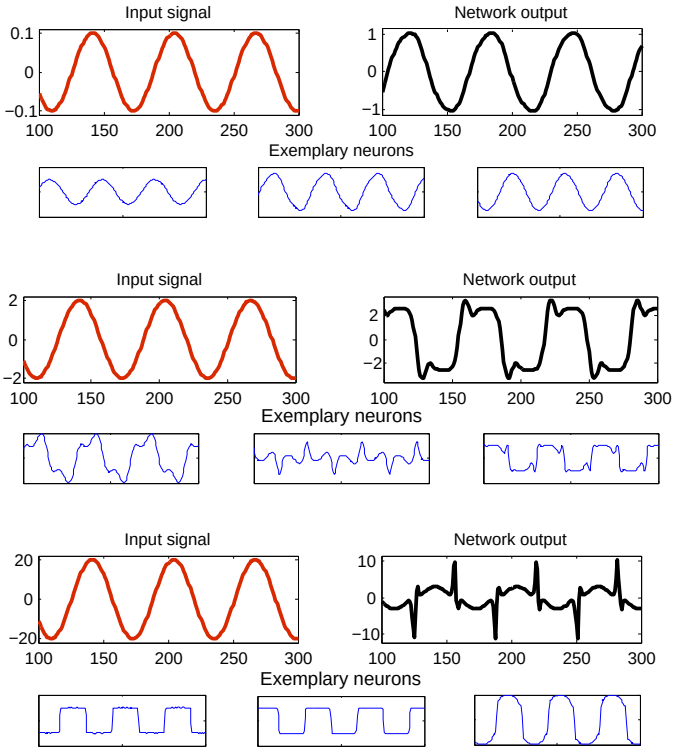

Figure 3.7: Input scaling and resulting reservoir mode (from top: linear, non-linear, binary). Example for simple sine signal on the input.

tion. Especially high dimensional multivariate input may require corrections to the scaling. Furthermore, adjustment of input scaling and spectral radius can determine whether the reservoir is input driven or rather intrinsic-state driven. Low-volume external input combined with large spectral radius will make the neurons regulated mostly by the inputs coming from the neighboring neurons, and hence the entire network will be less reactive and more autistic. On the other hand, high-volume external input and large spectral radius will increase the risk of instability.

Conclusions The purpose of this short section was to illustrate selected aspects of reservoir dynamics, but also to justify design choices that will be appearing in futher parts of the project. As we have seen, there are several parameters that can be adjusted to improve the model. Those adjustments are very task specific, and "no-free-lunch" rule applies. However, based on experience from

reservoir analysis, several good design principles can be derived. In particular, we specified decent estimates for starting values of the parameters, which will often lead to reasonably good performance. In some experiments we might indeed want to optimize those values, to find the optimal model. However in many others cases, our task will be to illustrate certain problem or regularity, and not necessarily to find the state-of-art model. In such cases, we will simply resort to those good-guess parameters, and concentrate on other aspects of the experiment.

3.6.2 ESN training for time series forecasting

Having introduced the essential paradigms and observed some basic dynamic features of the reservoir, let us now define a typical training task. We will generate training/validation/testing datasets, construct an ESN network and train the readout layer to make a single-step ahead prediction. We will observe the results both visually and in terms of MSE error, as defined before. This example will give us insight into the routine, that will be many times repeated in similar form through the rest of the thesis, when we address such issues as parameter optimizations, committee approach and financial forecasting.

Data preparation MG70 time series (as defined in section 3.4) will be used for the purposes of this experiment. The first step is to split available data to training samples, validation samples (optionally), and testing samples. We assume the limited dataset of 800 training samples and another 400 for testing. From the training dataset the first 50 samples will be used to wash out the initial random state of network, next 600 for the actual training and 150 for validation (see Fig.3.8). In fact, validation data can be omitted, but in such case we loose the ability to estimate generalization ability of the predictor. The validation becomes essential in case if we need to optimize regularization parameter of the ridge regression, or other parameters of the system. After validation is finished, the validation samples can be concatenated with the training dataset, and network system can be retrained, so that no data is wasted. This will be our common routine. The training data is assumed to be unknown *a priori*, and therefore cannot be used for any design or optimization decisions. A teacher dataset also needs to be created, which in this case corresponds to the input data shifted by one position backwards. The data is scaled and normalized to the range $[-1, 1]$.

Network preparation In the next step, ESN model is constructed following the procedure outlined in section 3.2. In this experiment we do not concentrate

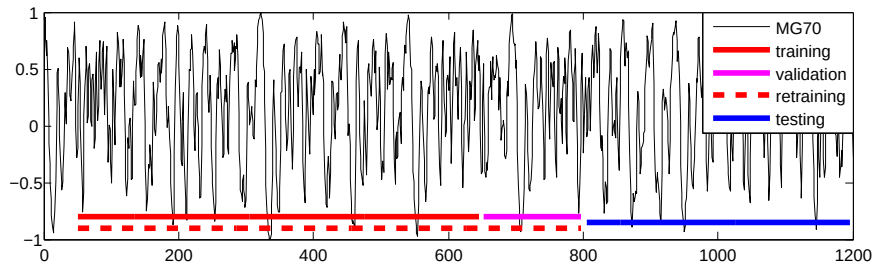

Figure 3.8: Mackey-Glass (τ=70) time series used in the experiment - split of samples into training/validation/testing datasets.

on optimization parameters but take typical values $N = 400$, $p = 0.9$ and $c = 0.05$, as justified in section 3.6.1. The network will receive one input time series (MG70) and constant bias signal. Only one readout will be trained to make a single-step ahead prediction of the target (teacher) signal.

Training and regularization The training starts from feeding the available data to the network inputs, collecting the state matrix (see section 3.1), and finally computing the output weights vector W_{out} with regression. We could use simple least-squares regression with pseudo-inverse matrix calculation ([1]) for this purpose. This can however lead in certain circumstances to over-fitting, as we shall see in the next sections. Basically, over-fitting is a problem of excessively large weights, resulting from precise matching the model to the training dataset, including its random noise and oscillations. It has a negative impact on generalization ability.

Alternatively, we can employ ridge regression method [24], which involves regularization. The regularization parameter λ will penalize large weights, leading to more conservative outputs. Although training error will increase, the validation error will be reduced due to changing the ratio between the variance and bias components of the output error. This is known as a bias-variance trade-off. Finding the optimal bias-variance ratio for a given task requires however a careful optimization of the regularization parameter λ. Too large λ will excessively shrink the response, while too small λ will give results comparable to those of simple least-square regression.

The optimization of regularization parameter is accomplished by multiple training of the readout matrix W_{out}, in every iteration incrementing λ with logarithmic step 0.5 in the range $\left[10^{-8}, 10^{2}\right]$. The resulting performance is evaluated on the validation dataset. For better accuracy, cross-validation scheme can be used.

The regularization parameter resulting with lowest validation error is considered as the optimal one (λ_{opt}), and is later used to retrain the network - this time on a concatenation of training and validation datasets. The regularization will be our common routine when working with single-model predictors. However, as we shall see in Chapter 4, advantage of regularizing individual networks is not that obvious in case of combined models, such as averaging committees.

Results and discussion Having trained and optimized the network, let us now discuss the obtained results. At first, we shall look at the effect of regularization on generated the output weights vector W_{out}. We will compare our trained network with another, structurally identical network, which was trained with classic least-squares regression. Fig.3.9(top chart, solid line) shows the process of optimizing regularization parameter, i.e. searching for optimal value of λ. It is clearly visible, that the validation MSE error is minimal for $\lambda_{opt} = 10^{-0.5} \approx 0.32$. Taking either too high or too low values will degrade the performance. Using λ_{opt} to ridge-regress the output, we obtain superior performance of the system over that of the non-regularized ESN - the testing errors are indicated by dashed lines.

The two other charts of Fig.3.9 present weight vectors W_{out} obtained with ridge regression (middle chart) and least-squares regression (bottom chart). Ridge weights have significantly narrower Gaussian distribution, and lower mean of absolute values. This is the effect of regularization, that penalizes the large weights.

Fig.3.10 shows the input and output of the network, including 50 last samples of training data and 150 samples of validation data. At the time step $t = 650$ the first regression was performed, therefore we see clear change in the generated output, which starts to reflect the desired target signal. The output is now an optimal, linear combination of the activations of reservoir neurons. Several exemplary neurons are displayed below input/output plots, showing rich diversity of nonlinear transformations of the input signal.

To evaluate the model quality, we need to observe its prediction accuracy, i.e. how well it can model the testing range of the time series, that has not been exploited during the training. In other words, we observe how much the trained network output Y diverges from teacher signal D. In section 3.5 we introduced different error measures, that can describe prediction accuracy in a quantitative way. The mean-squared error (MSE) is the most important measure for our purposes, that will be frequently used all through this thesis, for design decisions, optimizations and comparative studies. Other error measures can be useful in certain situations, especially RMSE or MAE, that have the advantage of being

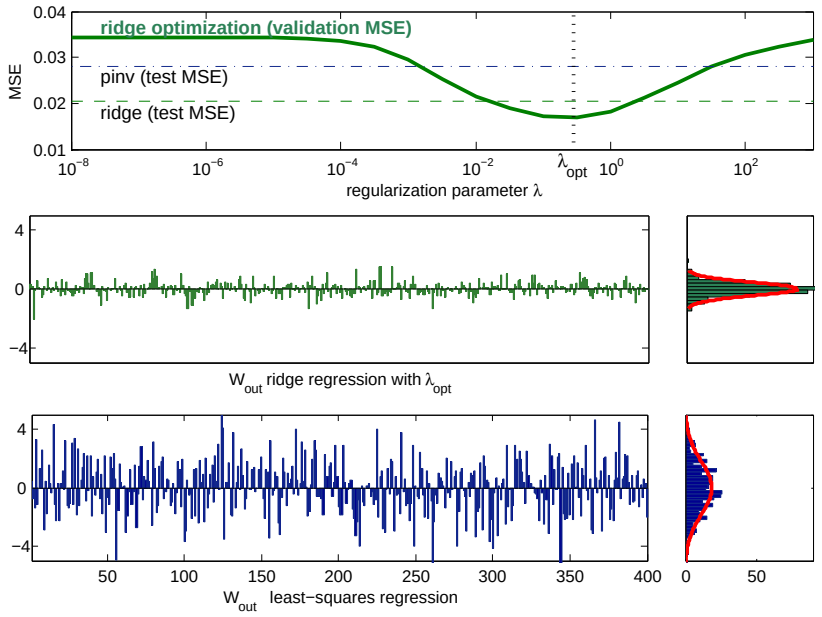

Figure 3.9: Optimization of regularization parameter λ (top), ridge regression weights distribution (middle), least-squares regression weights distribution (bottom).

measured in the same unit as the signal itself. Table 3.1 presents prediction errors that were obtained in the experiment. The errors for non-regularized model (trained with least-squares regression) and regularized model (trained with ridge with optimized λ) are compared.

Table 3.1:

Time series	MSE	RMSE	SMAPE	MAE
ridge optimized	0,020	0,143	18,95	0,113
least-squares	0,028	0,167	21,82	0,125

Fig.3.11 illustrates the output of regularized network, set together with the teacher signal. The first 200 samples of the training dataset are in focus. It seems that the network achieved reasonable accuracy and managed to capture chaotic dynamics of MG70. However, we shall soon see that we can obtain significantly better results, in terms of MSE error. First of all, careful optimization of network parameters would boost the performance, this will be a subject of section 3.6.4.

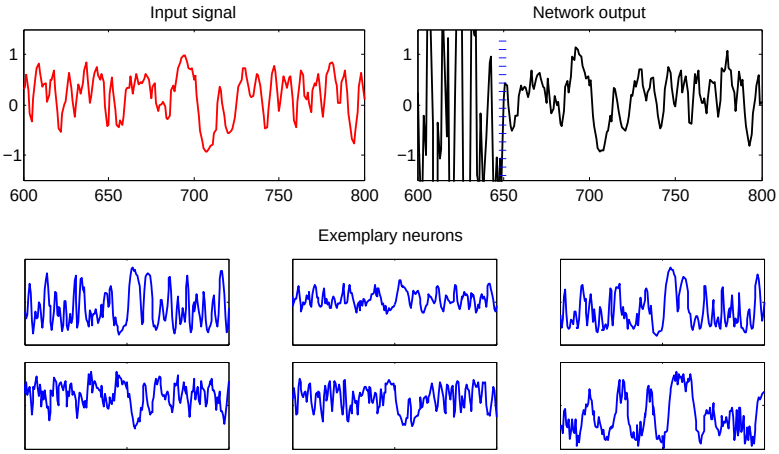

Figure 3.10: ESN training results - input, output, and exemplary neural activities. Effects of readout regression visible from time point $t = 650$.

Secondly, individual ESNs can be combined into voting collectives. We will treat the topic in detail in Chapter 4. Finally, in further experiments we will show that optimal prediction horizon k for cyclic or seasonal time series is often higher than 1. For instance, the optimal prediction horizon for MG70 is $k = 12$, which means that the best approach is to train the network to forecast 12-steps ahead.

3.6.3 Trajectory projection

So far we have been concentrating on prediction of the next value of given target time series, so that $y_t \approx d_t$, based on recent input history $U = \{u_1, ..., u_{t-1}, u_t\}$ available at time t. In particular, target signal may correspond to the next value of input signal, that is $d_t = u_{t+1}$. While in many prediction tasks such one-step-ahead prediction will be sufficient, some others will require a projection of entire trajectory of k subsequent values of target signal $Y_{traj} = \{y_t, y_{t+1}, ..., y_{t+k-1}\} \approx \{d_t, d_{t+1}..., d_{t+k-1}\}$. If this is the case, there are two alternative techniques that can be employed: feedback-loop recursive prediction and multiple-readout trajectory prediction.

Feedback-loop recursive prediction Using this technique, the trajectory is projected by recursive feeding recent the outputs to the network inputs, and

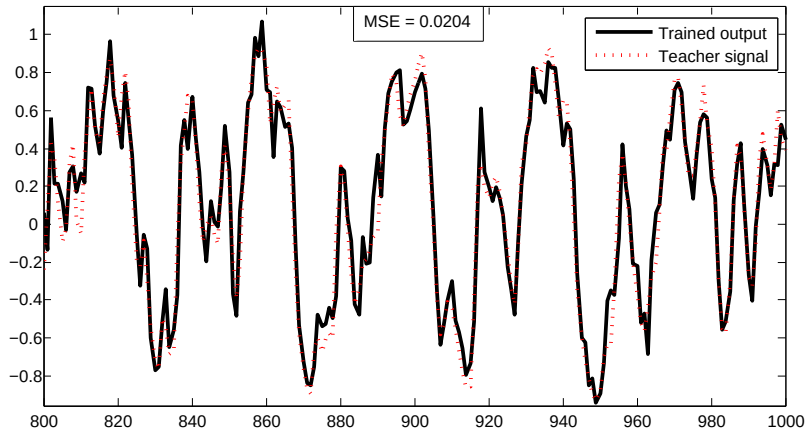

Figure 3.11: ESN training results - trained output vs. teacher signal. Comparison for the first 200 testing samples.

updating the states respectively, so that the network works in a closed-loop generator mode. In particular, the input term u_{t+1} in the equations 3.3 and 3.4 is replaced with the output term y_t. In this mode, it is essential that all the input terms are predicted by the output, so that $y_t \supseteq \widehat{u}_{t+1} \approx u_{t+1}$, where $\widehat{u}_{t+1}$ is a prediction of next input value. If the input is multivariate, it implies that multiple readouts need to be trained.

The advantage of the method is its conceptual simplicity and speed of training in case of univariate or low-dimensional inputs. In certain cases it may have better prediction accuracy than the alternative method, especially in long-term trajectory projections if one-step-ahead forecast can be done precisely. The drawbacks of the method is that all the input signals need to predicted, even if they are not a part of the target signal. Moreover, prediction error is propagated in every step. The error will quickly accumulate if the data is noisy, especially in case of multivariate input, where every predicted variable introduces additional error term.

Multiple-readout trajectory projection In the contrary to feedback-loop, this technique immediately predicts the entire output trajectory. This is achieved by training k independent outputs, each regressed to predict $i'th$ step-ahead value of the target signal, where $i \in \{1, ..., k\}$. In other words, k-dimensional output corresponds to predicted future trajectory of the target signal, up to k steps ahead.

The significant advantage of the method is that it helps to identify the prediction horizon that will give statistically most accurate prediction (critical point), which is task-specific and often different that 1 due to cyclic or seasonal dependencies. We will demonstrate in section 3.6.4 that prediction error in such critical points can be significantly lower than in the surrounding points. Another benefits of the method are clear if we consider high-dimensional multivariate input. Only the target signal must be predicted with this method, in the contrary to feedback-loop trajectory predictions. On the other hand, the method will not be very practical if very long trajectories need to be predicted, since that would require training of many readouts.

Exemplary trajectory The multiple-readout method will be preferred in our further work, for several reasons. First of all, both MG time series and financial data series contain cyclic and seasonal dependencies that will desirable to detect. Secondly, financial time series involve substantial noise, what together with multivariate input makes feedback-loop trajectory forecasting rather inefficient.

Fig. 3.12 shows 30-steps trajectory projections for various time series. For each time series 50 trajectories are plotted, computed by independent echo state networks. The precision of projections deteriorates along with complexity of the tasks (MG20-MG70), what is reflected by increasing variance of individual responses. Furthermore, the regularized networks (left-hand side plots) are more precise and display lower variance, than the non-regularized networks (right-hand side plots). While this is desired in case of individual networks, the issue is rather non-trivial in committee settings, where member variance can give certain benefits. We shall discuss those issues in Chapter 4.

Optimal prediction horizons (critical points) Observation of trajectory projections indicates that variance of individual network outputs is dependent on prediction horizon, however the uncertainty of prediction is not necessarily growing monotonically as the prediction horizon increases. To justify this statement, we will repeatedly generate 30-steps-long trajectories (such as showed by Fig.3.12) through entire testing dataset, what will result in 370 projections on the sample ranges $801, ..., 830, 802, ..., 831, ..., 1171, ...1200$. Moreover, all the projections will be repeated by employing 50 independent networks. Finally, for each prediction horizon $h \in \{1, ..., 30\}$ statistical prediction error will be computed by averaging through all 371 projections and 50 networks, so as to eliminate random influence of data subsets or networks. As a result, we obtain statistically credible evaluation of prediction MSE as a function of prediction horizon and hence we can estimate the optimal prediction horizons (see Fig.3.13).

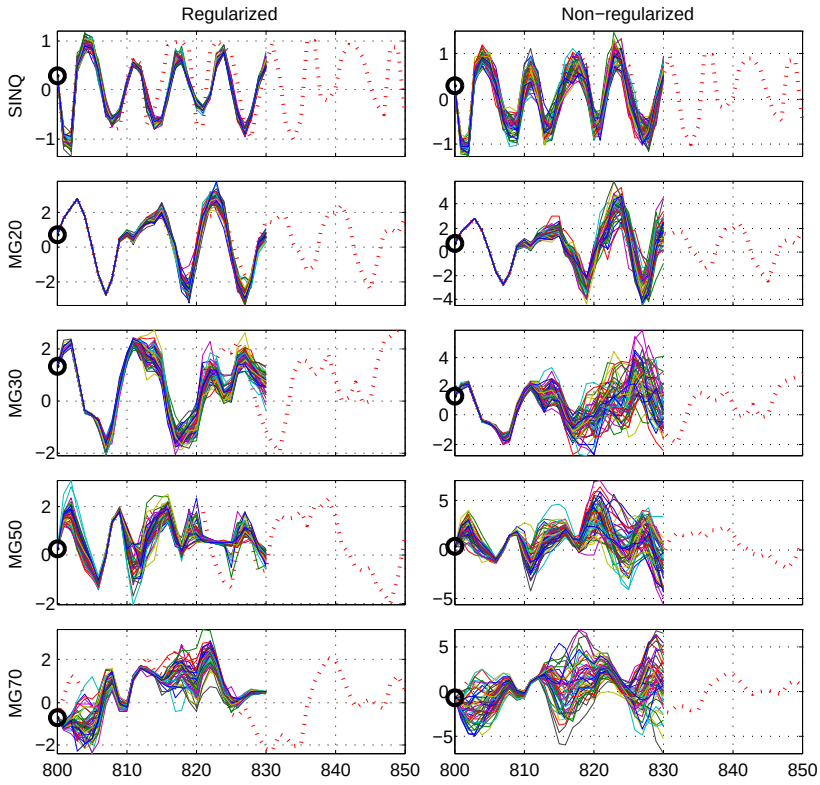

Figure 3.12: Trajectory projection with multiple-readout method. Left column presents regularized ESNs, right non-regularized. Each chart plots trajectories generated by 50 independent ESNs. Red dotted line denotes target signal.

Observing the results, we clearly see that the best prediction accuracy is not always achieved with one-step-ahead prediction ($h = 1$). Instead, it depends the dynamics of the underlying system and on the predictor characteristics. The former could be estimated with help of autocorrelation function, however to account for the latter we need to perform experiments as described above. Besides, by training multiple readouts we automatize the process and do not need to apply additional autocorrelation analysis.

Often the time series with seasonal or cyclic dependencies have the optimal prediction horizon for $h > 1$. This is also the case of the MG timeseries, due to $x_{t-\tau}$ component (see Table 3.2). We will call those prediction horizons as "optimal prediction horizons" or "critical points" and denote them with h_{opt}. Those points will be observed with special attention in further experiments,

because they give the minimum mean squared error of the prediction. Therefore they would be primarily used for the target signal reconstruction. We can easily identify those points, since as a common routine we will train multiple readouts for subsequent prediction horizons, just as we do while projecting trajectory with multiple-readout method.

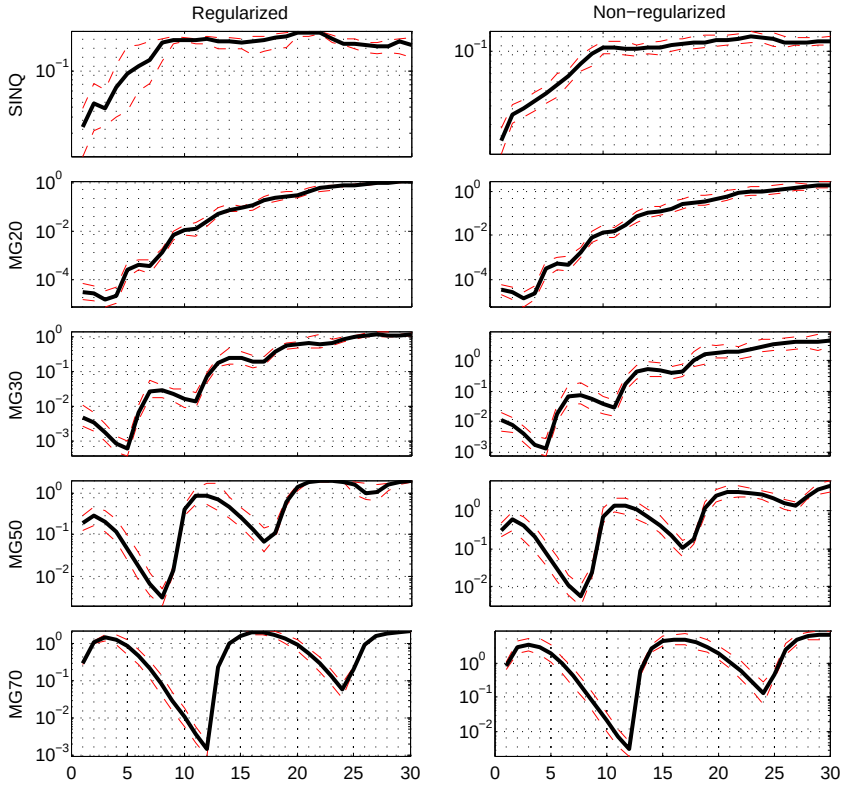

Figure 3.13: MSE error as a function of prediction horizon h_{opt}. Each chart is generated by averaging through 371 trajectories trajectories generated by 50 independent ESNs. Red dotted lines denote min/max errors achieved for given h by one of the ESNs.

Combining methods Considering possibility of h_{opt} being greater than 1, in certain cases it could be beneficial to combine the feedback-loop recursive method with multiple-readout trajectory projection method. The optimal prediction horizon h_{opt} should be first estimated with multiple-readout method.

Table 3.2: Optimal prediction horizons (h_{opt}) for selected time series.

Time series:	SINQ	MG20	MG30	MG50	MG70
Critical point:	1	3	5	8	12

Following that, the readout corresponding to h_{opt} can be used with feedback-loop method to project the future trajectory. In such case, in time step t the network receives on input its output from h_{opt} steps earlier, and produces new output y_t that will be first used h_{opt} steps ahead. Strictly speaking, we would have $y_t \supseteq \widehat{u}_{t+h_{opt}} \approx u_{t+h_{opt}}$, where $\widehat{u}_{t+h_{opt}}$ is a prediction of the input value h_{opt} steps ahead. We keep this method in mind as a potential tool for projecting long trajectories, however we do not further elaborate it at the current stage.

3.6.4 Optimization and regularization

For the purpose of optimization considerations, we will use several distinct Mackey-Glass time series - MG20, MG30, MG50, MG70 with time lags $\tau = \{20, 30, 50, 70\}$, respectively, as defined by equation 3.8. Instead of producing arbitrarily long time series, we assume limited availability of data, which is often the case in the real engineering applications. We assume to have 800 data samples available for the training and another 400 samples to test the system. We split the data in proportions 50/750/400, meaning that initial 50 samples were used to flash out random initial states of ESNs, 750 samples were used for members training, and 400 samples served to evaluate the performance. In case of the regularized ESNs, the last 150 samples of the training data set are used for cross-validation estimation of optimal λ, as was discussed in section 3.6.2. Another time series that will be used as a benchmark is a sine-based signal, defined earlier by equation 3.10. This time series is an example of linearly generated data without noise, but displaying interesting oscillating behavior. It is not trivial to forecast accurately, unless the training data is sufficiently long. Here we split the data in exactly the same proportions as in case of MG datasets - 50/750/400.

Reservoir size and regularization In section 3.6.2 we have already discussed training and regularization, taking arbitrary network as an example. Now we shall present more general and statistically correct analysis. In the first step, we shall look at two essential factors influencing ESN performance - model size (reservoir size, N) and regularization, while holding spectral ra-

dius and connectivity ratio on fixed levels, that is $\rho = 0.9$ and $c = 0.05$. Since there is no regularization, we should be able to see the effect of over-fitting if the reservoir size becomes too large, in particular if the number of neurons gets close to the number of training samples available. Remembering that our training data is limited to 750 samples, we will consider reservoir sizes in the range $N \in 100 \div 800$, which should suggest us the optimal size but also illustrate the effect of over-fitting. Fig.3.14 (left-hand side plots) presents performance of individual nonregularized ESNs on the Mackey-Glass time series (results limited to the optimal prediction horizons, see Table 3.2). Indeed, we see that testing error starts to increase when the reservoir grows above its optimal size, which varies between 250 and 500 neurons, depending on the time series. The over-fitting is particularly visible when $N \sim 750$, since the state-collecting matrix becomes nearly square and regression tends to excessively amplify the noise of training data set. It corresponds to 30-60% of the available training samples. In other words, if the number of model parameters is equal to the number of training samples, regression almost directly maps the training data samples onto the outputs, at the cost of oversized weights and degraded generalization ability. Interestingly, if the number of neurons is increased further, the excessive neurons begin to regularize the outputs, what is reflected by decreasing testing error when $N > 750$. We will come back to this aspect in more detail when we discuss committee optimization in Chapter 4.

For the contrast, the right-hand side plot presents performance of the identical ESNs, but this time optimally regularized. We observe asymptotically decreasing testing error and larger optimal values for the reservoir sizes. Furthermore, the regularized ESNs achieve lower MSE than the non-regularized. However, as we shall see in Chapter 4, it does not always apply to ensembles of ESN models. For know we conclude that both reservoir size and regularization have essential influence on ESN performance. Since reservoir size may display regularizing effect, we will usually consider reservoir size N in conjunction with regularization parameter λ when optimizing.

Spectral radius and connectivity ratio The other two parameters that we shall consider together, are connectivity ratio c and spectral radius ρ. The dependence between them was already discussed in section3.6.1. We mentioned that spectral radius should be set low enough to ensure stability of the system, and high enough to provide sufficient memory effect (echo property). Adjustment of c with invariant ρ balances the relation between density and average strength of connections. We shall look now at how those factors influence the performance of ESN network in particular prediction tasks.

The same time series will be used as previously - MG20, MG30, MG50, MG70.

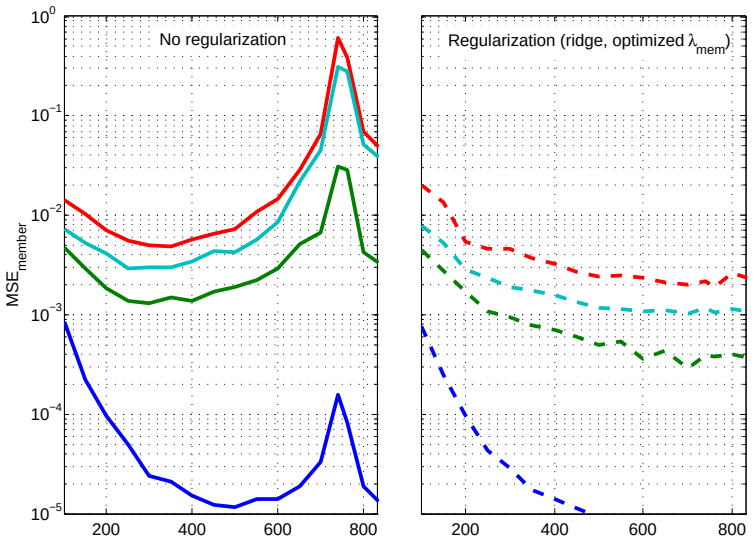

Figure 3.14: ESN performance. Examples for following time series (from top): MG50, MG70, MG30, MG20.

The data split into training/testing samples, as well as the training routine will be identical like before. For each of the time series the optimal reservoir size will be selected and kept invariant, based on results of the previous experiment (Fig. 3.14, left-hand side). We shall constrain now to non-regularized networks. The spectral radius will be varied in the wide range $\rho \in [0.2, 1.3]$ and connectivity ratio in the range $c \in [0.01, 0.20]$. We will perform a grid search of the parameters yielding optimal performance, in terms of MSE on the testing dataset. For each combination $\{\rho_i, c_i\}$ five networks will be trained and evaluated, and the final MSE error will be averaged through all the trials. The results are summarized by Fig.3.15.

The results lead to several conclusions. First of all, it is clearly visible that there is an optimal range of spectral radii, approximately $\rho \approx 0.5 \div 1.0$, and either lower or higher values will degrade performance due to reasons pointed out earlier. Secondly, varying connectivity ratio between $1\% - 20\%$ has minimal impact on the final performance. This justifies our decision to keep c invariant in the further experiments, and instead concentrate on optimizing other parameters like reservoir size, regularization parameters, and in certain cases spectral radius. The connectivity ratio will be fixed on $c = 0.05$. Finally, we observe significant robustness of the reservoir network, in respect to connectivity ratio and spectral radius. The similar efficiency is achieved for relatively wide ranges of c and ρ. As a result, it is relatively easy to obtain the reasonably good reservoir for

given task, and optimizations are only needed if the task requires state-of-art performance.

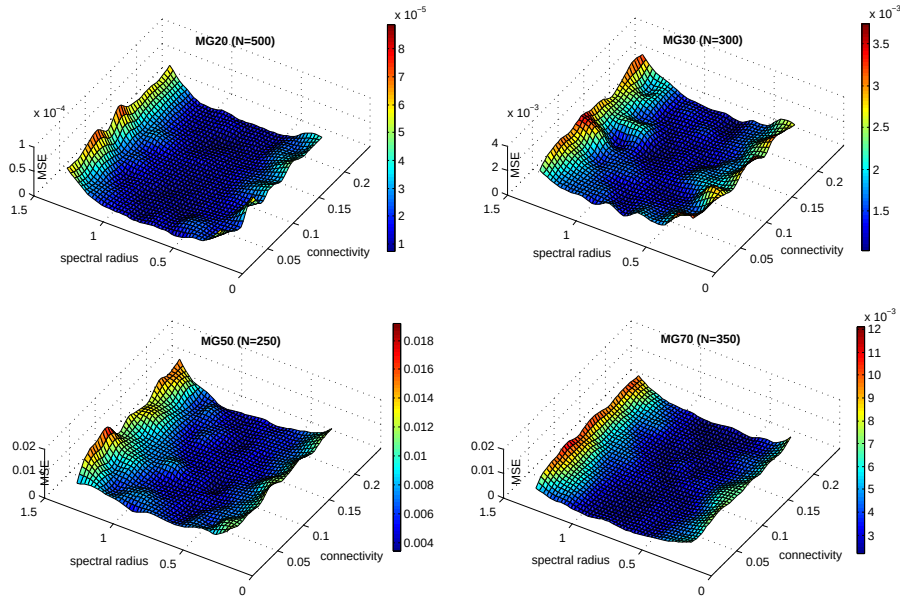

Figure 3.15: ESN optimization of spectral radius and connectivity. Results for MG20, MG30, MG50, MG70 tasks, with optimal reservoir sizes $N = \{500, 300, 250, 350\}$ respectively.

Other parameters, such as scaling and shifting the inputs, could be optimized in the similar way. However, exhaustive grid search is computationally demanding, and therefore other techniques are often applied - for instance analytical methods, heuristics, or evolutionary algorithms [9, 20].

3.7 Summary

In the Chapter we discussed the most important aspects of echo state networks, such as stability issues and reservoir dynamics, training methods, regularization and optimizations. We certainly did not exhaust the topic and neither was it our objective. Instead we refer the reader to numerous publications treating the ESN design issues in detail (section 3.1). Our goal was to introduce and give insight into the model that constitutes the base element of the voting committees, that we introduce in Chapter 4. Furthermore, we introduced and discussed several

routines and concepts, that will be appearing repeatedly through the rest of the thesis, such as cross-validation, ridge regularization and over-fitting, prediction error metrics, trajectory projection, critical prediction horizons.

Chapter 4

Reservoir Committee Methods

The purpose of this Chapter is to organize populations of echo state networks into larger structures, that we will refer to as "reservoir committees"[1]. Committees, as well as mixtures of experts, were already reported to be used in conjunction with echo state networks, but little research was done to thoroughly compare different types of such structures and focus on important design considerations related to this approach.

In the engineering applications it is common to train multiple models to solve given task (possibly varying parameters), and then use the best found model as a predictor. Other approach is to utilize all (or selected subgroup) of trained models by combining them into a voting committee. In majority of cases, the system output is computed simply by averaging individual outputs. Since reservoir is randomly initiated every time, each model will display slightly different dynamics and temporal characteristics, and therefore averaging will help to eliminate variance component from the error. However, some of the models will be more adequate to given task than the others. This difference will be even more visible in case of heterogeneous committees, where members differ in terms of parameters used to generate reservoirs, or in terms of inputs they receive. Theoretically,

[1] Note that the terms: ensemble, committee and (voting) collective are used interchangeably and refer to the same concept

it should be possible to exploit such variance to predict target signal more accurately. An efficient ranking algorithm should properly identify best fitting members and reward them with higher weights. Simultaneously it should filter out or suppress the outlier members that perform particularly poor. As a result, different members will contribute to final forecast with different intensities. In certain circumstances such solution can be more efficient than distributing equal votes to all the members without considering their performance.

Those approaches will be evaluated in the subsequent sections of this chapter. We start from short review of methods for combining models in section 4.1. Following that, in section 4.2 we introduce ranking algorithms that constitute interesting alternative to averaging committees. Section 4.3 presents results of comparative empirical studies. We discuss here different types of committees and highlight several important design considerations. Section 4.4 concludes the Chapter.

4.1 Committees and combining models

Various techniques were developed to improve prediction performance by combining several models into larger scale hybrids. Below we briefly review this methodology. We do not constrain to ESN models. In fact, any simple or complex model can be used as a member of such hybrid, including linear regression models, classifiers, probabilistic models, neural networks, expert systems or even constants.

The most common approach is a simple averaging committee [27]. An averaging committee combines M members, either homogeneous or heterogeneous models, and produces output by averaging the individual outputs:

$$y_{com}(x) = \frac{1}{M} \sum_i y_i(x) \tag{4.1}$$

Averaging committee is easy to create and in many cases will perform surprisingly well. By incorporating a range of similar members and averaging their outputs, the prediction error can be minimized due to reduced variance of individual outputs. Averaging acts as a regularizer and by smoothing the output it prevents overfitting. It can be shown that in the ideal case, assuming uncorrelated member errors with zero average, the committee can reduce prediction error by factor of M [25]. This however never happens in practical situations

and the error correlation can be quite significant. Nevertheless, the committee usually outperforms average members, and even the best fitting members.

To construct a committee, we need a selection of members trained on the same task, but varied in terms of dynamics. The variance can be obtained either by diversification of the members intrinsic structure, or with help of bootstrap aggregation, where the members are structurally identical by trained on different subsets of the training data. In case of ESN models, such variance arises naturally because every time when a new network is created, its reservoir is generated randomly with respect to given parameters.

The concept of committee can be easily generalized by replacing simple output averaging with weighted averaging . The output of such a generalized committee [27] is defined as:

$$y_{com}(x) = \sum_i \omega_i y_i(x) \tag{4.2}$$

where ω_i is a weight associated with $i'th$ member and can be selected arbitrarily with constraint $\sum \omega_i = 1$. In practice, finding efficient committee weights can be a non-trivial task. Similar problems need to be faced as in case of model training, in particular over-fitting. In fact, estimation of committee weights will be the subject of section 4.2 of this paper.

Another interesting combination of models is known as adaptive mixture of experts [28]. Similarly like a committee it combines M members, but the overall output is determined by input-dependent gating function. The members are trained as experts in particular regions of input space, while the gating function is trained to select appropriate expert depending on which region current input belongs to. The output of mixture of experts is defined analogously to that of generalized committee, except that constant weights are replaced with input-dependent mixing coefficient $\alpha_i(x)$:

$$y_{mix}(x) = \sum_i \alpha_i(x) y_i(x) \tag{4.3}$$

The gating function may have 'hard' or 'soft' form, i.e. only one best-matching expert is selected, or several are selected, each with different weight. The concept of mixtures is taken even further with hierarchical mixtures-of-experts [29]. In fact, ESN based mixture of experts was already investigated in [23], where the expert functionality as well as the gating function were delegated to distinct

echo state networks. In another work [16], mixture of experts was modelled with help of lateral inhibition between competing ESN experts.

Another way of connecting models is represented by boosting [30, 31] or corrective chains. With boosting method, models are trained in sequence in the way that every next iteration takes consideration to the training dataset, but also to the error information coming from the previous iteration. For instance, the original data samples can be associated with weighting coefficients, which depend on the performance of the previous iteration predictor on those samples. In this way, subsequent predictors become more sensitive to those regions of the data that were poorly learnt by the previous ones. Final response is computed either as a weighted average of all members in the chain, or as the response of the last member in the chain, which in theory should account for the expertise of the previous ones. Boosting method was shown to perform well even if base models are non-optimal, weak learners. Corrective chains have similar purpose as boosting, yet work in slightly different way. Again models are trained in sequence, but every subsequent member intends to predict the error term of the previous one. Hence, when the training is finished, adding the outputs of the subsequent models to the output of the first one should compensate for its error. Promising results were reported after implementing this method with ESN as the base model [22].

For completeness we should also mention decision trees, or tree-based models, that are primarily used in classification tasks of limited complexity. Those models split input space into finite number of regions, and assign a model (usually simple, e.g. constant) to each of them. Once the tree is constructed, the appropriate model for given input is found after traversing the tree, what corresponds to taking sequential discriminative decisions in the input space.

Some other solutions, that do not fit to one of the aforementioned categories, can be generally referred to as modular or hierarchical systems. Those systems implement divide-and-conquer strategy to split a problem into a set of related subproblems, and handle each of them with a separate model. The models can be independent or arbitrarily connected, including complex structures with recursive dependencies. The approach has a large potential, and in theory can simulate any complex, chaotic system with arbitrary accuracy. On the other hand, it introduces multitude of additional parameters and therefore training and optimization can be complicated. Prior domain expertise is usually required for successful implementation.

As we can see, wide range of possibilities have been investigated to boost accuracy by combining multiple models. Some of the hybrid solutions overpass the boundaries and hence can not be clearly classified to just one of those categories. The family of combined models becomes even richer if we consider variety of

models that can be utilized as base components, sometimes in heterogeneous ensembles.

The models that we investigate in this thesis belong to the group of simple committees and generalized committees (Eq.4.2). Apart from averaging committees we will investigate several algorithms to determine committee weights ω_i with the purpose of improving performance. Since the sole task of those algorithms is to rank the members, we will refer to the resulting committees as "ranked committees". If we consider the ranked committees in a broader perspective and allow them to have heterogeneous members and multivariate inputs differently distributed between the members, they can be also classified as a mixture of expert with input-invariant mixing coefficient.

4.2 Ranking algorithms

Strictly speaking, the task of a ranking algorithm is to determine a committee weights vector $W_{com} = [\omega_1, \omega_2, ..., \omega_M]$ where ω_i is a weight (or voting share) assigned to $i'th$ member of the committee[2]. The vector W_{com} should be normalized so that $\sum \omega_i = 1$. In that way, the final committee output can be computed as a weighted average of the member outputs:

$$Y_{com}(t) = W_{com} \times Y_{mem}(t) \tag{4.4}$$

where $Y_{mem}(t) = [y_1(t), y_2(t), ..., y_3(t)]^T$ is a vector of member outputs at time t.

Each of the proposed algorithms consists of three phases: (1) training and optimization of the members, (2) evaluation of committee ranking vector, and (3) retraining the members on entire dataset including validation samples. The main challenge is to find efficient ranking vector W_{com}. In case of simple averaging committee, the vector simply assigns equal weights to all M members: $W_{com} = \left[\frac{1}{M}, \frac{1}{M}, ..., \frac{1}{M}\right]$. Below we propose alternative methods to determine W_{com}.

[2]Committee weights vector can be also referred to as "ranking vector" or "voting share vector"

4.2.1 Exponential ranking algorithm

The exponential method ranks the members based on their validation performance. Therefore, the first step involves training all the members and computing the corresponding validation errors, as explained in section 3.2. Either regularized or non-regularized networks can be used, and the advantages of both will be discussed later. If regularization is performed, it will use the same validation data as the committee ranking algorithm.

Once all the committee members are trained, the ranking can be computed. The validation errors $mse_i^{(val)}$ of the members are combined into error vector $MSE^{(val)}$. The committee weight vector W_{com} can now be computed according to the following formula:

$$W_{com} = exp\left(-\alpha \cdot \frac{MSE^{(val)}}{stddev\left(MSE^{(val)}\right)}\right) \tag{4.5}$$

where $stddev\left(MSE^{(val)}\right)$ is the standard deviation of members validation errors, and α is a free parameter, that we will refer to as "democracy factor". The voting vector W_{com} is then normalized, so as to serve as a weighted average over the member outputs:

$$W_{com} = \frac{W_{com}}{\sum \omega_j} \tag{4.6}$$

The purpose of $stddev\left(MSE^{(val)}\right)$ divider is to maintain voting shares relatively low even in case of large variance of validation errors, which may be the case especially if the committee contains heterogeneous members. The democracy factor α on the other hand determines whether the committee behaves in more democratic manner (low α) or favors the best trained members (high α). In particular, if $\alpha = 0$ the committee will behave like a simple averaging committee. If $\alpha \to \infty$ the committee output will be driven by the single member, which obtained the lowest validation error. However, in most cases α will be a subject of further optimization, rather than being selected manually. For this purpose, the validation dataset is reused second time to find the optimal democracy factor α. This is achieved by multiple recomputation of the ranking W_{com}, committee outputs $Y_{com}(t)$, and committee validation error $MSE_c^{(val)}$, while in every iteration the value of α is modified with logarithmic step in the range $\left[10^{-4}, 10^{4}\right]$. The process is computationally fast, because the member outputs are already computed and hence no additional operations need to to be performed on the members. As a result, we obtain α_{opt} parameter that minimizes validation error,

and which we will consider as the optimal democracy factor to be used on the testing dataset.

In the final step, having estimated the ranking vector W_{out} and the democracy factor α, we reuse the validation dataset third time, to retrain the members. We concatenate the training and the validation datasets and repeat ridge regression for each member. Optimal λ_{mem}, which was previously estimated, is used in this process. In this way none of the available data is wasted.

The entire process of generating the exponentially ranked committee is schematically presented on Fig. 4.1.

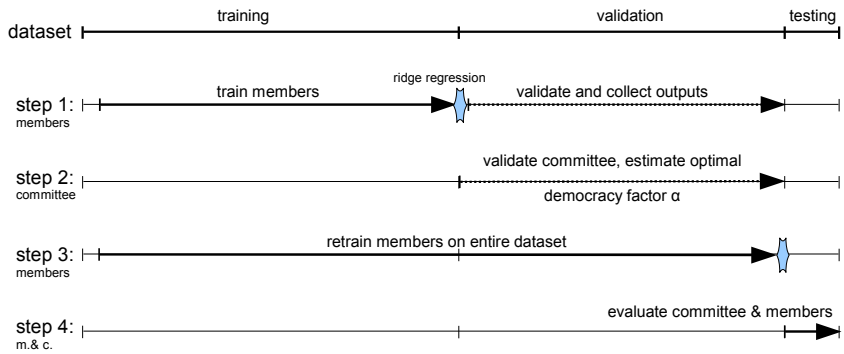

Figure 4.1: Exponential ranking algorithm - scheme. Note that regression in (1) and validation in (2) will be repeated multiple times to find optimal λ_{mem} and α, respectively.

The proposed ranking method is based on the assumption that there is a close correlation between member performance on the validation and the testing datasets. This is usually true for relatively stationary time series, especially if the validation dataset is representative and sufficiently long. For a typical heterogeneous committee (50 members with sizes $600 \div 1000$, spectral radius 0.8) trained on simple Mackey-Glass time series ($\tau = 30$) we observe significant correlation between members validation/testing errors $MSE_i^{(val)}$ and $MSE_i^{(test)}$ (see Fig.4.2). We should emphasize though, that the correlations will not necessarily be as obvious in case of highly noisy data, or non-stationary data where signal characteristics rapidly change over time, or if the validation dataset is not long enough.

Similarly, the democracy factor estimation will only be efficient, if for given committee and given task there is a close correlation between the optimal democracy level on the validation and the testing datasets, i.e. $\alpha_{opt}^{(val)} \approx \alpha_{opt}^{(test)}$. To verify

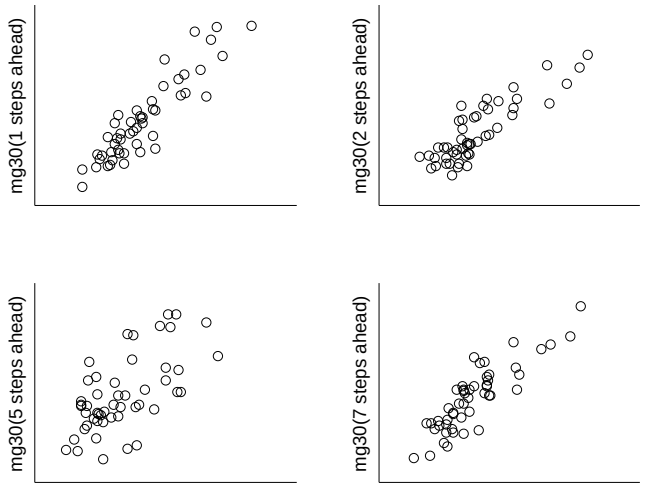

Figure 4.2: Validation/testing error correlation for committee members. Results for several exemplary prediction horizons (1-, 5-, 10- and 15-steps ahead)

this assumption we observe correlations between validation/testing committee errors $MSE_c^{(val)}$ and $MSE_c^{(test)}$ for a range of α values. For the same committee and time series as before, the clear correlation is obvious (see Fig.4.3). Modifying α initially reduces validation/testing errors up to a certain optimal point, and then starts to gradually increase it.

4.2.2 Regression ranking methods with cross-validation

Other ranking methods are based on regression of the member outputs to evaluate the optimal committee weights vector W_{com}. In the first step, the members are trained and optimized in the same manner as in case of exponential ranking methods. Afterwards, regression of the member outputs on a validation dataset is performed. The validation data should be selected as a distinct fraction of the training data, not used for members training. In this manner, committee regression will account for generalization ability of the members.The method conceptually promotes the ESN training method to the higher hierarchical level - it applies on the committee level the same regression algorithm that was previously used to train individual ESN members. The neural activation history is replaced with the member output history, and the resulting vector is the committee weights vector W_{com} instead of ESN output weights vector W_{out}.

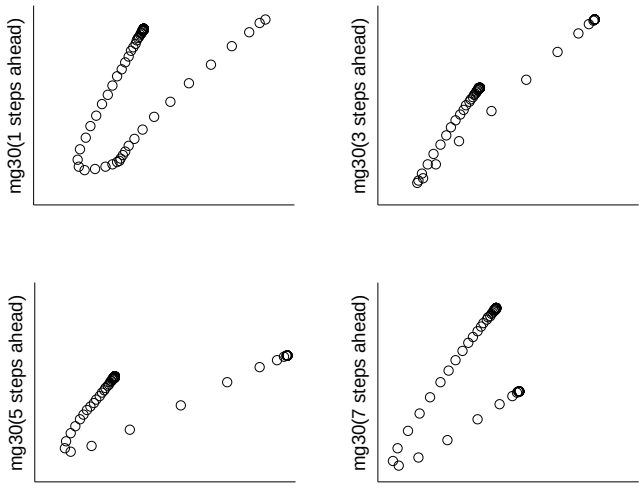

Figure 4.3: Validation/testing error correlation for range of alpha parameters. Results for several exemplary prediction horizons (1-, 5-, 10- and 15-steps ahead)

We found that simple least-square regression has a tendency to overfit the committee to validation data. Therefore we added regularization component λI to penalize large weights (ridge regression). This method however requires careful optimization of λ to work efficiently. This forced us to further split the validation dataset V into two subsets V_1 and V_2 - committee training and validation datasets correspondingly. We can now repeat regression on V_1 for the range of λ values, and use V_2 to find the λ value which ensures highest generalization ability of the committee, i.e. minimizes the committee mean-squared error on V_2 dataset. To make better use of usually limited dataset V, we apply k-fold cross-validation scheme. It means that V is splitted into k ranges of equal length, out of which one is used as V_2 and the rest as V_1. Evaluation is repeated k times, shifting V_2 in every iteration. Finally, the committee performance on V_2 is averaged over those k evaluations.

As a result of those operations, λ_{opt} is obtained, which will be considered as the optimal committee regularization parameter. Finally, committee ridge regression is repeated, this time on the entire validation dataset V and with the optimal regularization parameter λ_{opt}. In the very last step, all members are retrained on the entire dataset, including training and validation samples, to ensure that no data was wasted. The subsequent steps of the algorithm are illustrated by Figure 4.4.

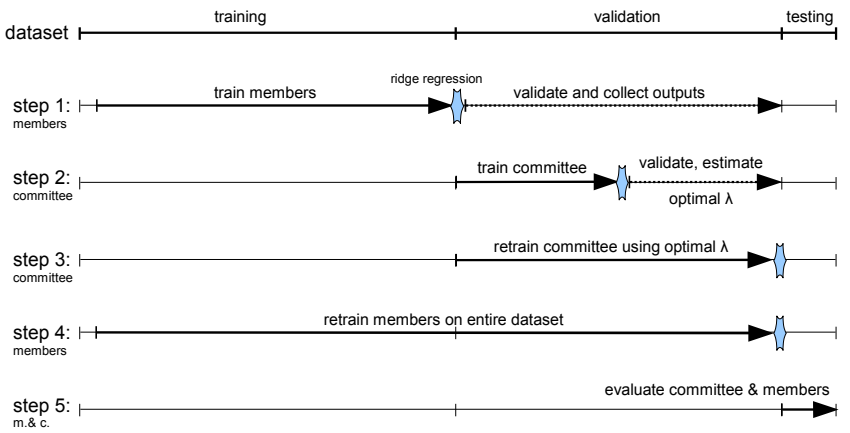

Figure 4.4: Regression based ranking algorithm - scheme. Note that regression in steps (1) and (2) will be usually repeated multiple times to find optimal λ_{mem} and λ, respectively.

The presented method gives promising results. However, in case of very noisy data or large prediction horizon, it has a tendency to excessively increase λ_{opt}. This will lead to underestimating the target signal values and "shrinking" the committee output. To deal with the problem, we introduce generalized ridge regression, defined as:

$$W_{com} = \left(Y^T Y + \lambda I\right)^{-1} \left(Y^T D + \lambda \omega_0\right) \tag{4.7}$$

where D is the desired output vector (teacher signal), Y is a matrix collecting outputs of committee members over the validation dataset, and ω_0 is a convergence vector, to which the regressed weights will be gradually converging with increasing λ. Note that setting ω_0 to vector of zeros will result in standard ridge regression. However, ω_0 can have any arbitrary values. The method gives us an opportunity to compute voting vector W_{com} as a trade-off between (1) averaging committee and regression ranked committee, or (2) exponentially ranked committee and regression ranked committee. In the first case, the convergence vector is set to $\omega_0 = \left[\frac{1}{m}, \frac{1}{m}, ..., \frac{1}{m}\right]^T$ being adequate to simple averaging committee. In the second case, the convergence vector is set to $\omega_0 = \omega_c^{(exp)}$, where $\omega_c^{(exp)}$ is exponentially ranked voting vector, as described in section 4.2.1.

4.2.3 Committee types and terminology

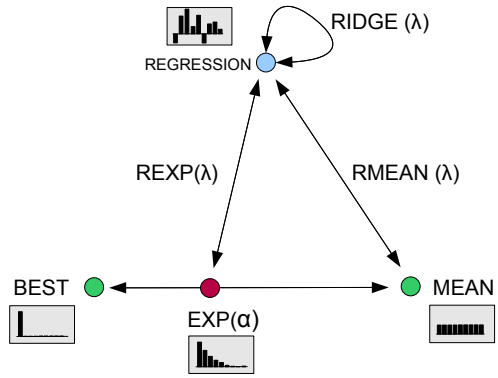

Figure 4.5: Committee ranking methods scheme

In the experimental studies we will compare several types of committees on different prediction tasks. Let us introduce the following terminology to refer to those methods:

- MEAN - simple averaging committee
- BEST - committee with output identical to the best performing member
- EXP - committee with exponential ranking
- RIDGE - committee with ridge regression
- RMEAN - committee with regression converging to mean weights
- REXP - committee with regression converging to exponentially estimated weights

Fig.4.5 illustrates direct relations between the methods. Adjustment of α with EXP algorithm determines how much the weights diverge from MEAN towards BEST. Adjustment of λ in case or regression based rankings determines how much the weights diverge from least-square regressed towards MEAN, EXP or RIDGE, respectively.

4.3 Experimental results

In order to evaluate the benefits of simple committees and generalized ranked committees, we performed numerous experiments on several different time series. In particular, Mackey-Glass time series and non-trivial sine-based signals, that we introduced earlier in section 3.4. Moreover, in every experimental setting we trained k independent outputs for every ESN member, each of them trained to predict $i'th$ step-ahead value of the input signal, where $i \in \{1, ..., k\}$. Since each committee member has k outputs, the committee ranking algorithms described in the previous section are repeated k times to produce k committee outputs (one for each time horizon). In other words, k-dimensional output can be understood as a projection of future trajectory of the target signal, up to k steps ahead. Such training scheme allows us to identify the prediction horizon that gives the best prediction accuracy, and helps to evaluate robustness of the compared algorithms. Exemplary committee outputs are illustrated by Fig.4.6 and will be discussed later.

In the experiments we will evaluate performance of different committee models. In particular, we will compare the ranking algorithms (averaging, exponential and regression based), simulate committees varied by size of the base model reservoirs $N \in 100-2000$, and evaluate committees with/without regularization of the base models. In any case, other parameters will be kept invariant. The committee size is set to $M = 50$ members, which is large enough to ensure high repeatability of training results. In case of the ranked committees, the number of cross-validation folds is set to $f = 8$. The connectivity ratio of member reservoirs is fixed at $c = 0.05$ and the spectral radius at $p = 0.9$. Direct input-output connections, as well as feedback connections are disabled. At the current stage we do not vary input data between the members, meaning that all of them receive identically scaled and preprocessed input signal (and constant bias signal). Splitting time series into training/validation/testing samples will be described later, when we specify the benchmarking time series.

In this Chapter, our primary goal is analysis and comparative study of committee approach in reservoir computing, rather than fine-tuning the system to achieve the record performance on particular tasks. Therefore the optimization of ESNs is constrained to only two essential parameters - reservoir size and regularization parameter, while holding other parameters on fixed, roughly adjusted levels, and we primarily concentrate on the benefits coming from the committee approach. We believe that different ESN configurations would unlikely change the general conclusions coming from our study.

In the following subsections, we first define the time series that will be used as benchmarks. Next, we present selected results of our comparative study,

which give some insight into the committee algorithms, and hence will lead us to several important conclusions and design principles.

4.3.1 Benchmark time series

Similarly like in Chapter 3, we will use several distinct Mackey-Glass time series - MG20, MG30, MG50, MG70 with time lags $\tau = \{20, 30, 50, 70\}$ respectively, as defined by equation 3.8 and a sine-based signal, defined earlier by equation 3.10.

Again we assume limited availability of data, and generate 800 data samples available for the training and another 400 samples to test the system. We split the data in proportions 50/750/400, meaning that initial 50 samples were used to flash out random initial states of ESNs, 750 samples were used for members training, and 400 samples served to evaluate the performance. In case of the ranked committees, the last 150 samples of the training data set are used for 8-fold cross-validation scheme (see Fig.4.1 and Fig.4.4).

In all cases we normalize the time series to the range $[-3, 3]$, which we found to be a reasonably good estimate, since it fits well to the nonlinear region of sigmoid activation function of the neurons. The normalization could be further optimized, but from our perspective the most important is that all the committees that we compare receive identically preprocessed inputs.

In the following experiments we focus primarily on the critical prediction horizons, for reasons explained in section 3.6.4. We can easily identify them, since we train multiple readouts, each for one subsequent prediction horizon.

4.3.2 Exemplary committee setting

Before we get to committee optimization and larger scale experiments, we shall first consider simple committee and observe its output as compared to individual members. Let us consider again the trajectory prediction task, that we discussed in section 3.6.3. We employed 50 independent ESN networks to project trajectories of the first 30 samples of the training data set. Having those 50 networks trained, we can construct a simple committee by combining them together. The committee output will be computed as arithmetic average of individual ESN outputs, for each of the prediction horizons $h = \{1, ..., 30\}$ independently (Eq.4.1).

Fig.4.6 shows the trajectory forecasted by committee in comparison to uncertainty of individual forecasts (shadowed area). We shall discuss the benefits of committee versus individual networks in the following subsections. For now we constrain to observation, that committee displays more conservative behavior and relatively good trajectory estimation, despite large variance of the base models. We shall see later that committee in fact efficiently reduces the variance component of the error.

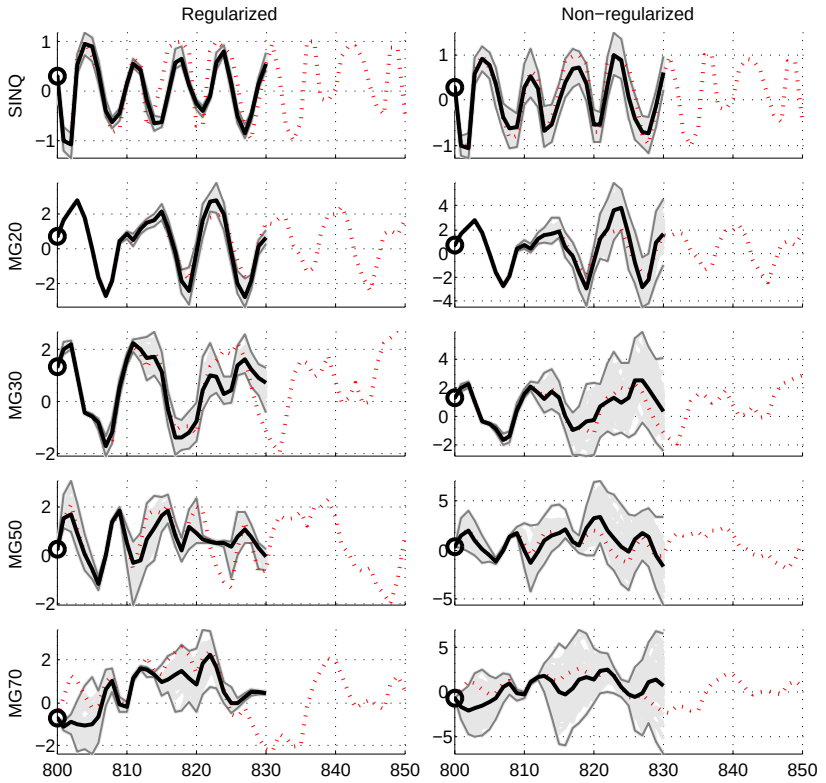

Figure 4.6: Trajectory forecasting by simple averaging committee of 50 members. The shadowed area corresponds to uncertainty of individual predictions.

4.3.3 Averaging committee with no members regularization

In the first experiments, we evaluate performance of simple averaging committees. Note that members are trained with least-squares regression, that does not

involve any form of regularization. Such choice will be justified in the further sections.

We have already observed the influence of reservoir size, and regularization, on the performance of single ESN network in section 3.6.4. Let us now consider ensembles of the non-regularized members. For each value of N let us construct corresponding committee comprising 50 members (i.e. 50×100, 50×150, ... , 50×800)[3], and observe their performances in terms of mean squared error on the testing datasets (Fig.4.7). Solid lines relate to the committees, while dashed lines to the corresponding best-found members. Again we restrict observations to the optimal prediction horizons, which have the highest significance for accurate prediction.

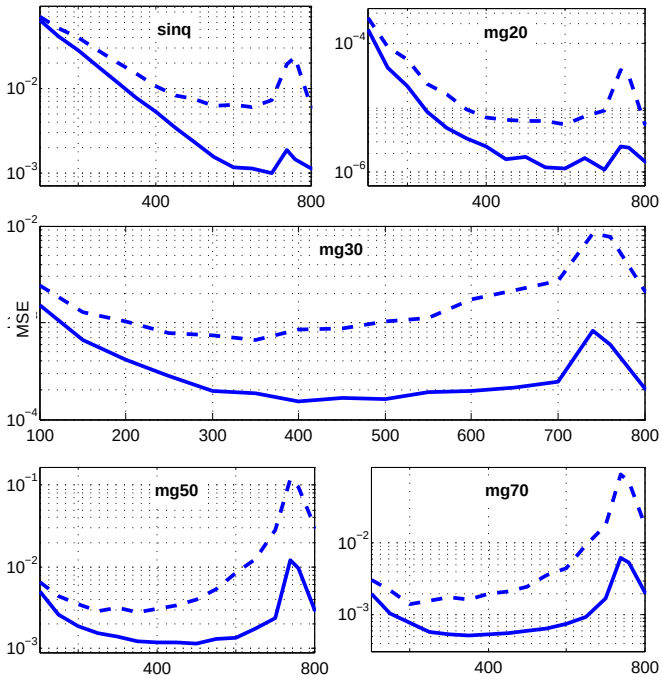

Figure 4.7: Performance of averaging committee (solid line) as a function of reservoir size, assuming no members regularization. Dashed line corresponds to the best individual network.

The first observation is a large performance gain of the simple committees over

[3]In fact, for each setting three committees will be simulated and their results averaged. More repetitions could be considered, but due to high computational requirements of the experiments, and due to highly repeatable committee results, we constrain to three iterations.

Table 4.1:

Time series	MSE_{min} (best network)	MSE_{min} (committee)	Committee gain:
SINQ	5,92E-03	1,00E-03	83,0%
MG20	5,61E-06	1,12E-06	80,1%
MG30	6,69E-04	1,53E-04	77,1%
MG50	2,77E-03	1,16E-03	58,2%
MG70	1,41E-03	5,13E-04	63,6%

the individual networks, for any value of reservoir size (see Table 4.1). The significance of this observation is even higher if we consider that individual networks that we use as benchmarks are the truly best ones, i.e. those networks out of 50 committee members that achieved the lowest MSE on the testing datasets. However, since the testing data set is not known *a priori*, the identification of those optimal members can be difficult, even with multi-fold cross-validation scheme. Gains of the committees over validation-estimated best networks is usually higher.

Another observed regularity is that the optimal reservoirs found for committee setting are slightly larger than the optimal reservoirs for individual networks. When the individual networks already begin to experience over-fitting, the committee error continues to decrease towards its minimum. This indicates that by averaging the individual outputs committee efficiently smoothens the final prediction and removes negative effects of individual variances. Since this observation is consistent across all analyzed time series, it can be considered a good starting point if searching for the optimal committee reservoir needs to be performed, once the optimal reservoir for individual network is known.

The results allow us to argue for robustness of the ensemble approach. First of all, even the committees without the optimal reservoir size yield better performance than the globally best individual networks. Benefits are clear for wide range of reservoir sizes, even the non-optimal ones from individual perspective ($N \sim 600 \div 700$). Secondly, advantages of the committee are not limited to critical points, but generalize well to all other prediction horizons ($k = 1, ..., 20$), as shown by Fig.4.8. For brevity, the figure presents only the results for exemplary MG50. Previously found optimal reservoir size ($N = 500$) is used. The figure, apart from proving robustness, justifies selection of the critical points in our experiments (here: $k = 8$ for MG50).

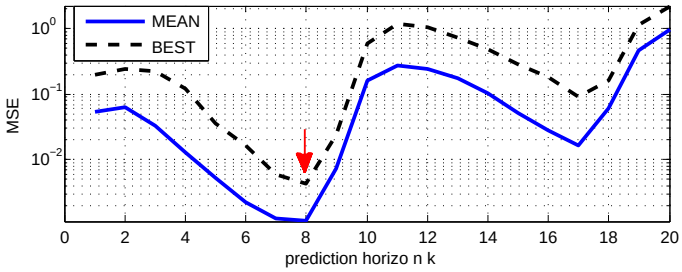

Figure 4.8: Committee vs. best network for range of prediction horizons $k = 1, ..., 20$. The arrow marks critical point $k = 8$.

4.3.4 Large-reservoir committee and self-regularization

In the previous experiments, we observed the peak error due to over-fitting in the point where the number of base model parameters (i.e. reservoir neurons) approached to the number of available training samples. However, adding more neurons tends to self-regularize reservoir. The excessive neurons prevent least-squares regression from direct mapping the training data onto the output weights. We shall observe now how far this self-regularization extends, by looking at performance achieved with large-reservoir committees[4]. Similarly like in the previous experiments, we shall compare the committee MSE with the best-found member MSE, this time including large reservoirs, i.e. $N \in 100 - 2400$. The results are shown by Fig.4.9.

When the reservoir size increases, MSE tends to fall asymptotically. However, in case of more chaotic time series (MG50 and MG70), it never reaches a new minimum. It seems that the non-regularized members suffer from excessive fitting to random noise and oscillations of the training dataset, and even with very large reservoirs are not able to extract more information about signal from limited training samples. On the other hand, small-reservoir committees ($N = 500$ and $N = 350$ respectively) display considerably higher prediction accuracy, at the same time being faster to train, optimize and exploit.

In case of the remaining time series, increasing reservoir size allowed to improve minimum MSE. However, error decrease was moderate and achieved rather marginal gains (see Table 4.2). Therefore, depending on application requirements, a trade-off between limited potential gains and significantly higher computational requirements should be considered. Additional cost of training,

[4]"Large reservoir" is a relative expression. Here we refer to a relative size of the reservoir against the training data set, and we consider given reservoir large if the number of neurons exceeds the number of training samples.

optimization, and exploitation of the large-reservoir committees can be substantial. Moreover, the committee approach requires training entire populations of networks in every iteration of the optimization process. The optimal size of reservoir can be very large, provided that we have many training samples in disposal. Therefore we conclude with a remark that it should be carefully assessed whether it is worth to search for global optimum, or instead consider locally optimal reservoir ($N <$ training data set), which in many cases can offer comparable, or sometimes better, prediction accuracy.

In any case, the general conclusions from 4.3.3 remain unchanged - the large-reservoir committees continue to outperform the best individual networks, even though magnitudes of those gains are slightly reduced for high values of N.

Table 4.2:

Time series	MSE_{min} (small reservoir)	MSE_{min} (large reservoir)	Gain
SINQ	1,00E-03	7,78E-04	22,5%
MG20	1,12E-06	7,14E-07	36,0%
MG30	1,53E-04	9,59E-05	37,2%
MG50	1,16E-03	1,26E-03	-8,9%
MG70	5,13E-04	7,49E-04	-46,0%

4.3.5 Averaging committee with members regularization

Let us now consider committees composed of the regularized members (like those analyzed before, Fig.3.14, right-hand side plots), where λ_{mem} is optimized in the validation phase of the training, as described in section 3.6.4. Intuitively, one may assume that an ensemble of efficient members will yield better prediction accuracy than that of non-regularized, weakly adapted networks. However, as we shall see, this assumption is valid only in certain circumstances.

We performed similar experiments in section 4.3.4, this time with the purpose of comparison of the committees composed of the regularized and non-regularized members, including both small and large reservoirs. Fig.4.10 summarizes the results.

Interestingly, for small reservoir the regularized committee usually does not outperform the non-regularized one. The non-regularized committees is more accurate, even though performance of the individual members is quite poor. The non-regularized committee performance converges quickly to its local minimum (for $N < 750$), which in many cases constitutes as well the global minimum. In the contrary, the regularized committee initially performs weaker, even though the individual networks are optimized and relatively efficient. This observation can be justified by the fact, that members regularization by increasing λ_{mem} shrinks theirs outputs - while searching for the optimal bias-variance balance (see discussion in section 3.6.4), individual variances are reduced. It minimizes individual errors, but at the same time reduces the amount of information provided to the committee, and hence degrades the committee output. If regularization is forced low or completely omitted, all the responsibility for balancing variances of the weak members is delegated to the committee, what can be beneficial. Those observations are consistent with [26, 27].

However, we observed that the regularized committees can be advantageous for larger values of N, since the error decreases asymptotically and is resistant to overfitting when $N \sim 750$. In certain cases, if the reservoir is sufficiently large, the regularized committee can outperform the non-regularized one. This will be usually the case for more complex and chaotic time series (here: MG50 and MG70, see Table 4.3), which on one hand require larger reservoir to capture the underlying dynamics, and on the other hand carry more random noise and chaotic dynamics, what in turn brings up the risk of over-fitting especially if training data is limited. In such cases, non-regularized committee will not be the optimal one.

We should be aware though of significantly higher computational cost associated with regularization of the members. The optimization of λ_{mem} requires that each committee member is trained r times, where r is the number of iterations in searching for the optimal regularization parameter, plus the final regression using the entire training dataset. In our experiments, we searched for λ_{mem} in the range $\left[10^{-8}, 10^{2}\right]$ with logarithmic step of 0.5, resulting with $r = 21$. In case of the large-reservoir committees comprising multiple members, the additional workload associated with such optimization can be substantial. Global gain over the non-regularized, small-reservoir committees is usually not large, therefore the same trade-off considerations apply here as discussed before (4.3.4).

We conclude with statement that prior regularization of base models is usually not necessary if the models are to be combined into a voting ensemble. In some cases such regularization can harm committee prediction accuracy, and in any case it will be significantly more demanding in respect to computational resources. It is often more beneficial and economical to delegate regularization responsibility to committee level. However, in case of certain complex tasks,

Table 4.3:

Time series	MSE_{min} (non-regularized)	MSE_{min} (regularized)	Regularized gain:
SINQ	7,80E-04	1,45E-03	-86,1
MG20	7,44E-07	9,12E-07	-22,6
MG30	1,02E-04	1,05E-04	-2,9
MG50	1,16E-03	9,81E-04	15,2
MG70	5,13E-04	4,56E-04	11,1

we can benefit more from employing large-reservoirs if we optimally regularize the members. Such approach will be recommend if: (1) peak performance, not computational cost, is the priority, and (2) prediction task is noisy and chaotic, and training data is limited (thus giving hope for reasonable size of optimal "large-reservoir"). This can make regularized committees appropriate for financial forecasting, elaborated in Chapter 5. In other cases, using simpler, small-reservoir committees with non-regularized members will yield comparable efficiency and require less computational effort.

4.3.6 Ranked committee

In the final experiment we shall benchmark the ranked committees introduced in Section 4.2 against the simple averaging committees that we were discussing so far. In the preliminary evaluation we found that the ranking algorithms work better with the regularized members. In case of the non-regularized models the individual variances are considerably larger, making it non-trivial for the ranking methods to make correct assessment and appropriately assign W_{com} weights. Fig.4.11 shows the MSE results of the ranked committees versus corresponding averaging committees. For clarity we plot only the lines related to the best and the worst ranking results, and thick lines corresponding to the regularized, averaging committees.

As we see, advantages of the ranking methods are not always obvious. In general, the algorithms are tightly competing with simple averaging (MEAN). They often perform moderately better in the regularized ensemble setting, but rarely in the non-regularized ensemble. They will be therefore primarily applicable in the circumstances appropriate for the regularized ensembles, which were discussed earlier (4.3.5). Since there is a lot of numeric data associated with the experiment (due to five time series, twenty prediction horizons, wide range of reservoir sizes and four different ranking algorithms) we will present preprocessed numeric results in a selective way. Firstly, for each time series we will

compare the globally best ranked committees with the globally best averaging committee, in terms of MSE (see Table 4.4). Secondly, we compute percentual gain/loss of the ranked methods over MEAN method, averaged through all reservoir sizes. This will give better estimation of robustness of the solution. The presented results limit to the regularized committee setting and only the critical points are considered, for the reasons stated earlier.

Table 4.4:

Time series	MSE_{min} (averaging com.)	MSE_{min} (ranked com.)	Ranked com. gain:
SINQ	7,80E-04	8,20E-04	-5,1%
MG20	7,44E-07	6,91E-07	7,2%
MG30	1,02E-04	1,06E-04	-4,1%
MG50	9,81E-04	9,33E-04	4,9%
MG70	4,56E-04	4,46E-04	2,1%

Table 4.5:

Time series	EXP	REXP	RMEAN	RIDGE
SINQ	18,5%	29,8%	31,7%	31,7%
MG20	5,1%	4,7%	6,1%	4,4%
MG30	-1,8%	-2,9%	-0,6%	-0,8%
MG50	7,0%	7,2%	6,9%	6,7%
MG70	2,9%	3,9%	4,4%	4,8%

In three out of five time series, the ranked committees showed the globally best performance. The methods showed robustness against MEAN in the regularized ensemble. This indicates that they will be applicable in tasks where the individual forecasts are relatively smooth (larger bias but limited variance). It can be for instance due to long training datasets in relation to reservoir sizes, or due to prior deliberate regularization of members.

The computational cost of those gains is similar to that of the regularized large-reservoir committee discussed before, plus additional effort related to committee level cross-validation to estimate the optimal weight vector W_{com}. The latter component however is relatively low as compared to the members optimization and regularization. Committee regression is usually uses less computationally demanding than member regression because it uses less dimensional input (here: 50 members, instead of 100-2000 reservoir neurons) and less data samples (here: 150 validation samples instead of 600-750 training samples).

In order to get more insight into the effects of the ranking algorithm on members

selection, we choose an exemplary committee (50 members with reservoir of 1600 neurons, MG50 task), where ranking algorithms showed superiority over common methods, and observe it in more detail (Fig.4.12). The top chart displays sorted mean squared errors of the members, compared to the best ranking method found (RIDGE in this case, left-hand side bar). The middle chart sets together performance achieved by various ranking methods (first four bars), averaging committee result (fifth bar) and best individual networks, found during training and validation phase (the last two bars). Superior performance in this case is owed to selective distribution of the committee weights (W_{com}), computed by EXP, REXP and RIDGE methods respectively (three bottom charts). Different shapes of the weight distributions are observed. Two effects are noticeable: the outlier networks that perform weakest are suppressed by very low weights, (2) well-performing members are generally preferred and rewarded with higher weights. In case of REXP, the weights distribution is a trade-off between the other two distributions. It is comparable to the ridge, however due to convergence towards the exponential, it becomes less smooth and reaches higher positive values. REXP is a promising approach, since it usually efficiently balances between RIDGE and EXP (in this particular example, EXP performs weaker while REXP is close to optimal). Efficiency of EXP method can be higher in case of time series with high noise ratio, since the method is less prone to overfitting problem - it does not use the member outputs directly as the regression methods do, but instead it ranks the members only based on their validation mean squared errors. While EXP is clearly biased towards better members, RIDGE method is more distinct in combining experts and often grants high weights also to the moderately performing individuals.

It should be mentioned here that potential gains from employing the ranking algorithms may be more visible in heterogeneous settings[5]. Since this issue is beyond the scope of this work, we constrain ourselves only to few brief remarks. Generally speaking, increasing parametric variance of base models within committee (e.g. reservoir size, spectral radius or connectivity ratio) can have a positive influence on the final performance. Such variance can enhance feature extraction from the input variables and increase robustness of a collective predictor. By incorporating members with varied dynamic characteristics, a committee can efficiently adapt to handle wider range of different tasks (e.g. range of prediction horizons). In particular, connecting different subset of inputs to different members is a promising method. In case of univariate input each member would receive differently preprocessed (e.g. smoothed) signal or - in case of multivariate input - each of the members would be trained on different subsets of input variables, thus creating mixture-of-expert type of committee.

[5] To be precise, committees of ESN networks are naturally heterogeneous due to randomly constructed sparse reservoir. However, referring to heterogeneous committee we mean an ensemble of members characterized by variance of generic parameters, e.g. reservoir size, spectral radius, input scaling, selection of activation functions, etc.

Although we leave the benefits of heterogeneous reservoir committees as an open problem, it is reasonable to state that efficient ranking methods might be essential in such settings. Those techniques, among others, are subject of our further research.

The final conclusion coming from the observations can be expressed as follows: although a ranked committee approach does not always guarantee best performance, it should be considered as a potential model for the tasks where the top prediction accuracy has the highest priority, rather than design simplicity or minimization of computational efforts. Obtaining state-of-art performance may require some additional optimization efforts, in particular adjustment of the validation datasets provided to the ranking algorithms. The ranked committees will be particularly recommendable in two cases: (1) the member outputs are relatively smooth (either due to forced member regularization, or external constraints on maximum reservoir size, or due to large training data available in relation to reservoir maximum size) and (2) committee is composed of heterogeneous members of varying performance, and a method is required to reward the efficient members and suppress the weaker ones.

4.4 Summary

The purpose of this chapter was to evaluate alternative types of committees and discuss several issues related to their design. Independently trained echo state networks, that we introduced in 3, were used as a base model. The committees that we investigated used either simple averaging, or regression/exponential ranking methods and cross-validation scheme to optimally combine individual outputs into joint committee output. We presented comparative studies of several types of committees, analyzed their performance on different prediction tasks, and highlighted several challenges and trade-offs that need to be considered in the design process. Although reservoir-based committees were in the center of our interest, many of the results and conclusions can be generalized to the committee approach in broader perspective, regardless the structure of underlying base model.

Our research showed that in majority of cases simple averaging committees of non-regularized member will be the best choice, considering their close-to-optimal performance, simplicity of design, and low computational cost. Those methods will usually significantly outperform the individual models. The peak performance is often achieved with relatively small reservoirs, but already large enough to display signs of over-fitting, if considered individually. However, expanding the reservoirs to sizes larger than the available dataset may be even

more beneficial. It is essential though to avoid sizes close to the size of the training dataset, since it will result in square-like state collecting matrix and violent amplification of noise in regression (if regularization was not performed). In certain cases, especially with more chaotic timeseries, committees of regularized members may perform better. However their advantages are usually visible only for large sizes of reservoir, and hence the computational cost related with optimization of the members regularization can be substantial. Finally, the ranking algorithms can be applied to further boost performance - however their applicability is rather limited to cases where individual model outputs are relatively smooth (regularization, long training data sets available in relation to reservoir sizes used, etc.). The ranked committees increase model complexity and require more expertise and considerations in the training phase. Our results indicate that ranked committees result in good reduction of bias component of the error and hence are more suitable to regularized settings. They are less efficient in reduction of variance component of the error, and therefore are less applicable to the non-regularized ensembles, where simple averaging committee will be recommendable.

It should be noted that "no-free-lunch" rule applies here as in case of any other predictor system. Several parameters need to be carefully tuned to obtain state-of-art performance, like selection of committee type, number of members, split of available dataset into training/validation sets. As we showed in the range of experimental settings, different types of committees will be preferable depending on the characteristics of the task. If proposed ranked committees are to be employed, one should pay special attention to careful adjustment of the committee-level cross-validation scheme. The ranked committees are particularly sensitive to the proper choice of validation dataset - both in terms of length, and - in case of nonstationary data - representativeness. Failure at this stage will lead to performance inferior to that of the simple averaging committees. Finally, in order to obtain peak committee performance, optimizations must be primarily done on the member level. Selected aspects of ESN optimization were presented in 3, and for more insight into the problem we refer the reader to wide selection of related publications. Many of them we include in the References section.

Having discussed echo state networks and introduced a concept of reservoir-based voting ensembles, we will now advance from experimental environment to non-trivial engineering applications. In the following Chapter we intend to employ the ESN committee model for the purpose of financial time series forecasting.

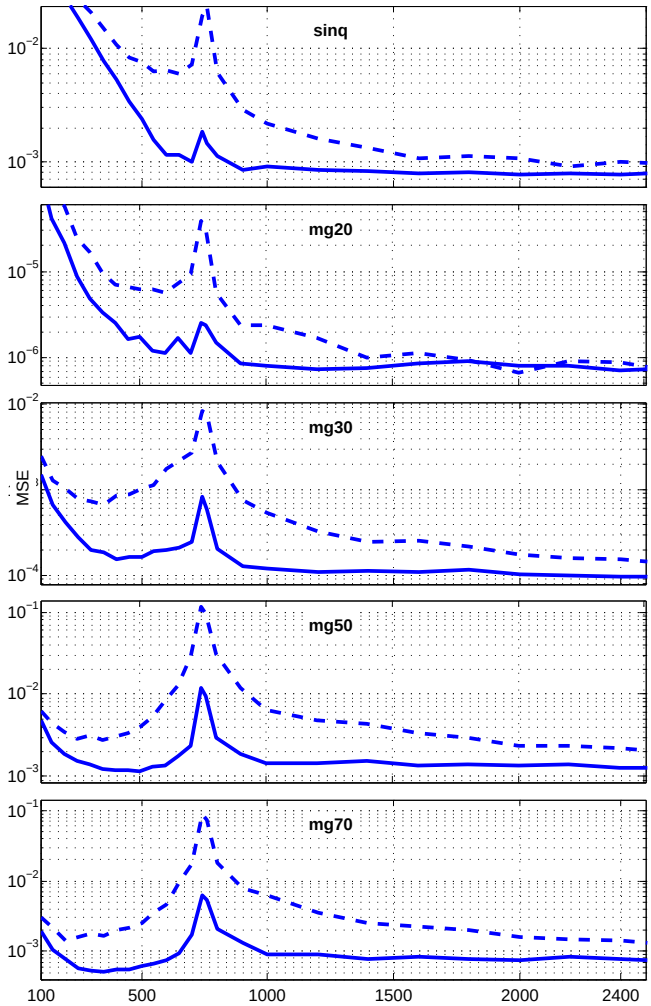

Figure 4.9: Performance of averaging committee (solid line), assuming no members regularization, including large-reservoir ensembles. Dashed line corresponds to the best individual network.

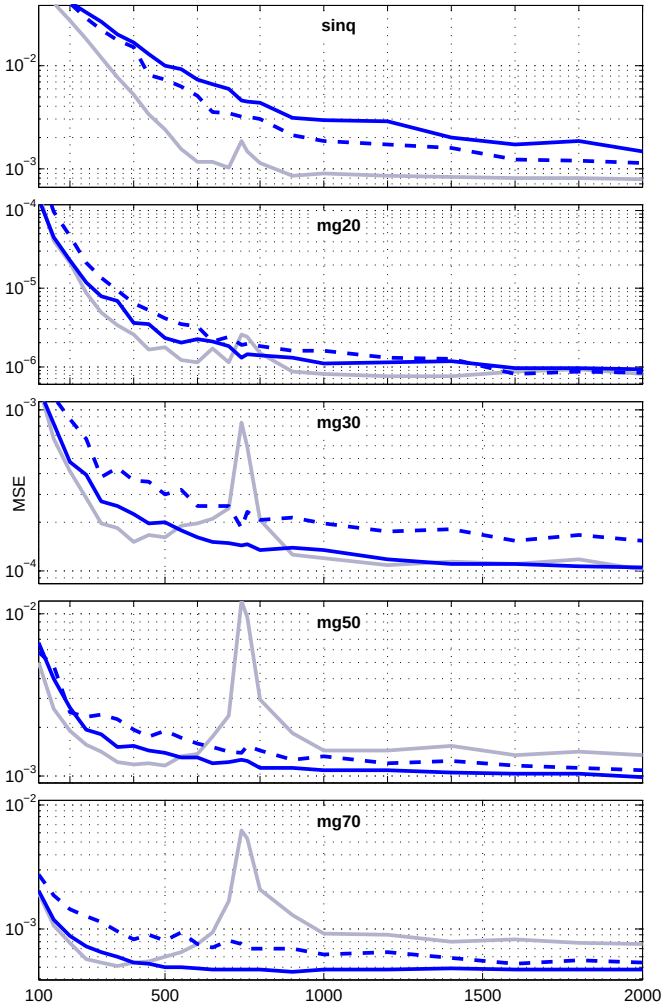

Figure 4.10: Impact of members regularization on committee performance. Committee of not regularized members (faded, gray), committee of regularized members (solid) and best-found, regularized member (dashed).

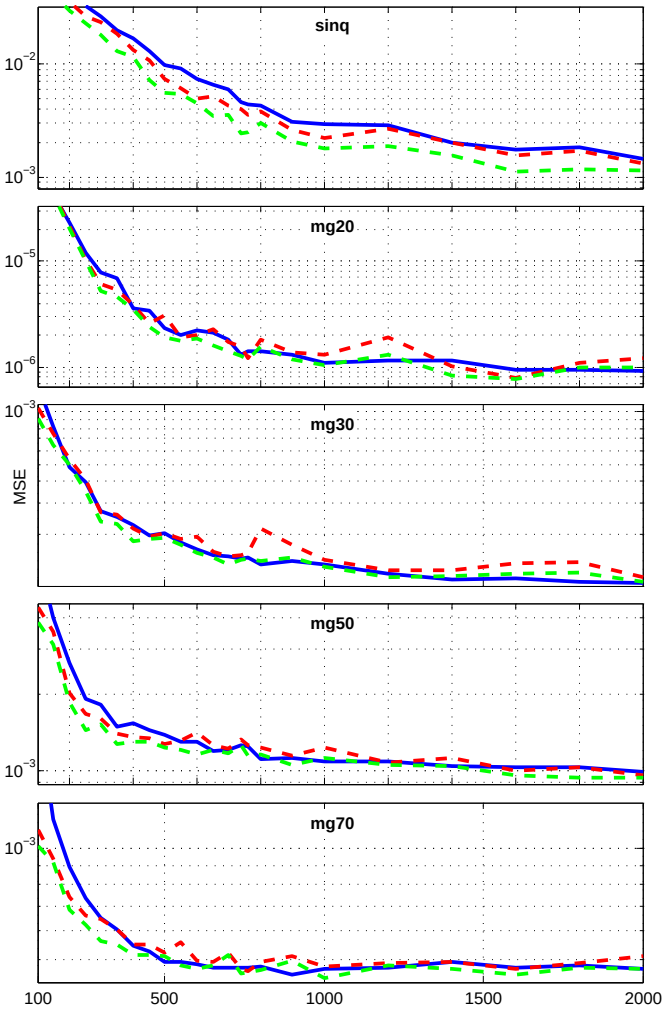

Figure 4.11: Committee with ranking algorithms. Averaging committees (solid line), best/worst ranking algorithms (dashed, green/red).

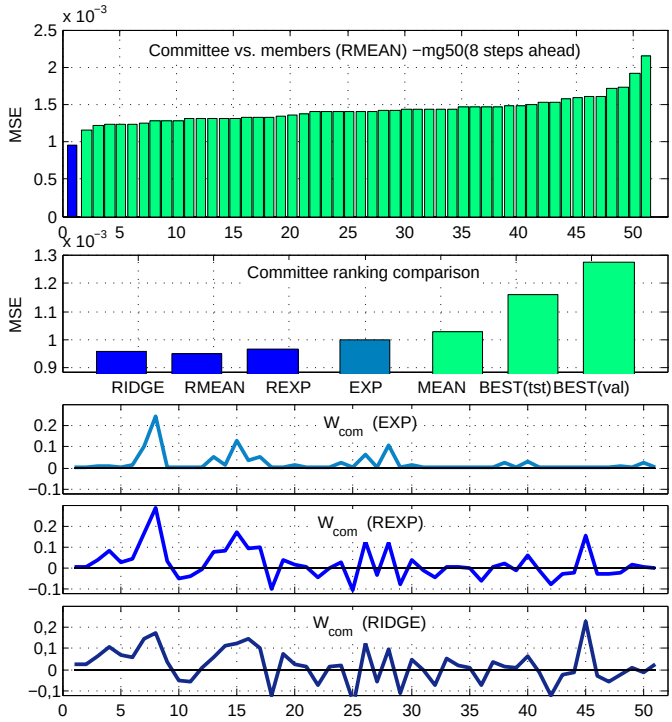

Figure 4.12: Exemplary committee performance details (MG50 time series, 50 ESNs of 1600 neurons). From top: performance comparison of ranking algorithm and individual members, performance comparison of ranking algorithms and MEAN, committee weights estimated by EXP, REXP, RIDGE algorithms respectively.

Chapter 5

Applications in Financial Domain

In this Chapter, the reservoir committees of echo state networks will be employed to the task of financial time series forecasting. The complex dynamics of financial markets and basic concepts related to the domain were already discussed in Chapter 2, where we also argued for selection of the data for further experiments. We will now address those issues more thoroughly. To begin with, section 5.1 will discuss the data related aspects. We found that careful preselection and preprocessing of data is particularly important in financial forecasting. The next two sections (5.2 and 5.3) introduce specific performance measures that will be of use and introduce simple alternative models, which will serve as benchmarks for the committee model. The benchmarking scheme will be also specified for the further experiments. Section 5.4 presents empirical results of our studies. The reservoir committees will be evaluated by means of their accuracy in prediction of the day-ahead market direction, and in terms of their profitability as an investment support tool. Furthermore, we shall define simple investment strategy and simulate theoretical capital flow, assuming that trained reservoir committee is used as automated trading system.

While optimizing the committee models to particular financial financial tasks, the experiences and concepts from Chapters 3 and 4 will be extensively used. We shall optimize model parameters and regularization parameters, apply cross-validation on both member and committee levels, and in general - intend to

minimize validation mean squared error. However from the global optimization perspective we will consider other measures of performance. For instance, maximization of correct classification of the next-day market direction will be the main objective.

5.1 Data selection and preprocessing

Financial forecasting is an example of domain, where data preprocessing and feature selection plays essential role. Even the optimally crafted predictor will not yield satisfactory results if the underlying data does not carry relevant and sufficient information. Therefore significant part of this Chapter is committed to data preparation.

In Chapter 2 we proposed, based on the preliminary domain analysis, a selection of important indicators that will be included as inputs or outputs in our financial simulations. The choice is certainly subjective and many other financial and economic variables could be considered, however due to project scope we needed to make certain constraints about the data selection. It should be emphasized though, that the very first step of any financial forecasting task focuses on carefull domain analysis and preselection of appropriate multivariate inputs, that may have direct or indirect impact on the target time series.

Furthermore, we have listed alternative sources and databases, which provide relevant financial time series in a downloadable form. After acquisition of the required time series, a lot of preprocessing is needed before the actual training can commence. Direct feeding of the raw values to the network input would either lead to unstable behavior of the system, or at least to unsatisfactory performance. Below we discuss the steps involved in data preparation, that will be applied to all financial time series used in further sections.

5.1.1 Format and conversion

For the practical reasons we will concentrate on daily resolution data. Higher resolution could be desirable in certain applications, such as high-frequency real-time trading algorithms, or intraday investment support systems. However the availability of intraday data is more limited and usually associated with provider costs. We constrain therefore to daily resolution of the data.

The historical financial time series are commonly offered in text or spreadsheet

files, where rows contains information related to subsequent days. Each row consists of several values - date, day-open price, day-close price, day-min price, day-max price and optionally - daily volume of transactions. Note that close-price and open-price of the subsequent days often vary. While we shall always consider close-price as the target forecast, the other variables will be beneficial as additional inputs, since they provide meaningful information about the shape of daily price level oscillations. The final value - volume - is reasonable to be included because it measures the significance of given price formation. Large transaction volume means that large capital was involved in generating the price movement, what indicates its high relevance.

The raw data files containing selected time series for period ranging from January 2009 (final phase of financial crisis) until August 2011 (time of finalizing this thesis) were imported to MATLAB environment and converted to appropriate matrix representation. The preliminary filtering was performed to identify and eliminate outliers (distant more than 3 standard deviations) and not-a-numbers (unrecognized characters, spelling errors in source files).

Fig.5.1 illustrates exemplary time series (American large-cap SP500 index). The top chart shows independent plots for open, close, min, max prices and the volume information below. The bottom graphs display two alternative types of charts, which are typical for representation of time series in financial domain - candlestick chart and highlow chart. They combine the open-, close-, min- and max-price information.

5.1.2 Synchronization of multivariate input

Special considerations need to be made if multivariate data is delivered to the system input. The variables in many cases describe prices of different assets or economic indicators, quoted on different international markets. It will commonly occur, that different countries have different calendar specifics and different distribution of the free-of-trade days. If this is the case, the input variables need to be synchronized to the target variable, so that the missing dates and prices are interpolated with appropriate values. In particular the open, close, min, max values of the interpolated sample will be set to the close-price of the preceding day. The volume will be set to zero. Such solution will indicate no activity on the given market during that day.

Similar corrections need to be made if one of the input variables have lower resolution - some of the macro-economic factors are updated weekly, monthly, or even quarterly. In such case the variables need to be interpolated to the target resolution, as explained above.

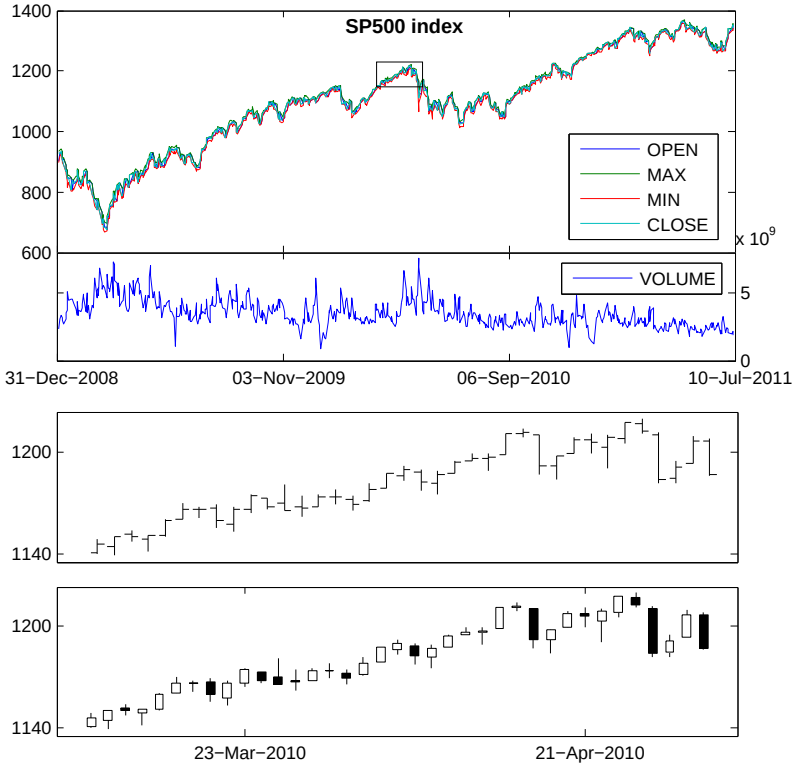

Figure 5.1: Exemplary financial time series - typical representations. The open,close, min, max, volume chart (top), candlestick chart (bottom left), high-low chart (bottom right).

5.1.3 Time-zone differences

The input will often comprise the assets and indicators quoted on different markets around the globe, therefore the time-zone issue emerges naturally. Table 5.1 lists the trading hours of the markets that are of our interest.

Differences in trading hours have certain effect on the preparation of the input variables. When defining the training setting, on one hand we need to provide the most recent input data, but on the other hand we must make sure, that all the input values are known before the opening hours of the target market. The most common situation is that to predict the market value for day $t + 1$, we will provide the vector of previous-day values of input variables - U_t. However

Table 5.1:

Market	City	Time-Zone	Trading hours [CET]
NIKKEI	Tokio	GMT +09:00	02:00am-09:00am
DAX	Frankfurt	GMT +01:00	09:00am-05:30pm
FTSE	London	GMT +00:00	09:00am-05:30pm
SP500	New York	GMT -05:00	03:30pm-10:00pm
EURUSD	-	-	24h/day (mon-fri)
USDJPY	-	-	24h/day (mon-fri)
GOLD	-	-	24h/day (mon-fri)
BRENT OIL	-	-	24h/day (mon-fri)

in certain cases, the input can contain the variables belonging to the same date as the target signal. For instance, if American market is the target variable, we can feed to the input the same day results of the Asian markets (NIKKEI closes at 9:00 CET, SP500 starts at 15:30 CET), but only the previous day results of the European markets (SP500 starts at 15:30 CET, DAX and FTSE close at 17:30 with 2 hours overlap).

In practice, the adjustments of time-zones reduce to appropriate shifting the input variable vectors in relation to the target vector (by default -1, in certain cases 0).

5.1.4 Trend elimination

One of the essential elements of financial data preprocessing is transforming the non-stationary time series into the corresponding stationary representation. A stochastic process $X(t)$ is said to be stationary if distribution of generated data is invariant in time. In particular, the variance and the mean of the time series remain constant:

$$\sigma(t) = E\left[(X(t) - \mu(t))^2\right] = const \tag{5.1}$$

$$\mu(t) = E\left[X(t)\right] = const \tag{5.2}$$

Another implication of stationarity is that the autocovariance and autocorrelation functions are only dependent on the time interval τ between the samples, but not on the time itself:

$$r_{XX} = \frac{\gamma_{XX}(\tau)}{\sigma_t^2} \tag{5.3}$$

where r_{XX} is a autocorrellation function, and γ_{XX} is a covariance function of stationary process.

The autocorrellation coefficients remain constant and depend only on the time lag. In other words, the explanatory relations in the time series and the dynamics of the underlying process do not vary in time. This makes it possible for linear regression based systems to efficiently model such time series.

The simplest method to make the time series stationary is based on differencing, so that the original time series $Y_t = [y_1, ..., y_k]$ are replaced with

$$Y_t' = [(y_2 - y_1), ..., (y_k - y_{k-1})] \tag{5.4}$$

Note that one sample is lost in this operation. Such differencing transformation will be often sufficient to make the mean value constant. However, in certain cases it may fail to stabilize the variance. Another method that can be used to address the issue is logarithmic differencing (Eq.5.5) or relative differencing (Eq.5.6). The latter will be adapted in this work.

$$Y_t^{log} = \left[log\left(\frac{y_2}{y_1}\right), ..., log\left(\frac{y_k}{y_{k-1}}\right) \right] \tag{5.5}$$

$$Y_t^{rel} = \left[\frac{(y_2 - y_1)}{y_1}, ..., \frac{(y_k - y_{k-1})}{y_{k-1}} \right] \tag{5.6}$$

The relative differencing enhances stationarity property of the time series and is convenient in performance evaluation, when the gain factor and capital flow need to be computed. Fig. 5.2 shows the exemplary time series before and after trend elimination. The right-hand side plots show the effects of relative differencing - zero mean distribution and relatively stable variance. The slightly increased variance in the initial period is due to high market volatility in the initial phase of recovery after the financial crisis in 2008.

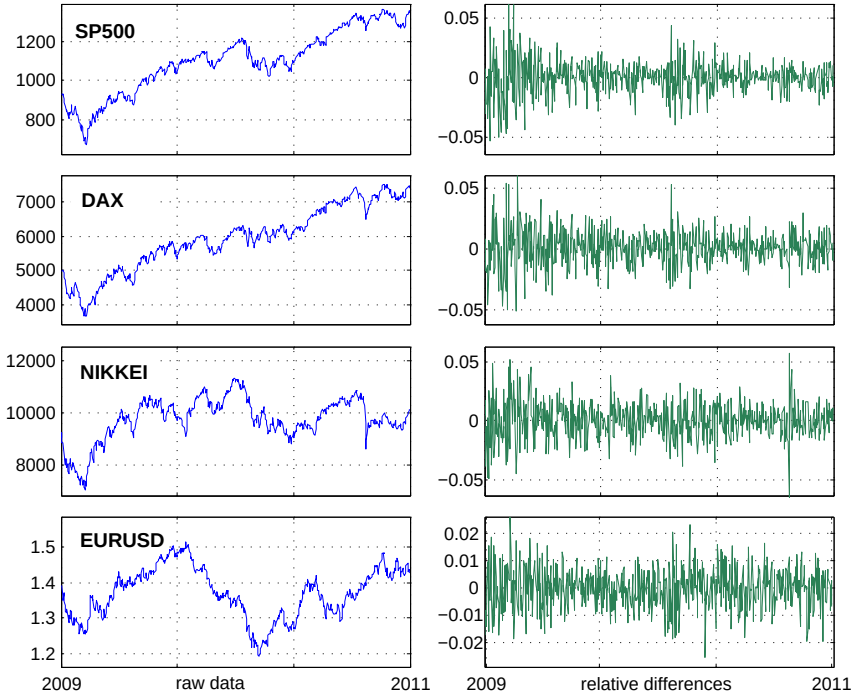

Figure 5.2: Trend elimination by relative differencing $y_i^{rel} = \frac{y_i - y_{i-1}}{y_{i-1}}$. Stable mean and variance of differenced values (right hand-side plots).

5.1.5 Scaling

After elimination of the trend with relative differencing, the resulting time series y" have low amplitude. Market indices rarely change by more than 3% per day, what corresponds to modest change of input value $\triangle u_i = 0.03$. It would result in linear behavior of reservoir neurons and entire network (we discussed this issue in section 3.6.1). In the financial forecasting however, a significant level of nonlinearity will be required. Therefore input variables will be scaled by factor $\beta = 50 - 400$ in all the experiments, so as to exploit the nonlinear region of the sigmoid activation function. Furthermore, optimal scaling depends also on dimensionality of the input. Although scaling factor can be approximated in advance, in many cases we will resort to global optimization.

5.1.6 Multivariate input considerations

In case of high dimensional, multivariate inputs there are several additional problems that might need to be addressed. First, it may be necessary to increase the input scaling to ensure stability of the system. If dimensionality of the input is d, a good heuristic is to scale down the inputs by factor $\beta \sim \sqrt{d}$.

Secondly, it can be beneficial to analyze correlations between time series and employ methods to reduce the effective dimensionality - for instance, ICA methods could be used for this purpose. Finally, it could be reasonable to distribute different subsets of inputs to different members of a voting ensemble, thus creating a mixture-of-expert type of committee. We do not utilize those methods at current stage due to limited number of inputs considered, however we bear in mind that they might be of interest if input dimensionality is extended in further research.

5.1.7 Technical analysis

Apart from applying the original time series (differenced and scaled) it is beneficial to consider linear transformations of the input data, what provides qualitative and quantitative description of signal characteristics. Such methods do not deliver any new information or features to the input, that could not be extracted from the raw data by the network itself. However, the transformations were suggested to enhance forecasting accuracy [7]. In finances there exist a large group of methods known under common name of technical analysis. Those methods usually provide additional statistical information about the time series and some of them are considered as leading indicators, often preceding trend changes. They are commonly used as independent tools, or as a part of larger decision support systems.

Whether such indicators are beneficial or not, and if yes - which of them should be selected, can be quite task specific. Fig. 5.3 presents several exemplary technical indicators that will be of our interest. They include signal smoothing by means of moving average (MA), moving average convergence divergence (MACD) indicator, relative strength index (RSI) and price rate of change (PROC). Some of those indicators will need further preprocessing, such as differencing and - almost always - appropriate scaling. The methods are commonly known in financial analysis, therefore instead of elaborating them further we refer the reader to comprehensive literature on the subject [41].

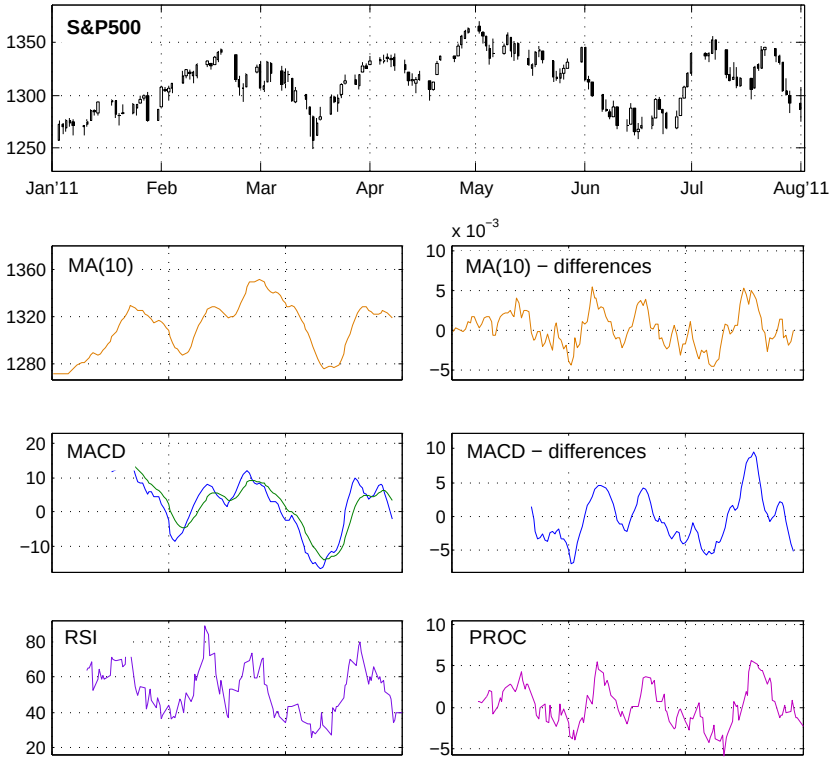

Figure 5.3: Technical analysis indicators. From top: original time series, moving average (MA) with and without differencing, moving average convergence/divergence (MACD) with and without differencing, relative strength index (RSI), price rate of change (PROC).

5.1.8 Selection of training/testing ranges

The final issue is the correct selection of training/validation/testing ranges from datasets. The issue will be discussed in detail in section 5.3, where we specify the benchmarking environment. For now we shall only mention, that selection of training data turns out to be a non-trivial problem in case of non-stationary processes in financial domain. Maximization of training period will not necessarily lead to performance increase, since the dynamic characteristics of the signal change over time. This is due to political and economical changes, long-term cycles, technology advancement.

Currently, the common approach in financial forecasting is to utilize primarily

the data sets since the end of the recent financial crisis (beginning of 2009), which gives approximately 600 trading days. This gives a reasonable amount of samples to work with, especially in multivariate settings where other time series contribute to the input (global market indices, macro-economic variables, currency exchange rates).

5.2 Domain-specific measures of performance

Several additional performance measures will be used in conjunction with financial forecasting tasks. The basic measures will be still used (especially MSE) for the purpose of model training, optimization, committee ranking generation, etc. However to assess system profitability and performance in the financial domain additional specific metrics will be needed. From the investment point of view, it is more important whether the system correctly predicts the direction of future market movements, rather than exact absolute next-day values. It is also needed to evaluate how often the system makes the correct guess, and whether it forecasts properly the significant market movements. To account for those requirements, we introduce additional performance metrics: hit ratio (HR), total return (TR).

5.2.1 Hit ratio

The hit ratio (HR) measures the relative number of correct guesses of the predictor about the next-day market direction. Consider testing data set $\mathcal{D}_t$ of length n and a predictor $\mathcal{P}$ producing n indications about subsequent values of $\mathcal{D}_t$. If we denote by ρ_c the number of correct indications and by ρ_f the number of failures, so that $\rho_c + \rho_f = d$, then the hit ratio is defined by:

$$HR = \frac{\rho_c}{n} \tag{5.7}$$

In particular, the system is 50% accurate if $HR = 0.5$ and perfectly accurate if $HR = 1$.

5.2.2 Total return

The total return (TR) is related to HR measure, however it includes the actual gains due to the correct decisions, and losses due to the wrong ones. It assumes that the investor initiates transaction (opens a position[1]) according to systems indications. TR computes subsequent gains/losses and estimates the relative profit that the investor would achieve, provided that he followed regularly the system indications.

Consider testing data set $\mathcal{D}_t = [d_1, ..., d_n]$ of length n representing target time series, and predictor $\mathcal{P}$ producing n indications about subsequent values of $\mathcal{D}_t$. We denote by $\rho = [\rho_1, ..., \rho_n]$ the vector of predictor indications, where $\rho_i \in \{-1, 1\}$, and $\rho_i = 1$ indicates prediction of market growth on $i'th$ day, and $\rho_i = -1$ indicates prediction of fall on $i'th$ day. The total return is computed by:

$$TR = \prod_{i=1}^{n} \left(1 + \rho_i \frac{d_i - d_{i-1}}{d_{i-1}}\right) - 1 \tag{5.8}$$

The total return is significant from the investment support perspective, because it illustrates the theoretical predictor's profitability. In case of hit ratio measure, value $HR > 0.5$ does not necessarily indicate that the system is profitable on given testing period $\mathcal{D}_t$, because even few wrong decisions may cause large losses. However $GF > 1$ shows that the system indeed generated profit. For instance, $TR = 0.2$ indicates 20% profit measured achieved in the testing period. For simplicity we neglect the broker costs related with buy/sell transactions. However, the TR measure is still simplified, in a way that it assumes the investment positions being opened with previous-day close-price, while in reality the predictor's decision will just be generated before next-day open-price. Therefore the method illustrates theoretical gain which can be different from the real one, in either direction. To make the evaluation more realistic, we derive two other measures - TR with 'buy-and-hold' strategy and TR with 'day-trading' strategy.

The day-trading variation of the method, denoted as TR_{dt}, assumes that investor opens position according to predictor indications in the beginning of the trading day and always closes the position at the end of the day (close-price). In

[1] On future markets (contracts on future prices) one can open 'short position' or 'long position'. The former allows to take profits in case of market growth, the latter in case of market fall. In a way, buy/sell transactions are symmetric. In case of short position the sell operation precedes the buy operation.

this setting, the gains and losses are computed as a difference between the close-price and open-price for given day, rather than close-prices of two subsequent days:

$$TR_{dt} = \prod_{i=1}^{n} \left(1 + \rho_i \frac{d_i^{close} - d_i^{open}}{d_i^{open}}\right) - 1 \tag{5.9}$$

The buy-and-hold variation of the method, denoted as TR_{bah}, assumes that investor changes a position to the opposite one only when the system indicates such change. When the indication is not changed during certain period, the investor simply holds current positions (in contrary to day-trading approach where position is always closed at the end of the day). TR_{bah} is computed with expression:

$$TR_{bah} = \prod_{i=1}^{n} \left(1 - \psi_i \rho_i \frac{d_i^{open} - d_{i-1}^{close}}{d_{i-1}^{close}}\right) \left(1 + \rho_i \frac{d_i^{close} - d_i^{open}}{d_i^{open}}\right) - 1 \tag{5.10}$$

where $p_i \in \{-1, 1\}$ is predictor indication at time i, $\psi_i = p_i \cdot p_{i-1} \in \{-1, 1\}$ is a position-switch indicator at time i, and d_i^{open} denotes day open-price and d_i^{close} day close-price.

In fact, the total return measure is often expressed as the annual return, which makes the measure independent from actual length of the testing period. We therefore normalize the total return measures as follows:

$$TR = (1 + TR)^{\frac{252}{n}} \quad TR_{dt} = (1 + TR_{dt})^{\frac{252}{n}} \quad TR_{bah} = (1 + TR_{bah})^{\frac{252}{n}} \tag{5.11}$$

where n is the length of testing data set $\mathcal{D}_t$ and corresponds to number of trading days within the testing period. The numerator value of 252 corresponds to commonly accepted average number of trading days per year.

5.2.3 Capital flow simulation

Capital flow simulation (CF) is an analysis of capital change over time, based on TR measure. It essentially computes TR factor for each subsequent day of the testing dataset D_t, and thus illustrates how the profit/loss would accumulate

during the testing period of the simulation. The results are visualized graphically, including additional information such as predictor indications, actual price levels of the target asset, daily gain/loss, etc.

Furthermore, CF calculations can be used to determine additional performance metrics, such as period-maximum gain/loss, period minimum/maximum capital level, maximum number of subsequent gains/losses, etc. Those measures are often considered in financial domain, since they support a practical risk assessment.

5.3 Benchmarking environment

To evaluate the reservoir committee model in financial forecasting tasks, we will measure its performance in wide range of forecasting settings, and compare it to statistical ARIMA predictor and to simple naive strategies. Hit ratio (HR) and total return (TR) measures will be used, which reflect well the systems applicability as an investment support system. All the compared models will work as classifiers, generating decision about next day market direction based on the recent history, i.e. decision taken on day t about day $t+1$ will be denoted by $\hat{x}_{t+1|t} \in \{-1, 1\}$, where -1 corresponds to prediction of market fall and $+1$ to prediction of market growth on $(t+1)$'th day.

Several time series will be considered as a target forecast - SP500 index, DAX index and EURUSD rate (see sections 2.2 and 5.1), while selection of inputs will vary across the experiments and will be detailed later. Moreover, for each of the time series the training and evaluation cycles will be repeated multiple times with moving window approach, and the overall averaged results will be considered. This should give a statistically correct evaluation and eliminate the randomness due to data sets.

5.3.1 Reservoir committee as classifier

The reservoir committees will be trained and optimized in a usual way, as described in Chapters 3 and 4. Some of the training parameters will be kept invariant (number of members, size of training and cross-validation data sets), while the others will be a subject of optimization to find the most efficient solution (size of reservoir of base models, spectral radius, regularization parameter, input scaling, input data selection). Similarly like before, MSE error will be

utilized in the training and cross-validation phase, so that the models will be trained to approximate the value of the target signal.

However the task now is to determine the direction of the market change, rather than the exact amplitude of that change. Therefore we need to modify the network to solve classification task. Committee training, which corresponds to inference phase of a classification model, remains unchanged and is based on MSE minimization. We add however the second phase - taking optimal decision based on the prediction. Having transformed the predictor into classifier, we will use HR and TG performance measures for global optimization of the system and finding most profitable committee.

As we justified in section 5.1, in financial domain both the inputs and the outputs will be converted to relative differences. Consequently, the output y_t of the committee will be a prediction of the market change in the following day. The positive values of y_t indicate prediction of growth, while negative values predict fall. Therefore the classification decision is obtained naturally by considering the sign of the output y_t, so that $y_t = \hat{x}_{t+1|t} \in \{-1, 1\}$. Combining it with equations 3.4 and 4.2, the committee decision about next day market direction is given by:

$$\hat{x}_{t+1|t} = sign\Big(Y_{com}(t)\Big) = sign\Big(W_{com} Y_{mem}(t)\Big) \tag{5.12}$$

where $Y_{com}(t)$ is the committee output at time t, W_{com} is the committee weights vector (see section 4.2) and $Y_{mem} = \big[y_{1,t}, ..., y_{i,t}, ..., y_{M,t}\big]'$ is a vector of individual network outputs $y_{i,t}$ at time t, computed according to:

$$y_{i,t} = f_{i,out}\left(W_{i,out} \cdot \begin{bmatrix} s_{i,t} \\ u_{i,t} \end{bmatrix}\right) \tag{5.13}$$

where $W_{i,out}$ is the trained output weights vector of the $i'th$ member (see section 3.2).

As a result we convert a reservoir committee predictor into a classifier, that maps the input history $u_1, ...u_t$ (represented by reservoir internal echo states s_t) into one of two classes: next-day market growth ($\hat{x}_{t+1|t} = 1$) or next-day fall ($\hat{x}_{t+1|t} = -1$). From the perspective of investment decision support, those classes correspond to BUY and SELL signals of the system. Of course the number of classes could be further extended to allow for more sophisticated investment strategies and risk adjustment. In financial domain it is common to use five classes STRONG BUY, BUY, HOLD, SELL, STRONG SELL. Since

our model is a combination of multiple base models, we have many possibilities of how to determine the classification decision. In the simplest case class assignment could be based on the amplitude and sign of the committee output Y_{com}. However we could also take advantage of the voting ensemble, and, for instance, classify signal to STRONG BUY only if both the committee generates BUY signal and the signal is backed by e.g. 80% of individual expert members.

In the following experiments and investment simulations presented in the further sections we shall constrain ourselves to use two-class classification, as specified by Eq.5.12. The benchmarking models, that will be used for comparison, will apply the same classification scheme to generate decisions.

5.3.2 Benchmarking models

For the purpose of benchmarking the reservoir committee system we will compare it to several commonly known methods. In particular, we will consider auto-regressive model (ARIMA) and two simple naive strategies.

Naive and naive-contrarian method The methods makes a simple assumption about the next day market direction based on the current day. The naive method assumes that the trend will be maintained, and the naive-contrarian method is the opposite. If $\hat{x}_{t+1|t}$ denotes decision about day $t+1$ made in day t, and $x(t)$ is market direction on day t, and the function takes positive value in case of market growth, and negative in case of market fall, then:

$$naive: \qquad \widehat{x}_{t+1|t} = \begin{cases} 1 & if\ x_t > 0 \\ -1 & if\ x_t < 0 \end{cases}, \qquad x_t, \in \{-1, 1\} \tag{5.14}$$

$$naive\, contrarian: \qquad \hat{x}_{t+1|t} = \begin{cases} -1 & if\ x_t > 0 \\ 1 & if\ x_t < 0 \end{cases}, \qquad x_t, \in \{-1, 1\} \tag{5.15}$$

Autoregressive models (AR, ARMA, ARIMA, VARIMA) The autoregressive integrated moving average model, described as $ARIMA(p, d, q)$, is a commonly used statistical tool to time series analysis, and constitute a fundamental part of Box-Jenkins modelling approach[42]. Values p,d and q denote

orders of autoregressive component, detrending and moving average component, respectively. $ARIMA$ is a generalized form of autoregressive moving average model $ARMA(p,q)$, which accounts for non-stationarity of time series. $ARMA$ model in turn is a combination of simple autoregressive $AR(p)$ and moving average $MA(q)$, and is defined by equation:

$$x_{t+1} = \sum_{i=1}^{p} \varphi_i x_{t-i+1} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i+1} + c + \epsilon_{t+1} \tag{5.16}$$

where c is constant, ϵ_t is white noise, x_{t-i} is a value of signal at time $(t-i)$, φ_i is a parameter of AR component, and θ_i a parameter of MA component.

The special case of $ARIMA(p,1,0)$ and $ARIMA(p,2,0)$ will be used as a benchmark in further sections, where p being the order of autoregressive model will be optimized to given task by means of cross-validation, similar to committee cross-validation scheme. Values of $d = 1$ relate to differencing raw data samples, and $d = 2$ to second order differencing, for instance relative or logarithmic differencing (as explained in section 5.1). In short, $ARIMA(p,2,0)$ will intend to predict next day relative market growth x_{t+1} based on p recent day values $x_t, x_{t-1}, ..., x_{t-p+1}$. Since we are interested in decision about direction, rather than exact amplitude of signal change, we compute $ARIMA(p,2,0)$ decision according to the following formula:

$$\hat{x}_{t+1|t} = sign\left(\sum_{i=1}^{p} \varphi_i x_{t-i+1} + c + \epsilon_{t+1}\right) \tag{5.17}$$

Finally, when dealing with multivariate inputs, the multivariate generalization of ARIMA will be employed, called vector ARIMA or simply VARIMA, which is obtained from equation 5.17 by replacing the variables x_i and parameters φ_i with vectors X_i and $\mathcal{G}_i$ of lengths equal to dimensionality of input. Since VARIMA can benefit from the same amount of information as the committee predictor the benchmark results will be more complete.

5.3.3 Performance and robustness evaluation

To obtain statistically correct evaluation of the model performance on financial time series, it is necessary to benchmark it on several prediction tasks. In particular, S&P500 and DAX indices and EUR/USD exchange rate will be

considered as target signals. Furthermore, for each task the evaluation should be performed on multiple independent (possibly overlapping) training/testing data sets, using moving window approach. The average performance over all the periods will be considered as the final measure of performance. In this way, random factors due to data sets will be eliminated, that could otherwise blur the results of evaluation.

All the financial time series used for benchmarking will be splitted into 27 overlapping periods, each of 2-years duration, overlapping by one week. Each of the periods will be further divided into training/validation/testing ranges of the length 300, 100, 120 days respectively. Those numbers are approximate, since exact number of trading days depends on of weekends and holidays distribution within the periods. Such scheme will be kept invariant through all the experiments, regardless target time series. The models will be trained and evaluated independently on each of the periods, and the statistical performance will be assessed as the average performance through all the trials.

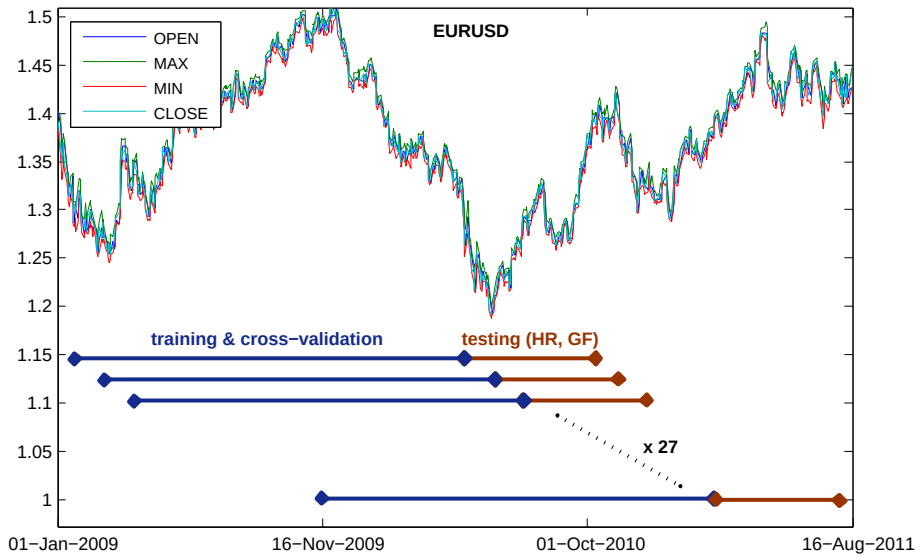

Figure 5.4: Benchmarking scheme. Moving window approach with one week lag.

5.4 Experiments

In this finalizing section the comparative results of the empirical studies will be presented, which give evaluation of reservoir committees in financial time series forecasting. We shall benchmark the model against well-known autoregressive models and naive strategies in the specified environment (section 5.3). The data will be preprocessed (section 5.1) before being supplied to the models. Experiences from sections 3.6 and 4.3 will be used to optimize the base ESN models and group them into committees. Finally, performance measures appropriate to financial forecasting tasks will be used to evaluate the system (section 5.2).

First, the general comparison will be presented for all the target time series. Secondly, we shall observe closer a particular committee predictor in exemplary case study, where details of prediction results will be discussed, and automated trading simulation will be presented. Finally, optimization aspects will be discussed, followed by general conclusions.

5.4.1 Comparative studies and evaluation

The comparative experiments will be carried out for S&P500, DAX, EURUSD time series, each splitted into 27 overlapping periods as discussed in section 5.3.3. The length of data sets and proportions between training/validation/testing data are kept invariant. The selection of input time series depends on the target time series and will be detailed later.

The compact committees of 25 ESN members will be considered. Connectivity ratio is constant $c = 0.05$. Other parameters (reservoir size, spectral radius, input scaling, regularization parameter) will be optimized with grid search, in the following ranges:

- Reservoir size $N = \{50, 100, 150, 200\}$
- Spectral radius $p = \{0.5, 0.7\}$
- Input scaling $d = \{100, 200, 300, 400\}$

For each combination of parameters five independent committees will be generated, and evaluated on the 27 data sets. The results (HR_{avg} and TR_{avg}) will be computed as an average over the five committee instances and over the 27 periods. In this way we eliminate the effect of random luck due to data set or reservoir structure. As a result we obtain a representative measure of 'goodness

of fit' of given parametric combination (N, p, d). The process is repeated for all the combinations (32 in total) and the final results allow us to pick the optimal configuration. The results will be visually and quantitatively evaluated.

It is important to emphasize, that apart from global optimization, every individual network is regularized with the purpose of minimization of cross-validation mean squared error (section 3.6.2). In similar way, each ranked committee is regularized as discussed in sections 4.2 and 4.3.6. The regularization parameters are varied in ranges $\lambda_{mem} = 10^0 - 10^4$, $\lambda_{com} = 10^{-5} - 10^4$. Once the optimal committee is trained, it is transformed into classifier and evaluated globally as was explained earlier.

Apart from preparation of ESN committees, we need to evaluate performance of the benchmarking models. The case of naive and naive contrarian strategies is trivial, since those models carry no parameters and depend only on the recent input. Concerning ARIMA(p,2,0) and VARIMA(p,2,0) models, we optimize the regression order p in the range $p = \{1, ..., 30\}$, evaluating the performance (HR_{avg} and TR_{avg}) as an average over the same 27 periods as in case of the committees. The best found model is taken as a benchmark. In case of ARIMA model, the input is one-dimensional and consists of previous day closing price of the same time series that is being predicted. In case of VARIMA model, the input is multivariate and identical to the input of the corresponding committee models.

In the following subsections we summarize the results obtained for S&P500 and DAX indices and the EURUSD exchange rate.

5.4.1.1 Standard & Poor's 500 Index (S&P500) - next-day direction forecasting

- Data: S&P500 index (01-Jan-2009 till 5-Aug-2011, splitted into 27 overlapping periods)
- Models: naive, naive contrarian, ARIMA, VARIMA, Reservoir Committees: MEAN, EXP, REXP, RMEAN, RIDGE
- Inputs: S&P500 open-price, max-price, min-price, close-price, S&P500 transaction volume, NIKKEI close-price EURUSD close-price
- Output: S&P500 next-day close-price (converted to direction indication)

Fig.5.5 presents the global results of the comparative study of the committee models and benchmarking models, for S&P500 time series. The hit ratio, which

we consider the most representative performance measure, is shown on the top chart, while the lower charts correspond to the annual returns. The bars in each column represent average results for given type of the model, the four left-most bars reffer to benchmarking models, while the five right-most to the various types of the committees (see section). Each bar is a result of computing 160 committees of 32 parametric combinations, and averaging the result over 26 testing data sets. The shaded tops of the committee bars illustrate difference between best- and weakest found configuration (N, p, d). The corresponding numerical results are presented in Table 5.2.

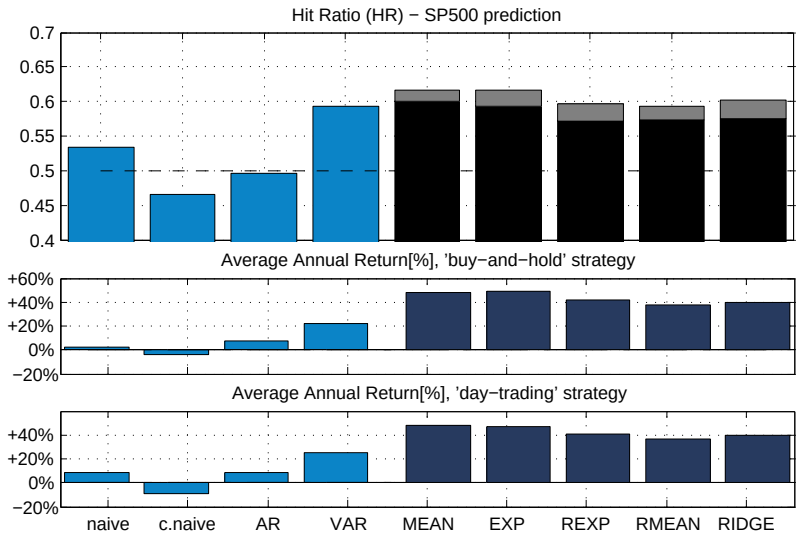

Figure 5.5: S&P500 next-day direction prediction - comparative results for different algorithms. From top: global hit ratio, corresponding capital return with buy-and-hold strategy, corresponding capital return with day-trading strategy.

5.4.1.2 Deutscher Aktien Index (DAX) - next-day direction forecasting

- Data: DAX index (01-Jan-2009 till 6-Aug-2011, splitted into 27 overlapping periods)
- Models: naive, naive contrarian, ARIMA, VARIMA, Reservoir Committees: MEAN, EXP, REXP, RMEAN, RIDGE
- Output: DAX next-day close-price (converted to direction indication)

Method	Hit Ratio	Annual Return[%] buy-and-hold	Annual Return[%] day-trading
naive	0,534	2,57	8,90
naive cont.	0,466	-3,42	-8,99
ARIMA	0,496	7,46	8,63
VARIMA	0,593	22,09	25,39
MEAN	**0,616**	48,52	**48,15**
EXP	**0,616**	**49,50**	47,01
REXP	0,597	42,03	40,61
RMEAN	0,592	37,18	36,77
RIDGE	0,603	40,15	39,96

Table 5.2: S&P500 prediction - comparative results.

- Inputs: DAX open-price, max-price, min-price, close-price, DAX transaction volume, S&P500 close-price, NIKKEI close-price, EURUSD close-price
- Models: naive, naive contrarian, ARIMA, VARIMA, reservoir committees: MEAN, EXP, REXP, RMEAN, RIDGE

Fig.5.6 presents the global results of the comparative study of the committee models and benchmarking models, for DAX time series. The corresponding numerical results are presented in Table 5.3.

Method	Hit Ratio	Annual Return[%] buy-and-hold	Annual Return[%] day-trading
naive	0,490	-4,29	2,84
naive cont.	0,510	2,43	-3,57
ARIMA	0,529	6,58	3,59
VARIMA	0,540	5,35	8,75
MEAN	**0,642**	36,06	8,74
EXP	0,636	**37,17**	**10,40**
REXP	0,623	31,17	9,98
RMEAN	0,621	27,11	6,75
RIDGE	0,599	24,21	5,89

Table 5.3: DAX prediction - comparative results.

5.4.1.3 Euro/US Dollar exchange rate (EURUSD) - next-day direction forecasting

- Data: EURUSD exchange rate (01-Jan-2009 till 5-Aug-2011, splitted into 27 overlapping periods)

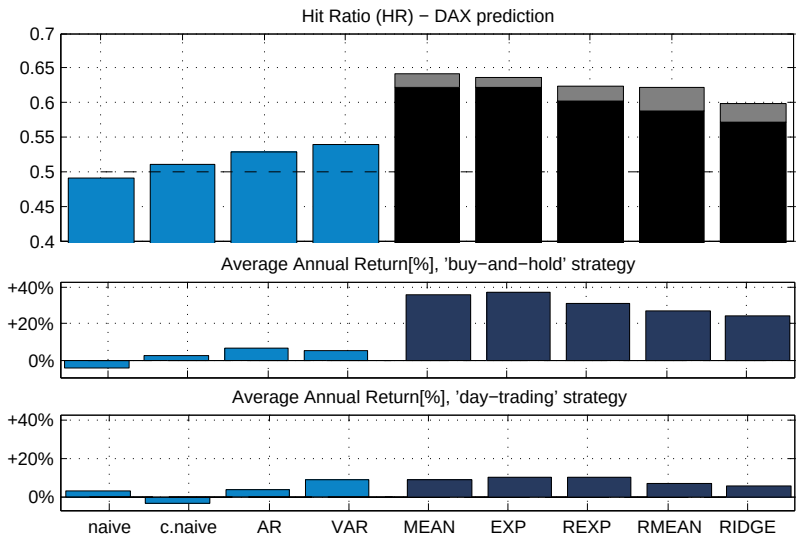

Figure 5.6: DAX next-day direction prediction - comparative results for different algorithms. From top: global hit ratio, corresponding capital return with buy-and-hold strategy, corresponding capital return with day-trading strategy.

- Models: naive, naive contrarian, ARIMA, VARIMA, Reservoir Committees: MEAN, EXP, REXP, RMEAN, RIDGE
- Output: EURUSD next-day close-price (converted to direction indication)
- Inputs: EURUSD open-price, max-price, min-price, close-price, S&P500 close-price, DAX close-price, NIKKEI close-price
- Models: naive, naive contrarian, ARIMA, VARIMA, reservoir committees: MEAN, EXP, REXP, RMEAN, RIDGE

Fig.5.7 presents the global results of the comparative study of the committee models and benchmarking models, for EURUSD time series. The corresponding numerical results are presented in Table 5.4.

5.4.1.4 Results summary

In the first two experiments we observed the superior performance of the committee-based models over the naive and autoregressive methods. The performance in

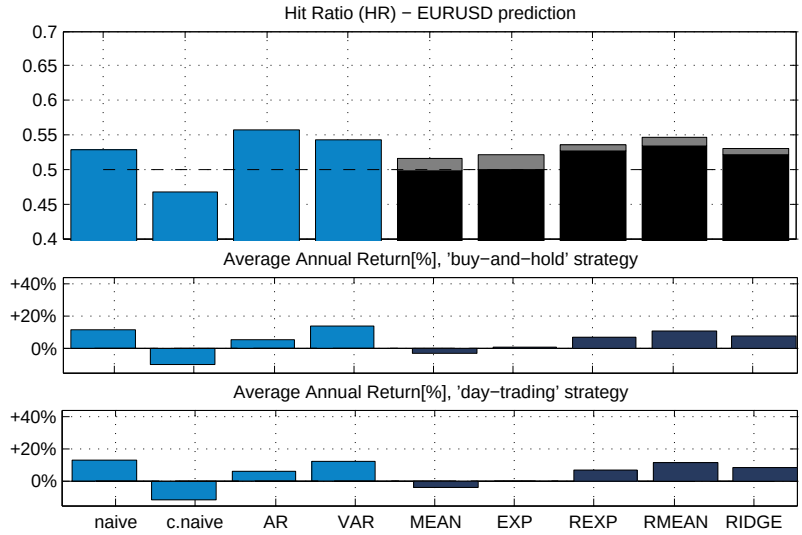

Figure 5.7: EURUSD next-day direction prediction - comparative results for different algorithms. From top: global hit ratio, corresponding capital return with buy-and-hold strategy, corresponding capital return with day-trading strategy.

terms of hit ratio HR_{avg} was particularly satisfactory for S&P500 and DAX indices, achieving 61% and 64% of properly indicated next-day market directions, what makes the results comparable with state-of-art solutions. The corresponding annual returns on investment (TR) indicate promising profitability of the proposed models, especially considering buy-and-hold strategy. Since the results are obtained by analysis of multiple training/testing data sets and averaging through multiple committees, it suggests that the similar performance is expected if the models are applied to forecasting the future data. Although there may occur periodic deviations, in the long run the performance will presumably tend to converge to the presented numbers, unless the market conditions dramatically change.

The results are less obvious in case of EURUSD exchange rate prediction. The committee predictors in fact maintained the above-average performance (51%-55%), however in this case the simplest methods (naive and univariate autoregressive) yielded slightly higher returns and hit ratio. The reason can lie in not optimal choice of multivariate input (what is also suggested by inferior performance of the multivariate VARIMA model). Indeed, the currency market is usually the first one to react on macroeconomical and political events, and thus causal impact of indices on the currencies is weaker than the other way round. The weaker results can be also due to data characteristics - not all the financial

Method	Hit Ratio	Annual Return[%] buy-and-hold	Annual Return[%] day-trading
naive	0,529	11,78	**13,56**
naive cont.	0,467	-9,94	-11,53
ARIMA	**0,558**	5,81	6,63
VARIMA	0,542	**13,79**	12,31
MEAN	0,517	-2,54	-3,66
EXP	0,521	1,26	0,57
REXP	0,536	7,21	7,29
RMEAN	0,546	11,24	11,61
RIDGE	0,531	8,14	8,96

Table 5.4: EURUSD prediction - comparative results.

timeseries are equally predictable and some are closer to follow a random-walk. The problem could be possibly addressed by (1) optimizing the selection of input data, and (2) changing the committee response from binary classification (section) to multiclass classification, and defining more sophisticated strategy than simple buy-and-hold and day-trading discussed here.

Considering different committee algorithms, the MEAN and EXP methods where slightly ahead of the regression methods, with MEAN leading in terms of hit ratio and EXP in terms of annual returns. Regression algorithms performance could be possibly enhanced by one of the two improvements: (1) extending the size of the committee (and hence dimensionality of regression coefficients) or (2) adjustments of committee level cross-validation scheme.

Furthermore, the experiments confirmed the stability and flexibility of ESN committee approach - we observed rather moderate spread between performances of the best-fitting and the weakest configurations, being no higher than 3-6% in terms of HR, even though the committees varied significantly in terms of reservoir sizes and input scaling. No outliers were identified among 160 committees representing each algorithm (32 configurations, 5 committees in each) in every prediction task.

5.4.2 Case study

Following the analysis of the global results, we shall now look into performance of selected committee model in particular prediction task. The MEAN committee will be considered, that in average yielded the highest accuracy in S&P500 forecasting.

Instead of averaging the results over all tested periods, let us now observe how

the committee performance varied in the subsequent periods (see Fig.5.8). Observation of HR variance from its mean value gives estimation of how much the final results are dependent on particular training setting, or in other words - how robust the model is. The figure also illustrates the importance of cross-evaluating the performance over range of data sets, especially in financial domain. Unfortunately, numerous examples can be encountered in literature where remarkable forecasting results are claimed, however based entirely on one particular time series and single, fixed training/testing period (in our case it would correspond to selecting the 4th period which yielded the highest accuracy of 66% and considering it our final result). Such measure though does not reflect the actual generalization ability of the model. (fig.5.8). Instead, all the global result discussed in section 5.4.1 are averaged through multiple models and multiple training settings.

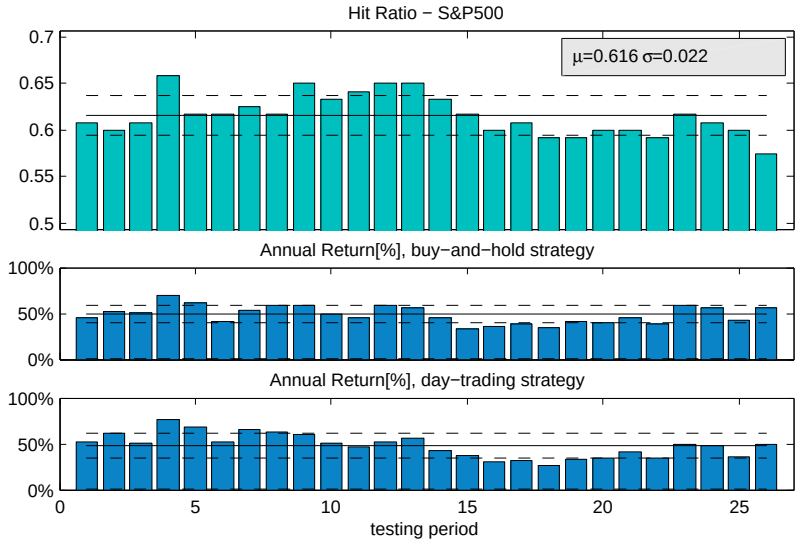

Figure 5.8: Prediction accuracy (HR) in subsequent, overlapping periods. Horizontal lines indicate the mean and one standard deviation distance from the mean. Bottom charts reffer to corresponding TR_{bah} and TR_{dt} returns.

Fig. 5.9 shows yet more detailed visualization of the prediction results, this time in terms of capital flow CF. For each of the analyzed periods (consisting of 120 trading days) we compute corresponding capital flow, which reflect how the cumulative gains/losses would develop on daily basis if investor followed the committee's indications in line with buy-and-hold strategy (section 5.2). In case of S&P500 prediction we observe incrementing lines of capital flow for all the periods, which is another indication for robustness of the committee approach.

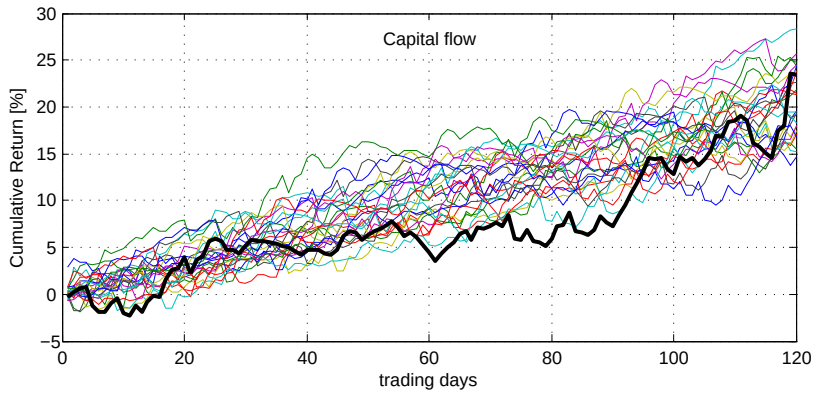

Figure 5.9: Projection of capital flow (CF) through 120 trading days, assuming buy-and-hold strategy. Each colored plot corresponds to one of the testing periods. The values at the last day correspond to the TR_{bah} returns. Thick line denotes the latest available period 24-Feb-2011 til 05-Aug-2011.

The thick line denotes the latest evaluated period (24-Feb-2011 til 05-Aug-2011), which was the most recent data at the time of writing this thesis. The final part of the highlighted plot increases very rapidly - it corresponds to the first week of August 2011, when S&P500 rapidly declined by nearly 15% within less than 2 weeks (in fact it continued the fall in the following days). Apparently the network managed to generate proper decisions and benefit from unusual volatility of the markets in that period. The similar corrective pattern of market movements occurred in May-June 2010, the period which was covered by the training data set. We shall observe the highlighted capital flow closer in the following section.

5.4.3 Trading simulation results

Suppose we have a trained committee model on financial data set $\mathcal{D}_{tr}$ and we would like to assess details of its performance on independent data set $\mathcal{D}_{test}$. From the financial perspective it would be interesting to observe the resulting capital flow versus the actual price of the forecasted asset, to assess the efficiency of the committee as an investment support tool. Furthermore, daily gains/losses resulting from system indications are also important, since they can be used to derive other measures of performance, such as distribution of returns, average return, maximum drawdown in the period, and many others. Such measures can give valueable assessment of the investment strategy and the associated risk.

From the analytical point of view, it would be interesting to observe system decisions along with the price chart of the predicted asset, and analyze how it reacts on certain price patterns, or how the predictor behaves in unusual market conditions.

For those reasons we developed a simple tool to perform a trading simulation for given predictor $\mathcal{P}$ and testing dataset D_{test}. Strictly speaking, only a vector of predictor indications $\rho = [\rho_1, ..., \rho_n]$, $\rho_i \in \{-1, 1\}$ is necessary, where each p_i indicates predicted market direction on $i'th$ day. In this way, we can connect any external model to the simulation and compare it with the committees.

Fig.5.10 illustrates the simulation results for the S&P500 index. In this setting, the committee was trained on the period 09-Jul-2009 til 23-Feb-2011 and tested on 24-Feb-2011 til 05-Aug-2011. The period includes rapid market decline that we mentioned in the previous section. The top charts illustrate committee decisions and resulting flow, both plotted together with the target asset price. Bottom charts present daily returns and their distribution. The performance of the committee seemingly outperforms market average. Moreover the gains are maintained both during market rises and declines.

Fig. 5.11 presents similar simulation results for DAX index. The committee achieves good performance with exception of the last days of strong market decline. However it manages to maintain the total return above zero level.

It should be emphasized that the results of the individual simulations are not sufficient for evaluation of given model. Instead they present details of one particular model on one particular data set. For instance, similar simulations were repeated 4320 times for each committee algorithm in order to generate the global results presented earlier in section 5.4.1 (27 training/testing data sets, 32 parametric combinations, 5 committees for each combination). However, the simulations can be helpful in the following tasks:

- Analysis (or debugging) of the model with the purpose of further optimization. For example, varying the inputs and observing how the system response changes can help to determine the optimal input combination for certain tasks, e.g. early crisis detection.

- Having the model optimized, the simulations can be used for final evaluation before application to real-time forecasting tasks (automated trading, section 2.1.6)

- Visualization of the final effects from the investment perspective, while hiding the complexity of the underlying committee.

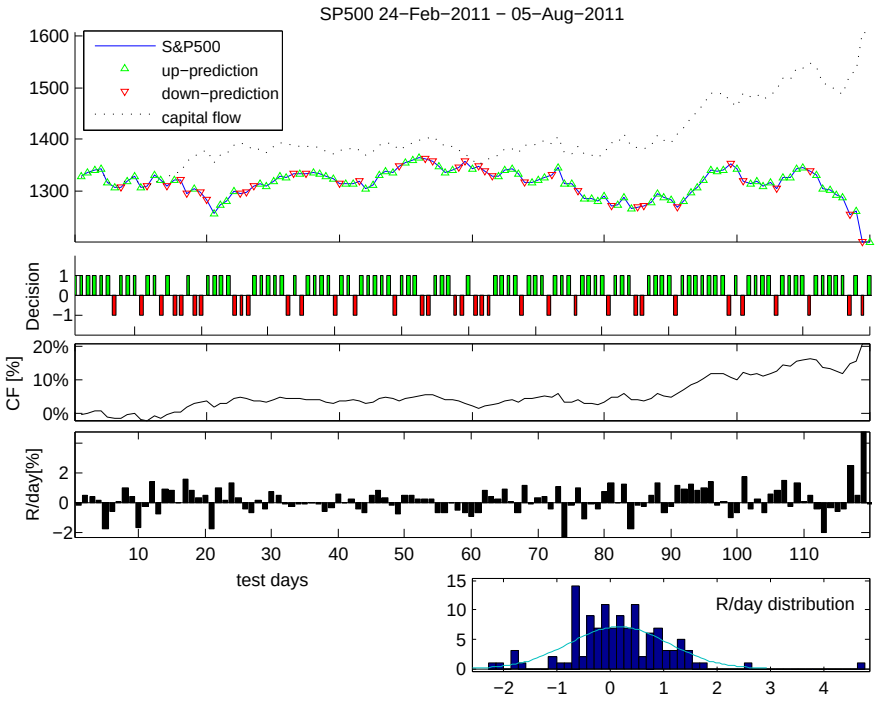

Figure 5.10: S&P500 trading simulation with reservoir-committee. From top: asset price compared with capital flow, decision vector, capital flow, daily returns, returns distribution. Triangular tags on the top chart indicate the predicted market direction on that day (i.e. committee output generated on the previous day).

5.4.4 Optimizations

In this final section we focus on influence of reservoir size and input dimensionality on the committee performance, in terms of HR. The results in financial domain are not always as explicit as it is with artificial time series, however we observed some noteworthy regularities.

5.4.4.1 Input dimensionality

Multiple committees were trained with four different combinations of inputs. The first input type is only one-dimensional and consists of the closing prices of

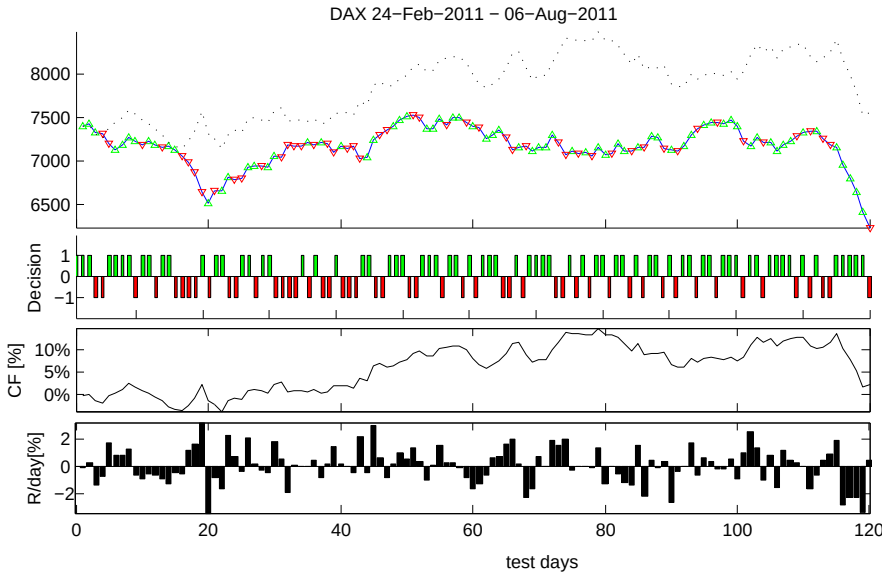

Figure 5.11: DAX trading simulation with reservoir-committee.

the target asset (here: S&P500). Subsequent configurations gradually increase dimensionality. The second type of input contains full information about previous prices of the target asset, that is open-, min-, max- and close-prices and transaction volume. The third input is already multivariate and apart from the target asset information contains also prices of NIKKEI index and EURUSD exchange rate. The fourth input type, apart from all the previous data, comprises also linearly preprocessed target asset, in particular: Relative Strength Index, Price Rate Of Change, and Moving Average Convergence-Divergence. The technical analysis (TA) indicators were presented in section 5.1.7. All three indicators are properly scaled to match with other inputs.

For every input type we train population of committees with varied input scale parameter $\beta = [50, 100, ..., 400]$, in that way accounting for the need of input scaling along with increasing dimensionality. For each combination of input type and scaling factor we compute HR results and present the results on Fig. 5.12. Note that HR is obtained similarly like before, by averaging through 26 testing data sets and through 5 committees.

The benefits from increasing input dimensionality are clearly observable, especially in case of RIDGE committees (bottom plots). The performance gain is particularly large after adding external signals - NIKKEI and EURUSD. Inter-

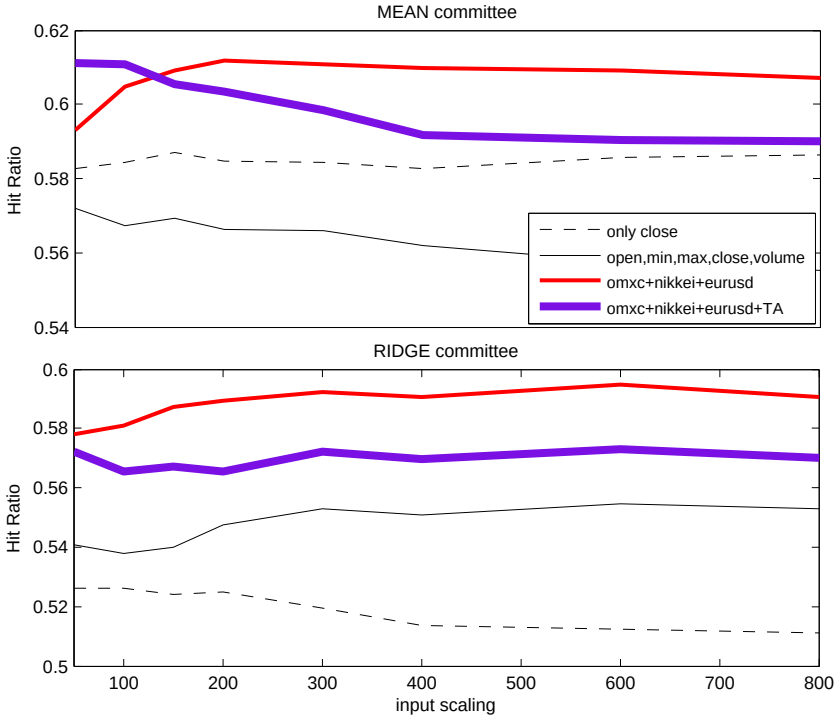

Figure 5.12: Influence of multivariate input on Hit Ratio performance.

estingly, additional TA indicators do not lead to significantly higher accuracy and in case of RIDGE committee they even degrade the performance. It is possible that higher reservoirs are needed to take advantage of high dimensionality of the input.

5.4.4.2 Large reservoirs

For the same prediction task (S&P500) multiple committees were trained, this time with varying reservoir sizes $N = \{100, 200, ..., 600\}$. HR is computed by averaging through 26 testing data sets and through 5 committees. Results are plotted on Fig.5.13.

Significant gain was observed for MEAN and EXP methods after increasing reservoir sizes from 100 to 400-600 neurons. The results are consistent with conclusions from Chapter 4 concerning large reservoirs. In current experiments

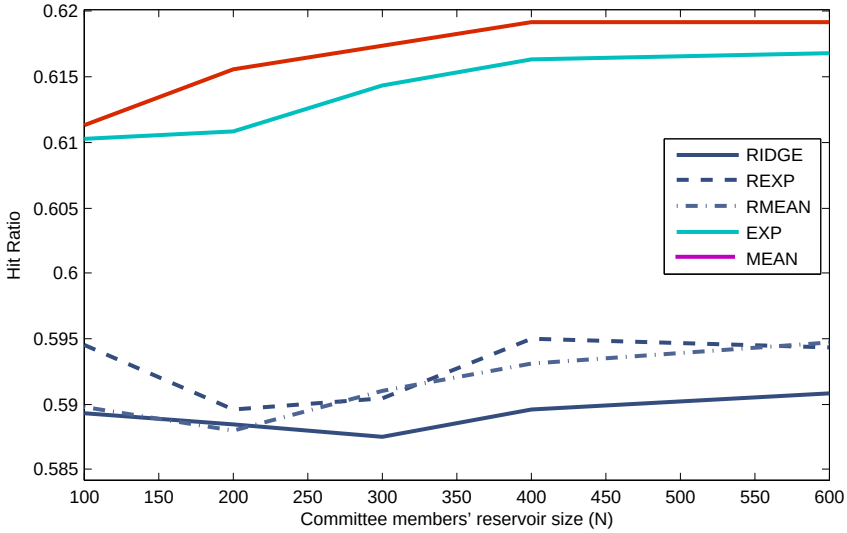

Figure 5.13: Influence of reservoir size of committee members on Hit Ratio performance.

we can consider reservoir large if the number of neurons is 400 or more, since this is the size of the training data set. It still remains to be verified whether large reservoirs increase the capacity sufficiently so that the system can benefit from higher dimensionality of multivariate inputs.

5.5 Summary

In this concluding section we applied the model developed earlier to non-trivial engineering task of financial time series forecasting. The financial domain is particularly challenging due to noisy and chaotic behavior of the data. We have touched upon many relevant issues, in particular data acquisition and preprocessing, adapting model to classification of next-day market direction, optimizations and defining simple investment strategies so that committee can be used as investment support system. Finally, we defined benchmarking and simulation environment, and tested the proposed models against classic autoregressive models and naive strategies. We obtained the promising results in two out of three large-scale comparative studies, comparable to state-of-art in the field. In the third experiment the committees performed reasonably good but did not outperform the autoregressive models.

Chapter 6

Conclusions

In the scope of the first part of the thesis we evaluated echo state network approach in the task of chaotic time series forecasting. Starting from analysis of single ESN models and artificial time series, we gradually advanced the concept to the committee level, which can be considered the central part of the thesis. Extensive comparative studies of different ensemble methods were performed and the final results allowed us to formulate several conclusions and design principles, in particular formulate the conditions in which certain types of the committees are more suitable than the others. In general, we conclude that relation between reservoir size, regularization scheme, and length of the training dataset constitute essential parameters needed to be considered in case of ESN committee forecasting. Our results showed that increasing reservoir size accompanied by regularization will almost always lead to asymptotic error decrease. The similar regularity was observed for both artificial and financial time series. The approach is therefore recommendable, unless the computational constraints are the issue. On the other hand, we presented examples where nonregularized committees of small reservoir networks act as efficient regularizers, and thus can yield comparable performance as the corresponding regularized committees, but with significantly lower computational cost involved.

The first part of the project resulted also in development of the Matlab framework to support ESN committees design and analysis. The implementation includes class representations of ESNs, topological ESNs and ESN-committees,

training algorithms, and diverse visualization tools, in particular for financial simulations. The framework will constitute a useful tool in further research.

The second phase of the project focused on application of the reservoir committee model to non-trivial task of forecasting the financial time series, characterized by non-stationary, chaotic behavior and large noise. Significant efforts were committed to domain analysis, data acquisition and preprocessing. The main conclusions coming from this part of the project relate to particular importance of data selection and preprocessing. Careful domain analysis is highly recommended to identify explanatory relations between financial markets, macroeconomic factors, currency exchange rates and other related variables, so as to construct informative, multivariate combination of signals for the system input. The data aspect is equally important, or perhaps more, than the model design itself. Furthermore, the grid search of optimal parameters and input configurations will be often inevitable in financial forcasting, since the predictor does not always respond intuitively to certain changes in experimental setups and hence derivation of good design principles is a hard problem.

Finally, we performed large-scale comparative studies of reservoir committees and classic autoregressive models in light of their applicability to financial forecasting. The proposed reservoir committees managed to achieve noteworthy results in next-day prediction of major global indices - S&P500 and DAX, and acceptable results in case of EURUSD rates. The subject was certainly not exhausted, and numerous ideas had to be put aside due to project constraints. However, our preliminary empirical studies give evidence that committees of echo state networks have potential to compete with state-of-art solutions in the field of financial forecasting.

It can be concluded that the main objective of the thesis, which is evaluation of reservoir committee methods and their applicability in financial forecasting, was fulfilled. Moreover, along with the project proceedings many ideas emerged of how to further elucidate the complex dynamics of the model as well as enhance its performance in financial domain. Those inspirations constitute promising basis for further research.

Bibliography

[1] Jaeger, H., The "echo state" approach to analyzing and training recurrent neural networks, Technical Report GDM Report 148, German National Research Center for Information Technology, 2001

[2] H. Jaeger, A tutorial on training recurrent neural networks. Covering bptt, rtrl, ekf, and the echo state network approach, German National Research Center for Information Technology, 2002

[3] Mass, W., T. Natschlager, H. Markram, Real-time computing without stable states: a new framework for neural computation based on perturbations, Neural Computation 14 (11) (2002) 2531-2560

[4] Jaeger H. and H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, Science, vol. 304, pp. 78-80, Apr 2004

[5] Ferreira A.A., Investigating the use of Reservoir Computing for forecasting the hourly wind speed in short -term, IEEE International Conference on Neural Networks, pp. 1649-1656, ICNN 2008

[6] Wyffels F., B.Schrauwen, A comparative study of Reservoir Computing strategies for monthly time series prediction, Neurocomputing 73(2010)1958–1964

[7] Fangwen Zhai, Xiaowei L., Zehong Y., Yixu S., Financial time series prediction based on echo state network, Sixth International Conference on Natural Computation, Vol.8, pp. 3983-3987, 2010

[8] Hartland Cedric, Bredeche N., Sebag M., Memory-enhanced evolutionary robotics: The echo state network approach, IEEE Congress on Evolutionary Computation, pp. 2788-2795, CEC 2009.

[9] Lukosevicius, M., H. Jaeger, Reservoir Computing approaches to recurrent neural network training, Computer Science Review, vol. 3, pp. 127-149, 2009

[10] Verstraeten, D., et al., An experimental unification of reservoir computing methods, Neural Networks, vol. 20, pp. 391-403, 2007

[11] Steil, J.J., Online reservoir adaptation by intrinsic plasticity for backpropagation-decorrelation and echo state learning, 2006

[12] Ozturk, M.C., D. Xu, J.C. Principe, Analysis and design of echo state networks, Neural Computation 19, 111–138, 2007

[13] Jiang, F. et al., Unsupervised learning of echo state networks: balancing the double pole. Proceedings of the 10th annual conference on Genetic and evolutionary computation, USA, 2008.

[14] Qingsong Song, Zuren Feng, Effects of connectivity structure of complex echo state network on its prediction performance for nonlinear time series, Neurocomputing 73, pp. 2177–2185, 2010

[15] Zhidong Deng and Yi Zhang, Collective behavior of a small-world recurrent neural system with scale-free distribution, IEEE Transactions on Neural Networks, Vol. 18, No. 5, September 2007

[16] Xue, Y., Yang, L., Haykin, S., Decoupled echo state network with lateral inhibition, IEEE Transactions on Neural Networks, January 2007

[17] Dutoit X., B. Schrauwen, J. Van Campenhout, D. Stroobandt, H. Van Brussel, M. Nuttin, Pruning and regularization in reservoir computing, Neurocomputing 72, 1534–1546, 2009

[18] Kobialka, H., U. Kayani, Echo state networks with sparse output connections, ICANN 2010, Part I, LNCS 6352, pp. 356–361, 2010

[19] Jaeger H. et al., Optimization and applications of echo state networks with leaky- integrator neurons, Neural Networks, vol. 20, pp. 335-352, 2007

[20] Ferreira A.A., Ludermir T.B., Evolutionary strategy for simultaneous optimization of parameters, topology and reservoir weights in Echo State Networks, The 2010 International Joint Conference on Neural Networks (IJCNN), 2010

[21] Roeschies, B., C. Igel, Structure optimization of reservoir networks, Oxford University Press, Vol. 18 No. 5, pp. 635-668, 2009

[22] Webb, R.Y.: Multi-layer corrective cascade architecture for on-line predictive echo state networks, Applied Artificial Intelligence, 22, pp. 811–823, 2008

[23] Babinec, S., J. Pospichal, Gating echo state neural networks for time series forecasting, ICONIP 2008, Part I, LNCS 5506, pp. 200–207, Springer, Heidelberg, 2009

[24] Hoerl A., R.Kennard, Ridge regression: biased estimation for non orthogonal problems, Technometrics 42(1970) 55–67.

[25] Bishop, C.M., Pattern recognition and machine learning, Springer, 2006

[26] Bishop, C.M., Neural networks for pattern recognition, Oxford University Press, 1995

[27] Perrone M.P., L.N. Cooper, When networks disagree: ensemble methods for hybrid neural networks, Artificial Neural Networks for Speech and Vision, pp. 126-142, 1993

[28] Jacobs, R.A. at al., Adaptive mixtures of local experts, Neural Computation 3(1), pp. 79-87, 1991

[29] Jordan, M. I., R. A. Jacobs, Hierarchical mixtures of experts and the EM algorithm, Neural Computation 6(2), pp. 181-214, 1994

[30] Freund, Y., R. E. Shapire, Experiments with a new boosting algorithm, Thirteenth International Conference on Machine Learning, pp. 148-156, 1996

[31] Friedman, J. H., Greedy function approximation: a gradient boosting machine, Annals of Statistics 29(5), pp. 1189-1232, 2001

[32] Davidson A., How the Global Financial Markets Really Work. The definitive guide to understanding international investment and money flows, The Times, 2009

[33] Murphy J.J., Intermarket Technical Analysis – Trading Strategies for the Global Stock, Bond, Commodity and Currency Markets, John Wiley & Sons, 1991

[34] Blanchard O., Macroeconomics, Third Edition, Prentice Hall, 2003

[35] Lipsey L.G., Chrystal K.A., Economics, Tenth Edition, Oxford University Press, 2004

[36] Kathy L., Day trading the currency market, John Wiley & Sons, 2005

[37] Dunsby A., "Commodity Investing", John Wiley & Sons, 2008

[38] Rob Iati, The Real Story of Trading Software Espionage, AdvancedTrading.com, July 10, 2009

[39] Kovalyov S., Programming in Algorithmic Language MQL4, http://book.mql4.com

[40] Fama, E., Efficient Capital Markets: A Review of Theory and Empirical Work, Journal of Finance 25 (2), pp. 383–417, 1970

[41] Achelis S.B., Technical Analysis from A to Z, McGraw Hill, 2001

[42] Box, G.E.P., Jenkins, G.M., Time Series Analysis, Forecasting and Control, Holden-Day, San Francisco, 1970/1976