

MASARYKOVA UNIVERZITA

**Proceedings of the
11th Summer School of Applied Informatics**

**Sborník příspěvků
11. letní škola aplikované informatiky**

**Bedřichov
12.–14. 9. 2014**

Editori: Jiří Hřebíček
Editors: Jan Ministr
Tomáš Pitner

Techničtí editoři: Jiří Kalina
Technical editors: Ing. Lenka Hefková

Odborní garanti letní školy: Prof. RNDr. Jiří Hřebíček, CSc., MU, IBA
Expert guarantos: Doc. RNDr. Tomáš Pitner, Ph.D., MU, FI

Programový výbor:
Editorial and programme board:

Doc. RNDr. Ladislav Dušek, Dr. – Masarykova univerzita, Brno, CZ
Doc. Ing. Josef Fiala, CSc. – Vysoká škola podnikání, Karviná, CZ
Doc. Ing. Alena Kocmanová, CSc. - Vysoké učení technické v Brně, CZ
RNDr. Jaroslav Ráček, Ph.D. – IBA CZ, s.r.o., CZ
Prof. Andrea Rizzoli – University of Applied Sciences and Arts of Southern
Switzerland, Lugano, CH
Dr. Gerald Schimak - Austrian Institute of Technology, Vienna, AT
Doc. Ing. Jan Žizka, CSc. – Mendelova univerzita v Brně, CZ

Slovo úvodem

11. letní škola aplikované informatiky navázala na předchozí letní školy aplikované (environmentální) informatiky v Bedřichově, které se zde konají od roku 2002 s výjimkou roku 2004, kdy se letní škola konala v Šubířově a roku 2005, kdy byla hlavní akcí Masarykovy univerzity (MU) v oblasti environmentální informatiky na 19. mezinárodní konferenci Informatics for Environmental Protection - EnviroInfo 2005 s nosným tématem Networking environmental information a kterou hostila Masarykova univerzita ve dnech 7. až 9. září 2005 v brněnském hotelu Voroněž. V letech 2006 až 2013 se letní školy konaly opět v Bedřichově v penzionu U Alexů (nyní pension Artis).

V letošním roce 11. letní škola aplikované informatiky se konala ve dnech 12. až 14. září 2014 opět v Bedřichově v penzionu Artis. Úroveň 11. letní školy pozvedla skutečnost, že se jí zúčastnili a měli přednášky zahraniční profesori prof. Ralf Denzer, president Environmental Informatics Group (EIG), z Hochschule für Technik und Wirtschaft des Saarlandes, v Saarbrücken, Německo, prof. Melius Widenman z Cape Peninsula University of Technology z Kapského města v Jižní Africe a dr. Bruno Rossi, hostující profesor na Fakultě informatiky Masarykovy univerzity. Všemi přednáškami 11. letní školy prolínala skutečnost, že aplikace moderních informačních a komunikačních technologií pro životní prostředí (potažmo environmentální informatiky) jak v České republice (ČR), tak i mezinárodně v Evropské unii (EU) a ve světě se zaměřuje na podporu eEnvironmentu, Jednotného informačního prostoru pro životního prostředí (SISE – Single Information Space in Environment for Europe) a Sdíleného informačního systému pro životní prostředí (SEIS – Shared Environmental Information System), které podporují naplňování nové politiky v budování informační společnosti EU a ČR, které přinesla „Digital Agenda for Europe“ v rámci vize nové Evropské komise „eEUROPE 2020“. Jde zejména o přeshraniční informační služby v rámci eEnvironmentu. Jedná se o sdílení monitorovaných a zpracovávaných dat a informací o atmosféře, povrchových i podzemních vodách, odpadech, půdě, biodiverzitě, atd. pomocí Globálního monitorovacího systému životního prostředí a bezpečnosti (GMES – Global Monitoring for Environment and Security). Tyto informační služby umožňují efektivnější a přesnější sledování aktuálního stavu životního prostředí a udržitelného rozvoje v Evropě, dále pak jeho modelování a simulaci jeho dalšího vývoje.

Za hlavní přínos 11. letní školy považujeme skutečnost, že na letošním ročníku setkali studenti s významnými světovými vědci, kteří se s nimi podělili jak o své znalosti, tak i jejich pohledy na výuku aplikované informatiky ve světě. V příspěvcích z letní školy jsou zastoupeni nejen doktorandi a učitelé z Masarykovy univerzity (Institut biostatistiky a analýz, Přírodovědecká fakulta a Fakulta informatiky), ale i z Mendlovy univerzity (Agronomická fakulta) a Vysoké školy báňské – Technické univerzity (Ekonomická fakulta) v Ostravě. Jejich příspěvky ve sborníku přispívají k tomu, že se letní škola stala širokým interdisciplinárním odborným fórem v rámci České republiky. Dále je důležité, že několik příspěvků je publikováno v anglickém jazyce, který podtrhuje mezinárodní význam sborníku.

Těžiště projednávaných otázek na letní škole bylo především v detailní diskusi věnované řešení projektů v oblasti aplikované informatiky a účasti zahraničních profesorů Ralfa Denzera, Meliuse Weidenmana a dr. Bruna Rossiho.

V Brně dne 31. listopadu 2014

Jiří Hřebíček
Jan Ministr
Tomáš Pitner
Editoři

Obsah

Movement of the Cooling-Warming border over territory of the Czech Republic	
<i>Stanislav Bartoň, Renata Osičková</i>	8
Forecasting Financial Volatility: Application of Customized Neural Network Combined with ARCH Model into Time Series Modelling	
<i>Lukáš Falát, Dušan Marček</i>	22
A Case Study of Failure Parameter Estimation in Software Reliability Models	
<i>Stanislav Chreň, Barbora Bůhnová</i>	43
Platform for medical curriculum innovation: The role of specialized vocabularies	
<i>Martin Komenda</i>	70
Towards Flexible Intelligent Building Data Analysis	
<i>Adam Kučera, Tomáš Pitner</i>	77
Modelling and Forecasting of WIG20 stock index	
<i>Dušan Marček</i>	84
An influence of changing conditions within the EU ETS system on the steel company	
<i>Jan Ministr, František Zapletal</i>	92
The Architecture of Semantically Partitioned Complex Event Processing	
<i>Filip Nguyen, Tomáš Pitner</i>	106
Matematické základy optimalizace svozové trasy komunálního odpadu	
<i>Michal Petřík, Stanislav Bartoň</i>	113

Movement of the Cooling-Warming border over territory of the Czech Republic

Stanislav Bartoň, Renata Osičková

Department of Technology and Automobile Transport, Mendel University in Brno,
Zemědělska 1, 613 00 Brno, Czech Republic
barton@mendelu.cz

Abstract

In this paper, authors are processing and analyzing values of the average monthly temperatures recorded at 34 meteorological stations since January 2003 till December 2013 that are uniformly distributed in the territory of the Czech Republic. At first, statistical relevance of each term of used regression function is evaluated by using the sum of squared residuals per degree of freedom. At second, function containing only significant terms is tested on the edge of statistical reliability equal to 99%, 95% and 90% with respect to its individual terms. The Fisher-Snedecor function was used to determine really important terms of the evaluated function. Final function enables computation coefficients of 7 regression functions, that explain recorded data in 7 time steps of 5 year intervals. These functions were used to determine border positions splitting areas of the Czech Republic into zones of local warming and cooling. Recorded data are plotted graphically and show that this border oscillates over the whole territory of the Czech Republic.

Abstrakt

V tomto příspěvku autoři zpracovávají a analyzují hodnoty průměrné měsíční teploty zaznamenané v 34 meteorologických stanic od ledna 2003 do prosince 2013, které jsou rovnoměrně rozloženy na území České republiky. Nejprve, je vyhodnocen statistický význam každého období použitých regresní funkce pomocí součet čtverců reziduí po stupních volnosti. Za druhé je testována funkce obsahující pouze významné termíny na okraji statistická spolehlivost rovnající se 99 %, 95 % a 90 % s ohledem na jejich individuální podmínky. Fisher-Snedecor funkce byla použita k určení skutečně důležitých podmínek hodnocených funkcí. Poslední funkce umožňuje výpočet koeficientů 7 regresních funkcí, které vysvětlují zaznamenané údaje v 7 krocích časového intervalu 5 let. Tyto funkce byly použity k určení pozice hranice rozdělení oblastí České republiky do zón místního oteplování a ochlazování. Zaznamenané údaje jsou zobrazeny graficky a ukazují, že tato hranice osciluje na celém území České republiky.

Keywords

Global warming, Mathematical modeling, Regression function, Linear correlation, Space and time coincidence, Temperature trends.

Klíčová slova

Globální oteplování, Matematické modelování, Regresní funkce, Lineární korelace, Prostor a čas, náhoda, Teplotní trendy.

1 Introduction

Problem of the global warming represents a widely discussed theme which is in the center of interest of the major part world population, for example [5], [10]. Many authors publish papers that accentuate the fact that the process of global warming is real and quite inevitable while some others wrote this is a disputable phenomenon and that the global warming is a mere fiction, [7] In this paper we present a mathematical study of the development of diurnal temperatures in the Czech Republic territory within

the period of the recent decade from January 2003 till December 2013. Using the Maple application, see [9], based on the method of least squares we have developed a regression function $T(t, x, y, h)$, which explains the dependence of temperature on time, geographical position and height above the sea. To determine the significance of the regression function, members that have been tested with a confidence interval of 90%, 95% and 99% have been used in the Fischer-Snedecor function (4). The resulting functions allow us to calculate the coefficients of the regression function, which is used at the end and are necessary for determining the motion boundary warming and cooling in the area of the Czech Republic. Calculations were made for seven five-year cycles, and gradually shifted by one year.

2 Material and Methods

Data concerning average monthly temperatures, recorded within the period of the last ten years in 22 selected meteorological stations, are normally available on the Internet, [1]. As far as further 12 stations are concerned, similar data can be reached from the graphs that are available at the web page, [2].

The Czech Hydrometeorological Institute collects data about the daily temperatures, which are measured and recorded at higher number of meteorological stations than we do in our case study and for a long time period. These data, however, can be obtained only on the base of paid membership and for that reason they are not available for wider public.

Nevertheless, data recorded in 34 available meteorological stations cover the Czech Republic territory adequately and in a satisfactory manner. The minimum airline distance between two stations is 12 km while the maximum does not exceed 54.7 km. Data presented in this paper informs about an exact geographical location of the station, about its altitude and also about average monthly air temperatures, see Tab. 1. Temporary data are expressed as a yearly fractions and the time $t = 0$ corresponds with the 1st January 2003. In case that temperature partial data are excluded or missing, the temperature is rewritten by $-99\text{ }^{\circ}\text{C}$. Stations with incomplete data are highlighted in red, stations in Group 1, or in blue, Group 2. Data from Group 2 are reconstructed from graphs, see Tab. 1.

Table 1: Example of the data

		Břez, Tuř (České Bud)	Doksany	Holešov	Hradec Kr.	Chelb	Chrástov	Klatov	Kuchařov	Liberec	Lysá hora	Milešovka	Mošnov	Olomouc	Praha, Ka	Příbram	Nová Ves	Ústí nad L.	Pec pod Skálkem	Kostelní K	Ústí nad O	Strážnice	Svitávka	Heřmanice		
		241	388	158	224	278	483	1 118	430	334	398	1 322	833	250	210	232	530	725	375	824	534	569	602	178	593	328
		49°09'36"	48°07'42"	50°07'30"	49°19'20"	50°03'04"	50°04'26"	49°04'06"	49°23'27"	48°52'57"	50°06'12"	49°22'46"	50°03'17"	49°04'54"	49°04'33"	50°04'03"	49°04'58"	50°06'34"	50°04'02"	50°04'13"	49°04'24"	49°09'36"	49°08'49"	48°53'57"	50°01'59"	48°58'27"
date	t [year]	16°01'44"	14°28'05"	14°01'13"	17°04'24"	15°50'19"	12°04'12"	13°06'54"	13°08'08"	16°05'11"	15°01'27"	18°26'52"	13°55'53"	18°07'18"	17°07'04"	14°25'02"	15°45'45"	13°29'08"	14°02'08"	15°43'44"	15°04'47"	15°26'21"	16°25'20"	17°20'17"	17°24'04"	16°58'03"
15.1.2003	0.04	-2.1	-1.4	-0.7	-2.2	-1.7	-2.2	-4.8	-1.4	-1.9	-2.9	-7.4	-4.5	-2.6	-2.5	0.0	-3.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
14.2.2003	0.12	-2.7	-3.2	-2.5	-3.0	-3.6	-4.5	-5.8	-4.3	-3.1	-4.6	-8.0	-4.9	-4.2	-3.3	-1.5	-5.5	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
15.3.2003	0.20	4.7	5.2	5.1	4.0	4.4	4.0	0.9	4.8	5.0	2.8	-1.8	2.5	3.3	3.8	6.5	3.1	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
15.4.2003	0.29	5.1	8.7	9.4	8.4	8.4	8.8	3.4	7.8	8.6	6.6	1.1	5.2	8.0	8.7	9.8	6.5	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
15.5.2003	0.37	17.2	16.4	16.4	16.7	16.6	14.1	11.1	15.3	17.1	14.4	10.5	12.4	16.2	16.8	17.0	14.7	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
15.6.2003	0.45	21.3	20.6	20.7	20.6	20.8	19.1	16.1	20.4	21.5	18.7	13.6	17.1	20.7	21.1	21.7	18.6	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
15.7.2003	0.54	20.3	19.8	19.5	19.7	19.6	17.8	14.8	19.0	21.0	18.0	12.8	16.2	19.8	20.1	20.7	18.1	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
15.8.2003	0.62	22.5	21.5	21.5	21.2	21.1	19.8	17.6	21.2	23.2	19.1	14.8	18.7	20.4	21.5	22.7	20.4	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
15.9.2003	0.71	15.4	14.1	14.5	14.7	14.7	13.0	10.4	13.3	15.2	13.1	8.9	12.1	14.4	14.6	15.8	13.5	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
15.10.2003	0.79	6.9	5.9	6.3	6.8	5.9	4.6	1.5	5.5	6.7	4.8	-0.6	2.2	6.4	6.3	7.2	5.2	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
15.11.2003	0.87	5.9	4.9	5.2	6.0	5.9	4.3	2.9	4.4	5.9	5.7	1.5	2.7	6.1	5.6	6.2	5.3	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
15.12.2003	0.96	0.2	-0.2	0.4	0.6	0.3	0.0	-1.6	0.1	0.1	0.2	-3.6	-2.2	0.9	-0.5	1.6	-0.7	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
15.1.2004	1.04	-3.3	-2.0	-2.9	-3.2	-3.5	-2.7	-5.7	-1.9	-2.7	-3.4	-9.1	-5.3	-3.4	-3.9	-1.8	-4.1	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
14.2.2004	1.12	1.0	2.3	3.4	0.7	1.2	1.4	-2.7	1.9	1.8	0.5	-6.0	-1.5	0.5	-0.1	3.3	0.1	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
15.3.2004	1.20	3.8	3.3	4.6	3.7	4.0	2.7	-1.1	2.6	3.7	3.1	-3.0	0.7	3.4	3.4	5.1	2.2	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
15.4.2004	1.29	10.7	9.6	11.0	10.5	10.1	8.1	4.7	9.2	10.7	6.8	2.7	6.8	9.8	10.5	11.1	8.5	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
15.5.2004	1.37	13.1	12.5	13.3	13.0	12.5	10.9	7.1	11.8	13.3	11.2	5.0	8.6	13.0	13.3	13.6	11.1	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
15.6.2004	1.46	17.0	16.3	17.4	16.6	16.4	15.4	11.1	15.7	17.2	15.1	9.8	13.1	16.6	16.9	17.6	15.4	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
15.3.2012	9.20	7.3	6.5	6.6	6.4	-99.0	5.6	2.7	6.5	7.0	4.8	-0.8	4.3	5.3	6.6	8.5	4.8	3.8	6.3	1.5	5.7	5.3	4.9	5.0	2.7	4.7
15.4.2012	9.29	10.7	9.2	9.9	10.6	-99.0	7.7	4.5	9.1	9.9	8.2	3.1	6.1	10.4	10.3	11.0	7.9	5.9	8.7	3.8	8.3	7.9	8.7	11.8	6.7	8.5
15.5.2012	9.37	16.9	15.0	16.2	16.1	-99.0	14.1	10.6	14.9	16.0	14.3	9.2	12.7	15.4	16.5	17.6	13.8	12.2	14.9	10.7	14.1	14.0	15.2	14.5	12.1	14.4
15.6.2012	9.46	19.6	18.0	18.4	18.9	-99.0	16.3	13.0	17.6	18.8	15.9	12.0	14.1	18.4	18.9	20.3	16.5	13.7	16.4	13.3	16.9	16.9	17.5	18.8	15.3	17.1
15.7.2012	9.54	21.0	18.7	19.7	20.2	-99.0	17.5	13.7	18.1	20.2	17.7	14.4	15.6	20.3	20.9	21.6	17.9	15.3	18.0	15.0	17.6	17.7	18.8	18.1	16.9	18.5
15.8.2012	9.62	21.3	18.9	20.0	19.9	-99.0	18.4	14.5	18.5	20.4	17.2	13.9	16.6	19.5	20.4	21.9	18.2	15.7	18.9	14.1	18.4	18.4	18.4	19.9	16.0	18.0
15.9.2012	9.71	16.2	14.0	14.5	15.6	-99.0	12.9	9.9	13.6	15.3	13.1	9.6	11.8	14.8	15.3	15.7	13.2	11.4	13.7	9.5	13.5	13.4	13.5	15.8	11.4	13.3
15.10.2012	9.79	9.7	8.5	8.2	8.9	-99.0	6.9	5.6	8.0	8.7	7.5	4.7	5.6	8.8	8.9	9.2	7.4	5.7	7.5	4.9	7.3	7.3	7.9	8.7	6.4	7.5
15.11.2012	9.88	6.5	5.2	5.2	7.0	-99.0	3.4	3.0	4.7	5.4	5.3	1.5	2.2	6.5	6.6	6.5	5.0	2.4	4.4	2.5	4.5	4.8	5.9	2.9	4.5	6.1
15.12.2012	9.96	-1.4	0.7	0.9	-1.3	-99.0	0.8	-2.4	0.4	-1.6	0.9	-5.2	-2.8	-1.4	-2.0	1.4	-1.9	-2.6	-1.4	4.0	-0.3	-2.2	-1.9	2.8	-3.8	-3.0
15.1.2013	10.04	-1.3	-0.1	-0.3	-2.0	-1.2	-1.6	-3.9	-99.0	-1.3	-2.3	-6.0	-99.0	-2.5	-1.7	0.7	-99.0	-3.9	-1.6	-3.9	-1.6	-2.1	-1.9	-1.8	-3.9	-2.5
14.2.2013	10.12	0.5	-0.6	0.3	0.0	-0.2	-2.1	-5.5	-99.0	-0.4	-1.8	-5.7	-99.0	-0.4	0.4	1.0	-99.0	-3.9	-1.3	-4.0	-1.4	-1.7	-0.8	0.6	-2.5	-0.9
15.3.2013	10.20	1.3	1.0	0.4	1.2	0.1	-1.1	-3.3	-99.0	0.6	-1.5	-5.7	-99.0	0.3	1.1	1.5	-99.0	-3.5	-1.3	-4.1	-0.5	-0.5	-0.6	1.9	-2.8	-0.2
15.4.2013	10.28	10.5	9.5	9.8	9.8	9.1	7.7	4.2	-99.0	9.9	7.8	3.8	-99.0	9.1	10.3	10.9	-99.0	5.6	8.4	3.6	8.3	8.2	8.4	10.1	6.0	8.3
15.5.2013	10.37	14.1	12.9	13.7	14.0	13.3	11.1	7.8	-99.0	13.4	12.0	7.7	-99.0	13.8	14.6	14.0	-99.0	9.3	12.2	9.1	11.9	11.6	12.7	14.4	10.7	12.9
15.6.2013	10.46	17.9	16.9	17.7	17.2	17.2	15.3	11.9	-99.0	17.2	15.5	11.1	-99.0	17.1	18.0	18.4	-99.0	13.2	16.1	13.0	15.8	15.7	16.4	17.7	14.4	16.7

2.1 Topography

The first step is to recompute GPS coordinates of the stations into 3D coordinates. As a z coordinate is taken distance from the tangent plane to the Earth at the average of the GPS coordinates of the perimeter of the Czech Republic. Axe x is oriented parallel with west-east direction, axe y is parallel with the north-south orientation. GPS coordinates and measured data are saved in the file **Data_3.sav**. File **CR.sav** contains coordinates of the perimeter of the Czech Republic. Kilometre is used as a distance value for x and y coordinates, coordinate z , equal to height h is measured in meters. Result is displayed in the Fig. 1.

```
> restart; Digits:=15; with(plots): with(LinearAlgebra): with(Statistics):
> read "CR.sav": CR:=map(u->[u[2],u[1]],CR): n:=nops(CR);
TCR:=add(u,u=CR)/n;
> CR:=map(u->u-TCR,CR): R:=12745591*0.5;
> CR:=map(u->evalf([R*cos((TCR[2]+u[2])*Pi/180)*u[1]*Pi/180000,
    R*u[2]*Pi/180000]),CR):
> G0:=spacecurve(map(u->[u[1],u[2],0],CR),color=magenta):
> read "Data_3.sav": T:=Data[3][5..8]: Jmeno:=[]: Vyska:=[]:
Sirka:=[]: Delka:=[]:
> for i from 4 to nops(Data) do;
    Jmeno:=[Jmeno[],Data[i][1]]; Vyska:=[Vyska[],Data[i][2]];
    Sirka:=[Sirka[],Data[i][3]]; Delka:=[Delka[],Data[i][4]];
end do;
> Nj:=nops(Jmeno);
> Sirka:=map(v->evalf((v[1]+v[2]/60+v[3]/3600)),map(u-
>sscanf(convert(subs(176=32,
    39=32,44=32,34=NULL,convert(u,bytes)),bytes),"%d %d %d"),Sirka)):
> Delka:=map(v->evalf((v[1]+v[2]/60+v[3]/3600)),map(u-
>sscanf(convert(subs(176=32,
    39=32,44=32,34=NULL,convert(u,bytes)),bytes),"%d %d %d"),Delka)):
> Stanice:=zip((u,v)->[u,v]-TCR,Delka,Sirka):
> Stanice:=map(u->evalf([R*cos((TCR[2]+u[2])*Pi/180)*u[1]*Pi/180000,
    R*u[2]*Pi/180000]),Stanice):
> X:=map(u->u[1],Stanice): Y:=map(u->u[2],Stanice):
> z:=sort(map(u->u[1],CR)): Xmin:=z[1]; Xmax:=z[-1];
> z:=sort(map(u->u[2],CR)): Ymin:=z[1]; Ymax:=z[-1];
> G1:=pointplot3d(zip((u,v)->[u[],0],Stanice,Vyska),style=point,
    symbol=circle,color=black):
> G2:=pointplot3d(zip((u,v)->[u[],v],Stanice,Vyska),style=point,
    symbol=circle,color=blue):
> G3:=spacecurve({zip((u,v)-
>[[u[],0],[u[],v]],Stanice,Vyska)[1]},color=brown):
> G4:=textplot3d([seq([op(j,op(1,G2))],cat(" ",Jmeno[j])),j=1..Nj]),
color=khaki,align={above,right}):
```

```
> display({G0,G1,G2,G3,G4},orientation=[-95,30],
axes=framed,labels=["x [km]","y [km]","h [m]"]);
```

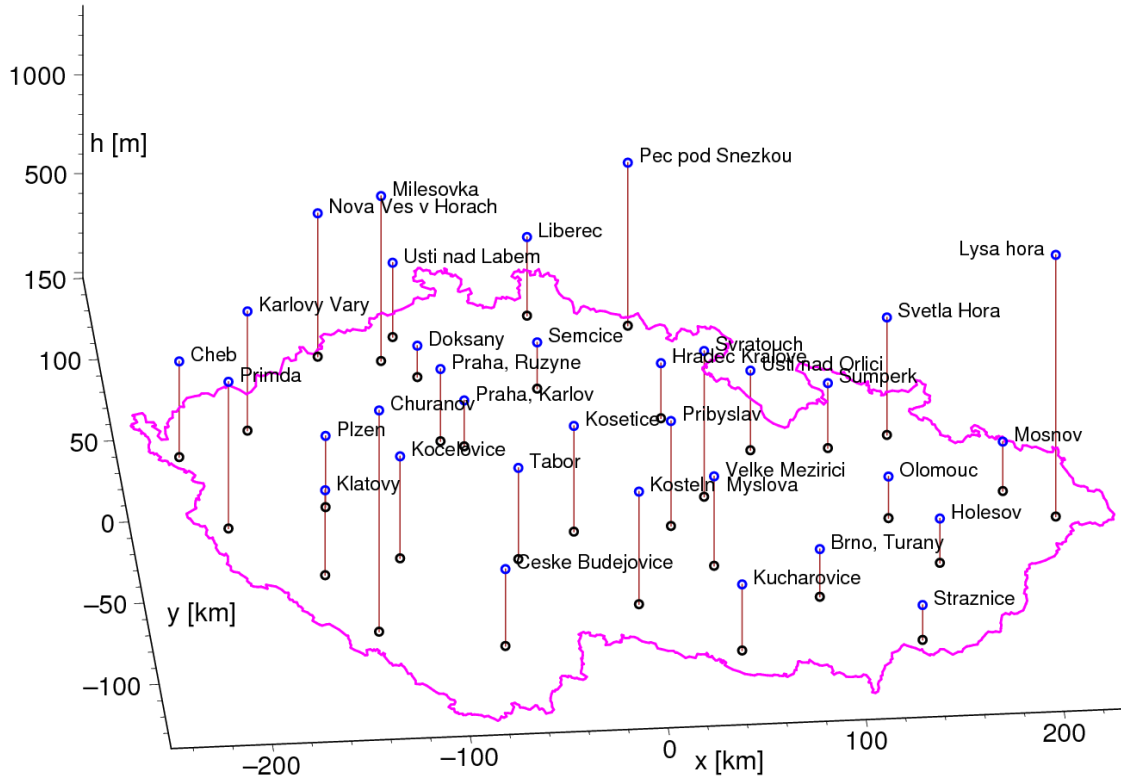


Figure 1: Positions of the meteorological stations

2.2 Regression function

As a regression function was used function $F(t,x,y,z)$, where $k = 2\pi$ see (1).

$$F(t, x, y, h) = c_1 x \cos(kt) + c_2 x \sin(kt) + c_3 x + c_4 xt + c_5 xt^2 + c_6 y \cos(kt) + c_7 y \sin(kt) + c_8 y + c_9 yt + c_{10} y^2 + c_{11} h \cos(kt) + c_{12} h \sin(kt) + c_{13} h + c_{14} ht + c_{15} ht^2 + c_{17} \cos(kt) + c_{18} \sin(kt) + c_{19} t + c_{20} t^2. \quad (1)$$

Coefficients $c_1 - c_{20}$ were computed by the least squares method to minimise the sum of squared residues on one degree of freedom – SQR_1 with a precision of 15 significant digits, see (2).

$$SQR_1 = \frac{\sum_{i=1}^N (F(t_i, x_i, y_i, h_i) - T_i)^2}{N - p}, \quad (2)$$

where N = number of measurements, t = time, $[x, y, h]$ = spatial coordinates, p = count of F function members, F = seeking regression function, T = measured temperatures.

```
> f:=expand((x+y+h+1)*(cos(k*t)+sin(k*t)+(1+t)^2)); ROW:=[op(f)];
> k:=evalf(2*Pi); N:=[];
> for i from 1 to Nj do;
>   teploty:=Data[i+3][5..-1];
>   Radek:=subs(h=Vyska[i],x=X[i],y=Y[i],ROW):
```

```

> Tt:=zip((u,v)->`if`(u<=-99,[v,u],NULL),Data[i+3][5..-
1],T);N:=[N[],nops(Tt)];
>
G[i]:=display({plot(Tt,style=point,symbol=circle,symbolsize=15,color=red),
plot(Tt,color=red)}): #display(%);
> M[i]:=map(u->evalf(subs(t=u[1],Radek)),Tt): V[i]:=map(u->u[2],Tt);
> Tau[i]:=map(u->u[1],Tt): print(Jmeno[i],nops(Tt));
> end do:
> A:=Matrix([seq(M[j][],j=1..i-1)]); B:=Vector([seq(V[j][],j=1..i-1)]);
> b:=convert(B,list): Nu:=nops(b): Sol:=LeastSquares(A,B);
> b2:=convert(evalm(A.Sol),list): Sol:=convert(Sol,list);
> FCE:=add(w,w=zip((u,v)->u*v,ROW,Sol));
FCE := 10.601 - 0.0010497 x cos(6.2832 t) - 0.0013391 x sin(6.2832 t) - 0.0020117 x
- 0.000095775 x t + 0.000018122 x t2 + 0.0015086 y cos(6.2832 t)
- 0.00044694 y sin(6.2832 t) - 0.0021058 y - 0.00039920 y t + 0.66412 10-7 y t2
+ 0.0012082 h cos(6.2832 t) - 0.00041306 h sin(6.2832 t) - 0.0053368 h
- 0.00013071 h t + 0.98070 10-5 h t2 - 10.499 cos(6.2832 t) - 2.4806 sin(6.2832 t)
+ 0.17162 t - 0.015358 t2

```

2.3 Verification

The accuracy of calculation was then verified with a precision of 36 digits in force according to another algorithm, based on the QR decomposition, see [6].

```

> Digits:=36;
> QR:=proc(M::Matrix,B::Vector) local n, r, q, k, j, p, R;
n:=ColumnDimension(M)+1; r[1,1]:=sqrt(add(u^2,u=Column(M,1)));
q[1]:=map(u->u/r[1,1],Column(M,1));
for k from 2 to n do;
if k<n then p:=Column(M,k); else p:=B; end if;
for j from 1 to k-1 do; r[j,k]:=add(w,w=zip((u,v)->u*v,q[j],p));
p:=p-q[j]*r[j,k];
end do;
r[k,k]:=sqrt(add(u^2,u=p)); q[k]:=map(u->u/r[k,k],p);
end do;
R:=Matrix(n,(i,j)->r[i,j],shape=triangular); p:=Column(R,n)[1..n-1];
R:=SubMatrix(R,1..n-1,1..n-1); R:=MatrixInverse(R); convert(R.p,list);
> end proc;
> Sol2:=QR(A,B); Sol-Sol2;

```

[-0.15455 10⁻¹⁶, 0.21017 10⁻¹⁸, -0.22928 10⁻¹⁷, -0.17196 10⁻¹⁸, 0.35846 10⁻¹⁹,
 -0.23839 10⁻¹⁶, 0.41674 10⁻¹⁷, 0.24270 10⁻¹⁷, 0.46773 10⁻¹⁸, 0.12575 10⁻¹⁹,
 -0.11477 10⁻¹⁶, -0.52563 10⁻¹⁸, 0.48513 10⁻¹⁷, -0.84839 10⁻¹⁸, 0.12777 10⁻¹⁸,
 0.21591 10⁻¹⁴, 0.81535 10⁻¹⁵, -0.53290 10⁻¹⁴, 0.58891 10⁻¹⁵, -0.87587 10⁻¹⁶]

Both calculations differed only within the selected numerical precision, which means that their relative difference was of the order of 10⁻¹³%. The coefficient of linear correlation for each station for the entire period 2003-2013 ranges from 0.961 to 0.975.

```
> LC0:=Correlation(b,b2); DT:=sqrt(add(u^2,u=b-b2)/Nu);
      LC0 := 0.970023869152986   DT := 1.85780197744508

> bbw:=CurveFitting[LeastSquares](b2,b,w,curve=c1*w+c0);

> Gb1:=plot(zip((u,v)->[u,v],b2,b),style=point,symbol=cross,
      symbolsize=4,color=red):

> Gb2:=plot([bbw,bbw+DT,bbw-DT],w=min(b2)..max(b2),color=blue,
      thickness=[3,1,1],linestyle=[1,2,2]):

> display({Gb1,Gb2},labels=["Teploty vypoctene [C]",
      "Teploty namerene [C]",labeldirections=[horizontal, vertical]);
```

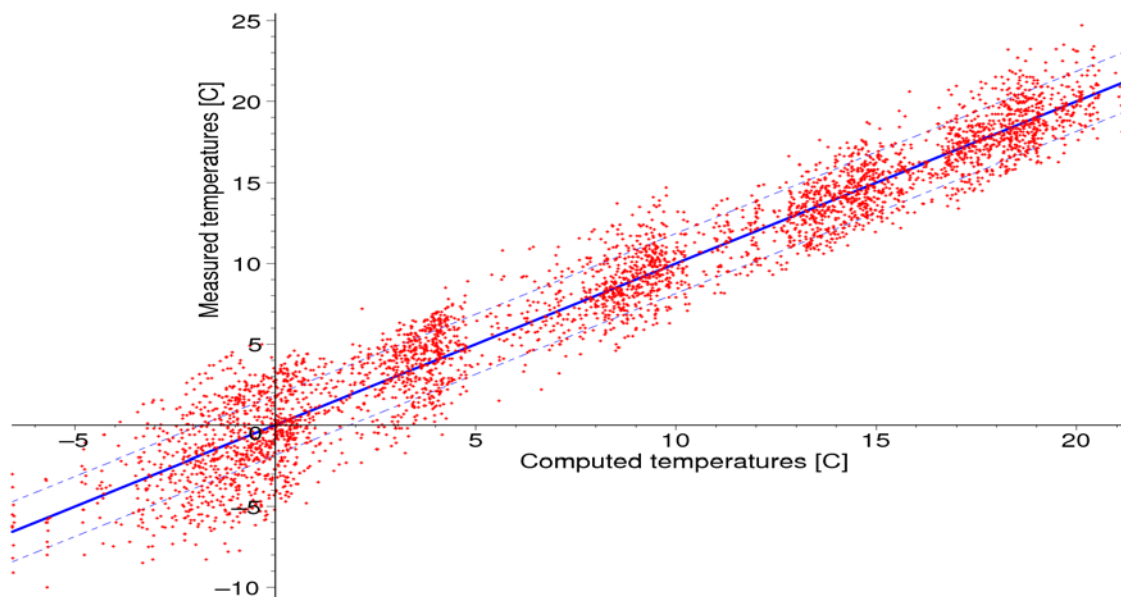


Figure 2: Comparison of measured and computed temperatures

```
> Cor:=[]; sigma:=[];

> for j from 1 to Nj do;
  RT[j]:=map(u->evalf(subs(x=X[j],y=Y[j],h=Vyska[j],t=u,FCE)),Tau[j]);
  Cor:=[Cor[],Correlation(RT[j],V[j])];
  sigma:=[sigma[],sqrt(add(u^2,u=RT[j]-V[j])/N[j])];
  g[j]:=plot(subs(x=X[j],y=Y[j],h=Vyska[j],[FCE-
sigma[j],FCE,FCE+sigma[j]]),
```

```

t=Tau[j][1]..Tau[j][-1],color=blue,thickness=[1,3,1],linestyle=[2,1,2]):
end do:
> MinC,MaxC:=min(Cor),max(Cor); MinS,MaxS:=min(sigma),max(sigma);
      MinC, MaxC := 0.961886745128666, 0.975726002082668
      MinS, MaxS := 1.71579406850540, 2.11933762875770
> Cmin:=seq(`if`(Cor[j]=MinC,j,NULL),j=1..i-1);
> display({G[Cmin],g[Cmin]},title=cat(Jmeno[Cmin]," : Lineární Korelace =
",
      convert(evalf(Cor[Cmin],5),string),", smer. odchylka =
      ",convert(evalf(sigma[Cmin],5),string)), labels=["t [roky]","T [C]"],
      labeldirections=[horizontal, vertical]);

```

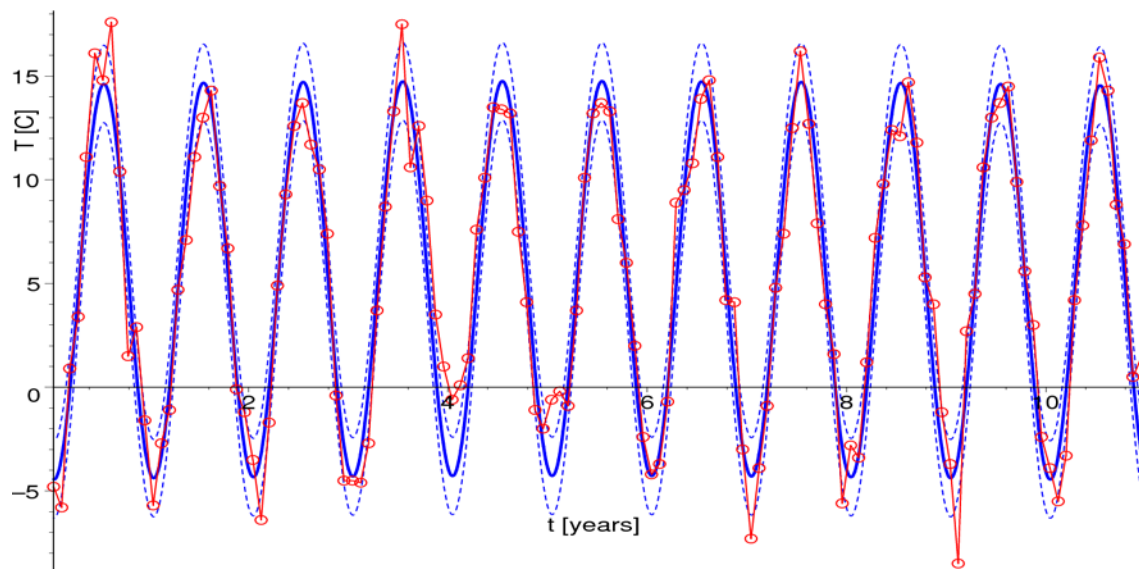


Figure 3: Graphical comparison of measured and computed temperatures in Churanov – the worst linear correlation

The average correlation coefficient of linear spatial temperature distribution in the Czech Republic is 0.931. This means that the regression function can be considered as satisfactory.

```

> Nr:=nops(Data); Nt:=nops(T); Cort:=[ ];
> for i from 1 to Nt do;
      Tr:=[seq(Data[j][i+4],j=4..Nr)]; FCEj:=evalf(subs(t=T[i],FCE));
      TP:=[seq(subs(x=X[j],y=Y[j],h=Vyska[j],FCEj),j=1..Nj)];
      TP:=zip((u,v)->`if`(u>-99,v,NULL),Tr,TP);
      Tr:=remove(has,Tr,-99.0); Cort:=[Cort[],Correlation(Tr,TP)];
> end do;
> MinC,MaxC:=min(Cort),max(Cort);
> Imin:=seq(`if`(Cort[j]=MinC,j,NULL),j=1..i-1);
> Imax:=seq(`if`(Cort[j]=MaxC,j,NULL),j=1..i-1);
> CorP:=add(u,u=Cort)/Nt;

```

```

> CorS:=sqrt(add(w^2,w=map(u->u-CorP,Cort))/Nt);
> Gc1:=plot(zip((u,v)->[u,v],T,Cort),labels=["t [roky]","Celkovy
    koeficient linearni korelace"],labeldirections=[horizontal,vertical]):
> Gc2:=plot([[T[Imin],Cort[Imin]]],style=point,symbol=circle,color=black):
Gc3:=plot([[T[Imax],Cort[Imax]]],style=point,symbol=box,color=black):
> Gc4:=textplot([[T[Imin],Cort[Imin],cat("
Minimum,t=",convert(evalf(T[Imin],3),
    string),"",Korelace=0",convert(evalf(Cort[Imin],3),string))]],
    color=black,align=right):
Gc5:=textplot([[T[Imax],Cort[Imax],cat("  Maximum,
t=",convert(evalf(T[Imax],3),
    string),"",
Korelace=0",convert(evalf(Cort[Imax],3),string))]],color=black):
> Gc6:=plot([CorP,CorP+CorS,CorP-CorS],t=T[1]..T[-1],
    color=blue,linestyle=[1,2,2]):
> Gc7:=textplot([1.4,CorP,cat("Prumer=0",convert(evalf(CorP,3),string))],
    color=blue,align=below):
> display({Gc1,Gc2,Gc3,Gc4,Gc5,Gc6,Gc7});

```

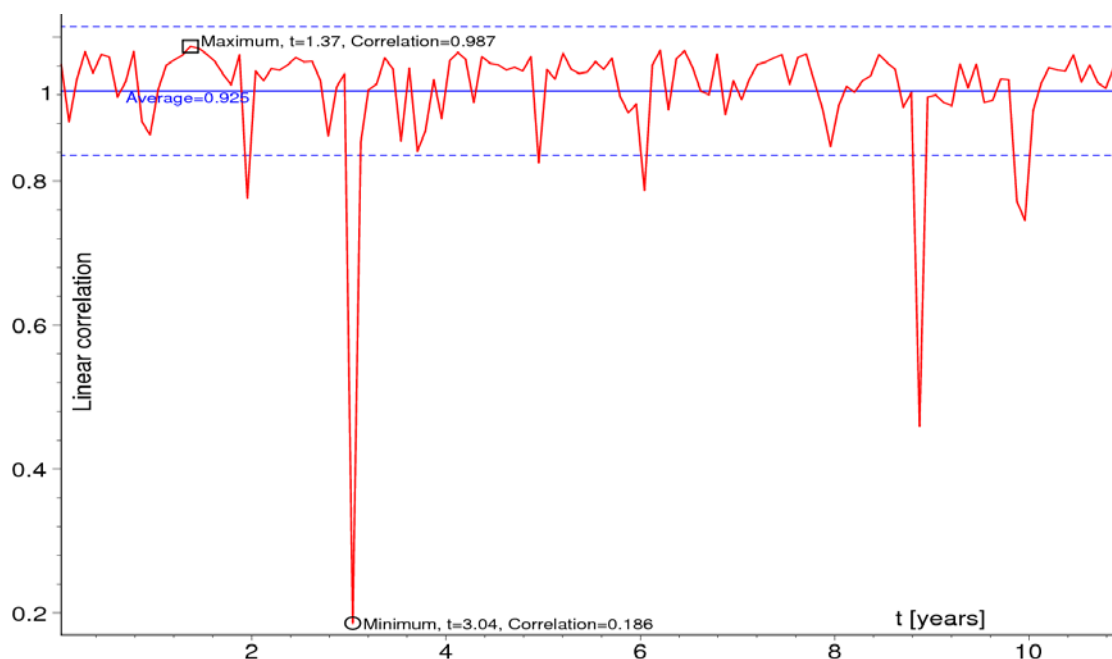


Figure 4: Time development of the spatial correlation between measured and computed temperatures

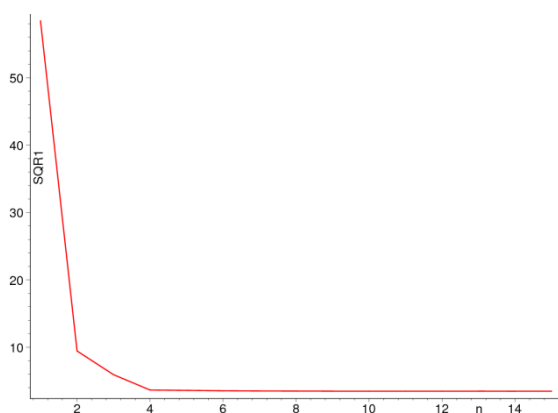
2.4 Determining of individual members of the regression function

For each member of the regression function (1) containing 20 members is calculated the sum of squared residuals per 1 degree of freedom, further shorten as **SQR1**. Value which is selected is the

one with the lowest **SQR1** value. This selected member is searched from the remaining 19 members, another member such that **SQR1** for two members was the lowest. It is being continued for each subsequent member by this manner until they are exhausted all the members or **SQR1** begins to rise. In this way, we find that members $2xt$, $y \sin(kt)$, yt^2 , $2t$, t^2 are meaningless.

```
> ROWF:=[]; LQ:=[]; LC:=[]; Nu:=nops(b); n:=nops(ROW0);
> sqr:=infinity; nu:=1; N:='N'; Go:=true;
> while Go do;
  Go:=false;
  for i from 1 to n do;
    A3:='if'(ColumnDimension(A00)=0,Matrix(Column(A0,i)),
      Matrix([A00[],Column(A0,i)]));
    sol3:=LeastSquares(A3,B); b3:=convert(evalm(A3.sol3),list);
    lc:=Correlation(b,b3); sq:=add(w^2,w=b-b3)/(Nu-nu);
    if sq<sqr then N:=i; sqr:=sq; lcr:=lc; SOL:=convert(sol3,list);
      Go:=true; end if;
  end do;
  if Go then
    LQ:=[LQ[],sqr]; LC:=[LC[],lc]; ROWF:=[ROWF[],ROW0[N]];
    F3:=add(w,w=zip((u,v)->u*v,ROWF,SOL));
    A00:='if'(ColumnDimension(A00)=0,Matrix(Column(A0,N)),
      Matrix([A00[],Column(A0,N)]));
    A0:=DeleteColumn(A0,N); ROW0:=subsop(N=NULL,ROW0);
    nu:=nu+1; n:=nops(ROW0); N:='N';
  end if;
end do;
```

Graph showing **SQR1** as a function of the number of members is shown below, see Fig. 1. From this graph it is clear that with the increasing number of sum of squared residuals on 1 degree of freedom to the 15th member decreases. Graph showing **SQR1** for two consecutive functions that differ from each other about member is for 5th to 15th member shown below, see Fig. 5.



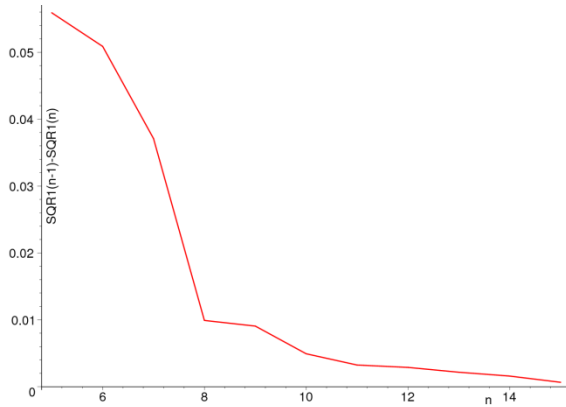


Figure 5: Decreasing of the sum of squared residuals per degree of freedom and detail for last eight terms of the function (5)

Fischer-Snedecor function $F(z)$, (3), see [3], [8], is used for testing and determining the significance of members of the regression function with more coefficients compared to simpler regression function. Level of the uncertainty is α .

$\int_0^q F(z) dz = 1 - \alpha$, where:

$$F(z) = \frac{\left(\frac{\kappa}{n}\right)^{\left(\frac{\kappa}{2}\right)} z^{\left(\frac{\kappa-1}{2}\right)}}{B\left(\frac{\kappa}{2}, \frac{n}{2}\right) \left(1 + \frac{z \kappa}{n}\right)^{\left(\frac{\kappa+n}{2}\right)}}, \quad \begin{array}{l} \kappa = \text{difference of count of parameters} \\ \text{of the functions} \\ n = \text{count of measurements} \end{array} \quad (3)$$

Fig. 1. This feature, with accuracy $1-\alpha$, tells us how to change the statistical significance of the function, if we add more function members. More complex function is statistically significant if it satisfies the condition (4). Selection of the most suitable model is performed on the basis of a test which is based on the inequality (4), see tables XVIII 4a-4c in [4].

$$\frac{S_R(1) - S_R(2)}{\frac{p_2 - p_1}{S_R(2)}} \geq q, \quad \text{where:} \quad \begin{array}{l} S_R(1) = \text{residual sum of squares of a simple model} \\ S_R(2) = \text{residual sum of squares of a complex model} \\ p_1 = \text{number of coefficients of a simple model} \\ p_2 = \text{number of coefficients of a complex model} \\ p_2 - p_1 = \kappa \equiv l \end{array} \quad (4)$$

Fig. 2. Test to determine whether a more complex model (multivariable) is better than the simpler model is done with respect to the equation (4). The more complex function is omitted if $F > F_{1-\alpha}(p_2-p_1, n-p_2)$. The number of operands of functions that correspond to the reliability of $1-\alpha = 90\%$, $1-\alpha = 95\%$ and $1-\alpha = 99\%$, is shown in Fig. 6.

```
> Fkznz:=unapply(1/Beta('k'/2,'n'/2)*('k'/'n')^('k'/2)*'z'^((('k'-2)/2)
/ (1+'z'*'k'/'n')^((('k'+n')/2),'k','n','z'));
> LF:=[seq((LQ[i]*(Nu-i)-LQ[i+1]*(Nu-i-1))/LQ[i+1],i=1..nops(LQ)-1)];
> L99:=fsolve(Int(evalf(Fkznz(1,Nu-1,q)),q=0..u)=0.99,u);
> L95:=fsolve(Int(evalf(Fkznz(1,Nu-1,q)),q=0..u)=0.95,u);
> L90:=fsolve(Int(evalf(Fkznz(1,Nu-1,q)),q=0..u)=0.90,u);
> nu1:=8; nu2:=14;
```

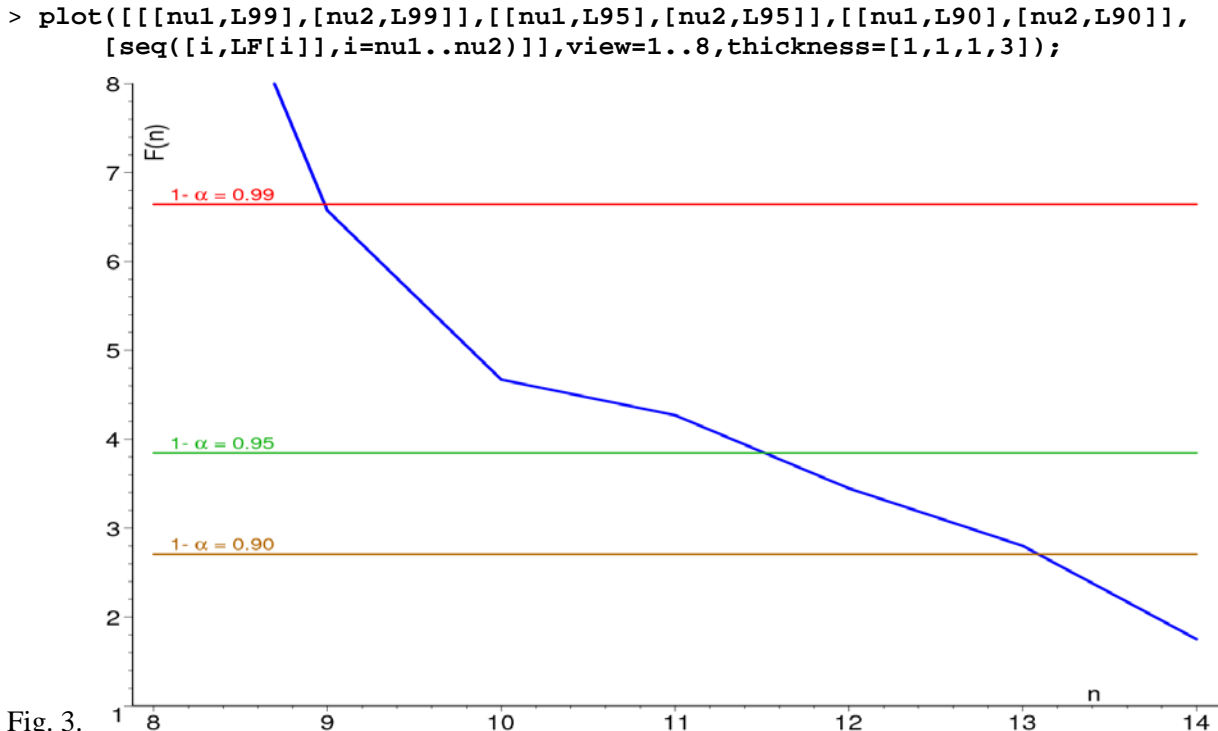



Fig. 3.

Figure 6: Graph showing the number of members and reliability values

2.5 Non-periodic functions

Fig. 4. Periodic components $\sin(kt)$ and $\cos(kt)$, which are removed of these functions, are appropriate to the individual reliability. Then a derivation function by time is done, see (5).

```
> U99:=LF[8]; V99:=evalf(Int(evalf(Fkznz(1,Nu-9,q)),q=0..U99));
> U95:=LF[11]; V95:=evalf(Int(evalf(Fkznz(1,Nu-12,q)),q=0..U95));
> U90:=LF[13]; V90:=evalf(Int(evalf(Fkznz(1,Nu-13,q)),q=0..U90));
> F99:=add(w,w=zip((u,v)->u*v,ROWF[1..9],SOL[1..9]));
> F95:=add(w,w=zip((u,v)->u*v,ROWF[1..11],SOL[1..11]));
> F90:=add(w,w=zip((u,v)->u*v,ROWF[1..13],SOL[1..13]));
> f99:=remove(has,F99,[sin,cos]);
> f95:=remove(has,F95,[sin,cos]);
> f90:=remove(has,F90,[sin,cos]);
```

$$f_{99} = 10.3200 - 0.0059h - 0.0004yt - 0.0022x$$

Fig. 5.

$$f_{95} = 10.3200 - 0.0059h - 0.0004yt - 0.0022x - 0.0002ht^2$$

Fig. 6. (5)

$$f_{90} = 10.3200 - 0.0059h - 0.0004yt - 0.0022x - 0.0002ht^2 + 0.0001ht$$

Fig. 7. These derivatives are set equal to zero, see (6). Furthermore, the coordinates y are calculated and the exact positions of 3 boundaries for a 5-year intervals are found.

```
> f99t:=diff(f99,t); Y99:=solve(f99t=0,y):
> f95t:=diff(f95,t); Y95:=solve(f95t=0,y):
```

```
> f90t:=diff(f90,t); Y90:=solve(f90t=0,y):
```

$$\hat{f}_{99} = -0.0004 y t$$

Fig. 8. $\hat{f}_{95} = -0.0004 y t - 0.0022 x - 0.0004 h$

Fig. 9. (6)

$$\hat{f}_{90} = -0.0004 y t - 0.0022 x - 0.0004 h t + 0.0001 h$$

3 Results and Discussion

Fig. 10. The functions corresponding reliability $1-\alpha = 0.99$ passes through the center of gravity of the Czech Republic. This fact shows the graph in Fig. 7 – positions on the map of the Czech Republic and Fig. 8 shows average y position of the boundaries in seven five-years intervals. Maple computation is very similar to the steps mentioned on page 3 and that is why it is not presented here.

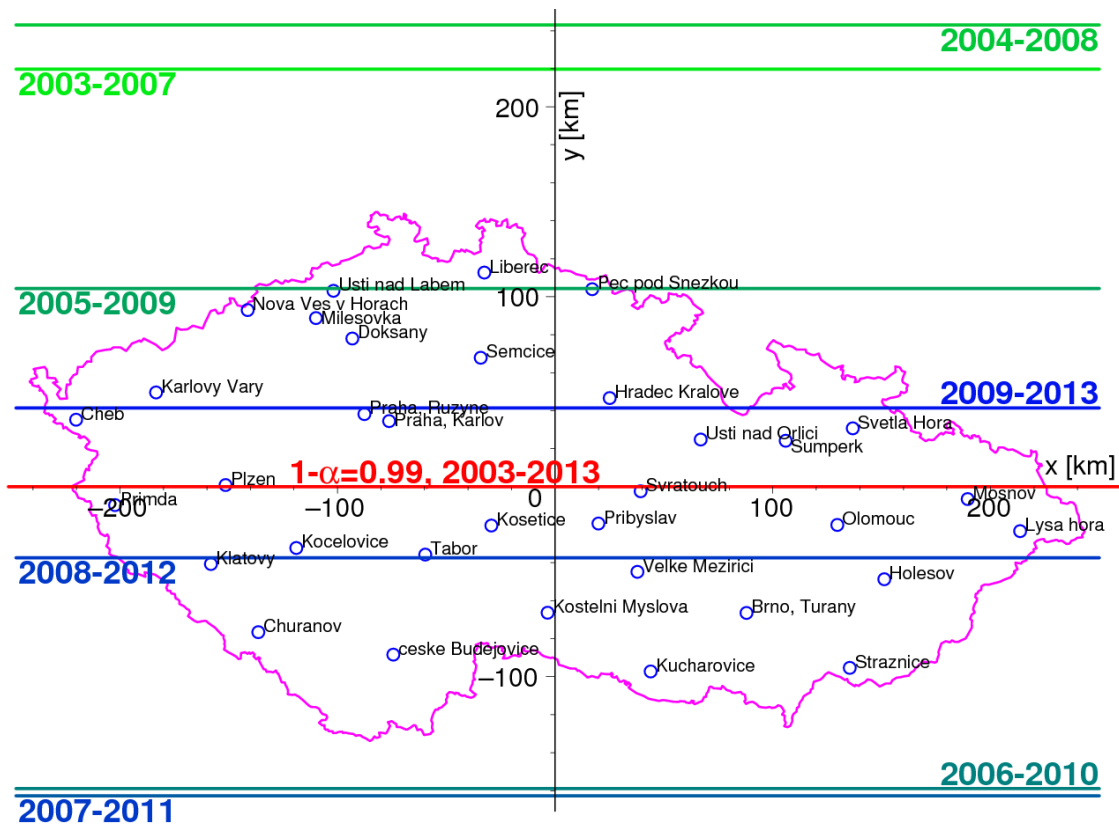


Fig. 11. 2007-2011

Figure 7: Average cooling/warming border position in 5 year intervals in the Czech Republic

From the calculations mentioned above it is clear that function with 9 members explains 99% of the data, the function with 13 members explains 95% of the data and with 14 members explains 90% of the measured data. The boundaries corresponding to the 95% and 99% reliability are plotted in the Fig. 7.

In our case, we work with 95% reliability, for which the corresponding function contains 14 members. This reliability is designed for more complex models.

The boundary, corresponding to 99%, is stable and passes through the center of gravity of the Czech Republic. The functions corresponding to 99% reliability contain 9 members and is time independent.

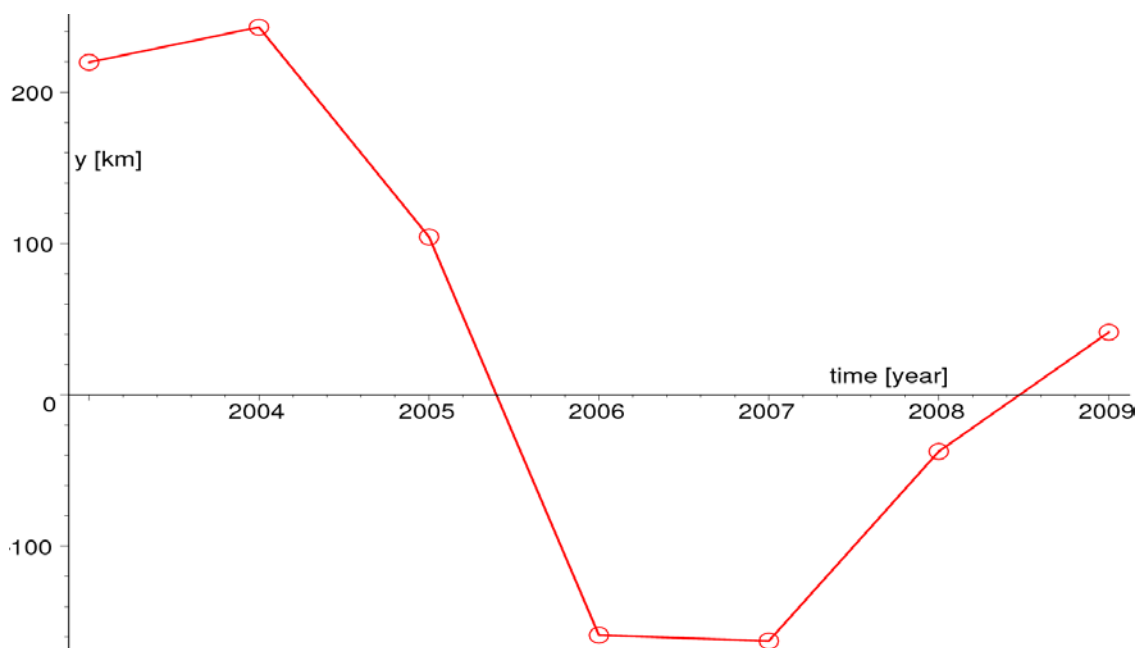


Figure 8: Y coordinate of cooling/warming border as a function of time

4 Conclusion

There is a border between areas where the warming and cooling are occurring in the area of the Czech Republic. If we use a simpler function, then a position of this boundary has coordinates $y = 0$ (the border passes through the center of gravity of the Czech Republic) and it is not dependent on time. This feature is able to explain 99% of the measured data. From the properties of the more complex function, it follows that the position of the boundary is moved over the whole area of the Czech Republic, see Fig. 4 and Fig. 5.

Acknowledgement 1

The authors want to express their thanks to Mrs. Pýchová of the Czech Hydrometeorological Institute for information about publicly accessible meteorological data.

Acknowledgement 2

The research has been supported by the project TP 4/2014 “Analysis of degradation processes of modern materials used in agricultural technology“, financed by IGA AF MENDELÚ.

5 References

- [1] WEB1, http://www.czso.cz/csu/2012edicniplan.nsf/kapitola/0001-12-r_2012-0200
- [2] WEB2, http://portal.chmi.cz/portal/dt?action=content&provider=JSPTabContainer&menu=JSPTabContainer/P4_Historicka_data/P4_1_Pocasi/P4_1_9_Mesicni_data&nc=1&portal_lang=cs#PP_Mesicni_data
- [3] WEB3, <https://ucnk.ff.cuni.cz/bonito/nahoda.php>
- [4] Anděl, J.: Matematická statistika. SNTL, Praha, 1985, pgs. 327-329
- [5] Barros, V., 2006, Globální změna klimatu. Praha, 165 s., ISBN: 80-204-1356-1

- [6] Gander W., Hřebíček J., Solving problems in Scientific Computing Using Maple and MATLAB. Springer-Verlag Berlin Heidelberg 1993, 1995. Germany, 315 p. , ISBN: 3-540- 58746-2
- [7] Klaus, V., 2009, Modrá, nikoli zelená planeta., Dokořán. Praha, 164 s., ISBN 978-80-7363-243-4
- [8] Meloun, M., Militký, J.: Kompendium statistického zpracování dat., Academia, Praha, 2006, pg. 677, ISBN 80-200-1396-2
- [9] MAPLE, User Manual, 1st edition, Maplesoft, 2007, 396 p., ISBN 978-1-897310-20-3
- [10] Potter Thomas D., Colman Bradley R., 2005: Handbook of weather, climate, and water: dynamics, climate, physical meteorology, weather systems, and measurements., Hoboken, NJ, USA, 2003, 973 p., ISBN: 9780471721604, 9780471214908

Forecasting Financial Volatility: Application of Customized Neural Network Combined with ARCH Model into Time Series Modelling

Lukas Falat

The Faculty of Management Science and Informatics, University of Zilina
Univerzitná 8215/1, 010 26 Zilina, Slovakia
lukas.falat@fri.uniza.sk

Dusan Marcek

VŠB-TU Ostrava, Department of Applied Informatics
Sokolská třída 33, 701 21 Ostrava 1
Dusan.marcek@vsb.cz

Abstract

In this paper, we investigate the volatility dynamics of EUR/GBP currency using neural network approach. We suggest the ARCH-RBF model that combines information from ARCH statistical model with RBF neural network. We also use a large number of statistical models as well as different optimization techniques for RBF network such as genetic algorithms or clustering. Both in-sample and out-of-sample forecasts are evaluated using appropriate evaluation measures.

Keywords

time series, forecasting, ARCH, EUR/GBP, currency, finance, volatility, artificial neural network, RBF

Abstrakt

V tomto článku jsme zkoumali dynamiku volatility měny EUR/GBP pomocí neuronových sítí. Navrhujeme ARCH-RBF model, který kombinuje informace z ARCH statistického modelu s RBF neuronové sítě. Používáme také velké množství statistických modelů, stejně jako různých optimalizačních technik pro RBF sítě například genetické algoritmy nebo clustering. Oba vzorky prognózy jsou vyhodnoceny pomocí vhodných hodnotících měr.

Klíčová slova

časové řady, prognózy, ARCH, EUR/GBP, měna, finance, volatilita, umělé neuronové sítě, RBF

1 Introduction

Volatility is an extremely important factor for risk management, for asset allocation, and for taking bets on future volatility. A large part of risk management is measuring the potential future losses of a portfolio of assets (volatility modelling provides a simple approach to calculating value at risk of a financial position in risk management), and in order to measure these potential losses, estimates must be made of future volatilities and correlations. In asset allocation, the Markowitz approach of minimizing risk for a given level of expected returns (see Ref. 1) has become a standard approach, and of course an estimate of the variance-covariance matrix is required to measure risk. Perhaps the most challenging application of volatility forecasting, however, is to use it for developing a volatility trading strategy. Option traders often develop their own forecast of volatility, and based on this

forecast they compare their estimate for the value of an option with the market price of that option. The simplest approach to estimating volatility is to use historical standard deviation, but there is some empirical evidence, which we will discuss later, that this can be improved upon.

Various approaches to volatility modelling have been suggested in the econometric and financial literature. In the following we will provide a brief overview of developments in the literature starting with the autoregressive conditional heteroskedasticity (ARCH) models [2]. Bollerslev [3] introduced the generalized ARCH (so called GARCH) model. Later, as time went on, many extensions of the GARCH model have been introduced in the literature since: e.g. GARCH-in-mean (GARCH-M) models [4], EGARCH models [5], Threshold ARCH (TARCH) and Threshold GARCH (TGARCH) [6] and Power Arch (PARCH) models [7] just to name a few. A number of studies have focused on optimal model specification and the performance of various GARCH models in financial markets providing no clear-cut results (such as [8]).

Also, as computer science has developed, techniques of machine learning started to apply in the domain of financial forecasting. Gooijer and Hyndman [9] proved that artificial neural networks had the biggest potential in time series forecasting. Therefore, various types of neural networks have been used for forecasting future values of high frequency financial data such as [10] or [11].

This study examines various models that can be used in forecasting volatility, to evaluate their respective performance. One of the main reasons for finding the appropriate volatility model is that volatility, as a representation of risk, plays an important role in an investor's decision making process. Volatility is not only of great concern for investors but also policy makers and regulators who are interested in the effect of volatility on the stability of financial markets in particular and the whole economy in general. Finally, volatility estimation is an essential input in many VaR (Value at Risk) models, as well as for a number of applications in a firms market risk management practices.

This paper concerns with forecasting volatility and it is divided into eight chapters. Chapter two presents the statistical ARCH/GARCH models used for volatility forecasting. In chapter three, data we use for our tests are presented. In chapter four we perform the GARCH volatility modelling on our tested data and in chapter five we present the neural network approach as well as our ANN model for volatility forecasting. In chapter six the results are presented and discussed.

2 Methods and Models

2.1 ARIMA & ARCH

The major breakthrough in the history of statistical modelling came with publishing a study from Box & Jenkins [17]. In this study authors integrated all the knowledge including autoregressive and moving average models into one book. From that time the ARIMA (AutoRegressive Integrated Moving Averages) models have been very popular in time series modelling for a long time as O'Donovan [18] showed that these models provided better results than other models used in that time. It is therefore no surprise that for more than 20 years Box-Jenkins ARMA models have been widely used for time series modelling. The models published in [17] are autoregressive models (AR) and moving average (MA) models. Let y_t be a stationary time series that is a realization of a stochastic process. Then, general formula of ARMA(p,q) model can be expressed as follows

$$y_t = \xi + \sum_{i=1}^p \phi_i y_{t-i} + \varepsilon_t - \sum_{j=1}^q \theta_j \varepsilon_{t-j} \quad (1)$$

where ξ is a constant, (ϕ_1, ϕ_2, \dots) are autoregressive parameters, $(\varepsilon_t, \varepsilon_{t-1}, \varepsilon_{t-2}, \dots)$ are independent random parts. If the model is correct, residuals are to form the white noise process. The model is composed of two parts –autoregressive (deterministic) part expressing the linear dependence on previous values of the dependent variable y_t ; and stochastic part represented by moving averages. If the series is not stationary, ARIMA models must be used. Let y_t be a time series and let d be the

order of differentiation; y_t will be called ARIMA(p,d,q) process if its d^{th} differences produce ARMA(p,q) process. ARIMA can be formally defined as

$$\Phi(B)(1-B)^d y_t = \mu + \Theta(B)\varepsilon_t \quad (2)$$

It is also obvious that if d equals zero, ARIMA equals just simple ARMA process. The whole process of Box-Jenkins statistical modelling, which is performed through Box-Jenkins analysis, has more steps and it is described in details in [3].

Financial volatility

The most common way in expressing the risk is the volatility. Financial volatility, which is present in dynamic economic markets like stock market or forex market and which plays an important role in financial forecasting as well as financial risk analysis, has some very unique features. First of all, it is its stochastic character. Moreover, financial time series exhibit a characteristic known as volatility clustering in which large changes tend to follow large changes, and small changes tend to follow small changes. Volatility is hence clustered in time and therefore it has persistence character. Resulting from this, actual variance is dependent on the previous variances and the time series is characterized by the time-variant conditional variance, also called clustering of variances.

Another feature of financial volatility is mean reversion. Volatility is often persistent and so has a long memory. In the long term period the volatility oscillates around its long-term mean which results in the fact that all long-term forecasts are to converge to its long-term mean value. So even though financial time series can exhibit excessive volatility sometimes, volatility will finally settle down to a long run level.

It has been also experimentally proved that the distribution of many high frequency financial time series usually have fatter tails than a Gaussian distribution. A phenomenon of fatter tails is also called as excess kurtosis.

The weakness of ARIMA models in modeling financial time series is the inability to model stochastic non-constant volatility having the features we described above. In [2] Engle suggested the solution by creating so called ARCH (Autoregressive Conditional Heteroskedastic) models which assume heteroskedastic variance of ε_t . Let y_t be a standard stationary AR(p)

process defined as in (1) and let a_t be a random part of this model and hence, is a white noise process and has a constant unconditional variance. Let also assume that $|\phi_i| < 1$ for $i = 1, 2, \dots, p$.

According to Engle (see [2]), the model will become more confident and the predictions will be more precise if it is dependent on a conditional variance of ε_t

$$y_t = \sum_{i=1}^p \phi_i y_{t-i} + \varepsilon_t \quad (3)$$

where the expected value of ε_t is zero and can be transformed into the form

$$\varepsilon_t = e_t \sqrt{h_t} \quad (4)$$

where ε_t is the residual part of the model having mean value equalled to zero, e_t is a white noise process $\sim N(0,1)$ and h_t is a function of conditional heteroskedastic variance of random part defined as follows

$$h_t = \alpha_0 + \sum_{j=1}^p \alpha_j \varepsilon_{t-j}^2 \quad (5)$$

According to [2], the standard ARCH model of p order is defined by equations (4) and (5) and should be used for conditional variance modeling. The conditional variance in the ARCH(p) model is a function of the past squares of random variable e_t (which can be understood as an arrival of new information in particular time moments). The ARCH model described above is able to model the

basic properties of financial volatility such as volatility clustering, stochastic properties of volatility, mean reversion, fat tails etc.

Bollershev [3] suggested the generalized form of ARCH model called GARCH (Generalized Autoregressive Heteroskedastic Models) where conditional variance of h_t depends on the previous conditional variances. The general GARCH(p,q) model can be formally defined as

$$\varepsilon_t = e_t \sqrt{h_t} \quad (6)$$

$$h_t = \alpha_0 + \sum_{j=1}^q \alpha_j \varepsilon_{t-j}^2 + \sum_{i=1}^p \beta_i h_{t-i} \quad (7)$$

where $\{\varepsilon_t\}$ is a sequence of error parts, $\{e_t\}$ is a white noise process and h_t is a function of conditional variance. It is also necessary that $\alpha_j > 0, j = 1, 2, \dots, q$ and $\beta_i \geq 0$ for $i = 1, 2, \dots, p$.

There exists a number of GARCH extensions, each of them are used to model some unusual property of volatility. EGARCH created by Nelson [5] is an implementation of leverage effects. As asymmetric influence of new information is another feature of financial volatility. EGARCH is able to model this feature of volatility. The leverage effect implemented in EGARCH expresses the asymmetric impact of positive and negative changes in financial time series. It means that the negative shocks in price influence the volatility differently than the positive shocks at the same size. This effect appears as a form of negative correlation between the changes in prices and the changes in volatility. EGARCH models leverage effects in the form

$$\log h_t = \alpha_0 + \sum_{i=1}^p \alpha_i \frac{|\varepsilon_{t-i}| + \gamma_i \varepsilon_{t-i}}{\sigma_{t-i}} + \sum_{j=1}^q \beta_j h_{t-j} \quad (8)$$

The leverage effect is present as follows: if ε_t is positive (there is "good news"), the total effect of ε_{t-i} is $(1 + \gamma_i) \varepsilon_{t-i}$. However, if ε_{t-i} is negative (there is so called "bad news"), the total effect of ε_{t-i} is $(1 - \gamma_i) |\varepsilon_{t-i}|$. Resulting from this, in EGARCH model bad news usually have larger impact on the volatility. (value of would be expected to be negative). For details see [19].

The basic GARCH model can be also extended to allow for leverage effects. This is performed by treating the basic GARCH model as a special case of the power GARCH (PGARCH) model proposed by Ding, Granger and Engle (see [7]):

$$\sigma_t^d = \alpha_0 + \sum_{i=1}^p \alpha_i (|\varepsilon_{t-i}| + \gamma_i \varepsilon_{t-i})^d + \sum_{j=1}^q \beta_j \sigma_{t-j}^d \quad (9)$$

where d is a positive exponent.

Before (G)ARCH modeling, one has to find out if heteroskedasticity is really present in series. According to Engle [2], the presence of heteroskedasticity is tested by ARCH test which supposes a non-existence of ARCH. This tests uses the Lagrange Multiplier (LM) statistics [19].

3 Experiment

3.1 Statistical Modelling

This paper focuses on financial time series of daily close prices of EUR/GBP exchange rate. The data we used, covered the historical period from October 31, 2003 to October 31, 2013 ($n = 2610$ daily observations).^{*} The graphical characteristics of the series is illustrated in Figure 1. Due to validation of our models, data were divided into two parts. The first part included 1306 observations (from 10/31/2008 to 10/31/2008) and was used for training (quantification) of our models. The second part of data (11/1/2008 to 10/31/2013), counting 1304 observations, was used for model validation by making one-day-ahead ex-post forecast. These observations included new data which had not been

^{*} The data was downloaded from the website <http://www.global-view.com/forex-trading-tools/forex-history>.

incorporated into model estimation. We used so many data in the validation phase in order to guarantee the validation robustness of our models. The reason for validation was to find out the real prediction power of the models; there was an assumption that if the model could handle to predict data from ex-post set, it would be able to predict values of a currency pair in the real future.

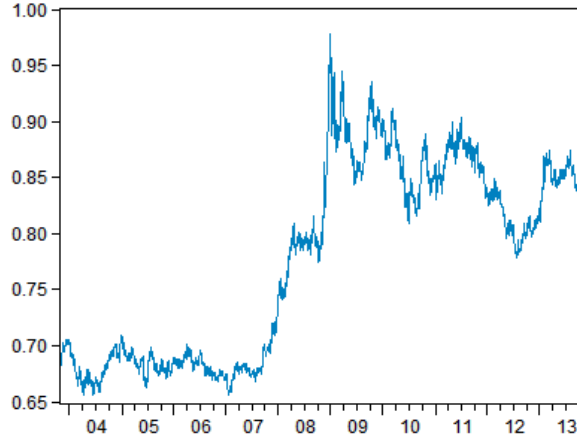


Figure 1: Time Series of daily close prices of EUR/GBP currency (October, 2003 – October, 2013).

Finally, in order to evaluate the characteristics for quantified model as well as to compare the real forecasting performance of our proposed models, the numerical characteristic for assessing models called Mean Squared Error (MSE) was used.

$$3 \quad MSE = H^{-1} \sum_{h=1}^H \left(\hat{Y}_{n-H+h} - Y_{n-H+h} \right)^2 \quad (10)$$

where h is the forecasting horizon, H is the total number of predictions for the horizon h over the forecast period, \hat{Y} is the estimated value and Y is the original value of the series.

The empirical statistical analysis, which was performed according to Box-Jenkins [17], focused on the original and differentiated series of daily observations of EUR/GBP currency pair covering a historical period from October 31, 2003 to October 31, 2008. Figure 2 and Figure 3 illustrates the original series as well as differences of the training set respectively. As stated in the previous section, we only used observations from training set for statistical modeling. Statistical modelling was performed in the Eviews software.

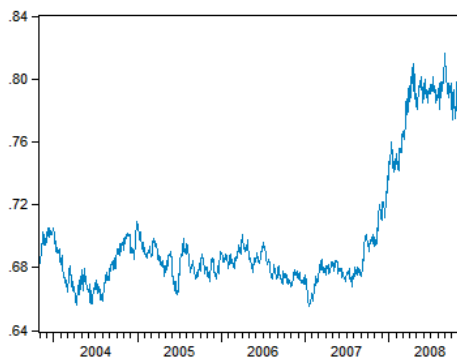


Figure 2: EUR/GBP – training set (original)

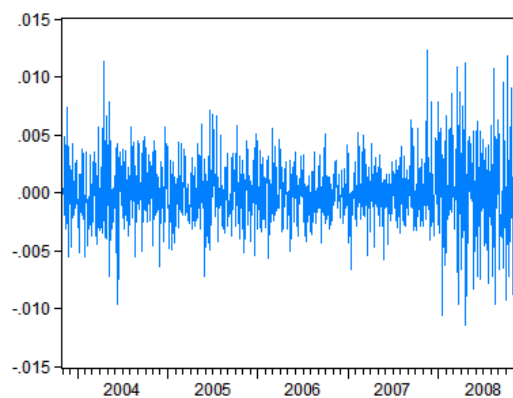


Figure 3: Differences of EUR/GBP – training set

Unit root tests results (see [21][22][23][24]) presented in the Table 4 (Appendix A) showed that this series was not stationary. In order to stationarize the series, it was differentiated. As seen from the Table 4, unit root tests confirmed that the differentiated series became stationary which had been a necessary condition in Box Jenkins modelling. By analyzing autocorrelation (ACF) and partial autocorrelation functions (PACF) of the differentiated series of EUR/GBP (see Table 5, Appendix A), there were no significant correlation coefficients (on $\alpha = 0.05$). Due to that we supposed that first differences of the original series formed a white noise process. In that case, the original series would have formed random walk process (RWP) as RWP is $I(1)$ process. Assuming the differences of the original series formed a white noise process, we selected $AR(0)$ as the basic Box-Jenkins model. Ljung-Box Q-statistics (see Table 5, right side) confirmed this assumption and the applicability of $AR(0)$ process as the correlations were statistically not significant. However, the assumption of normality of residuals of $AR(0)$ was rejected at 0.05 significance level (see Table 5, Appendix A). The observed asymmetry might have indicated the presence of nonlinearities in the evolution process of residuals. This nonlinearity was also confirmed by graphical quantiles comparison (see Figure 4) and a scatter plot of the series which did not appear to be in the form of a regular ellipsoid (see Figure 5). In addition, BDS test rejected the random walk hypothesis (see Table 7 Appendix A) as the BDS statistic was greater than critical value at 0.05 level.

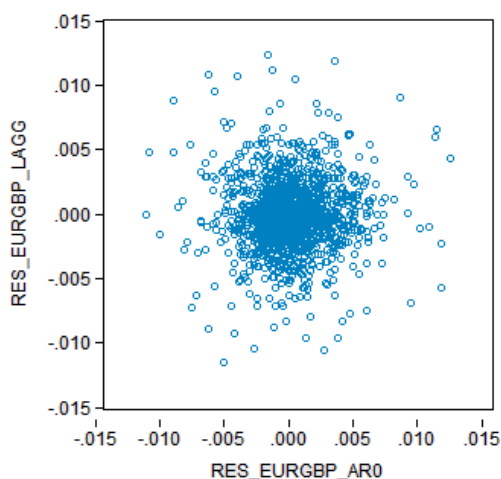


Figure 4: Scatter plot of EUR/GBP residuals variations.

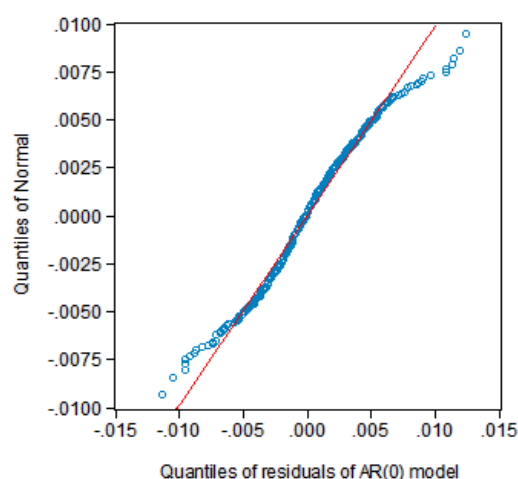


Figure 5: Quantiles of EUR/GBP residuals vs theoretical

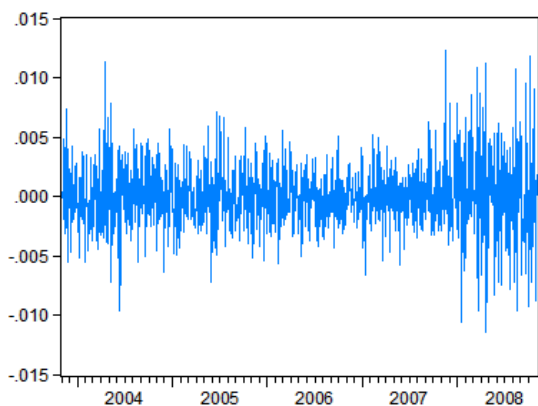


Fig. 6. Evolution of residuals of $AR(0)$ model.

Therefore, other tests had to be performed in order to correctly model this series. We noted that the residuals of AR(0) (see Figure 6) were not characterized by a Gaussian distribution (see Table 6, Appendix A). The asymmetry might have indicated non-linearities in the residuals. When looking at the graph of residuals (see Figure 6), one could observe the variability of these residuals could have been caused by the non-constant variance. Residual with small value followed another residuals with a small value. On the other hand, residual with a large value usually followed a residual with another large value. However, this is not typical for a white noise process. Therefore, this assumption lead us to think about stochastic model for volatility. The suitability for using stochastic volatility model was also accepted by performed heteroskedasticity test. ARCH test (see Table 6, Appendix A) confirmed the series was heteroskedastic since the null hypothesis of homoskedasticity had been rejected at 5% and so the residuals were characterized by the presence of ARCH effect which was quite a frequent phenomenon at financial time series. Therefore, we applied a stochastic volatility model into the basic model. According to correlogram of squared residuals of EUR/GBP differences (see Table 8, Appendix A) we quantified ARCH(4) model for volatility.

After quantification of ARCH(4) model, the residuals were characterized by the absence of conditional heteroskedasticity: the ARCH-LM statistics were strictly less than the critical value at 5%. In addition, the standardized residuals tested with Ljung-Box Q-test (see Table 9, Appendix A) confirmed there were no significant coefficients in residuals of this model. Finally, the final ARCH(4) volatility model is defined as follows

$$\begin{aligned}\sigma^2 = & 0.00000438 + 0.104930\varepsilon_{t-1}^2 + 0.101053\varepsilon_{t-2}^2 \\ & + 0.150503\varepsilon_{t-3}^2 + 0.085457\varepsilon_{t-4}^2\end{aligned}\quad (11)$$

Table 10 in Appendix A states the numerical characteristics of ARCH(4) model as well as Student's t-statistics for its parameters.

3.2 Modelling with Neural Networks

Other techniques have started to apply in the domain of time series forecasting recently One of the reason was the study by Bollershev [3], where he proved the existence of nonlinearity in financial data. Non-linearity modelling was one of the drawbacks of Box-Jenkins models. Today, according to studies such as that by Gooijer [9], artificial neural networks (ANN) are the machine learning models having the biggest potential in forecasting financial time series. This is due to the fact that these models are extremely helpful in modelling non-linear processes which have a priori unknown functional relations or this system of relations is very complex to describe mathematically (see [25]). ANN is based on human neural system and is an universal functional black-box approximator of non-linear type [26, 27 and 28). The reason for attractiveness of ANNs for financial prediction can be found in the work of Hill et al. [29]. Here, the authors showed that the ANNs worked best in connection with highfrequency financial data. The competitive performance of ANN is also documented on a large number of time series (see [30][31]). In this part we show a new approach of estimation of forecasting function for conditional volatility modelled by feedforward neural network of RBF type combined with genetic algorithms as well as statistical ARCH models.

A fully connected feed forward neural network was selected to be used as the forecasting function, due to its conceptual simplicity, and computational efficiency [32]. Our implemented neural network consisted of three layers, except for the input and output, there was also one hidden layer. We proposed the architecture of the neural network (see Figure 7) with only one hidden layer due to the fact that according to Cybenko theorem [33] the network with one hidden layer is able to approximate any continuous function. This hidden layer made a previous nonlinear transformation of the data so as to facilitate resolution of the problem in hand such as regression, classification, etc. The neural

network used for this research was the network of RBF type [34]. This network is one of the most frequently used networks for regression [32]. RBF, as well as multilayer perceptron (MLP, which is a predecessor of RBF) have been widely used to capture a variety of nonlinear patterns (see [35]) thanks to their universal approximation properties (see [36]); in other words, thanks to their capacity to approximate any continuous function provided that they have a sufficient number of hidden units (neurons).

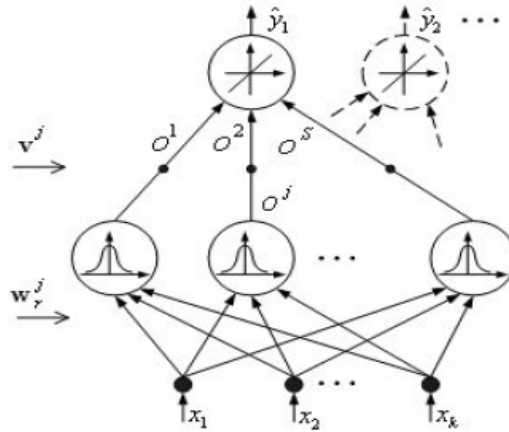


Figure 7: The architecture of used RBF neural network

The structure of our RBF neural network is defined by its architecture (processing units and their interconnections, activation functions, methods of learning and so on). In Figure 7 each circle or node represents the neuron. The neural network consisted of an input layer with input vector \mathbf{x} and an output layer with the output value \hat{y}_i .

The layer between the input and output layers is normally referred to as the hidden layer and its neurons as RBF neurons. Here, the input layer is not treated as a layer of neural processing units. One important feature of RBF networks is the way how output signals are calculated in computational neurons. The output signals of the hidden layer are

$$o_j = \psi_2(\|\mathbf{x} - \mathbf{w}_j\|) \quad (12)$$

where \mathbf{x} is a k -dimensional neural input vector, \mathbf{w}_j represents the hidden layer weights, ψ_2 are radial basis (Gaussian) activation functions. Note that for an RBF network, the hidden layer weights \mathbf{w}_j represent the centres \mathbf{c}_j of activation functions in the hidden layer.

The second parameter of the radial basis function, the standard deviation, is estimated as K , ($K \geq 1$) multiple of the mean value of quadratic distance among the input vectors and their cluster centres. The value of K is regarded as the rate of overlapping in the distribution of input data (see [37]). The output layer neuron is linear and has a scalar output given by

$$\hat{y} = \sum_{j=1}^s v_j o_j \quad (13)$$

where v_j are the trainable weights connecting the component of the output vector \mathbf{o} . Then, the output of the hidden layer neurons are the radial basic functions of the proximity of weights and input values. A serious problem is how to determine the number of hidden layer (RBF) neurons. The most used

selection method is to preprocess training (input) data by some clustering algorithm. After choosing the cluster centres, the shape parameters σ_j must be determined. These parameters express an overlapping measure of basis functions. For Gaussians, the standard deviations σ_j can be selected, i.e. $\sigma_j \sim \Delta c_s$, where Δc_s denotes the average distance among the centers. Finally, the RBF network computed the output data set as

$$\hat{y}_t = G(\mathbf{x}_t, \mathbf{c}, \mathbf{v}) = \sum_{j=1}^s v_{j,t} \psi_2(\mathbf{x}_t, \mathbf{c}_j) = \sum_{j=1}^s v_{j,t} o_{j,t} \quad t = 1, 2, \dots, N \quad (14)$$

where N is the size of data samples, s denotes the number of the hidden layer neurons. The hidden layer neurons received the Euclidian distances ($\|\mathbf{x} - \mathbf{c}_j\|$) and computed the scalar values $o_{j,t}$ of the Gaussian function $\psi_2(\mathbf{x}_t, \mathbf{c}_j)$ that form the hidden

layer output vector \mathbf{o}_t . Finally, the single linear output

layer neuron computed the weighted sum of the Gaussian functions that formed the output value of \hat{y}_t .

To optimize the outputs of the network and to maximise the accuracy of the forecasts, we had to optimize parameters of ANN. The most popular method for learning (i.e. adapting parameters) in multilayer networks is called back-propagation invented by Bryson and Ho [38]. However, there are some drawbacks to back-propagation. One of them is the convergence of this algorithm - it generally converges to any local minimum on the error surface, since stochastic gradient descent exists on a surface which is not flat. Due to this reason, we also used the combination of back-propagation with the standard unsupervised technique called K-means (see [39]). K-means algorithm, which belongs to a group of unsupervised learning methods, is a nonhierarchical exclusive clustering method based on the relocation principle. The most common type of characteristic function is location clustering. The K-means was used in the phase of non-random initialization of weight vector w performed before they were adapted by back-propagation. i.e. before the phase of network learning. We assumed that in many cases it was not necessary to interpolate the output value by radial functions, it was quite sufficient to use one function for a set of data (cluster), whose center was considered to be a center of activation function of a neuron. We also supposed that after K-means performed, weights should have been located near the global minimum of the error function and lower number of epochs were supposed to be used for network training.

The values of centroids were used as an initialization values of weight vector w . To find the weights \mathbf{w}_j or centres of activation functions we used the following adaptive (learning) version of K-means clustering algorithm for s clusters:

Step 1. Randomly initialise the centres of RBF neurons

$$c_j^{(t)}, j = 1, 2, \dots, s \quad (15)$$

where s represents the number of chosen RBF neurons (clusters).

Step 2. Apply the new training vector

$$x^{(t)} = (x_1, x_2, \dots, x_k). \quad (16)$$

Step 3. Find the nearest centre to $x^{(t)}$ and replace its position as follows

$$c_j^{(t+1)} = c_j^{(t)} + \lambda(t) (x^{(t)} - c_j^{(t)}) \quad (17)$$

where $\lambda(t)$ is the learning coefficient and is selected as linearly decreasing function of t by $\lambda(t) = \lambda_0(t)(1 - t/N)$ where $\lambda_0(t)$ is the initial value, t is the present learning cycle and N is number of learning cycles.

Step 4. After chosen epochs number, terminate learning. Otherwise go to step 2

The above learning method based on the clustering algorithm is regarded as one of the granular method presenting the bottom-up granulation (see [40]). Input vectors are combined into larger overlapping granules (clusters) described by cluster's centres and the standard deviations.

Since back-propagation also features some other problems such as "scaling problem" we decided to implement genetic algorithm as an learning technique for our RBF neural network too. Therefore, in our implementation of ANN, back-propagation was altered by the genetic algorithm (GA) as an alternative learning technique in the process of weights adaptation. Adopted from biological systems, genetic algorithms, which are algorithms for optimization and machine learning, are stochastic search techniques that guide a population of solutions towards an optimum using the principles of evolution and natural genetics [41]. They are based loosely on several features of biological evolution [42]), have become a popular optimization tool in various areas. They require five components to be met [43]:

1. A way of encoding solutions to the problem on chromosomes. In the original genetic algorithm an individual chromosome is represented by a binary string. The bits of each string are called genes and their varying values alleles. A group of individual chromosomes is called a population.
2. An evaluation function which returns a rating for each chromosome given to it.
3. A way of initializing the population of chromosomes.
4. Operators that may be applied to parents when they reproduce to alter their genetic composition. Standard operators are mutation and crossover (i.e. recombination of genetic material).
5. Parameter settings for the algorithm, the operators, and so forth.

Genetic algorithms are characterized by basic genetic operators which include reproduction, crossover and mutation [44]. Given these genetic operators and five components stated above, a genetic algorithm operates according to the following steps [45]:

1. Initialize the population using the initialization procedure, and evaluate each member of the initial population.
2. Reproduce until a stopping criterion is met. Reproduction consists of iterations of the following steps:
 - a) Choose one or more parents to reproduce. Selection is stochastic, but the individuals with the highest evaluations are usually favored in the selection.
 - b) Choose a genetic operator and apply it to the parents.
 - c) Evaluate the children and accumulate them into a generation. After accumulating enough individuals, insert them into the population, replacing the worst current members of the population.

When the components of the genetic algorithm are chosen appropriately, the reproduction process should continually generate better children from good parents. The algorithm can then produce populations of better and better individuals, converging finally on results close to a global optimum. Additionally, GA can efficiently search large and complex (i.e., possessing many local optima) spaces to find nearly a global optima [45]. In addition to that, genetic algorithm does not have the same problem with scaling as back-propagation. One reason for this is that it generally improves the current best candidate monotonically. It does this by keeping the current best individual as part of their population while they search for better candidates. Moreover, supervised learning algorithms suffer from the possibility of getting trapped on suboptimal solutions. Genetic algorithms are generally not bothered by local minima. The mutation and crossover operators can step from a valley across a hill to an even lower valley with no more difficulty than descending directly into a valley. So GA enables the learning process to escape from entrapment in local minima in instances where the back-propagation algorithm converges prematurely.

To create a genetic algorithm, a number of parameters was required: a method of encoding chromosomes, the fitness function used to calculate the fitness values of chromosomes, the population size, initial population, maximum number of generations, selection method, crossover function, mutation method. The implementation of the genetic algorithm we used for weight adaptation was as follows. The chromosome length was set according to the formula: $D * s + s$, where s is the number of hidden neurons and D is the dimension of the input vector. A specific gene of a chromosome was a float value and represented a specific weight in the neural network. The whole chromosome represented weights of the whole neural network. The fitting function for evaluating the chromosomes was the mean square error function (MSE). The chromosome (individual) with the best MSE was automatically transferred into the next generation. The other individuals of the next generation were chosen as follows: By tournament selection individuals were randomly chosen from the population. The fittest of them was then chosen as a parent. The second parent was chosen in the same way. The new individuals was then created by crossover operation. If the generated value from $(0,1)$ was lower than 0.5 the weight of the first parent at the specific position was assigned to the new individual. Otherwise, the new individual received the weight of the second parent.

4 Results and Discussion

For volatility (variance) modelling we followed several studies in the literature (see [46][47][48]) and measured the volatility of EUR/GBP currency by its squared daily first differences:

$$\hat{\sigma}_t^2 = \Delta_y^2 \quad (18)$$

In our tests, we used one-step-ahead, frequently called as static, forecasts, i.e. the horizon of predictions was equal to one day. As we said, we used MSE (Mean Square Error) and RMSE (root mean square error) numerical characteristics for assessing all models.

Firstly, we estimated and tested the ARCH(4) model for volatility defined in (11). However, we also tested some other statistical models modelling conditional variance such as GARCH(1,1) model [2] which is supposed to be so-called universal model in financial domain. We also tested EGARCH(1,1,1) defined in (8). Important to remember that the estimation of these models was only based on 1306 in-sample observations, in order to make ex-ante predictions with remaining 1304 observations. We used the Marquardt optimization procedure for finding the optimal values of ARCH/GARCH parameters. Initial values of parameters were counted using Ordinary Least Squares (OLS) method and these values were then optimized by iterative process consisted of 500 iterations. Convergence rate was set to 0.0001. The forecasting ability of particular networks was measured by the MSE criterion of ex post forecast periods (validation data set).

As for models based on neural networks, we implemented three models, each of them was an implementation of feedforward neural network of RBF type [34]. In addition to that we implemented three different optimization techniques for adaptation of weights (parameters) of this network – genetic algorithm, standard back-propagation algorithm (BP) as well as a combination of K-means clustering combined with the back-propagation (KM+BP). We implemented all of these algorithms and models by ourselves using the JAVA programming language. The approximation as well as forecasting results measured by MSE were calculated analogously as in the case of ARCH/GARCH models.

As for the inputs into our ANN models, we used the information from the statistical models, particularly from ARCH(4). Looking at the equation of the ARCH(4) model we knew that the conditional variance was dependent on the previous four lagged squared residuals. Therefore we used this information to construct a hybrid neural network (so called ARCH-RBF neural network) which used the residuals from ARCH(4) model to compute the outputs (variance). This approach is shown in Figure 9.

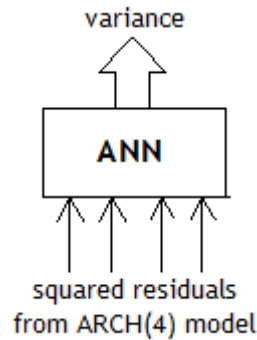


Figure 9: The suggested and implemented model of ARCH-RBF neural network.

For adaptation via BP we used learning rate of 0.1, 0.05 respectively. As for adaptive K-means, we used 10000 cycles and the learning rate of K-means adaptation was set to 0.1, 0.05 respectively. The number of clusters was set to the number of hidden neurons.

We also tested several configurations of genetic algorithms with different population size, different mutation rates as well as size of the first choice process. For this particular data the best mutation rate showed to be 0.1, 0.2 respectively. If performed, the specific gene (weight) of a chromosome was changed to a random value. The best population sizes were 500 and 1000 individuals. The first choice crossover was set with probability $p_1=0.02$ and the second one with $p_2 = 0.5$. The results for in-sample predictions are stated in Table 1 and results for out-of-sample predictions (ex-post predictions) are stated in Table 2.

Table 1: Prediction accuracy of tested models measured by MSE (in-sample predictions).

RBF configuration	Learning method		
	RBF (BP)	RBF (KM)	RBF (GA)
(4 – 3 – 1)	$5,3664 \cdot 10^{-3}$	$2,4832 \cdot 10^{-10}$	$2,5547 \cdot 10^{-10}$
(4 – 5 – 1)	$46933 \cdot 10^{-3}$	$2,2493 \cdot 10^{-10}$	$2,4505 \cdot 10^{-10}$
(4 – 7 – 1)	$2,6660 \cdot 10^{-10}$	$1,9481 \cdot 10^{-9}$	$5,5255 \cdot 10^{-10}$
(4 – 10 – 1)	$1,6411 \cdot 10^{-2}$	$4,5823 \cdot 10^{-9}$	$3,0536 \cdot 10^{-10}$
Error Distribution			
Statistical model	Gaussian	Student	GED

ARCH(4)	$2,2350 \cdot 10^{-10}$	$2,2360 \cdot 10^{-10}$	$2,2353 \cdot 10^{-10}$
GARCH(1,1)	$2,1718 \cdot 10^{-10}$	$2,1726 \cdot 10^{-10}$	$2,1722 \cdot 10^{-10}$
EGARCH(1,1,1)	$2,1644 \cdot 10^{-10}$	$2,1642 \cdot 10^{-10}$	$2,1643 \cdot 10^{-10}$

Table 2: Prediction accuracy of tested models measured by MSE (out-of-sample predictions).

RBF configuration	Learning method		
	RBF (BP)	RBF (KM)	RBF (GA)
(4 – 3 – 1)	$4,1471 \cdot 10^{-9}$	$3,9282 \cdot 10^{-9}$	$4,2028 \cdot 10^{-9}$
(4 – 5 – 1)	$4,1590 \cdot 10^{-9}$	$4,2222 \cdot 10^{-9}$	$4,1587 \cdot 10^{-9}$
(4 – 7 – 1)	$4,2156 \cdot 10^{-9}$	$1,1480 \cdot 10^{-8}$	$4,1170 \cdot 10^{-9}$
(4 – 10 – 1)	$4,5086 \cdot 10^{-9}$	$7,8121 \cdot 10^{-9}$	$4,9923 \cdot 10^{-9}$
Statistical model	Error Distribution		
	Gaussian	Student	GED
ARCH(4)	$3,8203 \cdot 10^{-9}$	$3,8054 \cdot 10^{-9}$	$3,8143 \cdot 10^{-9}$
GARCH(1,1)	$3,5058 \cdot 10^{-9}$	$3,5175 \cdot 10^{-9}$	$3,5099 \cdot 10^{-9}$
EGARCH(1,1,1)	$3,4809 \cdot 10^{-9}$	$3,4836 \cdot 10^{-9}$	$3,4851 \cdot 10^{-9}$

The standard back-propagation algorithm for weights adaptation showed to be a weakness of the network. The convergence was really slow (cca 5000 epochs) and in addition to that, resulting from many experiments with learning rate and the initialization by random weights, it generally converged to any local minimum on the error surface. Therefore there was no guarantee that the algorithm would converge to global minimum. In addition, this algorithm was very dependent on the initialized random weights. Due to this, generally a lot of more epochs was needed to achieve reasonable accuracy compared to K-means + BP.

Bearing in mind these disadvantages of BP, we also tested K-means, that was used in the phase of non-random initialization of weight vector w performed before the phase of network learning. Besides lower MSE at lower configurations of hidden neurons (three, five), another advantage of using K-means upgrade (but also GA) was the consistency of predictions. The standard deviation of these methods was uncomparably lower than the standard deviation when using the standard back-propagation. Moreover, the biggest strength of K-means was in the speed of convergence of the network. Without K-means, it took considerably longer time to achieve the minimum. However, when the K-means was used, the time (number of epochs) for reaching the minimum was much shorter (cca 500 to 1000 epochs). Therefore, the advantage of using K-means together with back-propagation is in the speed of adaptivity rather than in better predictions. However, one must bear in mind that K-means is a relatively efficient algorithm only in the domain of non-extreme values. Otherwise, other advanced non-hierarchical clustering algorithms must be used.

Having tested also genetic algorithm in weights adaptation, we found out the convergence was also considerably faster than at back-propagation. In addition to that, genetic algorithm did not have the same problem with scaling as back-propagation. One reason for this is that GA generally improves the current best candidate monotonically. It does this by keeping the current best individual as part of their population while they search for better candidates.

However, the accuracy results were not very different from the other two optimization techniques; in some cases the optimization methods based on BP were even more accurate. As according to the theory, genetic algorithms are not bothered by local minimum problem (since the algorithms operate on a population instead of a single point in the search space, they climb many peaks in parallel and

therefore reduce the probability of finding local minima) such as BP and as GA are also especially capable of handling problems in which the objective function is discontinuous or non differentiable, nonconvex, multimodal or noise; we expected better results than we got. This could, however, be caused due to non-optimized parameters of GA. Except for our experiments and tests with the best configuration of GA parameters, we also tested the optimization procedure stated in [49] which was in our case not very helpful. Maybe, testing some other optimization procedure for the best parameters of GA would lead to better results of genetic algorithm. The second reason could be that the standard unbiased crossover function was used. The biased crossover function stated in [45] could enhance our solution.

It is also to mention that the best results were achieved with lower number of neurons. Following from that one can deduce that for remembering the relationships in this time series it is enough to use smaller number of hidden neurons.

The final comparison of statistical as well as neural networks models is stated in Table 3.

Table 3: Final Comparison of out-of-sample (ex-post) predictions.
Numerical characteristics

Model	MSE E	RMSE E	Rank
RBF (4 - 3 - 1) (BP)	$4,1471 \cdot 10^{-9}$	0,00006439	6
RBF (4 - 3 - 1) (KM)	$3,9282 \cdot 10^{-9}$	0,00006267	4
RBF (4 - 7 - 1) (GA)	$4,1170 \cdot 10^{-9}$	0,00006416	5
ARCH(4) (Student)	$3,8054 \cdot 10^{-9}$	0,00006168	3
GARCH(1,1) (Gauss)	$3,5058 \cdot 10^{-9}$	0,00005920	2
EGARCH(1,1,1) (Gauss)	$3,4809 \cdot 10^{-9}$	0,00005899	1

Deducing from Table 3, we can say that on the validation set the best results were achieved with EGARCH(1,1,1) model. On the other hand, the worst results were achieved with RBF neural network combined with the standard back-propagation algorithm. However, the differences between the results were very small and the difference in results between the best and worst model is only about nine per cent.

Following from that, our suggested RBF hybrid neural network combined with ARCH inputs showed to be an efficient and accurate way of forecasting conditional volatility in financial domain. But, generally speaking, the statistical models achieved a little bit higher accuracy than the neural networks models. However, the difference was very small and the results were almost of the same accuracy. However, the achieved ex post accuracy of ARCH-RBF (best RMSE = 0.00006267) is still reasonable and acceptable for use in forecasting volatility which plays an important role in managerial decision processes in the finance area. Moreover, a little bit worse results of neural network models can be the result of the following factors:

- non-optimized parameters of genetic algorithm, which could cause a little bit worse final solution than expected
- back-propagation as the non-ideal optimization technique for parameters optimization
- The non-ideals inputs coming from the statistical ARCH(4) model
- The data we chose for our experiments were not “representative“. One can not eliminate the assumption saying that if we used other data for our experiments the neural network models would outperform the ARCH/GARCH models.

Coming from that, there are more options of how to upgrade this model in the future:

1) We could better „optimize“ the parameters of genetic algorithm. We could apply other known optimization procedure than [49] into our neural network models. It could improve the solution quite a lot.

2) Apart from the standard back-propagation algorithm it would be reasonable to use and implement the more advanced version of this algorithm (to avoid the imprisonment in the local minum). We could use some of the versions of adaptive back-propagation.

3) Probably the outputs of the neural network models would be more accurate if we did not use the inputs from ARCH model. We could use only the information from ARCH model and the residuals would come from the RBF itself. It could be done by implementing a version of recurrent ARCH-RBF neural network.

4) Probably, even better results could be achieved by implementing the recurrent RBF neural network based on GARCH model, not ARCH (so called recurrent GARCH-RBF).

5) The RBF model could be enhanced by implementing error-correction part, i.e. smoothing the error (residual) of the RBF neural network by using *m-period* weighted or exponential or simple moving average such as:

$$\varepsilon_t^{RBF} = e_t + u_t, \quad u_t \approx iid(0,1) \quad (19)$$

and

$$e_t = \sum_{i=1}^q \theta_i \varepsilon_{t-i}^{RBF}, \quad \sum_{i=1}^q \theta_i = 1 \quad (20)$$

5 Summary

The main reason for forecasting volatility for risk managers is that volatility is extremely important. It is due to the fact that a large part of risk management is measuring the potential future losses of a portfolio of assets (and volatility modelling provides a simple approach to calculating value at risk of a financial position in risk management. In this paper, we investigated the modelling of volatility dynamics of EUR/GBP exchange rate differences. We examined two types for volatility forecasting – the models based on statistics and neural network models. We evaluated the effectiveness of various volatility models with respect to forecasting market risk in the exchange rate market, EUR/GBP more specifically. While there is a stream of literature examining performance of models for volatility based on statistical approach, this is a pioneer study to particularly focus on forecasting volatility with neural networks.

For all our tests, the data were divided into training set and validation set. The models were quantified only on training set and then they were tested on out-of-sample prediction interval to evaluate their prediction power. The out-of-sample period, that has been tested in this study contained data from November, 1, 2008 to October, 31, 2013. The reason for doing that was that models that perform well in the considered out-of-sample period may well underperform in future periods, particularly when market conditions change. Both in-sample and out-of-sample forecasts were evaluated using statistical summary measures of model's forecast accuracy.

We evaluated three most common statistical models for volatility forecasting – the universal GARCH model, the basic ARCH and EGARCH model which is able to model leverage effects. In addition to that, all three models were evaluated with Gaussian, Student and GED error distributions.

In case of models based on neural network approach, we suggested new model for forecasting volatility with neural networks – the ARCH-RBF neural network. We used the proxy metrics to calculate the actual volatility and so we were able to implement the neural network with inputs from the quantified ARCH model and developed the new approach for forecasting volatility. Moreover, we

constructed ARCH-RBF neural network with three different types of optimization techniques. Except for the standard back-propagation, we combined an K-means clustering into the RBF to achieve higher accuracy of the network. Both of the algorithms were used in the process of adapting weights of the network. The reason for incorporating other algorithms into the network was that the back-propagation was considered a weakness of the RBF. In addition, we also eliminated the back-propagation algorithm by using the genetic algorithm instead. In the final comparison of the selected optimization techniques both of these upgrades showed to be helpful in the process of creating more accurate forecasts of volatility and they should be definitely used instead of the standard back-propagation.

According to results we achieved, the statistical approach was better than the neural network models. However, the differences in accuracy of volatility forecasting were very small. None of the considered models performed significantly better than the rest with respect to the considered criteria. So the achieved ex post accuracy of ARCH-RBF neural network models is still reasonable and acceptable for use in forecasting systems that routinely predict volatility in managerial decision processes in the financial domain. Moreover, a little bit higher error could be caused by non-optimizing parameters of genetic algorithm, non-ideal inputs of ARCH model or just due to type of data we used.

On the other hand, there is definitely a reason for using ARCH-RBF neural network in the domain of volatility forecasting as this model showed to be a good tool in volatility forecasting. Neural networks are capable of providing information in the form of forecasts with an acceptable degree of uncertainty. They are relatively fast and have the ability to generalize. Moreover, the ARCH-RBF has such attributes as computational efficiency, simplicity, and ease adjusting to changes in the process being forecast. ARCH/GARCH models require more costs of development, installation and operation in a management system, management comprehension and cooperation, and often a lot of computational time.

Finally, accuracy of our suggested model could be improved by upgrading the model by some of the upgrades we discussed in the Results & Discussion chapter (such as recurrent version of ARCH-RBF, GARCH-RBF, Error-correction RBF, etc). There is a strong assumption that this upgrade can cause this model to have even a great accuracy advantage over the statistical models. We leave the investigation of these issues to future work.

Acknowledgements

This paper has been elaborated in the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by Operational Programme 'Research and Development for Innovations' funded by Structural Funds of the European Union and state budget of the Czech Republic.

6 References

- [1] H. M. Markowitz, Portfolio Selection, *The Journal of Finance*, 7 (1) 1952, 77–91.
- [2] R. F. Engle, Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation, *Econometrica*, 50(4) (1982) 987 – 1008.
- [3] T. Bollershev, Generalized Autoregressive Conditional Heteroskedasticity, *Journal of Econometrics*, 31 (1986) 307–327.
- [4] R. F. Engle, D. M. Lilien, R. P. Robins, Estimating time varying risk premia in the term structure: the ARCH-M model, *Econometrica*, 55(2) (1987) 391–407.
- [5] D. B. Nelson, Conditional Heteroskedasticity in Asset Returns: a New Approach, *Econometrica*, 59 (2) (1991) 347–370.
- [6] L. R. Glosten, R. Jagannathan, D. E. Runkle, On the relation between the expected value and

- the volatility of the normal excess return on stocks, *The Journal of Finance*, 48(5) (1993) 1779-1801.
- [7] Z. Ding, C. W. Granger, R. F. Engle, A Long Memory Property of Stock Market Returns and a New Model, *Journal of Empirical Finance*, 1 (1993) 83-106.
 - [8] P. R. Hansen, A. Lunde, A forecast comparison of volatility models: does anything beat a GARCH (1,1). *Journal of applied econometrics*, 20(7) (2005) 873-889.
 - [9] J. G. Gooijer, R. J. de Hyndman, 25 years of time series forecasting, *International Journal of Forecasting*, 22 (2006) 443-473.
 - [10] G. P. Zhang, Time series forecasting using a hybrid ARIMA and neural network model, *Neurocomputing*, 50 (2003) 159-175.
 - [11] D. Marcek, Granular RBF NN Approach and Statistical Methods Applied to Modelling and Forecasting High Frequency Data, *International Journal of Computational Intelligence Systems*, 2 (4) (2009) 353-364.
 - [12] C. F. Huang, D. Ruan, Fuzzy risks and an updating algorithm with new observation, *Risk Analysis*, 28(3) (2009) 681-94.
 - [13] D. Cox D, D. Hinkley, *Theoretical Statistics* (Chapman and Hall, London, UK, 1974).
 - [14] S. Weisberg, *Applied Linear Regression* (Wiley, New York, USA, 1980).
 - [15] D. Applebaum, D. Lévy, *Processes and Stochastic Calculus* (Cambridge University Press, Cambridge 2004).
 - [16] D. Marcek, Risk Scenes Of Managerial Decision-Making With Incomplete Information: An Assessment In Forecasting Models Based On Statistical And Neural Networks Approach, *Journal of Risk Analysis and Crisis Response*, 3 (1) (2013) 13-21.
 - [17] G. E. P. Box, G. M. Jenkins, *Time series analysis: forecasting and control*, (Holden-Day, San Francisco, 1976).
 - [18] T. M. O'Donovan, *Short Term Forecasting: An Introduction to the Box-Jenkins Approach* (Wiley, New York, NY, 1987).
 - [19] E. Zivot, J. Wang, *Modeling Financial Time Series with S-PLUS* (Springer Verlag, NY, 2005).
 - [20] S. A. M. Yaser, A. F. Atiya, Introduction to financial forecasting, *Applied Intelligence*; 6 (1996) 205–213.
 - [21] D. Dickey, W. Fuller, Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74 (1979) 427-431.
 - [22] G. Elliott, T. J. Rothenberg, J. H. Stock, Efficient tests for an autoregressive unit root, *Econometrica*, 64(4) (1996) 813-836.
 - [23] D. Kwiatkowski, P. Phillips, P. Schmidt, Y. Shin, Testing the null hypothesis of stationary against the alternative of a unit root; How sure are we that economic time series have a unit root?, *Journal of Economics*, 54 (1992) 159-178.
 - [24] P. C. B. Phillips, P. Perron, Testing for unit roots in time series regression. *Biometrika*, 75 (1988) 335-346.
 - [25] G. A. Darbellay, M. Slama, Forecasting the short-term demand for electricity: Do neural networks stand a better chance?, *International Journal of Forecasting*, 16 (2000) 71– 83.
 - [26] K. Hornik, Some new results on neural network approximation, *Neural Networks*, 6 (1993) 1069-1072.
 - [27] K. Hornik, M. Stinchcomber, H. White, Multilayer feedforward networks are universal approximations, *Neural Networks*, 2 (1989) 359-366.
 - [28] L. S. Maciel, R. Ballina, *Design a Neural Network for Time Series Financial Forecasting: Accuracy and Robustness Analysis* (2008)
 - [29] T. L. Hill, M. Marquez, R. O'Connor, Neural networks models for forecasting and decision making, *International Journal of Forecasting*, 10 (1994) 5-15.
 - [30] K. P. Liao, R. Fildes, The accuracy of a procedural approach to specifying feedforward neural networks for forecasting, *Computers & Operations Research*, 32 (2005) 2151-2169.
 - [31] G. P. Zhang and M. Qi, Neural network forecasting for seasonal and trend time series, *European Journal Of Operational Research*, 160 (2005) 501-514.
 - [32] D. Marcek, M. Marcek, *Neural Networks and Their Applications* (EDIS –ZU, Zilina, 2006).
 - [33] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Mathematics of Control, Signals and Systems*, 2(4) (1989) 303-314.
 - [34] M. J. L. Orr, *Introduction to Radial Basis Function Networks* (University of Edinburgh, 1996).

- [35] K. Hornik, M. Stinchcombe, H. White, Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, *Neural Networks*, 3 (1990) 551-560.
- [36] M. Leshno, V.Y. Lin, A. Pinkus, S. Schocken, Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, *Neural Networks*, 6 (1993) 861-867.
- [37] CH. M. Bishop, *Neural Networks for Pattern recognition*. (Oxford University Press, New York, 1995).
- [38] A.E. Bryson, H. Yu-Chi, *Applied optimal control: optimization, estimation, and control* (Blaisdell Publishing Company, 1969).
- [39] J. B. MacQueen, Some Methods for classification and Analysis of Multivariate Observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, (University of California, 1967), pp. 281-297.
- [40] Y. Y. Yao, Granular Computing for Data Mining. Kissimmee: Congres of Data Mining, Intrusion Detection, Information Assurance and Data Network Security 2006. Vol. 6241. 16th – 17th April 2006, pp. 624105.1-624105.12.
- [41] M. Dharmistha, Genetic Algorithm based Weights Optimization of Artificial Neural Network, *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 1(3) (2012).
- [42] J. H. Holland, *Adaptation in Natural and Artificial Systems* (University of Michigan Press, 1975).
- [43] L. Davis, *Genetic Algorithms and Simulated Annealing* (Pitman, London, 1987).
- [44] D. Whitley, Applying Genetic Algorithms to Neural Network Problems, *International Neural Network Society* (1988).
- [45] D. J. Montana, L. Davis, Training Feedforward Neural Networks Using Genetic Algorithms, *Proceedings of the 11th international joint conference on Artificial intelligence*, (1989), pp. 762- 767.
- [46] R. Reider, *Volatility Forecasting I: GARCH Models* (2009).
- [47] P. Sadorsky, Modeling and forecasting petroleum futures volatility, *Energy Economics*, 28(4) (2006) 467-488.
- [48] S. Truck and K. Liang, Modelling and Forecasting Volatility in the Gold Market, *International Journal of Banking and Finance*, 9(1) (2012) 47 – 78.
- [49] K. A. DeJong, W. M. Spears, An Analysis of the Interacting Roles of Population Size and Crossover in Genetic Algorithms, *Proc. First Workshop Parallel Problem Solving from Nature* (Springer-Verlag, Berlin 1990), pp. 38-47.

7 APPENDIX

Table 4. Unit root tests of EUR/GBP.

Test		Original series [p-value]		1 st differences [p-value]	
Augmented Dickey-Fuller		(I)	-1.042297 [0.9225]	(I)	-36.08501 [0.0000]
		(II)	-0.282886 [0.9249]	(II)	-36.10204 [0.0000]
		(III)	-1.408858 [0.8583]	(III)	-36.13347 [0.0000]
Phillips-Perron		(I)	1.169134 [0.9381]	(I)	-36.23934 [0.0000]
		(II)	-0.067288 [0.9510]	(II)	-36.28507 [0.0000]
		(III)	-1.123755 [0.9017]	(III)	-36.37293 [0.0000]
Test	Window	Spectral estimation method			

		Bartlett kernel				Quadratic spectral kernel			
		Original series		Returns		Original series		Returns	
		(II)	(III)	(II)	(III)	(II)	(III)	(II)	(III)
KPSS	Newey-West	2.057885 (0.463)	0.738516 (0.146)	0.281293 (0.463)	0.060591 (0.146)	4.078739 (0.463)	1.4573 91 (0.146)	0.265212 (0.463)	0.056470 (0.146)
	Andrews	0.706988 (0.463)	0.153498 (0.146)	0.229661 (0.463)	0.048072 (0.146)	2.538563 (0.463)	0.2234 34 (0.146)	0.229661 (0.463)	0.047929 (0.146)
H ₀ : Stationary Series Elliot-Rothenberg-Stock	Newey-West	19.93783 (3.26)	28.96699 (5.62)	0.048644 (3.26)	0.182369 (5.62)	18.89474 (3.26)	27.414 43 (5.62)	0.045855 (3.26)	0.169872 (5.62)
	Andrews	16.47829 (3.26)	23.99466 (5.62)	0.039740 (3.26)	0.145021 (5.62)	16.47612 (3.26)	23.995 56 (5.62)	0.039744 (3.26)	0.145054 (5.62)

(I): model without constant and deterministic trend (5%)

(II): model with constant and without deterministic trend (5%)

(III): model with constant and deterministic trend (5%)

Table 5. ACF and PACF of EUR/GBP 1st differences.

Autocorrelation	Partial Correlation	AC	PAC	Q-Stat	Prob
		1 -0.001 -0.001	0.0006	0.980	
		2 -0.042 -0.042	2.2682	0.322	
		3 -0.064 -0.064	7.6254	0.054	
		4 0.019 0.017	8.1118	0.088	
		5 0.016 0.010	8.4325	0.134	
		6 -0.009 -0.012	8.5504	0.200	
		7 -0.035 -0.032	10.154	0.180	
		8 -0.017 -0.017	10.547	0.229	
		9 -0.032 -0.037	11.874	0.221	
		10 0.062 0.057	16.974	0.075	

Table 6. Normality tests on distribution of residuals (first differences) and other main characteristics.

Skewness	Kurtosis	J.B.	A.D.	ARCH-LM statistic
0.217321	4.890953	204.7010	6.802221	11.64566
		[0.0000]	[0.0000]	[0.0000]

J.B. Jarque-Bera statistic, A.D. Anderson-Darling statistic

Table 7. BDS test results on the series of AR(0) residuals.

Dimension	Method of Fraction of Pairs			Standard Deviations Method		
	BDS Statistic	z-Statistic	Prob.	BDS Statistic	z-Statistic	Prob.
2	0.013801	5.920452	0.0000	0.007255	4.372953	0.0000
3	0.027191	7.416898	0.0000	0.008789	5.552197	0.0000
4	0.034976	8.095958	0.0000	0.006969	6.147898	0.0000
5	0.038974	8.746172	0.0000	0.004469	6.285917	0.0000
6	0.037880	8.906903	0.0000	0.002525	6.117297	0.0000
7	0.035050	9.087935	0.0000	0.001372	6.022175	0.0000
8	0.030937	9.171206	0.0000	0.000704	5.800715	0.0000
9	0.026972	9.391910	0.0000	0.000367	5.824869	0.0000
10	0.023471	9.802340	0.0000	0.000190	5.928569	0.0000

Table 8. Correlogram of squared residuals (1st differences).



Autocorrelation	Partial Correlation	AC	PAC	Q-Stat	Prob
		1 0.155	0.155	31.528	0.000
		2 0.167	0.147	68.199	0.000
		3 0.186	0.147	113.35	0.000
		4 0.150	0.092	142.97	0.000
		5 0.092	0.021	154.19	0.000
		6 0.119	0.054	172.65	0.000
		7 0.114	0.052	189.84	0.000
		8 0.101	0.040	203.15	0.000
		9 0.096	0.033	215.27	0.000
		10 0.082	0.016	224.09	0.000

Table 9. ACF and PACF of AR(0)-ARCH(4) residuals.



Autocorrelation	Partial Correlation	AC	PAC	Q-Stat	Prob
		1 0.025	0.025	0.7866	0.375
		2 -0.037	-0.038	2.6156	0.270
		3 -0.043	-0.041	4.9818	0.173
		4 0.023	0.024	5.6669	0.225
		5 0.012	0.008	5.8475	0.321
		6 0.000	-0.000	5.8475	0.440
		7 -0.033	-0.031	7.2797	0.400
		8 -0.032	-0.030	8.5939	0.378
		9 -0.049	-0.050	11.732	0.229
		10 0.049	0.047	14.921	0.135

Table 10. Characteristics of ARCH(4) model.

Variable	Coefficient	Std. Error	z-Statistic	Prob.
Variance Equation				
C	4.38E-06	2.92E-07	15.00211	0.0000
RESID(-1)^2	0.104930	0.025855	4.058421	0.0000
RESID(-2)^2	0.101053	0.028219	3.581006	0.0003
RESID(-3)^2	0.150503	0.033467	4.497066	0.0000
RESID(-4)^2	0.085457	0.022812	3.746185	0.0002
R-squared	-0.000862	Mean dependent var		8.22E-05
Adjusted R-squared	-0.003941	S.D. dependent var		0.002802
S.E. of regression	0.002808	Akaike info criterion		-9.001033
Sum squared resid	0.010248	Schwarz criterion		-8.981209
Log likelihood	5878.174	Hannan-Quinn criter.		-8.993597
Durbin-Watson stat	1.999299			

A Case Study of Failure Parameter Estimation in Software Reliability Models

Stanislav Chren and Barbora Buhnova

Masaryk University, Faculty of Informatics
Botanická 68a, 602 00 Brno
{chren, buhnova}@mail.muni.cz

Abstract

The reliability is an essential quality attribute of contemporary software. The analysis of software reliability is governed by formal analytical models whose accuracy depend strongly on the estimation of their input parameters. This paper aims to investigate the practical limitations of failure parameter estimation methods. Total of 11 methods is applied in a case study of the Mozilla Firefox software. Discovered issues pertaining to the application of the methods are discussed and possible improvements are outlined.

Keywords

software reliability, parameter estimation, data collection, testing, growth models, metrics, open source, software architecture, Mozilla Firefox

Abstrakt

Spolehlivost je atribut základní kvality moderního software. Analýza spolehlivosti softwaru se řídí formálními analytickými modely, jejichž přesnost závisí silně na odhadu jejich vstupních parametrů. Tento příspěvek se snaží zkoumat praktické omezení metody odhadu parametrů selhání. Celkem 11 metod se aplikuje v případové studii Software Mozilla Firefox. Jsou probrány zjištěné problémy týkající se používání metod a možná zlepšení jsou uvedeny.

Klíčová slova

spolehlivost softwaru, odhad parametrů, sběr dat, testování, modely růstu, metriky, open source, softwarová architektura, Mozilla Firefox

1 Introduction

An ever increasing portion of human activities is becoming dependent on computer systems. In many cases, a software failure can endanger human lives or cause extensive monetary losses. Therefore, the issue of software quality has been receiving a growing attention. In software engineering, reliability is the quality attribute associated with software failures. Over the last decades, many formal models for reliability analysis have been proposed. However, no matter how advanced the models are, the quality of their results is critically influenced by the quality of the input data and the accuracy of model parameters.

In our previous work [16], we investigated which data sources and methods can be utilised to estimate failure parameters of software architecture-based models. Many of the methods identified in [16] been introduced decades ago with different types of software and development processes in mind.

The goal of this paper is to build upon our work by evaluating the applicability of these methods on a contemporary large software system and identify their practical limitations.

The contribution of this paper is the assessment of challenges which the practitioners of the software reliability analysis have to overcome. This information might guide the selection of suitable failure parameter estimation methods based on the available data sources. Moreover, it can help to set up the

software development process so that the important data for given methods are collected in sufficient amount and quality.

In the case study, we consider multiple distinct failure parameter estimation methods which makes our work unique among others in the domain of software reliability analysis.

The paper is structured as follows. Section 2 gives an overview of related work in the area of failure parameter estimation, Section 3 provides necessary background in software reliability, Section 4 describes the classification of methods for failure parameter estimation, Section 5 details the case study, Section 6 discusses the results of the case study. Finally, Section 7 concludes the paper.

2 Related Work

There are several publications that discuss methods for failure data collection. First, there are authors who have surveyed the methods for failure parameter estimation and data collection. However, the scope of these papers is usually limited to a single domain of parameter estimation methods. Also the practical restrictions of the discussed methods are often not considered.

Dimov et al. [10] focus on the testing methodologies used in reliability analysis but they do not discuss how the testing data can be transformed into failure parameters. Murphy et al [26] provide an overview of data collection methods based on customer-related data, such as questionnaires, customer service calls or bug reports. However, they do not explain if and how the methods could be used to derive reliability model parameters. The Handbook of Software Reliability by Lyu [22] is a comprehensive source of both theoretical and practical knowledge on reliability analysis processes including the data collection and parameter estimation. Nevertheless, it has been published nearly two decades ago and focuses predominantly on black-box models and testing.

Second, there are papers, which introduce new reliability model, such as [8, 3]. Although they contain case study sections and they outline the parameter estimation, the proposed models are often applied on artificial examples and the consideration of practical issues is neglected.

On the other hand, there are large case studies [20, 16, 30], which attempt to perform detailed architecture-based reliability analysis and also discuss potential shortcomings of the whole process. The drawback of the current work is their limited selection of failure parameter estimation techniques. Our work attempts to complement them by evaluation of variety of data sources and parameter estimation methods.

Koziolek et al. [20] used a combination of bug report analysis and software reliability growth models in analysis of a complex industrial system. They also briefly mention other possible methods for parameter estimation, but they are not applied in a case study.

Popstojanova et al. [16] performed another large case study. They used the fault injection together with input domain model and utilized data from change logs to estimate the software parameters. Furthermore, they discussed validity of various simplifying assumptions that are often made during the reliability analysis.

Nguyen et al. [30] inspects their experience with unsuccessful application of SRGMs. They point out the limitations of the models with respect to the quality of input data.

3 Reliability Analysis

This section gives overview of basic principles and definitions used in software reliability analysis domain.

3.1 What is Reliability

Throughout the literature, the exact definition of reliability varies from source to source. According to the standard IEEE 1633-2008 [1] the reliability is defined as: "*The probability that software will not cause a failure of a system for a specified time under specified conditions.*"

In practice, we can encounter other definitions depending on a type of software application. The first definition of reliability is more useful for terminating applications, which act on user request. Sometimes, it is also called *reliability on demand*. For continuously running applications (such as embedded software), the reliability can be defined as "*The failure intensity during specific time interval*" (i.e. rate of failure occurrence for a given period of time) [22].

3.2 Reliability Models

In the domain of software reliability analysis, great number of formal models have been proposed so far [19]. Based on their view of the system, they can be broadly classified into two categories.

The *black-box* reliability models analyse the reliability of the whole application while ignoring its internal structure. They are employed in later stages of software development or they are used on systems that are already deployed. They focus mostly on quantification of failures and down-times. Their main representative are the Software reliability growth models (SRGMs).

The *white-box* reliability models (also called architecture-based models) consider internal structure of an analysed software in terms of individual components, interfaces and possibly the deployment hardware. Compared to the black-box models, they offer several advantages. First, they can be used in earlier stages of software development, especially at design time when the crucial decisions about the application architecture and implementation are made. They can be used to identify critical components that have the highest impact on overall application reliability. Additionally, it is possible to use them for *what-if* analysis to compare different architectural design decisions. The most advanced models are for example Palladio [4], scenario-based model by Yacoub et al. [33] or the model by Cheung [8].

We concentrate on the architecture-based models. Since the contemporary software is component-based, the prevalent research in the reliability analysis is also the most significant in the area of white-box models.

3.3 Reliability Model Parameters

The input parameters of architecture-based reliability analysis approaches can be classified into three categories: the failure parameters, behavioural parameters and execution environment parameters.

Failure parameters

The failure parameters describe the failure behaviour of an element in question (whole system, components, scenarios, methods etc.). Majority models use three types of failure models: The *probability of failure* is the probability that software will cause a failure of a system or component [22]. The *constant failure rate* is defined as the number of failure occurrences per unit of time [14]. The *time-dependent failure intensity* is a rate of change of expected number failures with respect to time [22].

Behavioural parameters

The behavioural parameters model the operational profile of the system. According to Musa [27], the operational profile is a quantitative representation of how the system will be used. A whole operational profile consists of multiple characteristics, such as customer profiles, user profiles or functional profiles. The most typical representation of the operational profile in the architecture-based models are the transition probabilities between components.

Environmental parameters

The environmental parameters describe the execution environment in which the software is deployed. It mainly consists of the hardware resources and network infrastructure. The reliability of the hardware resources tends to be expressed via the Mean Time To Failure (MTTF) and Mean Time To Repair (MTTR) attributes. Reliability in networks may be defined either as a probability of successful communication between a specified pair of nodes within the network, or as the ratio of correctly delivered data.

In our work, we focus on the failure class of parameters because we consider them to be the most challenging to estimate. They can be obtained from the largest variety of data sources and methods. The behavioural parameters can be reasonably estimated only by monitoring the user behaviour and collecting usage statistics from profiling. The environmental parameters are only rarely considered by the architecture-based reliability models and they are usually supplied by the hardware vendors and infrastructure providers.

4 Methods For Failure Parameter Estimation

This section provides an overview of the failure parameter estimation methods. Based on a more detailed analysis of these methods in our previous work [6] we devised a classification framework depicted in Figure 1. Each of the methods belongs to one of the seven categories on the left. For each category we distinguish three relevant dimensions:

- **WHAT – Reliability model parameters.** This dimension describes what kind of data the techniques collect or estimate based on the parameters of architecture-based reliability prediction models that shall be constructed. It can be either probability of failure or constant failure rate or time-dependent failure intensity (see Section 3.3).
- **WHERE – Sources of data collection.** This dimension characterizes the information sources exploited by the data collection techniques, i.e. where the analysed data comes from. There is a variety of data sources that are available across the whole development life-cycle. The design-time artefacts are typically represented by UML models. Late-stage artefacts are a source code, change logs or testing suite. Runtime artefacts are produced during the execution of the software and are represented by the coverage data, testing reports or application logs. User data are information about the failures submitted by people as bug reports. Additional data might be obtained from the previous versions of the software. In theory, the third party commercial off the shelf components (COTS) might be already certified for a certain reliability level. If none of the above is available, the information provided is often used.
- **HOW – Collection process.** This dimension represents the activities contributing to the data collection/estimation process, i.e. how the data collection techniques obtain and process the data. The process can comprise measurement activities, such as test executions, simulation or data filtration, the metrics collection activities and analytical activities which involve using mathematical functions and models to transform input set of parameters to the reliability-relevant estimates.

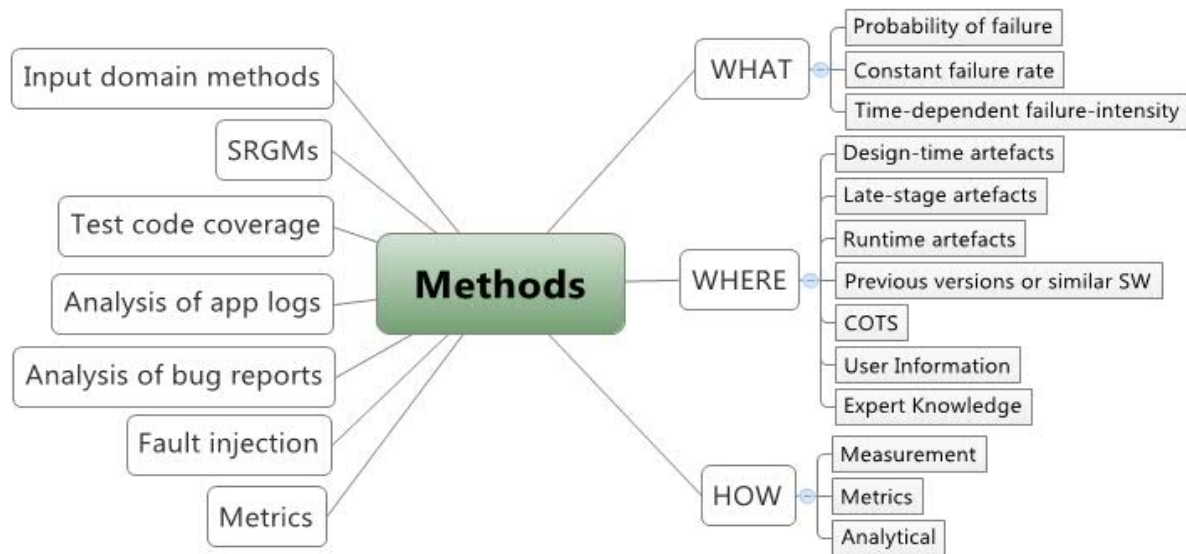


Figure 1: A classification scheme for failure parameter estimation methods

4.1 Input domain methods

The input domain techniques are based on random testing driven by the knowledge of system operational profile. First, the test cases are developed or generated according to a known distribution of the input elements (i.e. operational profile). Next, the sampled test cases are executed and the number of failed tests is recorded. Resulting ratio of number of failures to the total number of executions serves as an estimate for probability of failure. Depending on the test cases, we can use them to compute the probability of failure for the whole application (e.g. in case of functional testing suite), its components (if each component has separate test cases) and even the individual methods (e.g. unit tests).

The simplest method is the Nelson's model [29]. If the testing reveals no failures, the models based on Bayesian framework, such as the Miller's model [24] are more suitable.

4.2 Software reliability growth models

The basic idea behind the SRGMs is simple: If the history of fault detection and removal follows a certain recognizable pattern, it is possible to describe the mathematical form of the pattern. The function that represents this pattern is called mean value function $m(t)$, which is a cumulative number of faults detected by given time t . If we are able to fit it to the existing historical fault detection data, we can predict the future failure behaviour of a software. The derivation of the mean value function results in the time-dependent failure intensity.

Throughout the last 50 years, over 100 SRGMs have been introduced. The most common examples are the Goel-Okamoto model, S-shaped model or Littlewood-Verall model [22].

4.3 Test code coverage models

Test code coverage is a measure that describes the degree to which certain elements of the source code have been tested. In general, the methods based on the code coverage employ the same principle as SRGMs. They also use the mean value function to express the cumulative number of faults in given time that can be transformed to failure rate function. In fact, they usually use the same mean value

functions as SRGMs. The difference is that the parameters of these functions are estimated from the collected coverage data.

The general approach of relating the coverage growth to the reliability growth was described by Gokhale and Trivedi [15].

4.4 Analysis of application logs

The parsing of application logs is a popular approach for failure data reconstruction, which may be also employed for reliability-model parameter estimation. The logs are often voluminous, containing much information irrelevant from the reliability point of view. Hence the log files must be filtered first. Afterwards, the remaining entries are scanned to produce traces of system and user's failure related behaviour.

Unfortunately, a formal specification of the log parsing and filtration procedures have been missing so far. The exception is the method for analysis of web server application logs proposed by Banerjee et al. [2].

4.5 Analysis of bug reports

In a number of works, bug tracking databases have become an essential source for failure data collection [20, 34]. The reliability-model parameter estimation process in these approaches typically consists of two phases, where the first filters and analyses bug reports to collect raw failure data, which is then injected into an analytical model to compute the expected parameter values. The latter phase is most commonly based on Software Reliability Growth Models (SRGMs).

There are three distinct methods in this category. The method by Koziol et al. [20] focuses on filtration of the reports so that they adhere to the assumptions of the SRGMs. Brosch's [5] approach is based on qualitative analysis of the reports. The model by Nakagawa et al. [28] revolves around the classification of the reports according to their complexity.

4.6 Fault injection

A fault injection is a multi-purpose technique that is used for simulating events or conditions that are difficult to observe otherwise. The program with injected faults is usually examined to evaluate the fault propagation, latency, fault recovery mechanisms or test effectiveness in connection with mutation testing. The fault injection can be performed manually, which can be cumbersome in larger applications or it can be automated, for example by executing newer versions of the test cases on the older versions of the software [16].

A fault injection does not result in failure parameters on its own. It needs to be coupled with either SRGMs [7] or input domain models [16]. Another application of fault injection are the capture-recapture models represented namely by Mill's hypergeometric model [25].

4.7 Metrics

There are several ways of how metrics can be leveraged to estimate values of failure parameters. First is the model incorporating the defect density metric to produce the failure rate parameter [11].

Another technique is called GERT which is both an automated tool for collecting mostly source code related metrics and also a linear regression model for estimation of failure probability for the whole application or its parts [9].

For the early quality analysis of the software, the object oriented (OO) metrics have been developed and they can be used to estimate the risk-based reliability of a system or its components [17].

Finally, when the relationship between various metrics and their impact of failure parameters is uncertain and when a great deal of expert knowledge is involved, the Bayesian Belief Network (BBN) models seem most appropriate [31].

5 Case Study

In this section, we demonstrate the applicability of the methods analysed in [6]. We discuss the challenges and limitations we encountered during our experiments. The case study is based on the Mozilla Firefox web browser. First, we describe its architecture and available data sources. The rest of the section is devoted for the individual categories of failure data collection and parameter estimation methods.

5.1 Mozilla Firefox

Mozilla Firefox [13] is a free multi-platform open source web browser that is being developed by Mozilla Corporation. Presently, it is ranked as the third most popular web browser with almost 500 million users worldwide.

Architecture

The architecture of the Firefox is component-based and follows the layered architectural pattern. It provides interfaces from higher level components to interact with components on lower levels. The layered approach also allows components to be developed independently on each other. Moreover, since all interaction is done through interfaces only, one component implementation can be replaced with another without affecting the rest of the application. The Firefox's architecture consists of the following components (for graphical representation, see Figure 2):

- **User Interface (UI)** - provides means for user interaction with browser engine. It offers standard features the user might expect from a web browser, such as toolbars, printing, downloading etc. It is implemented with a set of packages named XPToolkit.
- **Browser and rendering engines** - the browser engine is responsible for initiating high-level browsing actions, such as loading the URL, reload page, etc. The rendering engine produces the visual representation of the web page. Some of its responsibilities like HTML, XML and JavaScript interpretation are delegated to lower-level components. In Firefox, both engines are part of Gecko component.
- **Data persistence** - manages local storage of user and browser data. It uses custom Storage API for interaction with internal SQL database.
- **Networking** - handles the communication and security over the network protocols, such as HTTP and FTP. In Firefox, all network related services are implemented by a Necko module.
- **JavaScript interpreter** - executes JavaScript code located within a webpage. The results are passed to the Gecko component. It is implemented by SpiderMonkey engine.
- **XML parser** - it is responsible for parsing of XML data. The Firefox uses various XML dialects. XUL is used for construction of user interface. RDF is used for data storage and the XPCOM for management of component objects. The Firefox utilises third-party XML parser Expat

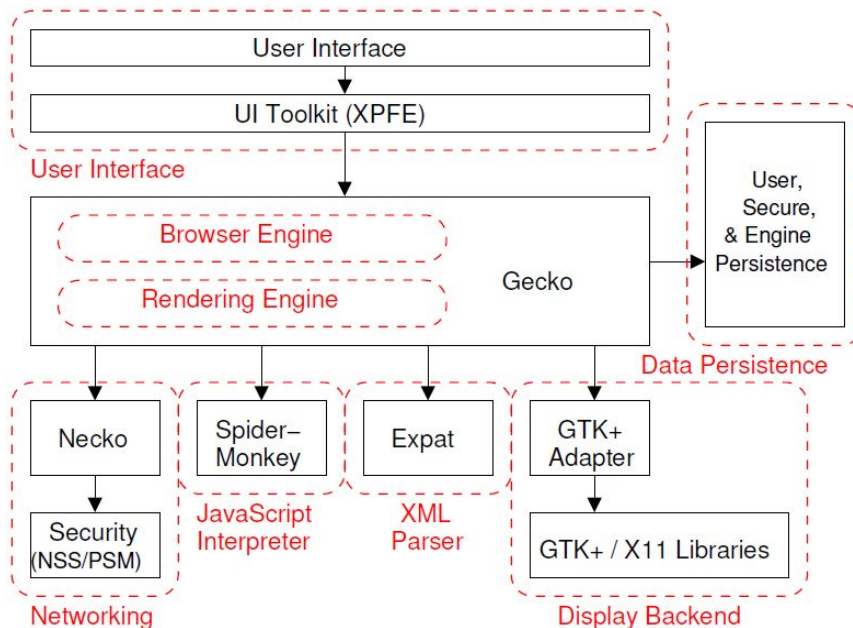


Figure 2: Overview of the Firefox's architecture [16]

Available data sources

Since Mozilla Firefox is a large open source application with an active community of developers, it offers variety of publicly available data sources:

- **Design time artifacts** – the official documentation and related Mozilla knowledge portal do not contain design time UML diagrams. However, the project schedule is being maintained that shows dates of past and planned future releases. This information was leveraged in bug filtering for SRGM application.
- **Late stage artefacts** – the source code of the Firefox is available and is well maintained. It can be managed with the Mercurial and Github version control systems. Alternatively the Mozilla provides the FTP access to current and past releases. The Firefox utilizes multiple types of testing suites with high emphasis on testing automation. Since the source code is maintained by multiple control version systems, the change logs are automatically generated. They can be also accessed via web interface.
- **Runtime artefacts** – the automated build system generates runnable night-builds based on the most recent code in the repository. For testing and debugging purposes the developers are encouraged to make their own customized builds of the code. The compilation is a time consuming process, which depends on the HW configuration of the machine the code is built on. In our case, the compilation times ranged from 66 minutes (CPU 3.3 GHz x3, 3GB RAM) to 220 minutes (CPU 2.2 GHz, 1GB RAM). The build system also keeps track of the test results for every night build. Until the February, 2013 also the weekly coverage reports generated by the automated testing were stored. The measured coverage elements were the line, function and branch coverages.
- **Previous versions or similar software** – as was mentioned before, the Firefox employs control version systems to manage the source code making the previous versions easy to

access. Regarding the similar software, the web browsers are a specific category of software. They all share common features, functionality and elements of user interface. Additionally, modern web browsers are based on the same conceptual architecture [18]. The main difference is usually the browser and rendering engine. Currently, the other most popular browsers are the Google Chrome, Internet Explorer and Safari [32].

- **COTS** – the Firefox does not use COTS per se. However, some of its components can be built as a stand-alone modules that can be reused in different applications. For instance the Gecko engine is also used in Avant web browser [12] and many user interface elements are shared among other Mozilla suite applications. On the other hand, Firefox incorporates the Expat XML parser, which is a component being developed by a third party. None of these components have been certified or evaluated separately with any quality attribute in mind.
- **User Information** – the bug reports are the main source of failure data. In Firefox, they are managed by the Bugzilla bug tracking database. The bug reports can be submitted by both developers and users. Currently, there is over 400,000 registered Bugzilla accounts. The developers are required to follow specific guidelines when submitting a bug, thus trying to prevent inconsistent or missing information. The data about operational profile are not available.
- **Expert knowledge** – we are neither members of the Firefox development team nor we have extensive knowledge of its implementation. While conducting this case study, we leveraged only the available data sources and documentation.

5.2 SRGM

When applying SRGMs, we have basically two choices regarding the source of input data. First are the failure reports obtained during the testing phase. The second are the problem (bug) reports that can be used as a replacement if the testing failure reports are not available.

The Firefox development process is agile in nature and does not have a separate testing phase. The testing is carried out regularly while the coding is still under way. Therefore, no separate long term testing reports are being maintained. Instead, all discovered faults and failures are reported as bugs in bug tracking database. Hence, we use the bug analysis approach for obtaining the input data for SRGMs.

Data filtration

First step is to filter the bug reports so that the input data correspond to the assumptions of SRGMs as closely as possible. We followed the filtration scheme described in [20].

In the case study by Koziol et al. [20], there were two components which did not have enough related reports in the bug tracking database. Therefore, we also performed the analysis for individual components according to the architecture outlined in Section 5.1. Since Koziol et al. suggested we should use reports for a particular release of the application, we performed the filtration for ten past releases in order to identify the release that could provide us enough data.

The results of our first attempt are shown in Table 1.

Release	UI	Gecko	Networking	JS Engine	XML Parser
20.0	0	7	1	2	0
21.0	0	2	0	3	0
22.0	1	9	0	4	6
23.0	0	14	0	5	2
24.0	2	2	1	10	1
25.0	0	5	3	1	2
26.0	0	2	0	1	1
27.0	0	3	1	0	0
28.0	1	2	0	0	0
29.0	0	1	0	0	0

Table 1: Counts of bug reports (critical and blocker severity) for particular releases and components

As can be observed, there is not enough data for any given component and given release. The most bug reports can be attributed to the Gecko component and the release version 23.0. For instance, the documentation of the CASRE tool recommends that there is data about at least 40-50 failures for reasonable analysis.

Considering the blocker and critical severities only, we obtained the reports describing the hard crashes of the Firefox. However, the failure does not have to manifest as crashes. Therefore, in order to obtain more data, we decided to relax the filtration parameters and to include the failures also with severities normal and higher. Selection of even lower severity classes is possible. However, this would also include reports describing requests for new features, convenience tweaks and other enhancements which do not represent failures. The failure counts for the severities with normal and higher are shown in Table 2.

Table 2: Counts of bug reports (severities normal and above) for particular releases and components

Release	UI	Gecko	Networking	JS Engine	XML Parser
20.0	11	18	2	3	1
21.0	3	14	0	4	2
22.0	5	30	0	6	8
23.0	5	26	2	7	3
24.0	9	11	1	13	2
25.0	5	15	9	6	6

26.0	5	14	3	3	2
27.0	6	10	2	2	3
28.0	16	17	2	1	2
29.0	32	25	0	2	5

As we can see, the number of reports is higher, namely for the Gecko and the User interface components. Although still far from optimal quantity, we further investigate the Gecko component in release version 22.0 to determine whether the data actually exhibit a reliability growth.

Application of SRGMs

Based on the data from Table 2, we try to apply the SRGMs on the Gecko component from the Firefox 22 release. We process the data in both time between failures (TBF) and failure counts formats in order to examine larger variety of SRGMs. The time between failures input data are shown in Table 3 and the failure count data are in Table 4.

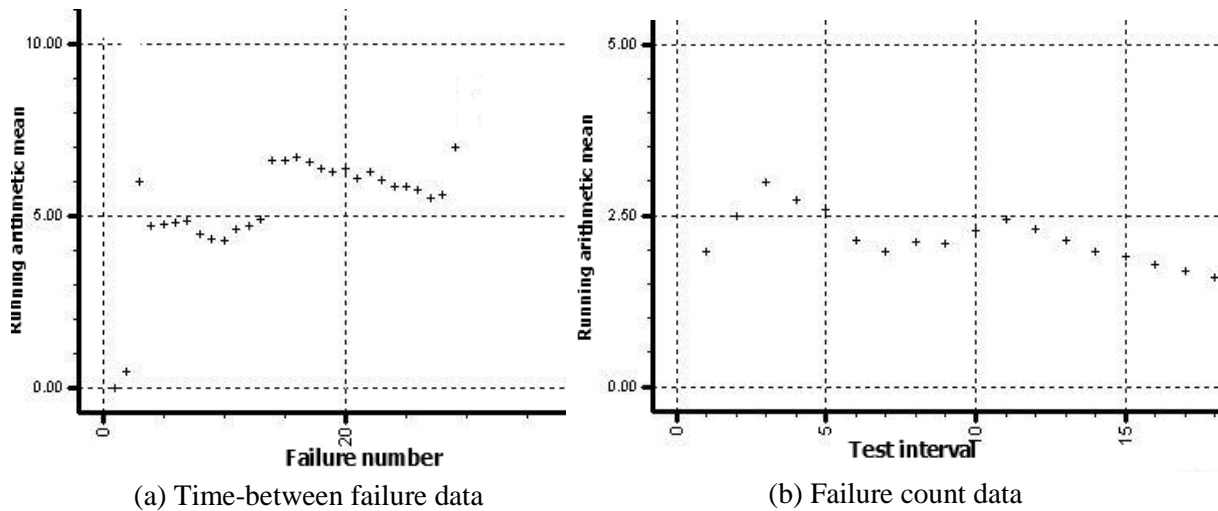
Before application of the SRGM it is recommended to perform a trend test to determine whether the data actually follow a reliability growth trend. We use the running average trend test. For TBF data, the average TBF should be growing with time in order to exhibit reliability growth trend. For failure count models, the trend should be decreasing as the number of failures should decrease with time. The trend test results are in Figure 3. As can be observed, only the failure count data show desired trend. Therefore, we continue the SRGM application with the failure count data.

Table 3: Time-between-failure data for Gecko component (Firefox 22.0)

Failure #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
TBF (Days)	0	1	17	1	5	5	5	2	3	4	8	6	7	29	7
Failure #	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
TBF (Days)	8	4	3	5	8	0	11	1	1	6	3	0	8	45	50

Table 4: Failure counts data for Gecko component (Firefox 22.0) in two-week intervals

Interval	1	2	3	4	5	6	7	8	9	10
Failures	0	1	17	1	5	5	5	2	3	4
Interval	11	12	13	14	15	16	17	18	19	
Failures	8	4	3	5	8	0	11	1	1	

**Figure 3:** Running average trend tests

For SRGM analysis we used the reliability modelling tool CASRE. Based on the input data, it allowed us to apply the Generalised Poisson model, the Non-homogeneous Poisson model and the Yamada S-Shaped model. The graph with the original data and the fitted SRGMs is shown in Figure 4.

Afterwards, we conducted a Chi-square test for the goodness-of-fit analysis. The results are shown in Table 5.

The results shows that the Generalised Poisson model is the best fit four data followed by the Yamada S-shaped model. The NHPP model does not even fit the data at the 5.% significance level. The reliability related results for the Generalised Poisson and Yamada S-shaped models are shown in Table 6.

Table 5: Results of goodness-of-fit analysis

Model	Chi-square	DOF	Significance
Generalised Poisson	5.593	2	6.1%
NHPP	13.055	3	0.45%
Yamada S-shaped	7.43	3	5.93%

Table 6: Results of SRGM analysis for Gecko component

Model	Reliability	Failure rate	Total failures (parameter a)	Proport. constant (parameter b)
Generalised Poisson	0.69	0.026	33.4	0.008
Yamada S-shaped	0.84	0.014	31.1	0.02

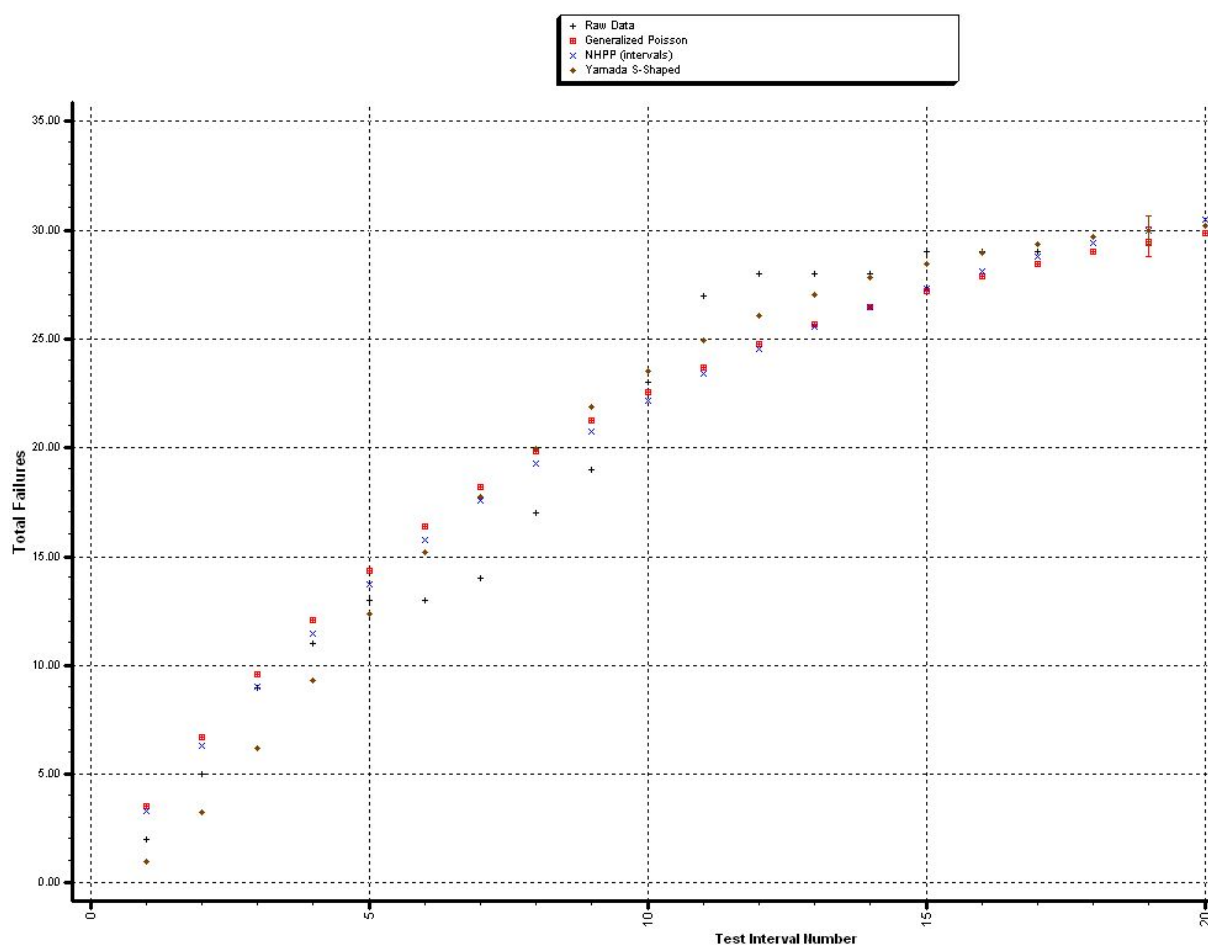


Figure 4: Application of SRGMs on failure count data

5.3 Bug Report Analysis

The bug analysis by Brosch [5] could be applied in our case. However, an in-depth knowledge of the Firefox and its components is required to provide the value for the proportional constant and to identify possible failure types. Additionally, the proposed approach requires detailed manual processing of the individual entries which is impractical for large scale application, such as the Firefox.

In theory, the error complexity model by Nakagawa et al. [28] is applicable. However, it also assumes an expert knowledge is involved in the categorization process. In their original work [28] it was done by five experts including the developers and the members of the quality assurance team.

This could be improved by devising such classification scheme that could leverage the filtration capabilities of the bug tracking database. Nevertheless, no such classification has been proposed neither by the original authors nor by anyone else.

Furthermore, there would be a problem with finding a mature application of similar type. In our case, it should be a stable web browser that has been in operation for at least two years. We cannot use the previous versions of the Firefox, since it has rather rapid development cycle with release every six weeks. That also goes for other web browsers, such as Chrome or Opera.

On the other hand the Microsoft's Internet Explorer could be a good candidate, since there is usually one to three years between releases. Unfortunately, the Internet Explorer is not an open source application. Although there is a "feedback" system where users can submit their problems, it lacks the filtration capabilities of the Firefox's Bugzilla. Moreover this system is used by the end users only and the reports usually lack sufficient technical information for proper classification of the reports. The internal bug tracking database used by the developers is not public.

The details about application of bug analysis method by Koziol et al. [20] have been already presented in Section 5.2.

5.4 Input Domain Models

The Firefox source code comes bundled with very extensive testing suites making the input domain models seemingly simple to apply. The main assumption for using this method is that the selection of test cases follows known operational profile. Unfortunately, in our case, the operational profile is not known. Nonetheless, some methods, such as Gert [9] used Nelson's model for estimating reference probability of failure based on the unit test executions that did not reflect the operational usage. Therefore, we also decided to investigate results from different testing suites.

Functional testing

First we executed the functional tests from the Mochitest suite. These tests focus on evaluation of the Gecko engine and the user interface component. Due to their large number, it is possible to divide the test execution into five smaller chunks. Our results from this testing are in Table 7.

Based on the results, the reliability of the Gecko engine according to the Nelson's model would be 99,987%. It clearly does not correspond with the observations from the bug report data and the subsequent SRGM analysis (Section 5.2) thus further confirming unsuitability of the Nelson's model in this case.

Table 7: Results from the Mochitest suite execution.

Test part	Execution time (minutes)	Total test cases	Passed	Failed	Prob. of failure (Nelson's model)	Prob. of failure (Miller's model)	Variance (Miller's model)
1	54	142 301	142 269	32	0.22E-3	-	-
2	25	216 362	216 327	35	0.16E-3	-	-
3	12	51 240	51 240	0	0	0.02E-3	3.8E-10
4	13	13 530	13 530	0	0	0.07E-3	5.4E-9
5	18	244 360	244 343	17	0.07E-3	-	-
Σ	122	667 793	667 709	84	0.16E-3	0.02E-3*	2.4E-10*
* For parts 3 and 4 only							

Unit testing

After the Mochitest functional testing we wanted to investigate the unit tests for the browser's components. We leveraged the JavaScript based xpcshell unit test suite as it is available for all

components and contains more test cases than standard compiled C++ unit tests. The results can be found in Table 8. Similarly to the Mochitest testing, also the unit tests exhibit very few failures.

Table 8: Results from the xpcshell test suite execution.

Component	Execution time (seconds)	Total test cases	Passed	Failed	Prob. of failure (Nelson's model)	Prob. of failure (Miller's model)
SpiderMonkey	95	63	62	1	0.02	-
Necko	225	279	277	2	0.01	-
XML Parser	13	10	10	0	0	0.08
XPTToolkit	2073	959	957	2	2.09E-3	-

Automated failure injection

In order to collect more data, we decided to try the automated fault injection technique from Popstojanova et al. [16]. It is based on executing test cases from newer version of the software against the older version.

We conducted our experiment on the Necko component by taking the xpcshell unit tests from the Firefox version 29.0 to the 22.0 beta version. The transfer consisted of the copying the newer test cases to the respective folder of the older version. Furthermore, the xpcshell.ini had to be modified to incorporate the newly added tests.

After the recompilation and execution of the test suite, we observed 218 failed test cases from the total of 283. The examination of the test logs revealed 75 cases whose failure was caused by testing the presence of the bug that was fixed in the newer version. Additionally, there were 10 failures referencing the functionality that was not yet implemented in the older version. The causes of the remaining failures could not have been identified from the test logs.

5.5 Code Coverage

For the code coverage analysis, we used the method proposed by Gokhale and Trivedi [15].

Source code preparation

First step was to instrument the source code of the Firefox with probes that would detect the visits in pieces of the code during the testing. The instrumentation was performed at compilation time. The Firefox source code was already bundled with the gcov coverage tool. All that needed be done was to enable the coverage measurement in the build configuration file.

After the coverage-enabled build was ready, we were able to execute the test cases. Depending on the selected tests, we could measure the coverage of the whole Firefox or its parts. We collected the coverage data for the network component Necko (Section 5.1) by executing its xpcshell unit tests. This test suite contains total of 279 test cases written in JavaScript.

We measured the line, function and branch coverage as these are the only coverage types offered by the gcov tool. In order to estimate the parameters of the coverage function $c(t)$ we wanted to observe the continuous growth of the coverage with testing progression. Unfortunately, the gcov and no other known unix-compatible coverage tools provides the analysis of coverage trends.

Test execution

To compensate for the limitation, the testing was initiated repeatedly. After a certain amount of elapsed time, we manually terminated the test execution. The intervals between the execution and termination were continuously increasing so that we could observe the coverage growth.

After each execution, the coverage data were collected using the lcov tool. The output was processed by genhtml tool to generate the graphical HTML reports.

The reports show the aggregated data for individual source code folders, but not for the whole component. Although there is a total coverage at the top of the report, it also incorporates statistics from additional files that were accessed during testing and which do not belong to the Necko component. Therefore, we further mined the reports for the Necko's data only. All the reports and extracted information is included in the electronic attachment of this thesis.

Coverage measurement results

The final coverage results are shown in Table 9. The graphical representations are in Figure 5. It can be seen that in our case, the difference between the time units is not significant. Therefore, in the following analysis we work only with the execution time.

Table 9: Coverage analysis results

Measurement no.	1	2	3	4	5	6	7
Execution time (sec)	75	131	247	305	425	484	603
Test executed	37	44	97	129	193	224	279
Lines total	86 714	86714	86 714	86 714	86 714	86 714	86 714
Lines covered	12 180	15 840	20 116	21 167	23 981	25 201	27 184
Lines coverage	0.140	0.183	0.232	0.244	0.277	0.291	0.313
Functions total	10 889	10 889	10 889	10 889	10 889	10 889	10 889
Functions covered	1 823	2 386	2 923	3 102	3 454	3 618	3 876
Functions coverage	0.167	0.219	0.268	0.285	0.317	0.332	0.356
Branches total	72 101	72 101	72 101	72 101	72 101	72 101	72 101
Branches covered	6 152	10 153	10 867	11 497	13 377	14 109	15 073
Branches coverage	0.085	0.141	0.151	0.159	0.186	0.196	0.209

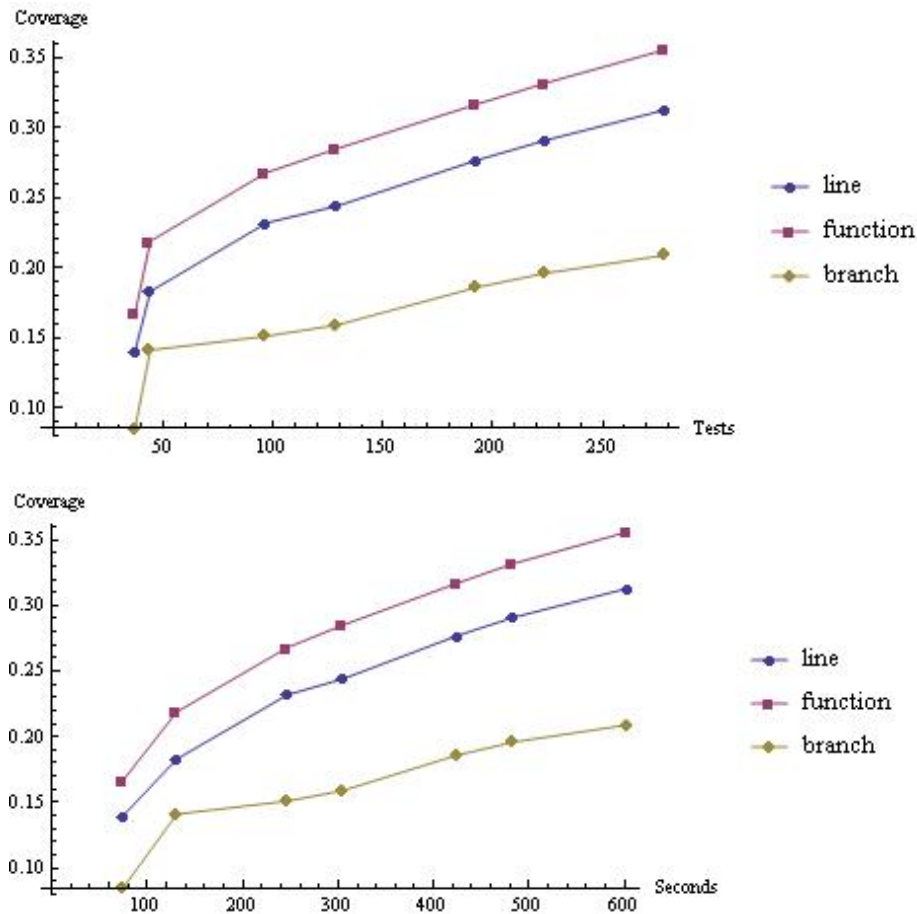


Figure 5: Overview of the measured coverage data

Parameter estimation

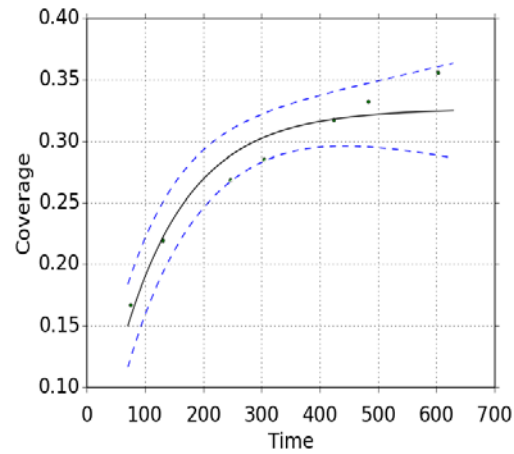
The next step was to determine the parameters for the coverage function $c(t)$. We selected the exponential model (Table 16) for further analysis as estimation of their parameters is the most straightforward. We performed it for all the coverage types. We used the least squares method for parameter estimation. The results are shown in Figure 6 and the numerical values are in Table 10.

Table 10: Parameter estimations for exponential model with goodness-of-fit test

Coverage type	Parameter b	Parameter g	Chi-square
Line	0.283	0.008	0.0018
Function	0.326	0.009	0.0019
Branch	0.192	0.008	0.0011

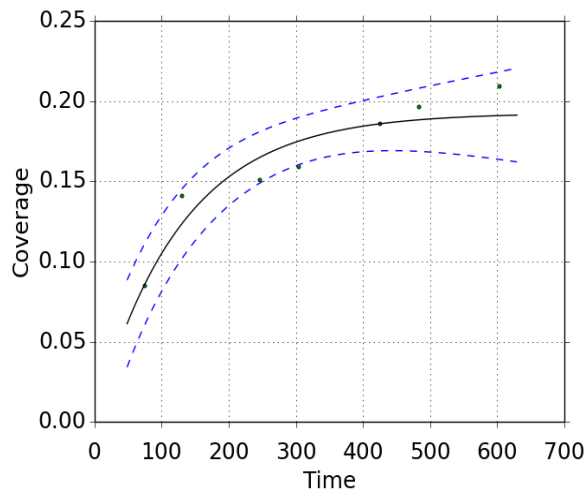
To finally obtain the failure intensity functions, we needed to determine the parameter a representing the total residual number of faults in the application. The authors suggested in [15] an approximation by dividing the number of detected faults with the measured coverage. In our case, the unit tests lead only to two failures. If we assumed that each failure was caused by different fault, the number of residual faults would be about 7. Unfortunately, we do not have means to verify this result. Under

different circumstances, we could use the SRGMs to estimate the total number of faults. However, as we showed in Section 5.2, the Necko component does not have enough data for proper SRGM analysis.



(b)

(a)



(c)

Figure 6: Overview of the fitted data with 95% confidence interval for (a) line, (b) function and (c) branch coverage

The final lesson learned from the coverage analysis was that the processing of coverage data is time- and computational resource-consuming process. On our first attempt we tried to measure the coverage

for the Gecko component by executing the Mochitest suite. The first thing we observed was that the test executions take twice as much time on the coverage enabled build compared to the standard one. For an initial data collection, we terminated the testing after five minutes. We let the lcov tool to collect the data which resulted in out of memory error after unsuccessful allocation of another 30 GB of the disk space that was needed for processing.

5.6 Metrics

In this section we look into application of metric based methods. For the rest of this section, the measurements are based on the Necko component.

Defect density metric

The methods for defect density estimation are partially applicable, albeit with uncertain outcome. The defect density approach by Lipow [21] would produce the defect density $D = 0.0027$ (i.e. the Necko would contain about 443 faults in its source code). This value could be further adjusted by the Malayia and Denton method [23]. However, it is not clear which values for the tabular coefficients such as the average skill level of the team or the CMM level of the organisation should be used. The final derivation of the failure rate from the defect density is problematic as well.

First, we do not have means of evaluation of the fault exposure ratio K . Although we could use the sensitivity analysis to iterate over the whole possible range for the K , we still would not be able to determine the second parameter F representing the linear execution frequency of the program. For its quantification we would need the number of the source instructions for the Necko component. That could be accomplished by compilation of the source code into the assembly language. However, the build system of the Firefox does not provide such option.

UML based analysis

The approach by Popstojanova et al. [17] was not usable in our case. The documentation of the Firefox does not include the necessary UML diagrams. Their method for estimating the component risk-factor revolves around the state machine diagram, which is not derivable from the source code and must be supplied by the domain expert.

Bayesian belief network

We could not apply the BBN, because there is no BBN published with complete definitions of the probability tables and we lack the necessary expert knowledge for their calibration.

GERT model

The GERT model [9] relies on source code, testing and OO metrics. The source code metrics and the OO metrics can be easily collected by appropriate tools. The main challenge comes with the evaluation of the testing metrics and obtaining the reference reliability values from unit testing. The GERT was designed with the Java language code and JUnit testing framework in mind. None of those apply for the Firefox (or the Necko component). The Necko uses multiple unit testing frameworks, namely the standard compiled C++ tests and JavaScript based xpcshell tests. The former test suite contains less test cases and all of them pass without failure leading to the constant reliability value of 1. The latter one has more test cases and couple of them result in failure. However, since they are written in JavaScript we cannot use them for collection of OO metrics.

Therefore, we proceeded with the analysis considering only the C++ unit tests. For the number of assertions metric, the `assert()` methods from the JUnit library were used originally by the authors. In our case, we substituted them with the counts of the `NS_ASSERTION()` macro, which serves the similar purpose. Additionally, the number of testing files constituted the number of test cases metric. The rest of the metrics was collected using the CCCC tool.

Since the GERT is intended for continuous application throughout the development, we collected the metrics for three different releases of the Firefox. The results for the raw metrics are shown in Table 11. Afterwards, we derived the STREW suite metrics (see Table 17) with the results in Table 12.

Table 11: Summary of Necko metrics

Metric \ Version	29.0	24.0	20.0
SLOC	163 936	136 983	132 831
TLOC	6 100	6 027	5 919
Assertions	54	77	80
Test cases	37	49	48
Classes _{test}	58	57	53
Classes _{source}	894	746	753
CComplexity _{test}	1 224	1 208	1 198
CComplexity _{source}	35 682	30 140	29 037
CBO _{test}	192	184	176
CBO _{source}	5 336	4 238	4 168
DIT _{test}	17	16	15
DIT _{source}	284	223	232
WMC _{test}	192	188	185
WMC _{source}	6 888	5 475	5 490

Based on the multivariate regression we derived the parameters a_i which serve as weights for the STREW metrics. The results of the regression are in Table 13.

Although, we were able to successfully derive the parameters, they cannot be considered accurate because of the small input sets and limited reference values for the probability of failure.

Table 12: Parameters from for the STREW metrics from the regression model

Metric \ Version	29.0	24.0	20.0
SM1	3.29E-4	5.62E-4	6.02E-4
SM2	2.26E-4	3.58E-4	3.62E-4
SM3	1.46	1.57	1.67
SM4	0.57	0.58	0.63
SM5	0.03	0.04	0.04

SM6	0.04	0.04	0.04
SM7	0.06	0.07	0.06
SM8	0.03	0.03	0.03

Table 13: STREW metrics for Necko component

STREW Metric	SM1	SM2	SM3	SM4	SM5	SM6	SM7	SM8
a_i	-165.841	-0.014	0.123	0.383	1.55	6.209	2.423	8.279

6 Discussion of Results

This section summarizes the results of our attempts to apply the failure parameter estimation methods on the Mozilla Firefox web browser. In total, we investigated 11 methods, out of which 4 were at least partially applicable, 7 were not applicable at all. We could not use any of these methods without reservations. The main obstacles which prevented successful evaluation of the methods can be classified into five main categories:

- **Missing information** – some of the input data required by the methods was not available at all or was present in insufficient amount.
- **Expert knowledge** – the methods required information provided by an expert (member of the development team) as an input.
- **Development process** – the methods assumed that specific software development strategy was used which was not adhered by Mozilla Firefox life-cycle.
- **Tool support** – the methods are built around a particular software tool or framework, which has limited support for different programming languages.
- **Software type** – the method was designed for a specific type of software which does not match the purpose of the Mozilla Firefox

The overview of the limitations for different categories of failure parameter estimation methods is found in Table 14.

Table 14: Summary of practical limitations for the failure parameter estimation methods

Method category	Limitations
Input domain models	Missing information
Software reliability growth models	Development process, Missing information
Test code coverage models	Missing information, Tool support
Analysis of application logs	Software type

Analysis of bug reports	Expert knowledge
Fault injection	Missing information
Metrics	Expert knowledge, Tool support, Development process, Missing information

For the input domain models, the crucial limitation was the lack of data about the Mozilla Firefox operational profile. Although, the number of test cases is sufficient and the models can be formally used, the underlying assumption about selecting the test cases according to the operational usage is violated and therefore reasonable interpretation of the results hardly possible.

Software reliability growth models could not be used on they own, because the development of Mozilla Firefox does not involve isolated and continuous testing periods which are assumed by SRGMs. Thus, instead of data collected from testing, we leveraged data from the bug tracking database. However, the filtration of bug reports based on the approach by Koziol et al. [20] did not result in sufficient amount of inputs for SRGMs. More data could be obtained by relaxing the filtration conditions. The effect of the different filtration schemes on the prediction accuracy can be investigated in the future work.

The test coverage model was applicable on a smaller scale. However, in the future it may benefit from more advanced tool support that would allow effective coverage collection from large components and that would be able to evaluate the trends in the coverage growth. Moreover, test coverage models still require more input data related to the discovery of faults for estimation of the rest of the input parameters.

The analysis of application logs was impeded by inappropriate software types. The only method in this category was tailored for analysis of web server application logs which are not produced by Mozilla Firefox. In the future, more general approaches for log analysis should be devised as application logs represent one of the most common and self-contained runtime artefacts.

One of the methods for bug report analysis was used together with SRGMs, but unfortunately, it was not able to provide desired amount of input data. The other two approaches in this category were not applicable due to their high dependency on the expert knowledge required for the bug report classification. This area could benefit from incorporating methods for automated bug reports analysis that have been previously used in the requirements engineering domain for prioritization of requirements in agile development [35].

The fault injection complemented the application of input domain models. We attempted the automated fault injection by executing the test cases from newer versions of the Mozilla Firefox on the older versions. However, for proper evaluation, more detailed logging mechanisms would need to be implemented in order to distinguish failures caused by injected faults from failures caused by a missing functionality.

The metrics-based methods seemingly suffered from largest variety of limitations. However, this category of methods is also the most diverse one. We were not able to use the Bayesian networks because of the lack of expert knowledge. The application of GERT model was problematic because of incompatible testing frameworks and programming languages. The GERT model also requires continuous collection of specific metric data during software development, which is not a current practice in Mozilla Firefox development. We managed to collect required data to some extent by

using multiple versions of the Mozilla Firefox, but for realistic analysis, more data would be needed. The UML-based method by Popstojanova et al. [17] was not applicable, because necessary UML diagrams are not maintained by the development team.

Overall, the two most frequent limitations were the missing or incomplete information and the expert knowledge. Issues stemming from the lack of expert knowledge could be alleviated by enlisting the help of the members of Mozilla Firefox development team.

Dealing with the missing data poses a bigger challenge. A possible direction of future research could be a proposal of methods for aggregation of information from several data sources in order to produce more accurate failure parameter estimates. The first step would be an examination of inaccuracies involved in using incomplete data. This could be addressed by employing uncertainty analysis and sensitivity analysis approaches. The former is used for quantification of uncertainties in the input data, uncertainties in the parameter estimation model and also the propagation of the uncertainties into the resulting estimates. Examples of methods for uncertainty analysis include interval arithmetic, probability theory, evidence theory, possibility theory [36]. The latter, including techniques like the method of moments or Monte Carlo simulation is used to identify the input parameters which have the highest impact on the resulting estimate [37].

7 Conclusion

Software developers have been tackling with growing demand for high quality software over the last couple of decades. Substantial effort has been devoted into research of formal models for analysis of software reliability. The main obstacle preventing the wide-spread adoption of these methods is the estimation of their input parameters. In this paper, we expanded upon our previous work involving analysis of the methods for failure parameter estimation by their usage in a case study. We pointed out their practical limitations and suggested possible courses of solving the problems we have discovered. Some of the issues might have a straightforward solutions, others still require a more in-depth research.

In our future work, we focus on dealing with insufficient or missing information via detailed uncertainty analysis. Additionally, we would like to investigate whether these issues can be remedied by utilization of data from previous versions of the software.

8 References

- [1] IEEE Recommended practice on software reliability. IEEE STD 1633-2008, pages c1–72, June 2008.
- [2] Sean Banerjee, Hema Srikanth, and Bojan Cukic. Log-based reliability analysis of software as a service (saas). In Proc. of ISSRE'10, pages 239–248. IEEE, 2010.
- [3] Nikola Benes, Barbora Buhnova, Ivana Cerna, and Radek Oslejsek. Reliability analysis in component-based development via probabilistic model checking. In Proc. of CBSE'12, pages 83–92. ACM, 2012.
- [4] F. Brosch, H. Koziol, B. Buhnova, and R. Reussner. Architecture-based reliability prediction with the palladio component model. IEEE Trans. on Software Engineering, 38(6):1319–1339, 2012.
- [5] Franz Brosch. Integrated Software Architecture-Based Reliability Prediction for IT Systems. PhD thesis, Karlsruher Institut für Technologie, Karlsruhe, Germany, 2012.

- [6] Barbora Buhnova, Stanislav Chren, and Lucie Fabriková. Failure data collection for reliability prediction models: a survey. In Proceedings of the 10th international ACM Sigsoft conference on Quality of software architectures, pages 83–92. ACM, 2014.
- [7] Mei-Hwa Chen, Aditya P Mathur, and Vernon J Rego. Effect of testing techniques on software reliability estimates obtained using a time-domain model. Reliability, IEEE Transactions on, 44(1):97–103, 1995.
- [8] Leslie Cheung, Roshanak Roshandel, Nenad Medvidovic, and Leana Golubchik. Early prediction of software component reliability. In Proc. of ICSE’08, pages 111– 120. ACM, 2008.
- [9] Martin Davidsson, Jiang Zheng, Nachiappan Nagappan, Laurie Williams, and Mladen Vouk. Gert: An empirical reliability estimation and testing feedback tool. In Proc. of ISSRE’04, pages 269–280. IEEE, 2004.
- [10] Aleksandar Dimov, Senthil Kumar Chandran, and Sasikumar Punnekkat. How do we collect data for software reliability estimation? In Proc. of CompSysTech’10, pages 155–160. ACM, 2010.
- [11] Norman E. Fenton and Martin Neil. A critique of software defect prediction models. IEEE Trans. on Software Engineering, 25(5):675–689, 1999.
- [12] Avant Force. Avant web browser. http://www.w3schools.com/browsers/browsers_stats.asp, 1999.
- [13] Mozilla Foundations. Mozilla Firefox. www.mozilla.org/firefox/, 2002.
- [14] Swapna S. Gokhale. Architecture-based software reliability analysis: Overview and limitations. IEEE Trans. on Dependable and Secure Computing, 4(1):32–40, 2007.
- [15] Swapna S Gokhale and Kishor S. Trivedi. A time/structure based software reliability model. Annals of Software Engineering, 8(1):85–121, 1999.
- [16] K. Goseva-Popstojanova, M. Hamill, and R. Perugupalli. Large empirical case study of architecture-based software reliability. In Proc. of ISSRE’05, pages 43–52. IEEE, 2005.
- [17] Katerina Goseva-Popstojanova, Ahmed Hassan, Ajith Guedem, Walid Abdelmoez, Diaa Eldin M Nassar, Hany Ammar, and Ali Mili. Architectural-level risk analysis using uml. IEEE Trans. on Software Engineering, 29(10):946–960, 2003.
- [18] Alan Grosskurth and Michael W Godfrey. A case study in architectural analysis: The evolution of the modern web browser, 2007.
- [19] Anne Immonen and Eila Niemelä. Survey of reliability and availability prediction methods from the viewpoint of software architecture. Software and Systems Modeling, 7(1):49–65, 2008.
- [20] H. Koziol, B. Schlich, and C. Bilich. A large-scale industrial case study on architecture based software reliability analysis. In Proc. of ISSRE’10, pages 279–288. IEEE, 2010.
- [21] Myron Lipow. Number of faults per line of code. IEEE Transactions on Software Engineering, (4):437–439, 1982.
- [22] Michael R Lyu et al. Handbook of software reliability engineering, volume 3. IEEE Computer Society Press CA, 1996.
- [23] Yashwant K Malaiya and Jason Denton. What do the software reliability growth model parameters represent? In Proc. of ISSRE’97, pages 124–135. IEEE, 1997.

- [24] Keith W. Miller, Larry J. Morell, Robert E. Noonan, Stephen K. Park, David M. Nicol, Branson W. Murrill, and Jeffrey M. Voas. Estimating the probability of failure when testing reveals no failures. *IEEE Trans. on Software Engineering*, 18(1):33–43, 1992.
- [25] HD Mills. On the statistical validation of computer programs. IBM Federal Syst. Div., Tech. Rep, pages 72–6015, 1972.
- [26] Brendan Murphy and Ted Gent. Measuring system and software reliability using an automated data collection process. *Quality and reliability engineering international*, 11(5):341–353, 1995.
- [27] J.D. Musa. Operational profiles in software-reliability engineering. *IEEE Software*, 10(2):14–32, 1993.
- [28] Yutaka Nakagawa and Shuetsu Hanata. An error complexity model for software reliability measurement. In *Proc. of ICSE'89*, pages 230–236. ACM, 1989.
- [29] Eldred Nelson. Estimating software reliability from test data. *Microelectronics Reliability*, 17(1):67–73, 1978.
- [30] Elisabeth A Nguyen, Carlos F Rexach, David P Thorpe, and Andrew E Walther. The importance of data quality in software reliability modeling. In *Proc. of ISSRE'10*, pages 220–228. IEEE, 2010.
- [31] JH Sigurdsson, LA Walls, and JL Quigley. Bayesian belief nets for managing expert judgement and modelling reliability. *Quality and Reliability Eng. International*, 17(3):181–190, 2001.
- [32] W3Schools. Browser statistics. http://www.w3schools.com/browsers/browsers_stats.asp, April 2014.
- [33] S. Yacoub, B. Cukic, and H.H. Ammar. A scenario-based reliability analysis approach for component-based software. *IEEE Trans. on Reliability*, 53(4):465–480, 2004.
- [34] Alan Wood. Software reliability growth models. Tandem Technical Report, 96, 1996
- [35] Singha Roy, Nivir Kanti and Rossi, Bruno. Towards an Improvement of Bug Severity Classification. In *Proc. Of SEAA'14*, IEEE, 2014.
- [36] Helton, J. C., Johnson, J. D., Sallaberry, C. J., and Storlie, C. B. Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliability Engineering & System Safety*, 91(10), 1175-1209. 2006.
- [37] Goseva-Popstojanova, K., and Kamavaram, S. Assessing uncertainty in reliability of component-based software systems. In *Proc. of. ISSRE'03*. IEEE. 2003

9 Appendix

Nelson's model

(1)

POF	probability of failure
n_f	number of failed test executions
n	total number of test executions

Miller's model

(2)

(3)

\hat{p} estimator of failure probability
 a, b parameters of $Beta(a,b)$ distribution
 t number of executed test cases without failure
 \hat{p}^2 variance of the estimate

Software reliability growth models

Model	Goel-Okumoto	Delayed S-shaped	Duane/Crow	Littlewood-Verall
$m(t)$				*
$\lambda(t)$				*
* ,				

Table 15: Examples of software reliability growth models

Test code coverage

Name	$c(t)$
Exponential	
Weibull	
S-shaped	
Log-logistic	
Log-poisson	
Log-normal	

Table 16: Overview of coverage functions**GERT model**

(4)

 POF probability of failures a_i regression parameter SM_i STREW metric**Table 17:** Metrics from STREW suite

Metric	Id
Test quantification	
	SM1
	SM2
	SM3
	SM4
Complexity and OO metrics	
	SM5
	SM6
	SM7
	SM8

Platform for medical curriculum innovation: The role of specialized vocabularies

Martin Komenda^{1,2}

¹ Institute of Biostatistics and Analyses, Faculty of Medicine, Masaryk University, Kamenice 126/3,
625 00, Brno

² Faculty of Informatics, Masaryk University, Botanická 68a, 602 00, Brno
komenda@iba.muni.cz

Abstract

The paper discusses a role of various specialized vocabularies from perspective of a technology enhanced medical education. We introduce the most widespread and the most commonly used medical oriented nomenclature and explore the benefits which the integration of MeSH thesaurus can provide. With the use of MeSH as an internal part of the OPTIMED platform, we aim to identify valid, novel, potentially useful patterns, which can significantly help evaluators to make right decisions and afterwards built well-balanced medical curriculum.

Key words

Medical education, specialized vocabulary, curriculum harmonization.

Abstrakt

Článek pojednává o roli různých specializovaných slovníků z pohledu technologie rozšířeného lékařského vzdělání. Je přestaveno nejrozšířenější a nejčastěji používané zdravotnické orientované názvosloví a prozkoumány výhody, které může poskytnout integraci tezauru MeSH. S využitím MeSH jako vnitřní součásti platformy OPTIMED je naším cílem je identifikovat platné, nové, potenciálně užitečné vzory, které mohou výrazně pomoci hodnotitelům učinit správná rozhodnutí a poté vytvořit vyvážené lékařské kurikulum.

Key words

Lékařské vzdělání, specializovaný slovník, harmonizace osnov.

1 Introduction

From the perspective of medical education, a number of unresolved challenges appear, including the proposal and broad acceptance of global standards for medical vocabulary. Specialized terminologies represent a set of mostly phrases, which are consistent across various international organization and particular medical domains. The use of approved thesaurus (such as UMLS, MeSH, SNOMED etc.) as a comprehensive nomenclature brings a mechanism for representing such formal and shared domain descriptions. It provides systematic and computer-processable collection of specialized terms, which can be used for annotate data with labels (metadata) indicating their meaning, thereby making their semantics explicit and machine-accessible. These vocabularies offer an interesting way how to find appropriate codes, terms, synonyms and definitions designed to complement the broad coverage of medical concepts. It has the great potential of improving the deal with information in the broadest sense possible, e.g. better search engines, more effective categorization or retrieval and analysis of data. During educational process, the controlled vocabularies implementation significantly influences normalization of achieved results, because students and teachers very often use different theoretical and clinical terms that in fact mean the same thing. For example, the terms heart attack, myocardial infarction, and MI abbreviation may represent the same meaning to academics, but, to a computer, they are all different. The main purpose of thesaurus concept integration is to allow users to search

one term and to find and retrieve learning activities that use synonymous terms. It also enhances the indexing and aggregation process of medical data across various medical fields of study and particular disciplines. In general, medical classification systems are used for variety of applications in medicine, public health and medical informatics including the reimbursement, e.g. statistical and data mining analysis, knowledge engineering and decision support system [1]. In our particular case, we concentrate on MeSH thesaurus and its implementation in higher education area within the process of medical curriculum harmonization at Masaryk University. This is a white paper, which concludes with requests for additional insights and information that will enable us to continue to better understand the domain of continual improvement of medical education with the use of modern information and communication technologies.

2 Medical vocabularies

The use of the specialized vocabulary of a particular domain (terminology nomenclature) is an important initial step of creating formalized knowledge representations as an essential part of educational process, especially in medical fields. These vocabularies follow the ratchet principle: it moves from basic understanding to thorough understanding, from simple to complex education [2]. When a virtual learning environment (VLE) turns to the task of consolidated educational data collection (for instance content management, curriculum mapping and planning, student engagement and administration, communication and collaboration domain), these vocabularies will present considerable challenges to standardize medical education. A prerequisite to more comprehensive categorization of educational content is the implementation of standardized terminology into the VLE systems. A primary aim is to overcome two significant barriers to effective retrieval of machine-readable information: the variety of names used to express the same concept and the absence of a standard format for distributing terminologies [3]. The purpose of medical vocabularies is to embody what has been known in the past about every phase of medicine [4]. These vocabularies continues to increase and grow, not only in its technological aspects, but also from the perspective of medical education quality, which is logically reflected in global level of health care. Below, the most widespread and the most **commonly used medical oriented nomenclatures are described.**

2.1 UMLS

Unified Medical Language System (UMLS) brings together many health and biomedical vocabularies, ontologies and standards to enable interoperability between computer systems. It was developed by National Library of Medicine covers the entire terminology domain by integrating more than 60 families of biomedical vocabularies. The three major components of UMLS are the Metathesaurus (repository of inter-related biomedical concepts covering extensive list of terms and codes from various vocabularies), the Semantic network (high-level categorization of Metathesaurus concepts) and the Specialist Lexicon (generating the lexical variants of biomedical terms) [5]–[7]. For the illustration, figure 1 shows UMLS Metathesaurus integrates sub -domains.

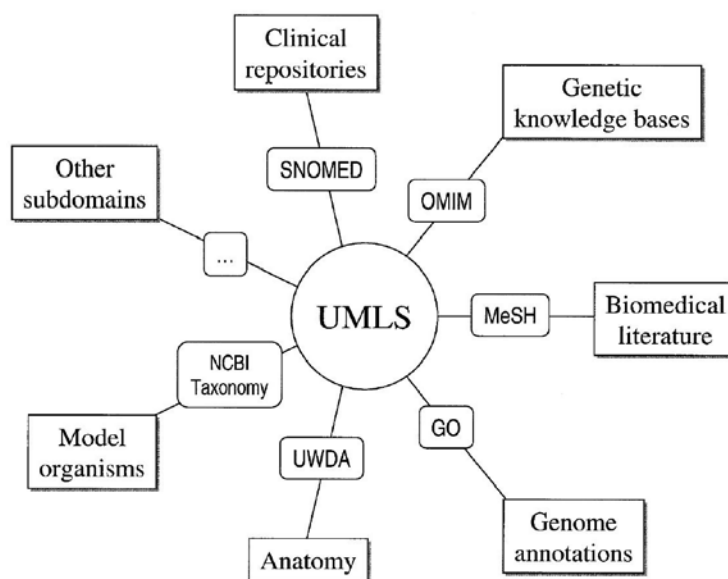


Fig. 1 – The various subdomains integrated in UMLS [5].

2.2 MeSH

Although geared specifically for information retrieval, MeSH (Medical Subject Headings) can almost be seen as a general purpose vocabulary with concepts from all areas of the biomedical domain. This classification is a rich and controlled vocabulary generated through an intense indexing process performed by examiners. Terms (namely descriptors) are assigned to documents to delineate their content at 16 different branches of specificity. The 2014 MeSH vocabulary is specifically composed by more than 27 thousands of descriptors, which are organised in a tree-like structure. Descriptors may be also complemented with one or more qualifiers. These terms further contextualise the meaning of the descriptors to which they are assigned in relation to the content of the considered document. MeSH vocabulary is often used to delineate samples of documents in a number of medical areas and, as discussed, at various levels of specificity [8]–[10]. The Czech translation of MeSH is being prepared by the Czech National Medical Library, which issues annual updates. Following a contractual agreement with the publisher, MeSH will also be used for the educational purposes.

2.3 ICD-10

ICD (International Classification of Diseases) is used to classify and code mortality information worldwide. The tenth revision (ICD-10) is more complex than the previous one (ICD-9). Specifically, ICD-10 contains more codes as well as re-structured chapters and changes in rules for coding. ICD-10 as a whole is designed to be a core classification for a family of disease- and health-related classifications. Some members of the family of classifications are derived by using a fifth or even sixth character to specify more detail. In others, the categories are condensed to give broad groups suitable for use, for instance, in primary health care or general medical practice [11], [12].

2.4 SNOMED-CT

SNOMED CT (Systematized Nomenclature of Medicine Clinical Terms) as a coding system for clinical problems was created by the merger, expansion and restructuring of two large-scale terminologies: SNOMED RT (Systematized Nomenclature of Medicine Reference Terminology) and

CTV3 (Clinical Terms Version 3). Because it contains more than 380 thousands concepts, with a total of about 800 thousands descriptions or terms, the practical use of SNOMED will demand new types of tools to search and navigate intuitively in the term collection. It is probably one of the most comprehensive, multilingual clinical healthcare terminology in the world, because it systematically supports the development of comprehensive high-quality clinical content in health records and provides standardized way to represent clinical phrases captured by the clinician and enables automatic interpretation of these. SNOMED CT is a clinically validated, semantically rich, controlled vocabulary that facilitates evolutionary growth in expressivity to meet emerging requirements [13], [14].

2.5 Emtree

Emtree is a hierarchically structured and controlled terminology for Biomedicine and related Life Science, which has been developed by Elsevier as a biomedical and pharmacological online database. It includes a whole range of terms for drugs, diseases, medical devices and essential life science concepts. Emtree thesaurus contains about 48 thousands preferred terms and 200 thousands synonyms in 15 main domains of drugs and diseases, organized in a multilevel hierarchy. All the terms, that are strictly synonymous with each other, are organized in concept-based structure [15]–[17].

2.6 RadLex

RadLex [18] is a standardized vocabulary of radiological terms, which includes highly detailed terms for anatomy, pathology, and radiological observations. It already contains over 8,000 anatomic and pathologic terms, many of which are not currently available in other controlled medical terminology system. Generally, RadLex was designed to fill in the gaps in other medical terminology systems, thereby creating a single source for medical imaging terminology. Another key distinguishing feature of RadLex is that it is designed to be continuously supplemented and updated with incorporation of new concepts, including harmonization with other popular medical vocabularies and term sets. The goal of this method is to establish a uniform, consistent terminology to improve communication of results and to better integrate clinical practice with education and the scientific literature [19].

2.7 Overview

The integration of various specialized nomenclatures, vocabularies, and terminologies allow more precise analysis and improvement of the educational data content. Furthermore, automated systems can apply the knowledge encoded with the use of mentioned taxonomies and human users can easily search and browse the available data in a straightforward manner. It will have both academic and clinical implications by enhancing the retrieval as well as the indexing of information. Besides introduced approaches, there exist many others vocabularies, which are specifically focused on particular biomedical discipline. For example GO (Gene Ontology), eVOC (**gene expression data**), OMIM (Online Mendelian Inheritance in Man), MEDIC (Merged Disease vocabulary), LDDb (London Dysmorphology Database), FMA (Foundational Model of Anatomy), LOINC (Logical Observation Identifier Names and Codes) and NCI thesaurus (cancer research).

3 The practical use: OPTIMED project

An OPTIMED platform describes and categorizes all the learning activities (lectures, seminars, clinical practices and self-study) in the theoretical and clinical sections of medical curriculum. This web-based tool is designed to capture the systematic transmission of medical/clinical knowledge to students during their courses in General medicine study field at Faculty of Medicine of Masaryk University. Overall, the OPTIMED system gives students, teachers, guarantors, curriculum designers and faculty management a detailed look at where specific topics and learning outcomes are addressed and how educational objectives are being met. As stated above, the implementation of standardized

vocabulary can significantly improve the quality of further analytical processing to understand stored data. We decided to integrate the one from available vocabularies for standardised work with key words. So we adopted the biomedical dictionary MeSH, where the main objective is to classify learning activities appeared in curriculum. In the past, key words were defined and structured in many forms and there was a growing need for their unification with respect to the international framework. The main requirement for standardized dictionary integration was regular updates of the Czech mutation, which MeSH fulfils as the only solution available. No other language mutations are foreseen at the moment, but a possible change should not bring too many complications in terms of the proposed structure [20]. Figure 2 represents a conceptual data model covering all essential attributes for curriculum description using MeSH thesaurus.

Currently, the OPTIMED system has provided a searching assistance with a rich set of lexical look-up facilities, which is based on MeSH's complex structure including terms with tree structured contexts (hierarchical locations).

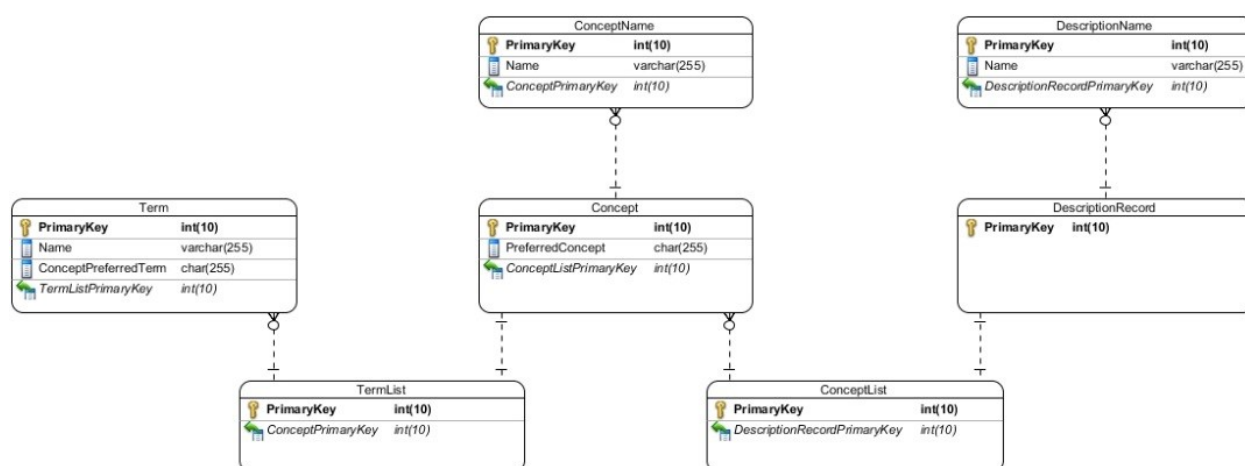


Fig. 2 – The data model of MeSH entities in the OPTIMED database.

4 Future visions

MeSH vocabulary implementation into the OPTIMED system causes specific validation tasks based on standardized terms that we would like to resolve by detailed analytical summarization. There are several interesting areas that selected analytical methods may answer: Coverage and relations between learning activities and terms, objective evaluation of term occurrence, model term integration into the education. The analytical outputs may significantly assist to find the potentially problematic areas and construct comprehensive reports or instructions for the subsequent global in-depth inspection. These extracted information from the OPTIMED curriculum management system serve as a supporting material for the evaluation process under the supervision of the expert committee and faculty management. In the future, an idea of Automatic Term Mapping (ATM) of medical curriculum data would be implemented. The ATM technique [21] was developed by US National Center of Biotechnology Information for mapping end-user queries to MeSH thesaurus and other search field. The basic aim will be to improve information retrieval in structured information: searching indexes instead of only the free text.

5 Conclusions

The OPTIMED system has provided huge amount of data records related to the medical curriculum: approx. 1350 learning units and 7100 learning outcomes, i.e. more than 2500 normated pages of text. For long-term guarantee of quality, a continuous inspection is needed. From the perspective of human cognition abilities, it is not possible to carefully read and verify all the curriculum content. We proposed of a formal database metadata arrangement including standardized MeSH vocabulary, which described the medical curriculum independent of subsequent implementation. It provides a background, which plays an essential role during validation analysis. This domain consists of data mining, data pre-processing, data analysis, and visualization. We aim to identify information rich data relations and offer global overview for simpler and easier of understanding curriculum structure.

Acknowledgement

The author was supported from the grants project OPTIMED - OPTImized MEDical education: horizontal and vertical connections, innovations and efficiency in practice reg. no: CZ.1.07/2.2.00/28.0042, which is funded by the European Social Fund and the state budget of the Czech Republic.

6 References

- [1] A. Holzinger, Biomedical Informatics: Discovering Knowledge in Big Data. Springer, 2014.
- [2] C. Adelman, “The Bologna Process for US Eyes: Re-learning Higher Education in the Age of Convergence.,” Inst. High. Educ. Policy, 2009.
- [3] D. A. Lindberg, B. L. Humphreys, and A. T. McCray, “The Unified Medical Language System.,” *Methods Inf. Med.*, vol. 32, no. 4, pp. 281–291, 1993.
- [4] J. Anderson, “The computer: medical vocabulary and information,” *Br. Med. Bull.*, vol. 24, no. 3, pp. 194–198, 1968.
- [5] O. Bodenreider, “The unified medical language system (UMLS): integrating biomedical terminology,” *Nucleic Acids Res.*, vol. 32, no. suppl 1, pp. D267–D270, 2004.
- [6] Unified Medical Language System (UMLS).” [Online]. Available: <http://www.nlm.nih.gov/research/umls/>. [Accessed: 22-Jan-2015].
- [7] “Performance evaluation of unified medical language system’s synonyms expansion to query PubMed,” *BMC Med. Inform. Decis. Mak.*, vol. 12, no. 1, pp. 12–17, Jan. 2012.
- [8] “Fact SheetMedical Subject Headings (MeSH®).” [Online]. Available: <http://www.nlm.nih.gov/pubs/factsheets/mesh.html>. [Accessed: 12-Jul-2012].
- [9] D. Rotolo and L. Leydesdorff, “Matching MEDLINE/PubMed Data with Web of Science (WoS): A Routine in R language,” *J. Assoc. Inf. Sci. Technol. Forthcom.*, 2014.
- [10] A. P. Davis, T. C. Wieggers, M. C. Rosenstein, and C. J. Mattingly, “MEDIC: a practical disease vocabulary used at the Comparative Toxicogenomics Database,” *Database*, vol. 2012, p. bar065, 2012.
- [11] W. ICD, “Classification of mental and behavioural disorders,” Edinb. Lond. Melb. N. Y. Tokyo Churchill Livingstone, 1991.

- [12] F. Gjertsen, S. Bruzzzone, M. E. Vollrath, M. Pace, and Ø. Ekeberg, “Comparing ICD-9 and ICD-10: The impact on intentional and unintentional injury mortality statistics in Italy and Norway,” *Injury*, vol. 44, no. 1, pp. 132–138, Jan. 2013.
- [13] P. Ruch, J. Gobeill, C. Lovis, and A. Geissbühler, “Automatic medical encoding with SNOMED categories,” *BMC Med. Inform. Decis. Mak.*, vol. 8, no. Suppl 1, p. S6, Oct. 2008.
- [14] P. L. Elkin, S. H. Brown, C. S. Husser, B. A. Bauer, and et al, “Evaluation of the Content Coverage of SNOMED CT: Ability of SNOMED Clinical Terms to Represent Clinical Problem Lists,” *Mayo Clin. Proc.*, vol. 81, no. 6, pp. 741–8, Jun. 2006.
- [15] C. Fluit, M. Sabou, and F. van Harmelen, “Ontology-Based Information Visualization: Toward Semantic Web Applications,” in *Visualizing the Semantic Web*, V. G. Ds. MSc and C. C. BSc MSc, Eds. Springer London, 2006, pp. 45–58.
- [16] T. Bekhuis, D. Demner-Fushman, and R. Crowley, “Comparative effectiveness research designs: an analysis of terms and coverage in Medical Subject Headings (MeSH) and Emtree,” *J. Med. Libr. Assoc.*, vol. 101, no. 2, pp. 92–100, Apr. 2013.
- [17] M. Taboada, R. Lalin, and D. Martinez, “An Automated Approach to Mapping External Terminologies to the UMLS,” *IEEE Trans. Biomed. Eng.*, vol. 56, no. 6, pp. 1598–1605, Jun. 2009.
- [18] C. P. Langlotz, “RadLex: A new method for indexing online educational materials 1,” *Radiographics*, vol. 26, no. 6, pp. 1595–1597, 2006.
- [19] M. W. Shore, D. L. Rubin, and C. E. Kahn Jr, “Integration of imaging signs into RadLex,” *J. Digit. Imaging*, vol. 25, no. 1, pp. 50–55, 2012.
- [20] M. Komenda, D. Schwarz, and L. Dušek, “Towards a System of Enhanced Transparency of Medical Curriculum,” 2013.
- [21] N. C. for B. Information, U. S. N. L. of M. 8600 R. Pike, B. MD, and 20894 Usa, “PubMed Help,” Dec. 2014.

Towards Flexible Intelligent Building Data Analysis

¹Adam Kučera, ²Tomáš Pitner

¹ Masarykova univerzita, Fakulta informatiky
Botanická 68a, 602 00 Brno
akucera@mail.muni.cz

² Masarykova univerzita, Fakulta informatiky
Botanická 68a, 602 00 Brno
tomp@fi.muni.cz

Abstract

Costs related to the operation of facilities (buildings and devices) represent a significant part of expenses of an organization. To analyse the effectiveness and performance of facilities, advanced software systems are available on the market. Such systems focus mostly on the analysis of financial data, human resources information, or static data concerning the buildings (e.g. room area). However, a large amount of precise, detailed and up-to-date information can also be gathered from building management systems (BMS) that integrate various building automation and electronic systems. The building operation data is currently unavailable for advanced analytical tools. This situation is largely attributable to two factors: inaccessibility of BMS data and missing semantic information. The paper proposes a middleware layer that facilitates a data retrieval and an analysis by providing data access and semantic interfaces and models for querying the building systems.

Abstrakt

Náklady spojené s provozem budov a jejich vybavení představují významnou část výdajů organizace. Na trhu je k dispozici široká škála nástrojů, které umožňují analyzovat efektivitu provozu. Běžně dostupné nástroje se však zaměřují zejména na analýzu na základě ekonomických dat, dat o lidských zdrojích nebo neměnných datech o budovách (rozloha apod.) Kromě těchto dat jsou však v moderních budovách k dispozici velké objemy přesných, detailních a aktuálních dat pocházejících z tzv. automatizačních systémů budov. Tato data jsou však v současné situaci pro analýzu provozu prakticky nevyužívána. To je způsobeno zejména dvěma faktory – obtížnou dostupností dat pro ostatní systémy a chybějícími sémantickými informacemi o významu shromažďovaných dat. Tento článek přináší návrh vrstvy „middleware“, která výrazně usnadňuje získávání dat z automatizačních systémů a jejich následnou analýzu. Toho je dosaženo poskytnutím příslušných rozhraní a modelů pro přístup k datům z automatizačních systémů a zjištění doplňujících sémantických informací.

Keywords

Facility management, Intelligent buildings, Smart buildings, Building automation, Building management systems, Systems integration, Ontology, Semantics.

Klíčová slova

Facility management, Správa budov, Inteligentní budovy, Chytré budovy, Automatizace budov, Systémová integrace, Ontologie, Sémantika.

1 Introduction

Each organization needs to ensure various aspects of its operation that are not directly involved in reaching its primary goal (e.g. providing service to customer or sell its products). The profession of *facility management* (FM) covers these aspects by supporting tasks such as space management, asset management, help desk and service desk, maintenance, energy monitoring, room reservations or hoteling. The facility management is defined by the International Facility Management Association (IFMA) in following words: “Facility management is a profession that encompasses multiple

disciplines to ensure functionality of the built environment by integrating people, place, process and technology.” [1]

When we look closely at the field of the FM, we can distinguish several systems and/or data sources that can be utilized in order to support and simplify tasks of facility management staff.

Widely used *Computer Aided Facility Management* (CAFM) systems cover most of the areas of the FM described above. A CAFM software serves as a data store and user interface for operational data such as assignment of employees to rooms, a log of maintenance plans, requests and tasks, or energy consumption data. CAFM systems offer advanced analytical tools for an evaluation of efficiency and performance of organization's operation based on a financial (energy consumption), spatial (occupancy planning) and technical data (maintenance).

A *Building Information Model* (BIM) is a data source that contains spatial information about building constructions (materials, dimensions) and locations (sites, buildings floors, rooms) and technologies installed in them (e.g. valves, pumps, a plumbing, lights, and power lines). Data from a BIM database serve as an input for CAFM systems. Spatial data is imported (and synchronized) into the CAFM system as “background data” for space management, occupancy planning, maintenance management and other tasks.

The last system related to the task of facility management is tightly connected to modern “intelligent” buildings. Such facilities incorporate wide scale of electrical automated systems such as a security system, an access control system, a fire alarm system, or a building automation system that controls mostly *Heating, Ventilation and Air Conditioning* (HVAC) devices. Mentioned systems offer variety of sensors and controllable devices. Monitoring and control of the building operation can be integrated into the *Building Management System* (BMS). The BMS provides environment that ensure services such as remote monitoring and control of a building operation, archiving historical data and event notification.

The paper proposes a middleware layer that will integrate BMS with other facility management software systems. More specifically, the middleware aims to simplify the development of end-user applications by providing convenient data sources and interfaces, enabling the developers to fully focus on analytical methods and user-oriented aspects of the applications.

2 Issues of Building Operation Data Analysis

In current state-of-the-art solutions, the integration of the BMS with the data in the CAFM and the BIM is missing or is simplified to a flat structure that cannot be effectively queried. One of the reasons that prevent deeper integration is the fact that the structure of the BMS data is determined by a network topology and not by semantic information (purpose) of provided data. As a result, advanced analytical features of the CAFM software are currently unavailable for the BMS data. This does not pose a problem for small installations up to dozens of devices, because a data retrieval and an analysis can be easily performed manually. However, for large sites (hundreds of devices, thousands of sensors), an amount of data prevents effective gathering of needed information with no advanced querying tools available. However, the BMS contains a large amount of a precise, up-to-date and detailed data which are valuable for a building operation analysis and cannot be obtained any other way.

2.1 Current Workflow

In current environments, we can define two types of users – one group of users are facility managers who know which data they need for a building operation analysis, know the context, but are unable to get the data from the systems. The other group of users are building operators which have capabilities to get the data from the BMS (even if the task comprises large amount of “manual” work), but don’t have enough time, knowledge, competence or authority to fully evaluate the building operation and make long-term decisions based on the results.

Current workflow for a BMS data analysis is thus too complicated and inflexible. A responsible staff member (facility manager) asks building operators to get the needed data and agrees with them on the data output format. Building operator gathers the addresses of respective data points according to the request and then extracts the data for each of the data points. Next, a conversion to the defined format takes place. This process is not automated and has to be repeated every time the report is needed.

Advanced applications with convenient user interface that will hide low-end aspects of the task (gathering the data point addresses, extracting the data from the database, conversion to the interchange format) will allow facility managers to obtain the data directly from the system without the need of human work of the building operators.

2.2 Problem Complexity

Development of such applications is very demanding. The system has to cover each of the aspects of the task:

- Data retrieval;
- Definition of data semantics;
- Analysis;
- User interface and experience.

To accomplish the above mentioned tasks, experts from several fields are needed. The problem comes mainly with the two first points. They require expertise in the fields of building automation protocols and even building technologies (e.g. HVAC devices) itself, which is not common among IT experts. The vendors of building automation systems focus mostly on development of software that can be used for management, programming of the system and for an every-day operation of the building technologies, rather than on analytical features.

2.3 Proposed Solution

Thus, to overcome the issue of problem complexity, the goal of the research is to propose a middleware layer that will largely simplify development of advanced applications for the field of a building operation analysis. The middleware layer will provide APIs allowing access to the building operation data as well as needed semantic information based on integration of the BMS and the BIM data.

3 State of the Art

One of the main concerns of the facility management is evaluation of organization’s operation performance and efficiency. Since the aim of the research is to provide tools for evaluation of a building operation, the following section provides an overview of benchmarking methods and approaches in the facility management.

The facility management is undergoing a long-term process of standardization. In the context of the European Union, the domain is covered by the EN 15221 – Facility Management [2] standard issued

by the European Committee for Standardization. For the field of the space management, the EN 15221-6 provides guidelines for a measurement of dimensions, a categorization and evaluation of a “building performance” (e.g. a ratio of an overall area of a building and an area that can be leased). In the USA, The Building Owners and Managers Association (BOMA) provides a different set of standards known as the ANSI/BOMA Z65 – Standard Methods of Measurement [3], which focuses on the area of space management and specifically on the measuring methods and efficiency evaluation.

Benchmarking is a subject matter of the last part of the European standard listed as EN15221-7. The focus is put mostly on processes or services which can be easily outsourced, such as cleaning or maintenance. Evaluation of Key Performance Indicators (KPIs) then becomes essential for enforcing Service License Agreements with service providers. The document also covers benchmarking in other areas, including energy management.

Energy efficiency of facilities gains a significant interest from research groups, authorities and administration. In the European Union, the Directive 2002/91/EC on the energy performance of buildings [4] introduced the Energy Performance Certificate (EPC) that provides an A to G scale for a rating of energy efficiency of a facility.[5][6] However, the rating is based solely on evaluation of used materials, equipment, and design – it does not reflect actual energy consumption during a facility operation. The EPC is thus criticized for its inaccuracy (e.g. [7], [8], [9]). Considering energy consumption evaluation during a facility operation, different approaches (sets of KPIs) are proposed [10], [11]. Typically, the data needed for an evaluation are gathered from energy costs. If there were tools for BMS data extraction available, evaluation of the KPIs would be simplified, more precise (BMS provides greater level of the detail than an invoice for the whole building or site) and even completely new KPIs could be defined, taking into account other aspects such as temperature oscillations in a facility. In [12], Complex Event Processing tools are used for an energy consumption analysis and decision support for an industrial environment (i.e. a factory).

As stated in the previous section, the BMS contains vast amounts of precise, up to date and detailed data, which are very valuable for a building operation analysis, but development of business intelligence and decision support applications is very demanding because of complexity of the task. In our opinion, this is the reason why building operation data are not widely used for benchmarking in facility management.

4 Middleware Layer

The research topic we consider crucial for an efficient large-scale building operation analysis is to design and develop a middleware layer that will provide various applications both with the building operation data itself and with the semantic information about the BMS data meaning and the purpose.

On the Fig. 1: System overview, the whole systems architecture is shown. The thesis aims to cover the middleware layer depicted in the figure. The aim is to provide the missing link between the source systems (BMS, BIM) and the end user analytical applications (e.g. CAFM software) and to allow development of new tools that will be able to facilitate the building operation data. The architecture introduces the principles of the *Service Oriented architecture* (SOA) into the field of a building operation data access.

Provided proposed system components exist, developers of analytical or monitoring (fault detection) applications will be able to fully focus on front-end features (user interface, query definition wizards, integration with GIS services) and analytical features, and not on the core logic of data integration and retrieval. Advanced user interfaces are essential for the successful BMS data analysis as they will remove the gap between the facility managers’ knowledge of the overall (mostly financial) context of a building operation and technical knowledge needed for gathering the data from the BMS.

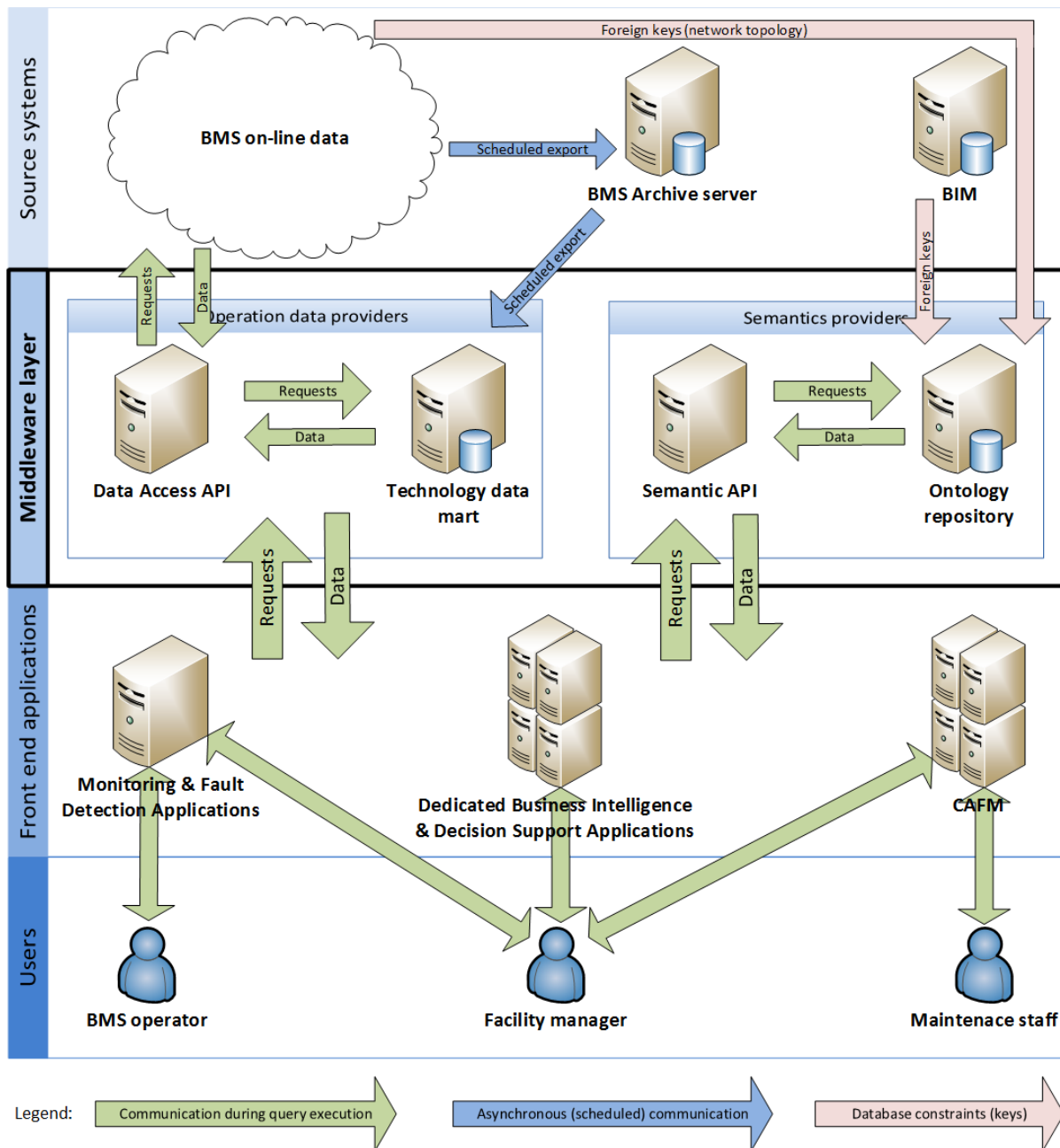


Fig. 1: System overview

To achieve this goal, we can define different aspects of the problem that need to be solved:

- **Architecture design** - The goal of an architecture design is to identify system layers and required components.
- **Query library & Domain specific operators** - The methodic background for the middleware design and development will be provided by examining typical use cases of a building operation analysis. The goal of this part of the research is to propose a set of query definitions with a high informational value that can be successfully used for a decision making support in order to enhance energy efficiency, lower operation costs and improve a working environment in the facilities. The domain specific operators are related mostly to the grouping

and filtering of data according to a position in a hierarchy tree of locations or device types. The operators will provide grouping and filtering capabilities based on data from the semantic model (see below).

- **Building operation data access** - One of the issues preventing an effective building operation analysis lays in the inaccessibility of the data that are available in the BMS network. To communicate with the building automation devices, the system must be capable of communication using building automation protocol (e.g. BACnet, KNX/EIB, MODBUS or LONWorks). In order to facilitate basic operations (mostly reading of data) from the BMS, we propose an API (see Fig. 1: System overview – *Operation data providers* container on the Middleware layer) that will provide an additional abstraction layer on top of the existing automation protocol stack implementations. This API will not require deep knowledge of the building automation domain and will provide convenient ways to access the data using well known and established communication methods (RESTful API, Web services or JSON).
- **Semantic model** - The semantic model enriches the BMS network addresses with additional information, describing a relation of the published BMS data with the real world elements such as locations (e.g. sites, buildings, floors, and rooms) and devices (e.g. sensors, air conditioning units, valves, and engines). The model also introduces other attributes to the BMS data, such as a measured physical quantity. For a formal representation of the model, *Web Ontology Language* (OWL) framework is used.
- **Semantic API** - The Semantic API will serve as an interface to the semantic model stored in the ontology repository. RDF based ontology repositories provide APIs (e.g. *Apache Jena*) and the SPARQL query language for accessing the data. However, the SPARQL language is specific to the field of ontology development and is not well known among developers. Queries to the ontology repository became overwhelmingly complex for developers that are not experts in an ontology design. Thus, the Semantic API will be proposed, based on the query library (see above). The API will provide support for typical queries to the ontology repository by defining query templates that can be called from the API and populated by parameters supplied by the end user application.
- **Front end applications** - Although front end applications (i.e. business intelligence tools, decision support tools, analytical applications, monitoring systems) are not subject of the proposed research, they are needed for credible evaluation of the results. There are several methods and techniques that we consider promising in a field of building operation analysis, such as Complex Event Processing methods, integration of BMS and GIS or deep integration with CAFM systems.

5 Conclusions and Future Work

Together, components of the middleware layer (Data access API, Semantic model, Semantic API) and methodical guidelines (Query library) will largely facilitate development of end user applications used for a building operation analysis. Possible types of end user applications were mentioned in the previous section.

The middleware layer is currently in the design and development phase. After the first development iteration will be completed, the middleware design will be evaluated by implementing one of the benchmarking methods introduced in the EN15221-7 standard (Benchmarking in facility management), namely the „Energy consumption per square feet/meter“ metrics proposed as a measure for energy efficiency evaluation. We expect that the results will show the benefits of an approach based on precise building operation data coming from building automation systems when compared with traditional approach based on economical data.

6 References

- [1] International Facility Management Association. What is facility management?, 2014. Available from: <http://ifma.org/about/whatis-facility-management>. Referred on August 24, 2014.
- [2] European Committee for Standardization. EN 15221 1-7 – Facility management, 2006-2012.
- [3] American National Standards Institute. ANSI/BOMA Z65 1-6 – Standard Methods of Measurement, 2009-2012. Overview of parts is available at <http://www.boma.org/standards/>.
- [4] European Union. Directive 2002/91/EC On the Energy Performance of Buildings, 2002.
- [5] Bart Poel, Gerelle van Cruchten, and Constantinos A. Balaras. Energy performance assessment of existing dwellings. *Energy and Buildings*, 39(4):393–403, 2007.
- [6] Luis Pérez-Lombard, José Ortiz, Rocío González, and Ismael R. Maestre. A review of benchmarking, rating and labelling concepts within the framework of building energy certification schemes. *Energy and Buildings*, 41(3):272–278, 2009. Farin, G. E., Hansford, D.: Mathematical Principles for Scientific Computing and Visualization. Natick, USA : A.K. Peters Ltd, 2008.
- [7] Lamberto Tronchin and Kristian Fabbri. Energy performance certificate of building and confidence interval in assessment: An italian case study. *Energy Policy*, 48(0):176–184, 2012. Special Section: Frontiers of Sustainability.
- [8] Henk Visscher, Dasa Majcen, and LCM Itard. Effectiveness of energy performance certification for the existing housing stock. In *RICS COBRA 2012, Proceedings of the Construction, Building and Real Estate Conference, Tempe, AZ: Arizona State University*, pages 130–148, 2012.
- [9] Nick Hogg and Chris Botten. A tale of two buildings – are EPCS a true indicator of energy efficiency? Published by Jones Lang LaSalle and Better Building Partnership, 2012. Available from <http://www.betterbuildingspartnership.co.uk/download/bbpjll---a-tale-of-two-buildings-2012.pdf>. Referred on August 25, 2014.
- [10] Katharina Bunse, Matthias Vodicka, Paul Schönsleben, Marc Brühlhart, and Frank O. Ernst. Integrating energy efficiency performance in production management gap analysis between industrial needs and scientific literature. *Journal of Cleaner Production*, 19(67):667–679, 2011.
- [11] John C Van Gorp. Using key performance indicators to manage energy costs. *Strategic planning for energy and the environment*, 25(2):9–25, 2005.
- [12] Konstantin Vikhorev, Richard Greenough, and Neil Brown. An advanced energy management framework to promote energy awareness. *Journal of Cleaner Production*, 43(0):103–112, 2013.

MODELLING AND FORECASTING OF WIG20 STOCK INDEX

Dusan Marcek

VŠB-TU Ostrava, Department of Applied Informatics
Sokolská třída 33, 701 21 Ostrava 1
Dusan.marcek@vsb.cz

Abstract

We examine the ARIMA-ARCH type models for the volatility and forecasting models of Polish WIG20 stock indexes based on statistical (stochastic), machine learning methods and an intelligent methodology based on soft or granular computing and make comparisons with the class of RBF neural network and SVR models. To illustrate the forecasting performance of these approaches the learning aspects of RBF networks are presented. We show a new approach of function estimation for nonlinear time series model by means of a granular neural network based on Gaussian activation function modeled by cloud concept. In a comparative study is shown that the presented approach is able to model and predict high frequency data with reasonable accuracy and more efficient than statistical methods.

KEYWORDS

Time series, ARCH-GARCH models, volatility, forecasting, neural networks, cloud concept, forecast accuracy, granular computing.

Abstakt

Zkoumáme modely typu ARIMA ARCH pro volatility a prognózy modelů polských WIG20 akciové indexy založené na statistických metodách (stochastické), učení stroje a inteligentní metodiky založené na měkké nebo granulí výpočetní a provádění srovnání s třídou RBF neuronové sítě a SVR modelů. Pro ilustraci jsou uvedeny předpovědi výkonu těchto přístupů aspekty učení RBF sítí. Ukazujeme nový přístup funkce odhadu pro nelineární časové řady modelu pomocí granulí neuronové sítě založené na funkci Gaussian aktivace vymodelovaných mraků konceptu. Ve srovnávací studii se ukazuje, že prezentovaný přístup je schopen modelovat a předvídat vysokofrekvenční údaje s dostatečnou přesností a efektivněji než statistické metody.

Klíčová slova

Časové řady, ARCH-GARCH modelů, volatility, prognózy, neuronových sítí, mraků konceptu, prognózy přesnost, granulované výpočetní techniky.

1 Introduction

Over the past ten years academics of computer science have developed new soft techniques based on latest information technologies such as soft, neural and granular computing to help predict future values of high frequency financial data. At the same time, the field of financial econometrics has undergone various new developments, especially in finance models, stochastic volatility, and software availability.

This paper analyses, discusses and compares the forecast accuracy from nonlinear models which are derived from competing statistical and Radial Basic Function (RBF) neural network (NN) specifications. Our motivation for this comparative study lies in both the difficulty for constructing of appropriate statistical Autoregressive/Generalised Conditionally Heteroscedastic (ARCH-GARCH)

models (so called hard computing) to forecast volatility even in ex post simulations and the recently emerging problem-solving methods that achieve low solution costs (soft computing).

In economics and in particular in the field of financial markets, forecasting is very important because forecasting is an essential instrument to operate day by day in the economic environment. In companies, medium and small enterprises, selecting an appropriate forecasting algorithms or methods is important in terms of forecast accuracy and efficiency. Therefore, it is important to search available information technologies to get optimum forecasting models.

The paper is organized in following manner. In Section 2 we briefly describe the basic methodology of ARIMA (Autoregressive Integrated Moving Average), ARCH-GARCH (Generally Autoregressive Conditionally Heteriscedastic) models. In Section 3 we present some RBF type models and models based on SVM (Support Vector Machine) for financial data. Section 4 analyses the data, builds statistical, RBF NN and SVR forecasting models. Section 5 puts an empirical comparison and assesses predictive accuracy of developed models. Section 6 briefly concludes.

2 Econometric, ARIMA and some ARCH-GARCH Models for Financial Data

The econometric approach adopted from early days of econometrics is referred to as “AER” or Average Economic Regression [1, 2] is concerned with the functional form of the multiple regression model in the form

$$y_t = \beta_0 + \beta_1 x_{1t} + \dots + \beta_p x_{pt} + u_t \quad (1)$$

where x_{it} represent a series of independent variables, β_0 regression intercept, β_i partial regression coefficients, for $i = 1, \dots, p$, u_t random error term, for $t = 1, \dots, N$.

In many cases economic theory do not give the assumption above the functional form of the model, or the assumption of independent errors and hence independent observations y_t is frequently unwarranted. If this is the case, forecasting models based on AER may be inappropriate. Box and Jenkins [3] developed a new modeling approach based on time series analysis and derived from the linear filter known as AR or ARIMA (AutoRegressive Integrated Moving Average) models. The fundamental aim of time series analysis is to understand the underlying mechanism that generates the observed data and, in turn, to forecast future values of the series. Given the unknowns that affect the observed values in time series, it is natural to suppose that the generating mechanism is probabilistic and to model time series as stochastic processes. An ARMA(p, q) model of orders p and q is defined as

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} \quad (2)$$

where $\{\phi_i\}$ and $\{\theta_i\}$ are the parameters of the autoregressive and moving average parts respectively, and ε_t is white noise with mean zero and variance σ^2 . We assume ε_t is normally distributed, that is, $\varepsilon_t \sim N(0, \sigma^2)$. ARIMA(p, d, q) then represents the d th difference of the original series as a process containing p autoregressive and q moving average parameters. The method of building an appropriate time series forecast model is an iterative procedure that consists of the implementation of several steps. The main four steps are: identification, estimation, diagnostic checking, and forecasting. For details see [3].

The first model that provides a systematic framework for volatility modelling is the ARCH model proposed by Engle [4]. Bollerslev [5] proposed a useful extension of Engle's ARCH model known as the generalised ARCH (GARCH) model for time sequence $\{\varepsilon_t\}$ in the following form

$$\varepsilon_t = v_t \sqrt{h_t}, \quad h_t = \alpha_0 + \sum_{i=1}^m \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^s \beta_j h_{t-j} \quad (3)$$

where $\{v_t\}$ is a sequence of IID (Independent Identical Distribution) random variables with zero mean and unit variance. α_i a β_j re the ARCH and GARCH parameters, h_t represent the conditional variance of time series. Nelson [6] proposed the following exponential GARCH model abbreviated as EGARCH to allow for leverage effects in the form

$$\log h_t = \alpha_0 + \sum_{i=1}^p \alpha_i \frac{|\varepsilon_{t-i}| + \gamma_i \varepsilon_{t-i}}{\sigma_{t-i}} + \sum_{j=1}^q \beta_j h_{t-j} \quad (4)$$

The basic GARCH model can be extended to allow for leverage effects. This is performed by treating the basic GARCH model as a special case of the power GARCH (PGARCH) model proposed by Ding, Granger and Engle [7]:

$$\sigma_t^d = \alpha_0 + \sum_{i=1}^p \alpha_i (|\varepsilon_{t-i}| + \gamma_i \varepsilon_{t-i})^d + \sum_{j=1}^q \beta_j \sigma_{t-j}^d \quad (5)$$

where d is a positive exponent, and γ_i denotes the coefficient of leverage effects [7]. Detailed procedure for parameter estimation of these models and investigate response investigation of equity volatility to return shock for WIG20 time series can be found in [8].

3 Soft Computing Models

In this section we briefly introduce two models belonging to soft computing methods: the RBF NN and SVR model. The first model show a new approach of function estimation for time series modeled by means a granular RBF neural network based on Gaussian activation function modeled by cloud concept (Cloud Activation Function - CAF [9]). We proposed the neural architecture according to Figure 1.

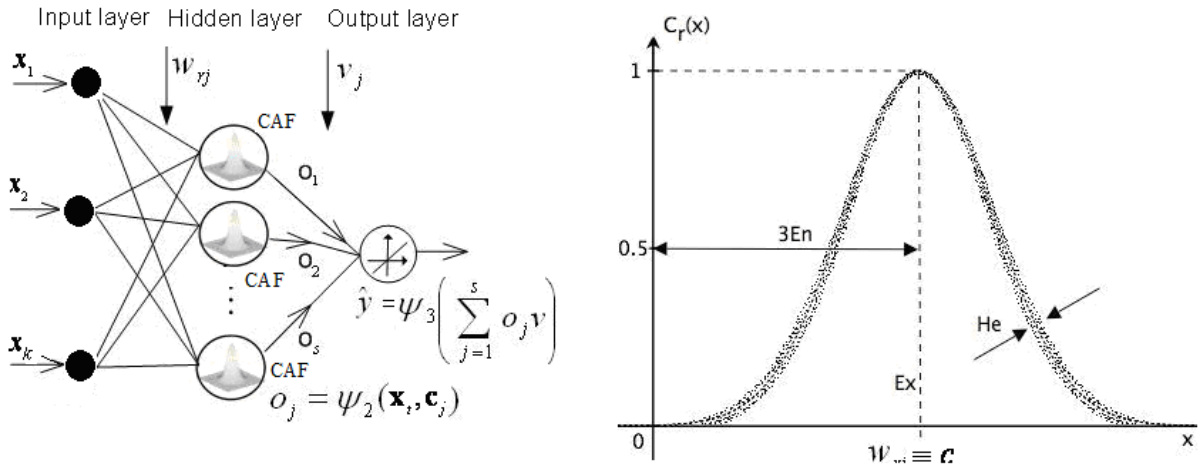


Figure 1. RBF neural network architecture (in the left). Cloud Activation Function (right).

The RBF network computes the output data set as

$$\hat{y}_t = G(\mathbf{x}_t, \mathbf{c}, \mathbf{v}) = \sum_{j=1}^s v_{j,t} \psi_2(\mathbf{x}_t, \mathbf{c}_j) = \sum_{j=1}^s v_{j,t} o_{j,t}, \quad t = 1, 2, \dots, N \quad (6)$$

where \mathbf{x} is a k -dimensional neural input vector, \mathbf{w}_j represents the hidden layer weights (parameters), ψ_2 are radial basis (Gaussian) activation functions, v_j are the trainable weights (parameters) connecting the component of the output vector \mathbf{o} . Weights can be adapted (estimated) by the error back-propagation algorithm. If the estimated output for the single output neuron is \hat{y}_t , and the correct output should be y_t , then the error e_t is given by $e_t = y_t - \hat{y}_t$ and the learning rule has the form

$$v_{j,t} \leftarrow v_{j,t} + \eta o_{j,t} e_t, \quad j = 1, 2, \dots, s, \quad t = 1, 2, \dots, N \quad (7)$$

where the term, $\eta \in (0,1)$ is a constant called the learning rate parameter, $o_{j,t}$ is the output signal from the hidden layer.

We replaced also the standard Gaussian activation (membership) function of RBF neurons with functions (see Fig. 1 right) based on the normal cloud concept [9] p. 113, see Fig. 1 right. Cloud models are described by three numerical characteristics: expectation (Ex) as most typical sample which represents a qualitative concept, entropy (En) and hyper entropy (He) which represents the uncertain degree of entropy. Then, in the case of soft RBF network, the Gaussian membership function $\psi_2(./.)$ in Eq. (6) has the form

$$\psi_2(\mathbf{x}_i, \mathbf{c}_j) = \exp\left[-(\mathbf{x}_i - E(\mathbf{x}_j)/2(En')^2)\right] = \exp\left[-(\mathbf{x}_i - \mathbf{c}_j)/2(En')^2\right] \quad (8)$$

where En' is a normally distributed random number with mean En and standard deviation He , E is the expectation operator. For details see [10].

Nonlinear SVR is frequently interpreted by using the training data set $\{y_k, \mathbf{x}_k\}_{k=1}^N$ with input data $\mathbf{x}_k \in \mathcal{R}^N$ and output data $y_k \in \mathcal{R}$ as follows

$$f(\mathbf{x}, \mathbf{w}, b) = \sum_{i=1}^N \mathbf{w}_i \varphi_i(\mathbf{x}) + b \quad (9)$$

where $\varphi_i(\mathbf{x})$ are called features (the input data are projected to a higher dimensional feature space). In order to perform SVM regression one optimizes the cost (empirical risk) function

$$R_{emp} = \frac{1}{N} \sum_{i=1}^N |y_i - f(\mathbf{x}, \mathbf{w})|_\varepsilon \quad (10)$$

which leads to solving of the quadratic optimization problem. More information can be found in [11].

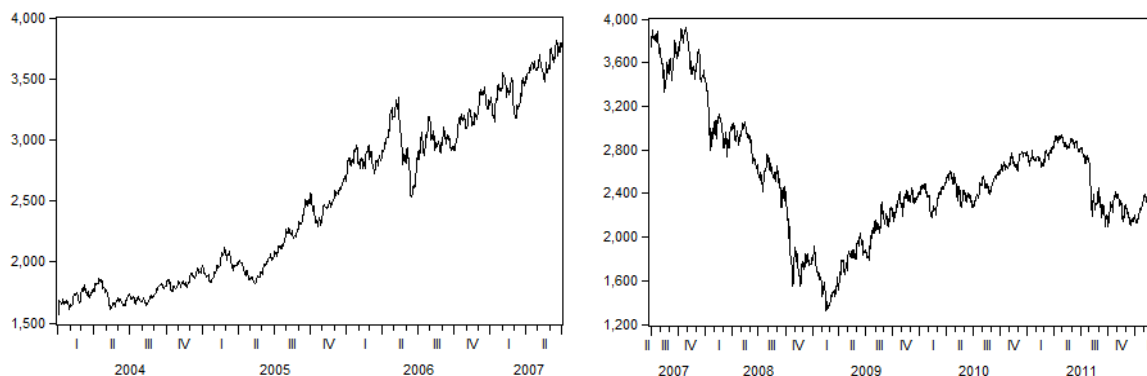
4 Building a statistical vs. soft computing prediction model for WIG20 values

To build a forecast model the sample period, we divided the basic period into two periods. First period (as the training data set) was defined from January 2004 to the end of June 2007, i.e. the time before the global financial crisis or pre crisis period, and the second one so called crisis and post crisis period (validation data set or ex post period) started at the beginning of July 2007 and finished by the March 8, 2012². Visual inspection of the time plot of the daily values of WIG20 index is given in Fig. 2. The daily time series depicted in Fig. 2 exhibits non-stationary behavior. However, as was confirmed by ADF test (see Tab. I) its first differencing become stationary.

² This time series can be obtained from <http://www.wse.com.pl>.

Table 1. The results of ADF test applied to the WIG20 time series values.

ADF test	t-statistics	p-value
Original WIG20 time series values	-1.881134	0.3415
WIG20 time series (first differences)	-45.55257	0.0001

**Figure 2.** Time series of the daily closing prices of WIG20 index. Left period (1.2004 – 6.2007), right period (7. 2007 – March 8, 2012).

Input (independent variables) selection is crucial importance to the successful development of an ARIMA/ARCH–GARCH model. Potential inputs were chosen based on traditional statistical analysis: these included the WIG20 indexes and lags thereof. The relevant lag structure of potential inputs was analysed using traditional statistical tools, i.e. using the autocorrelation function (ACF), partial autocorrelation function (PACF). According to these criteria the ARIMA(1,1,0) model was specified as follows

$$\Delta y_t = \xi + \phi_1 \Delta y_{t-1} + \varepsilon_t \quad (11)$$

where Δ is the difference operator defined as $\Delta y_t = y_t - y_{t-1}$. Estimated parameters of specified ARIMA(1,1,0) model are reported in Tab. 2.

Table 2. Estimated mean (Eq. 11) for WIG20 indexes.

Coeff.	Value	St. dev.	p-value	D-W
ξ	¹ 3.938 362	² 5.899 953	³ 0.504 ⁵	⁴ 1.994470
ϕ_1	0.047107	0.019045	0.0134	

As we mentioned early, high frequency financial data, like our WIG20 time series, reflect a stylized fact of changing variance over time. An appropriate model that would account for conditional heteroscedasticity should be able to remove possible nonlinear pattern in the data. Various procedures are available to test an existence of ARCH-type model. A commonly used test is the LM (Lagrange Multiplier) test. The LM test assumes the null hypothesis $H_0: \alpha_1 = \alpha_2 = \dots = \alpha_p = 0$ that there is no ARCH. The LM statistics has an asymptotic χ^2 distribution with p degrees of freedom under the null hypothesis. The ARCH-LM test up to 10 lags was statistically significant of the mean equation (10). For calculating the LM statistics see for example [4].

For estimation of the parameters of GARCH type model the maximum likelihood procedure was used and resulted into the following variance equation:

$$h_t = \alpha_0 + \sum_{i=1}^m \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^s \beta_j h_{t-j} = 675.0 + 0.125 \varepsilon_t^2 + 0.854 h_{t-1}. \quad (12)$$

Furthermore, to test for nonlinear patterns in WIG20 time series the fitted standardized residuals $\hat{\varepsilon}_t = e_t / \sqrt{h_t}$ were subjected to the BDS test. The BDS test (at dimensions $N = 2, 3$, and tolerance distances $\varepsilon = 0.5, 1.0, 1.5, 2.0$) finds no evidence of nonlinearity in standardized residuals of the WIG20 time series. Next, the variance model given by Eq. (12) was re-estimated considering that the residuals follow a Student's distribution, and after GED. The model with the lowest value of AIC fits the data best. Tab. 3 presents AIC, log likelihood functions (LL) in all cases.

Table 3. Information criteria and log-likelihood functions for re-estimated asymmetric variance models.

Model Criteria	PGARCH	EGARCH	Distribution
Period: 1.2004 - 6.2007			
AIC	9.5853	9.3838	Student's
LL	-4353.294	-4353.642	
AIC	5.5728	5.5717	GED
LL	-4347.636	-4348.162	

As we can see in Tab. 3, the smallest AIC has just the EGARCH(1,1) with GED distribution. After these findings we re-estimate the mean Eq. (10) assuming that the random component ε_t follow EGARCH(1,1) GED. The final estimated prediction model has the form

$$\Delta y_t = 2.0492 - 0.0462_1 \Delta y_{t-1} + \varepsilon_t \quad (\text{GED}) \quad (13)$$

Actual and fitted values of the WIG20 index calculated according to model (13) we can view in Fig. 3.

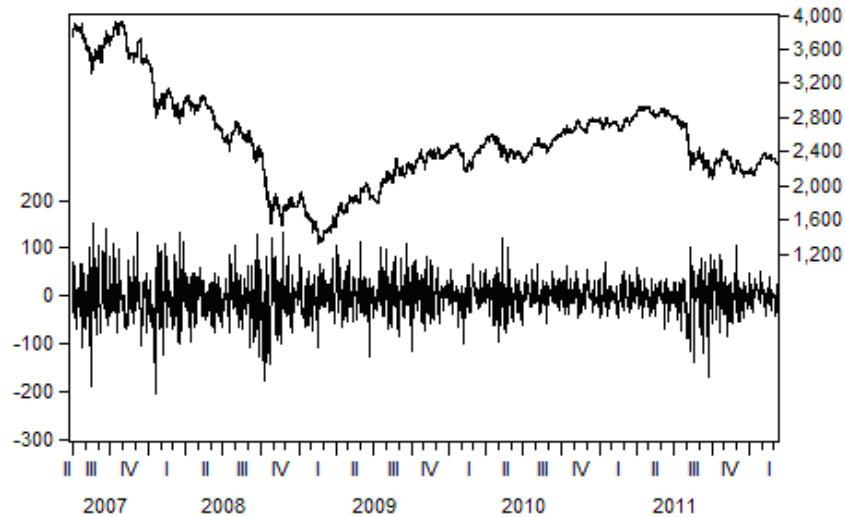


Figure 3: Actual and fitted values of the WIG20 index. Residuals are at the bottom. Actual time series represents the solid line, the fitted vales represents the dotted line (statistical approach).

The granular RBF NN was trained using the variables and data sets as the ARIMA(1,1,0)/EGARCH(1,1) model (13). In G RBF NN, the non-linear forecasting function $f(\mathbf{x})$ was estimated according to the expressions (6) with RB function $\psi_2(. /.)$ given by Eq. (8). The

detailed computational algorithm for ex post forecast RMSE values and the weight update rule for the granular network is shown in [10]. The fitted vs. actual WIG20 indexes for the validation data set are graphically displayed in middle of Fig. 4.

The prediction of WIG20 values for the post-crisis period was also done by SVR model using *gretl software*³. *Gretl software* is the implementation of Vapnik's Super Vector Machine [12] for the problem of pattern recognition, regression and ranking function. The fitted vs. actual values of the WIG20 time series for the validation data set (post-crisis period) are graphically displayed in Fig. 4 right.

5 Empirical Comparison and Discussion

Table in Fig. 4 left presents the summary statistics of each model based on RMSE and MAPE calculated over the validation data set (ex post period). This table shows the results of the methods used for comparison. The best performing method is G RBF NN followed SVR. A comparison between latest statistical and intelligent methods shows that intelligent prediction methods outperformed the latest statistical forecasting method. Further, from table in Fig. 4 left it is shown that both forecasting models used are very accurate. The development of the error rates on the validation data set showed a high inherent deterministic relationship of the underlying variables. Though promising results have been achieved with all approaches, for the chaotic financial markets a purely linear (statistical) approach for modelling relationships does not reflect the reality. For example if investors do not react to a small change in exchange rate at the first instance, but after crossing a certain interval or threshold react all the more, then a non-linear relationship between Δy_t and Δy_{t-1} exist in model (11).

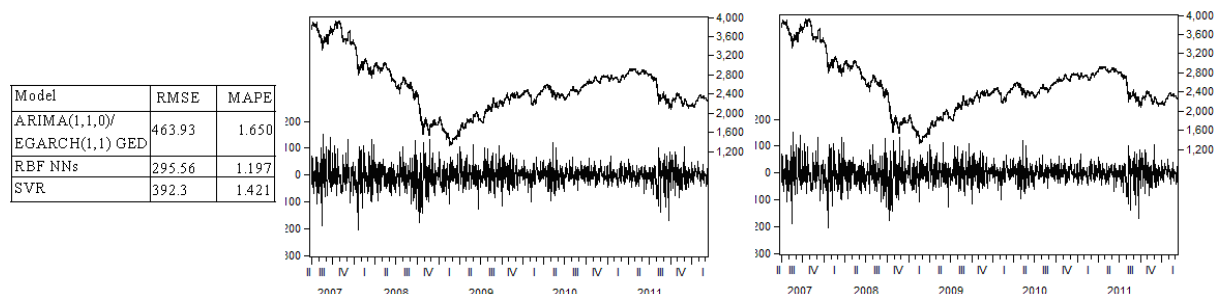


Figure 4. Table in the left presents the summary statistics of each model based on RMSE and MAPE calculated over the validation data set (ex post period). Actual and fitted values of the WIG20 index for G RBF NN (in middle)) and for SVR model (right). Residuals are at the bottom. Actual time series represents the solid line, the fitted vales represents the dotted line.

6 Conclusion

In managerial decision-making, risk and uncertainty are the central categories based on which the effects of individual variants are assessed, and subsequently the final decision is chosen from several variants. In the present paper we proposed two approaches for predicting the BUX time series. The first one was based on the latest statistical ARIMA/ARCH methodologies, the second one on neural version of the statistical model and SVR.

After performed demonstration it was established that forecasting model based on SVR model is better than ARIMA/ARCH one to predict high frequency financial data for the BUX time series.

³ See <http://gretl.sourceforge.net>

The direct comparison of forecast accuracies between statistical ARCH-GARCH forecasting models and its neural representation, the experiment with high frequency financial data indicates that all investigated methodologies yield very little MAPE (Mean Percentage Absolute Error) values. Moreover, our experiments show that neural forecasting systems are economical and computational very efficient, well suited for high frequency forecasting. Therefore they are suitable for financial institutions, companies, medium and small enterprises.

The results of the study showed that there are more ways of approaching the issue of risk reducing in managerial decision-making in companies, financial institutions and small enterprises. It was also proved that it is possible to achieve significant risk reduction in managerial decision-making by applying modern forecasting models based on information technology such as neural networks developed within artificial intelligence. In future research we plan to extend presented methodologies by applying fuzzy logic systems to incorporate structured human knowledge into workable learning algorithms.

7 Acknowledgement

This paper was supported within Operational Programme Education for Competitiveness – Project No. CZ.1.07/2.3.00/20.0296, and partially by project 7AMB14PL029.

8 References

- [1] Kennedy, P.: A Guide to Econometrics. Oxford, Basil, Blackwell,(1992)
- [2] Holden, K.: Developments in Dynamic Modelling in Economics. Proceedings of the Mathematical Methods in Economics. International Scientific Conference. VŠB TU Ostrava (1997) Marcek, D., Frano, M., Marcek, M.: Managerial Decision-making: Measuring and Manifestations of Risks and the Possibilities of their Reducing, Journal of Economics, Vol. 59/4, 2011, 392–408.
- [3] Box, G.E.P., and Jenkins, G.M.: Time Series Analysis, Forecasting and Control. San Francisco, CA: Holden-Day, 1970.
- [4] Engle, R.F.: Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kindom Inflation. Econometrica, Vol. 50, No. 4, 1982, 987–1007.
- [5] Bollerslev, D.: Generalized Autoregressive Conditional Heteroscedasticity, *Journal of Econometrics* Vol. 31, 1986, pp. 307–327.
- [6] Nelson, D.B.: Conditional Heteroskedasticity in Asset Returns: A New Approach, *Econometrica* 59 (2),1991, 347-370.
- [7] Ding, Z., Granger, C.W. and Engle, R.F.: A Long Memory Property of Stock Market Returns and a New Model, Journal of Empirical Finance, 1, 1993, pp. 83-106.
- [8] Sed'a, P.: Asymmetric Conditional Volatility Modeling: Evidence from Central European Stock Markets. 8th Int. Scietific C. on Firms and Financial Institutions, VŠB-TU Ostrava, Faculty of Economics, 2011, 375-383.
- [9] Li, D., and Du, Y.: *Artificial intelligence with uncertainty*. (Boca Raton: Chapman & Hall/CRC, Taylor & Francis Group, 2008.
- [10] Marcek, M., Pancikova, L. and Marcek, D.: *Econometrics & Soft Computing*. The University Press, Zilina, 2008.
- [11] Vapnik, V.: The support vector method of function. In: Nonlinear Modeling: Advanced Black-Box Techniques, Suykens, J.A.K., Vondewalle, J. (Eds.), Kluwer Academic Publishers, Boston: 1998, 55-85.
- [12] Vapnik, V.: *The nature of statistical learning theory*. Springer Verlag, New-York, 1995.

An influence of changing conditions within the EU ETS system on the steel company

¹František Zapletal, ²Jan Ministr

¹VŠB – Technical University of Ostrava, Faculty of Economics, Department of Systems Engineering
Sokolská 33, 701 00 Ostrava
frantisek.zapletal@vsb.cz

² VŠB - Technical University of Ostrava, Faculty of Economics, Department of Applied Informatics
Sokolská 33, 701 00 Ostrava
jan.ministr@vsb.cz

Abstract

Heavy industrial companies in the EU have to face specific environmental factor – carbon emission trading. This system is considered to be the main environmental tool of the European Union. The power of its influence is frequently discussed topic and it was researched within many studies. The aim of this paper is to assess an effect of two main factors of the EU ETS system – an amount of free allocated allowances and emission permit price. This is done using the parameterization and sensitivity analysis of the simple optimization model maximizing the total company profit and involving the emission trading factor. Analyses in this paper are performed on the data of one concrete steel company in the Czech Republic. An influence of chosen factors is assessed for the second and third phase of the EU ETS, i.e. from 2008 to 2014.

Keywords

EU ETS, emission trading, carbon market, EUA, mathematical programming.

Abstrakt

Těžké průmyslové podniky v EU musí čelit specifickému faktoru prostředí – obchodování s emisemi uhlíku. Tento systém je považován za hlavní nástroj pro životní prostředí Evropské unie. Jeho vliv je často diskutované téma a to byl zkoumán v mnoha studiích. Cílem této práce je zhodnotit vliv dvou hlavních faktorů systému EU ETS – bezplatné povolenky přidělené a emisní cena povolení. To se provádí pomocí Parametrizace a citlivosti analýzy jednoduché optimalizace modelu maximalizace zisku celé společnosti a zahrnuje obchodování s faktorem. Analýzy v tomto dokumentu jsou prováděny na datech jedné konkrétní ocelářské společnosti v České republice. Vliv vybraných faktorů se posuzuje pro druhou a třetí fázi EU ETS, tj. od roku 2008 do roku 2014.

Klíčová slova

EU ETS, obchodování, trh s uhlíkem, EUA, matematické programování.

1 Introduction

The aim of this paper is to analyze an influence of selected factors of the European emission trading system (EU ETS) on profit of participating companies. Companies must face many constraints on their production. Except of traditional economical and legislative restrictions, European industrial companies are affected also by legislative constraints protecting the environment. Thus, decision-making on production of these companies is influenced by environmental factors headed by carbon emissions trading. All analyses will be performed for companies of so called *carbon leakage* sector. This sector involves selected industrial branches where an amount of emissions released to the atmosphere is extra large. This sector was established by [3]. It is not necessary to involve also other companies to analysis of influence of an amount of freely allocated permits by the EU because these

remaining installations either do not currently get any such free allowances or at least it will be so in very short term. An influence of emission permit price is the same for both carbon leakage company and non-carbon leakage company. All the analyses will be performed using data of one Czech steel company. The steel sector is also involved in the carbon leakage sector, see [3].

Emission trading within the European Union is a frequent object of many researches but most of them have analysed EU ETS' factors only as a whole (e.g. system efficiency analysis, econometrical analysis for forecasting the price of allowances etc.) and only few of them investigate their influence on companies. Some optimization models maximizing companies' outputs with respect to the EU ETS have been already designed (e.g. [1], [6], [9] or [10]). One of them [9] will be also used for analyses in this paper. The main benefit of this paper is the complex assessment of the EU ETS' influencing factors for the whole EU ETS system's lifetime till September 2014.

The paper is organized as follows. After this short introduction, Chapter 2 containing basic principles of the EU ETS system follows. Legislative background and links between the system and companies are presented there briefly. Chapter 3 consists of the optimization model design and basic data on researched parametres of the EU ETS and their descriptive statistics. Results of performed analyses and their critical discussion can be found in the Chapter 4. Chapter 5 describes basic Software support of optimizing the company EU ETS system. Finally, the paper is ended by conclusions and suggestion of future possible research topics related with this paper.

2 Emission Trading Scheme of the EU (EU ETS)

The EU ETS is the main tool of the EU's environmental policy which was established in 2005 by [2]. The core idea of the system is that each ton of the CO₂ released to the atmosphere by company must be covered by one emission permit. That means that increase in production leads to increase in profit but, on the other hand, it will also cause an increase in need of emission allowances and thus increase in costs. Currently, the EU ETS involves more than 12 000 industrial installations inside the EU. A lifetime of the EU ETS is divided into phases – phase 1 (2005-2007), phase 2 (2008-2012) and current phase 3 (2013-2020). Conditions of emission trading have been changing gradually, see e.g. [7.]

Emission permit flows between a company and its environment are illustrated in Figure 1.

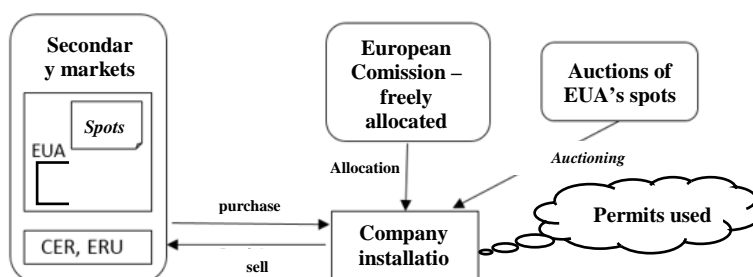


Fig. 1: Emission permit flows between a company and its environment

In the figure 1, it can be seen that three possible sources of allowances exist for companies. The main source for companies from the already mentioned carbon leakage sector (steel companies, paper mills, brick producers etc.) is a flow of freely allocated permits by the European Commission. Other two flows are common for all the companies in the EU ETS – additional permits can be purchased either on a secondary market or via emission auctions for a market price. On the other hand, only one possibility to sell unused permits is accessible for all the companies – secondary market where permits are traded. All flows mentioned above refer to two main conditions affecting companies –

market price of allowances and an amount of allowances granted to companies for free. These two parameters of the system will be analysed further.

Allowances of the EUA type (European Union Allowance) in the form of a spot is the basic and the most traded financial instrument of the EU ETS system. Above that, derivatives of EUAs also exist, but they are of interest rather for speculators on the financial market. Except of EUAs, companies can also use allowances which have their origin in the Kyoto worldwide emission trading system – CERs (Certified Emission Reduction). These CERs are highly beneficial for European companies because their price is much cheaper than in case of EUAs. The EU have come with 10% quota for using CERs by European companies to prevent companies from over-using CERs which would lead to EUA's price drop, see [2].

2.1 Secondary markets trading allowances and emission auctions

Secondary market has been allowed to use since the very beginning of the EU ETS system in 2005. It consists of many stock exchanges all around Europe (e.g. SendeCO2 in Spain, EEX in Germany, ICE in Great Britain etc.). Analysis of EUA allowance price dependency among various stock exchanges was performed in [7]. Almost perfect correlation (with the correlation coefficient greater than 99%) was proven. That is why only prices from one chosen stock exchange (SendeCO2) were used for analyses in this paper. Two reasons for this choice exist. First, SendeCO2 provides data on emission price on its websites clearly and for free. Second, this stock exchange (as one of few) enables also CERs trading.

Emission auction is a new channel for purchasing the permits launched in 2013 [3]. The reason for this change was the fact that conditions for allowances allocation had been changed for the third phase of the EU ETS system. The already mentioned research [7] proved that almost perfect correlation between prices at auctions and prices on the secondary market exists (with correlation coefficient exceeding 99%). That is why a possibility of emission auctioning will not be included in following analyses.

3 Optimization model and input data

In this chapter, the optimization model together with input data from the EU ETS system on allowance prices for further analyses are presented.

3.1 Optimization model maximizing the total profit of a company

The following deterministic optimization model is aggregation of models presented in [8] and [9]. Model is deterministic like the one in [8], on the other hand it enables using the CER type of allowances like the one in [9].

Model presumptions are as follows:

- all the model parameters are considered to be deterministic;
- model is static, for only one period;
- decision is made at very beginning of the period and it cannot be changed further;
- company always chooses a possibility of using the greatest possible amount of CERs (i.e. 10%) to cover its emissions.

The last presumption is supported by the fact that EUAs have never been cheaper than CERs so far. Currently, CERs are more than fifty times cheaper.

$$\max_{x,y} \{m^T y + (r - 0.9 \cdot e^T x) \cdot p^{EUA} - 0.1 \cdot e^T x \cdot p^{CER} - c\},$$

$$\begin{aligned} \text{z. p.} \quad & \mathbf{y} = (\mathbf{E} - \mathbf{A})\mathbf{x}, \\ & \mathbf{d}^c \leq \mathbf{y} \leq \mathbf{d}^e, \\ & \mathbf{x} \leq \mathbf{v}, \\ & \mathbf{x} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}, \end{aligned}$$

kde:

- $\mathbf{y} \in \mathbf{R}^n$ is a vector of product sales;
- $\mathbf{x} \in \mathbf{R}^n$ is a vector of company's production;
- $\mathbf{e} \in \mathbf{R}^n$ is a vector of carbon coefficients indicating an amount of CO₂ released by production of a one unit of a particular product;
- p^{EUA} is a price of the EUA allowance type;
- p^{CER} is a price of the CER allowance type;
- r is an amount of emission permits granted to company for free;
- $\mathbf{m} \in \mathbf{R}^n$ is a vector of company's margins;
- c stands for total fixed costs of a company;
- $\mathbf{E} \in \mathbf{R}^{n \times n}$ is a unit matrix;
- $\mathbf{A} \in \mathbf{R}^{n \times n}$ is a matrix of technical emission coefficients of a production;
- $\mathbf{d}^c \in \mathbf{R}^n$ is a vector of lower bounds for sales (given by already signed contracts)
- $\mathbf{d}^e \in \mathbf{R}^n$ is a vector of expected demands for the investigated period;
- $\mathbf{v} \in \mathbf{R}^n$ is a vector of company's production capacities.

The reduction of model variables would be possible by substituting of \mathbf{y} by \mathbf{x} (in accordance with the first model constraint). This form was chosen in order to keep the transparency of the model. Fixed cost (c) is not dependent on any model variable and thus it does not affect the result of optimization. Therefore, it would be possible to optimize the model excluding c and then to decrease the optimal value by the value of fixed costs in the end.

3.2 Input data for analyses

To be able to perform analyses declared in the introduction of this paper, many data are required as for example data of some concrete industrial company on its production and data related with the EU ETS system (amounts of freely allocated allowances and allowance prices). Figure 2 and figure 3 show a development of EUAs and CERs, respectively. In figure 4, it can be seen a development of amounts of allowances granted for free to the modelled company from the very beginning of the EU ETS system (2005) till the end of the third phase of the system (in 2020). Basic descriptive statistics of data sets mentioned above can be found in tables 1a-1c and 2. Many changes in EU ETS conditions were realised between the third phase and the second phase so these statistics are shown also separately for these periods (the second phase is shown in table 1b and the third phase in table 1c). All the data were taken from the Spanish stock exchange Sendeco2⁴. Data on amounts of freely allocated allowances were gained from the CarbonMarketData database⁵. The main difference between prices and these amounts is the fact that amounts are determined directly by the central authority and prices are determined by the market. That is why data till the end of the third phase of the EU ETS are already known.

All the outputs mentioned above were processed using the SPSS 22 software.

⁴ Sendeco2.com

⁵ www.carbonmarketdata.com

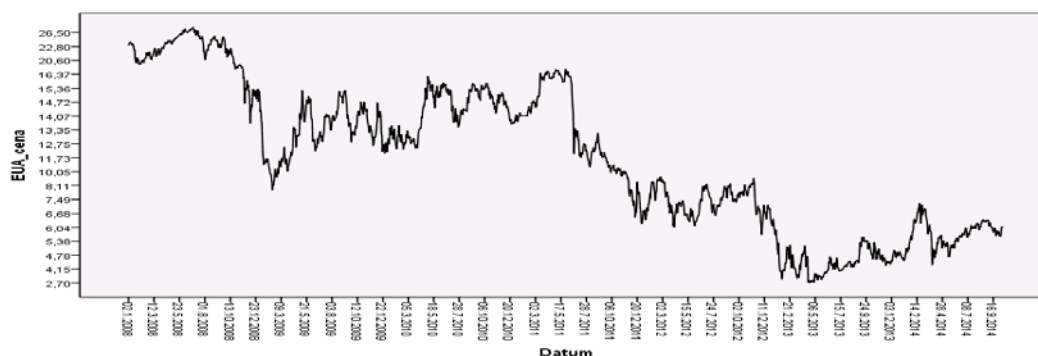


Fig. 2: EUA price development on the SendeCO2 stock exchange in the second and third phase of the EU ETS

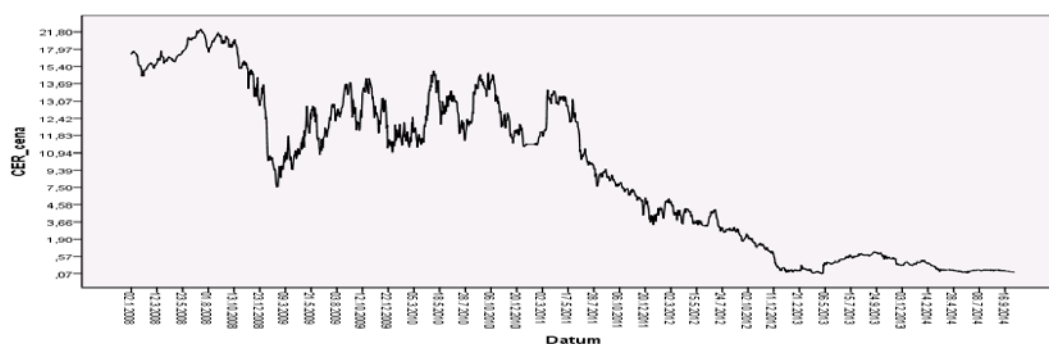


Fig. 3: CER price development on the SendeCO2 stock exchange in the second and third phase of the EU ETS

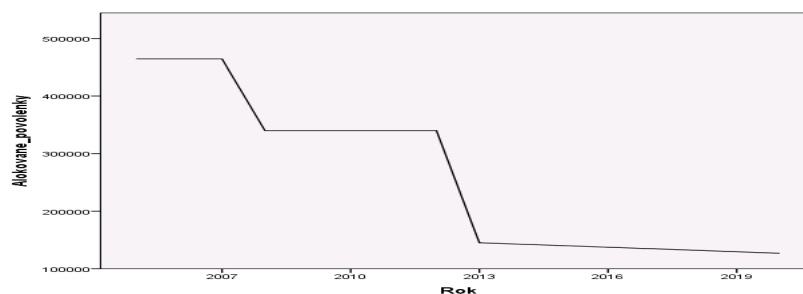


Fig. 4: Development of freely allocated amount of EUAs to the modelled company (2005-2014)

Tab. 1a: Descriptive statistics of allowance prices in the second and third EU ETS phases

	N	Range	Minimum	Maximum	Mean	Std. Deviation	Skewness		Kurtosis	
	Statistic	Statistic	Statistic	Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic	Std. Error
EUA	1721	25,60	2,70	28,30	11,5689	5,96453	,646	,059	-,292	,118
CER	1721	22,53	,07	22,60	8,1620	6,41197	,078	,059	-1,288	,118
Valid N (listwise)	1721									

Tab. 2b: Descriptive statistics of allowance prices in the second EU ETS phase

	N	Range	Minimum	Maximum	Mean	Std. Deviation	Skewness		Kurtosis	
	Statistic	Statistic	Statistic	Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic	Std. Error
EUA	1266	22,57	5,73	28,30	13,9231	5,20558	,630	,069	-,069	,137
CER	1266	22,45	,15	22,60	10,9838	5,07435	-,292	,069	-,434	,137
Valid N (listwise)	1266									

Tab. 3c: Descriptive statistics of allowance prices in the third EU ETS phase

	N	Range	Minimum	Maximum	Mean	Std. Deviation	Skewness		Kurtosis	
	Statistic	Statistic	Statistic	Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic	Std. Error
EUA	455	4,42	2,70	7,12	5,0184	,90409	-,052	,114	-,521	,228
CER	455	,65	,07	,72	,3107	,18098	,594	,114	-,997	,228
Valid N (listwise)	455									

Tab. 2: Descriptive statistics of amounts of allowances granted to the modelled company for free

	N	Minimum	Maximum	Mean	Std. Deviation
V2	16	126884	464505	261386,25	136764,008
Valid N (listwise)	16				

The data on the modelled company were provided by one steel company in the Czech Republic. This company requires to be kept in anonymity because of the data privacy. However, this fact does not influence results of analyses performed.

4 Model verification and sensitivity analysis

Results of optimization of the model established in the previous chapter are shown in the table 3. It can be seen that the modelled company is currently in loss of about 4 mil. EUR where about 10% of this loss is caused by the emission trading. These results correspond to values of input parameters current in September 2014 (EUA price equals to 6.2 EUR, CER price equals to 0.56 EUR and 145,098 permits granted to the company for free.

Now, sensitivity analyses of obtained results will be performed when the input data change in a range of historical data shown in tables 1a-1c and 2.

Tab. 3: Results of the steel company's profit optimization

		V1	V2	V3	V4	V5
[tons]	Production	431451.3	576840.5	665365.6	120000	30000
[tons]	Sales	0	12000	620545.6	120000	30000
[EUR]	Objective function optimum:	-4013185		Costs induced by allowances:	-406903	
[EUR]	Optimal value excluding fixed costs:	47638910.05				

A sensitivity analysis was performed using the parametrization in the MS Excel 2013 software.

Macro application in Visual Basic was designed to enable the batch running of optimization in the Solver tool of the MS Excel. Fig. 5 presents the program code of the macro. The macro optimizes the K3 cell while values of variables B5-F5 are changed. The parameter in the cell (3,16) is changed by a value of one step for each iteration. This step is determined by the value of the cell (3,19). *pocet* variable, which can be found in the cell (3,20), determines the number of iterations. Optimal values of variables are copied to rows below the model automatically.


```

Sub MakroR()
    Dim pocet As Integer
    pocet = Worksheets("List1").Cells(3, 20).Value
    For i = 1 To pocet
        Sheets("List1").Select
        ActiveWindow.SmallScroll Down:=-45
        SolverOk SetCell:="$K$3", MaxMinVal:=1, ValueOf:="0", ByChange:="$B$5:$F$5"
        SolverSolve UserFinish:=True
        SolverOptions AssumeNonNeg:=True
        SolverFinish KeepFinal:=1
        Range("P3").Select
        Selection.Copy
        Worksheets("List1").Cells(49 + i - 1, 1).Select
        Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
            :=False, Transpose:=False
        Range("B5:F5").Select
        Application.CutCopyMode = False
        Selection.Copy
        Worksheets("List1").Cells(49 + i - 1, 2).Select
        Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
            :=False, Transpose:=False
        Range("B3:F3").Select
        Application.CutCopyMode = False
        Selection.Copy
        Worksheets("List1").Cells(49 + i - 1, 7).Select
        Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
            :=False, Transpose:=False
        Range("K3").Select
        Application.CutCopyMode = False
        Selection.Copy
        Worksheets("List1").Cells(49 + i - 1, 12).Select
        Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
            :=False, Transpose:=False
        Worksheets("List1").Cells(3, 16).Value = Worksheets("List1").Cells(3, 16).Value + Worksheets("List1").Cells(3, 19).Value
    Next i
End Sub

```

Fig. 5: Program code for optimization model parametrization in Visual Basic

As mentioned above, conditions of the EU ETS system have been changing very quickly during its phases which influenced a development of both analysed parameters – emission permits' price and amounts of freely allocated permits to companies. That is why these factors will be further analysed both for each phase individually and for the second and third phase together.

Figures 6a-8 show the results of optimization. All these figures contain the visualization of dependency of optimal profit (“%UF” curve) and costs given by emission trading on a change of chosen parameter (value ranges of parameters were determined on the base of value range from previous periods. In order to keep higher clarity of the figures, values of dependent variables are given in percentage change in comparison with the current state (the current state is indicated by dotted vertical lines). Mean values of probability distributions (μ) of explanatory variables are indicated by dashed vertical lines. Borders of blue rectangles correspond to bounds of intervals $[\mu - \sigma; \mu + \sigma]$, where σ stands for a standard deviation.

Figures 6a-6c show an influence of changes in EUA prices (figure 6a for values of parameters according to the variation range in the second and third EU ETS phase together, figure 6b for values of the second phase only and figure 6c of the third phase only). Figures 7a-7c present an influence of changes in CER prices (figure 7a for values of parameters according to the variation range in the second and third EU ETS phase together, figure 7b for values of the second phase only and figure 7c of the third phase only). Finally, figure 8 demonstrates an influence of freely allocated allowances to the modelled company on the profit of this company.

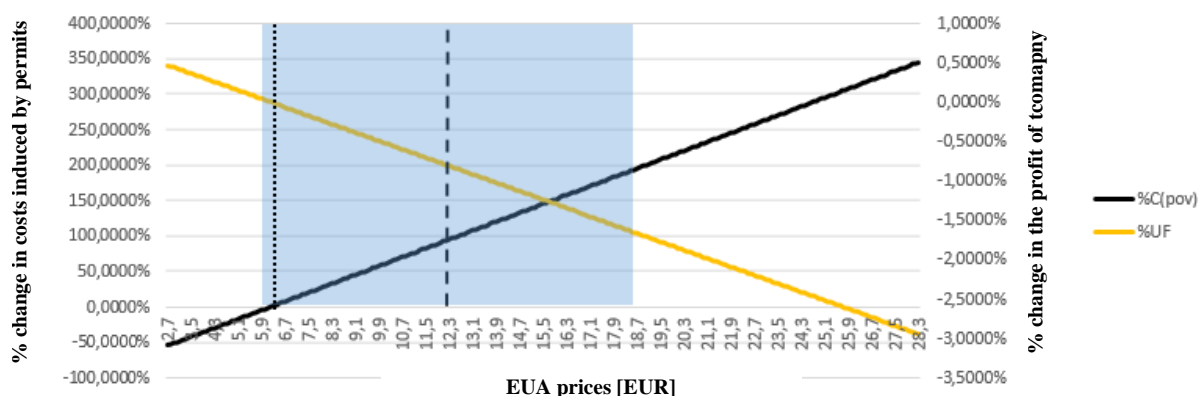


Fig. 6a: Influence of EUA prices in the second and third EU ETS phase on chosen measures of the modelled company

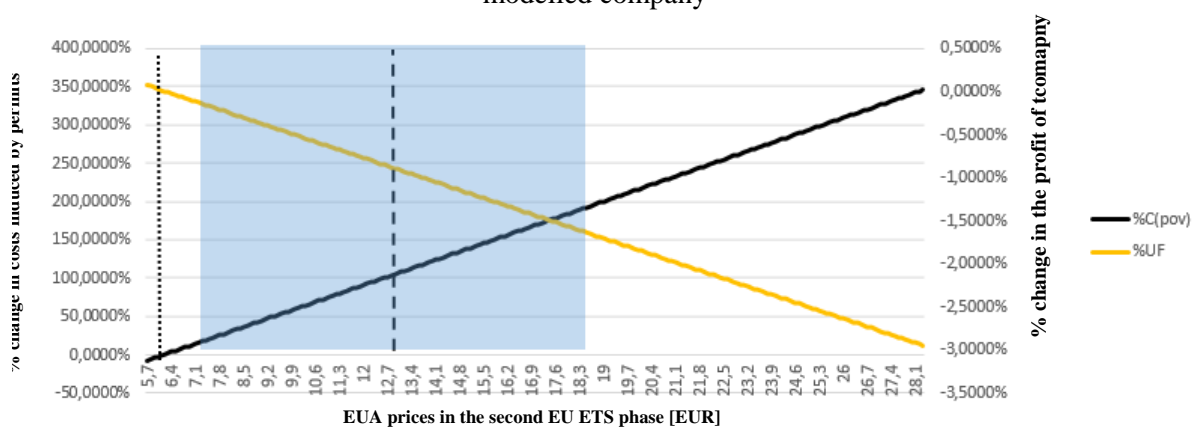


Fig. 6b: Influence of EUA prices in the second EU ETS phase on chosen measures of the modelled company

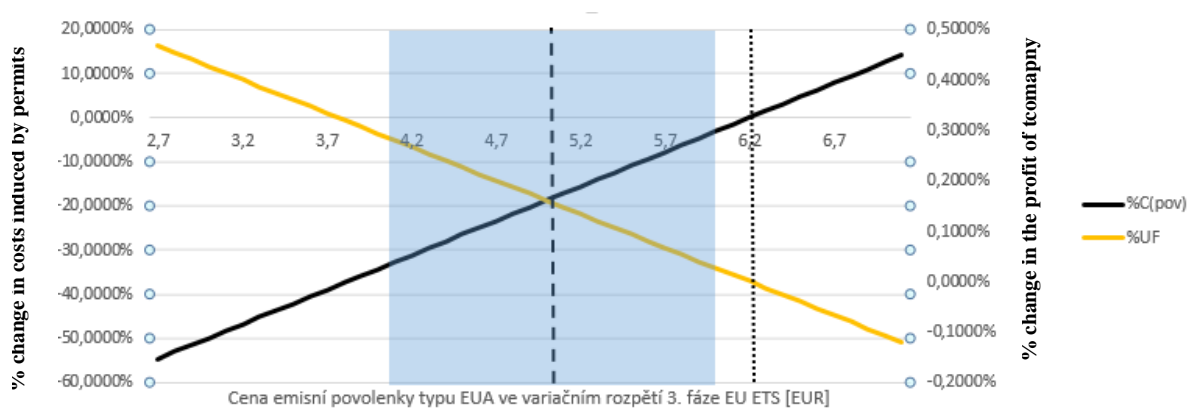
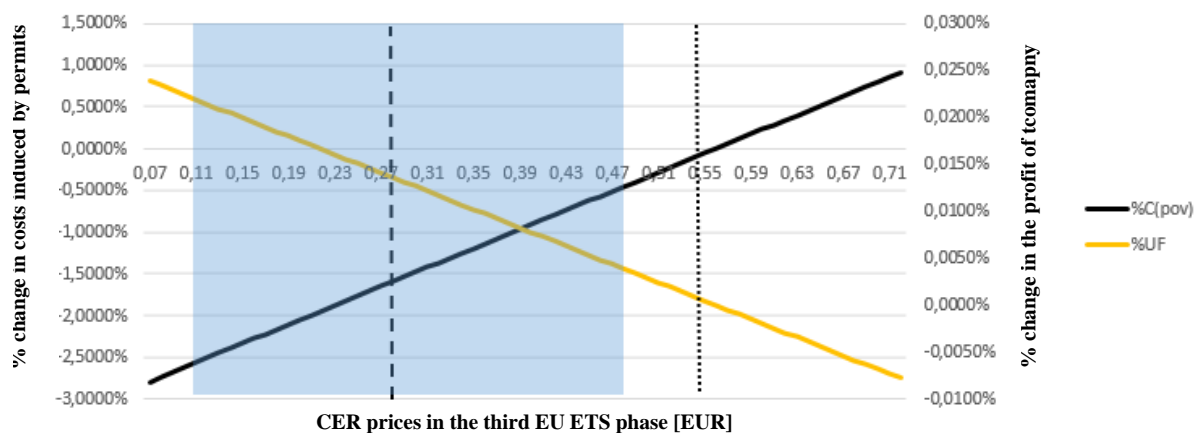
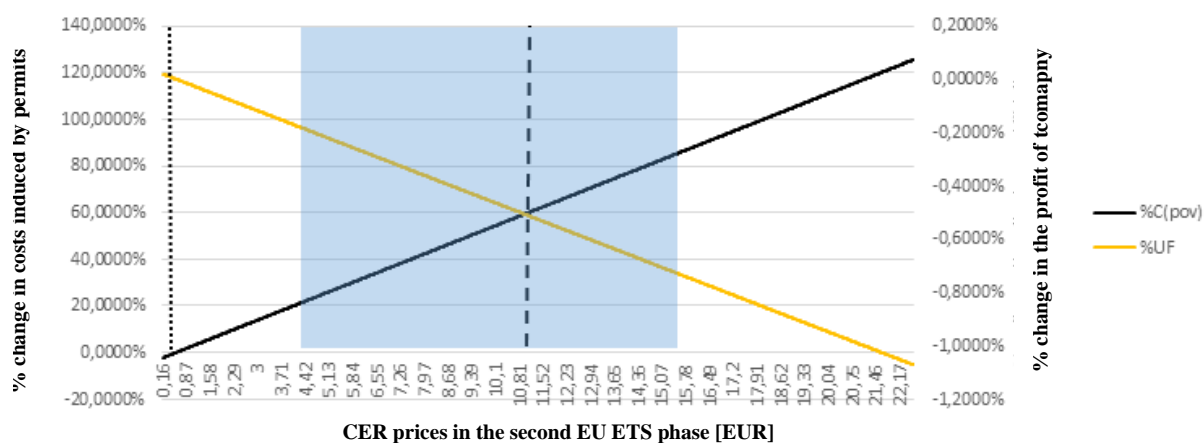
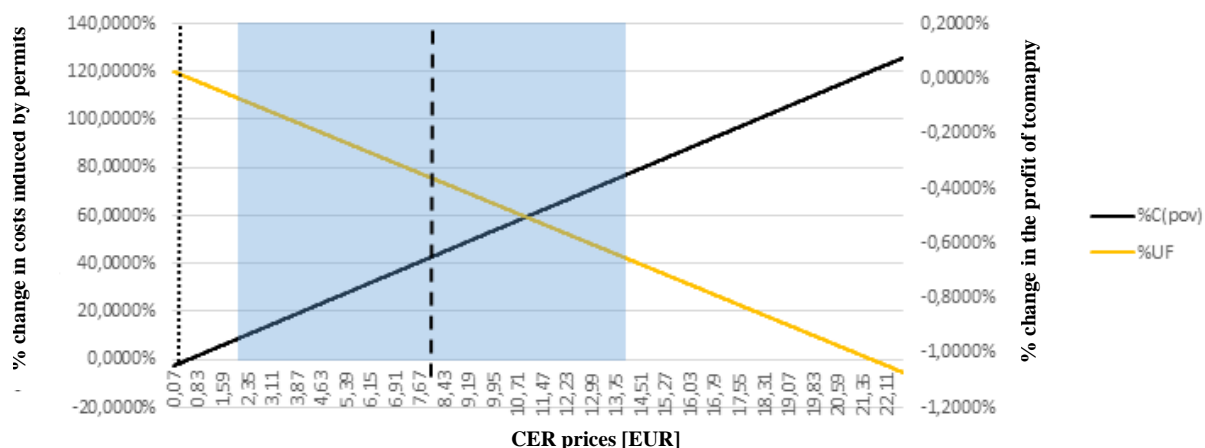


Fig. 6c: Influence of EUA prices in the third EU ETS phase on chosen measures of the modelled company



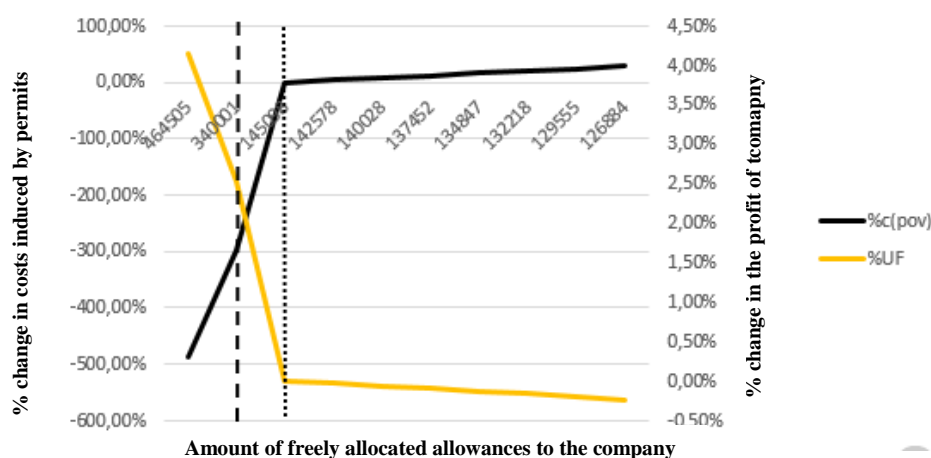


Fig. 8: An influence of the free allocated amount of allowances on selected measures of the modelled company

Figures 6a-8 demonstrate the dynamics of allowance prices' development. Analysis of individual EU ETS phases showed that current EUA and CER prices do not lie nearer than the standard deviation from the mean value (they lie out of blue rectangles in figures above). In comparison with the second phase, current prices are lower, and in comparison with the whole third phase, prices are higher, on the contrary. This fact emphasises a need of further analysis of potential allowance prices' impacts on companies.

Analyses showed that potential influence of EUAs is greater than in the case of CERs. An impact of EUA prices from both EU ETS phases on company's profit (without fixed costs) lies in $[-3; 0.46]\%$ in comparison with the current state. The same analysis but only taking variation range of values from the third EU ETS phase specified a shorter interval $[-0.12; 0.46]\%$. An influence of EUA prices on costs induced by emission trading can be expressed by interval $[-54; 345]\%$ while taking into account values from both phases and by interval $[-57; 222]\%$ for values only from the third phase. In financial terms, EUA prices can affect the profit by $[-1,404; 222]$ thousand of EUR for values from both EU ETS phases and by $[-57; 222]$ thousand of EUR for values from the third EU ETS phase only.

An influence of CER prices is weaker in comparison with EUA prices, especially due to the legislative restriction in the form of 10% limit for European companies which was already mentioned. Percentage change of company's profit (excluding fixed costs) in comparison with the current state taking into account values from both the second and third EU ETS phases lies in $[-1.07; 0.024]\%$. The same analysis but only taking variation range of values from the third EU ETS phase specified a shorter interval $[-0.078; 0.024]\%$. An influence of CER prices on costs induced by emission trading can be expressed by interval $[-2.79; 125.56]\%$ while taking into account values from both phases and by interval $[-2.79; 0.912]\%$ for values only from the third phase. In financial terms, CER prices can affect the profit by $[-510; 11]$ thousand of EUR for values from both EU ETS phases and by $[3.8; 11]$ thousand of EUR for values from the third EU ETS phase only.

The last analyzed factor was an amount of freely allocated permits and it turned out to be the most influencing EU ETS system's parameter. The data collected for all available years were used (that means the data for the period 2005-2020). It was discovered that the involved variation range would cause a change in profit in $[-0.24; 4.16]\%$ (that means in $[-112.9; 1980.3]$ thousand of EUR)

considering all else being equal. An influence of the same factor on costs induced by emission trading lies in [- 487.7; 27.8]% or [-112.9; 1980.3] thousand of EUR, respectively.

Although an amount of freely allocated permits was identified as the most influencing factor for companies, another important fact should be taken into account. Too high amounts of allowances were granted to companies especially in the first EU ETS phase. Corresponding potential great increase in company's profit and decrease in costs induced by emission trading are caused by these high values for the first EU ETS phase. However, values of amounts are already known till the end of the third trading phase (till 2020) and these values are going to decrease slightly for each following year. That is the reason why emission prices can be considered as a greater threat for companies than freely allocated EUA amounts. Because allocated amounts are known in advance, companies can adjust their decisions on production and investments also in advance. But emission prices are very difficult to predict. Presented risks could be also investigated by some specialised risk assessing methods in the frame of risk management like the FMEA method, see e.g. [5].

5 Software support of optimizing the company EU ETS system

For the a pilot development software support has been selected Unified Process object methodology. The software solution has been decomposed into next six Use Cases (see Fig. 9):

- *User Authentication* that provides user's basic authentication. Users have two roles (administrator and user of EU ETS) with different rights of access to system.
- *Purchases and Sell* that provides purchasing and selling of the emission permits on secondary markets and auctions including the obtain information about the development price of emission permits.
- *Free Allocation* that provides information on the number of emission allowances received free of charge.
- *Optimization* that optimizes emission trading by the above described optimization model due to the economic performance of the company.
- *Sensitive Analysis* that performs what if analysis on the basis of parameters set. This Use case fully utilizes the Excel program, especially the graphical interface.
- *Data Mart Administration* that manages enterprise data and external data sources which are necessary for the implementation of optimization and sensitivity analysis. The Class diagram is shown on Fig. 10.

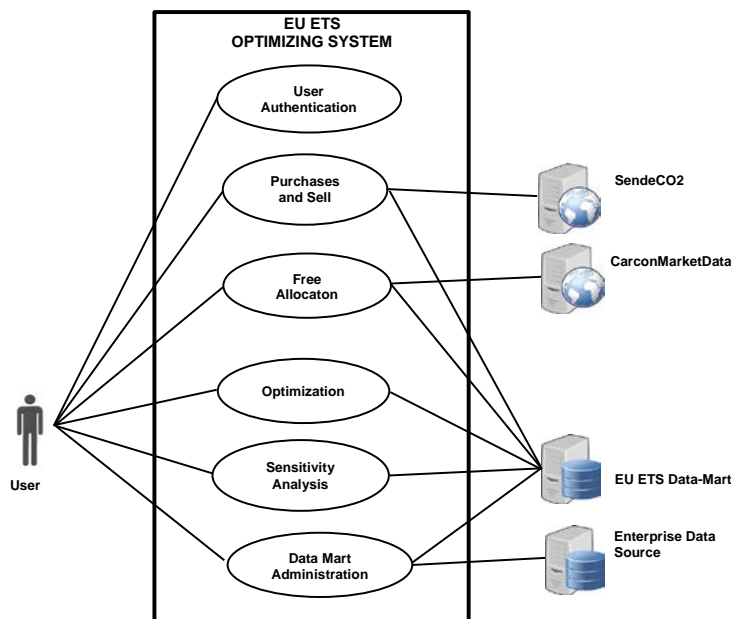


Fig. 9: Use Case Diagram of EU ETS Optimizing System

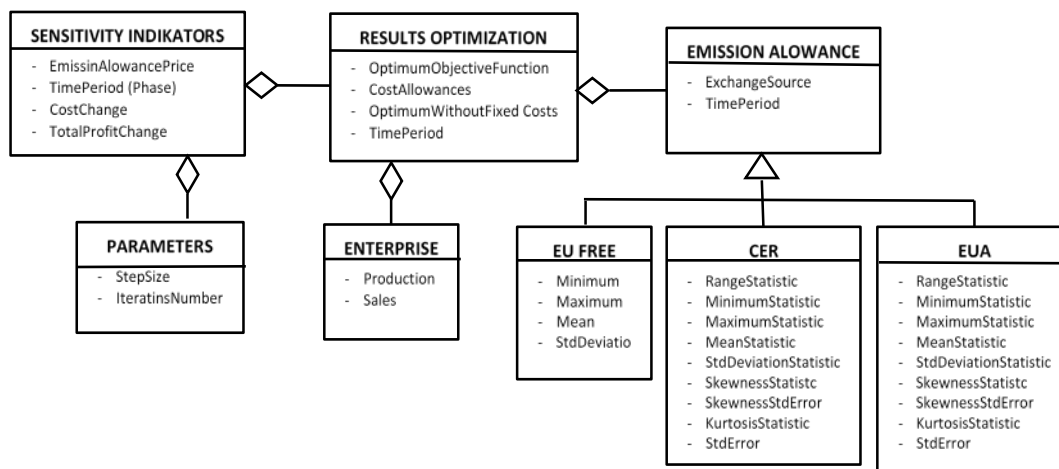


Fig. 10: Class Diagram of EU ETS data Mart

Individual Use Cases were implemented as separate software modules. The most complex module are Optimization module and Sensitivity analysis. The software solution of module Purchases and Sell and module Free Allocation have to be opened to external data sources on Internet because occurs frequent changes of data structures web pages as described in detail [4].

The software pilot project is currently in the testing phase, when is verified the software functionality and linkage to other production planning models that are used in the enterprise.

6 Conclusion

The aim of this paper was to analyze a power of influence of selected EU ETS factors on companies. Concretely, an influence of CER and EUA permit prices and amount of freely allocated permits was investigated. The sensitivity analysis was performed using the data of one steel companies in the Czech Republic. Very similar effect for the whole steel sector and for other sectors listed in the carbon leakage group within the EU can be assumed. All three analyzed factors had a higher impact in the second EU ETS phase than in the third one. Currently, the weakest influence on companies can be observed for CER permit price, especially due to its low mean value, low volatility and because of legislative restriction of CER's use for European companies. An amount of freely allocated permits is the most influencing factor whose advantage is the fact that its values are known in advance. Therefore, EUA permit price is considered to be the most important factor for companies which is influenced by a high uncertainty. According to the historical EUA prices in the third EU ETS phase, an increase in costs induced by emission trading up to 14% can come. On the other hand, these costs can be also decreased down to 54% under the effect of EUA prices and down to 2.8% under the effect of CER prices.

The presented research can be extended further for example by involving also non-carbon-leakage companies. Statistical dependencies between particular factors could be also included into analyses to make the results more realistic.

7 Annotation

This paper was created with financial support of SGS project SP2014/146 and the Operational Programme Education for Competitiveness – Project CZ.1.07/2.3.00/20.0296. The support is gratefully acknowledged.

8 Literature

- [1] Rong, A. & Landhelma, R.: CO₂ emissions trading planning in combined heat and power production via multi-period stochastic optimization. *European Journal of Operational Research*, 176, pp. 1874-1895, 2007.
- [2] Directive 2003/87/EC of the European Parliament and of the Council of 13th October 2003 establishing a scheme for greenhouse gas emission allowance trading within the Community and amending Council Directive 96/61/EC.
- [3] Directive 2009/29/EC of the European Parliament and of the Council of 23th April 2009 amending Directive 2003/87/EC so as to improve and extend the greenhouse gas emission allowance trading scheme of the Community.
- [4] Ministr, J., Racek, J. and Toth, D. (2012). Visualization of the Discussion Context from the Internet. *Proceedings of 20th Interdisciplinary Information Management Talks (IDIMT 2012)*. Linz: Trauner Verlag. pp. 297-303.
- [5] Řeháček, P. Risk management and FMEA. In proceedings: *Strategic Management and its Support by Information Systems*. Ostrava: VSB – Technical University of Ostrava, 2011, p. 154.
- [6] Tang, H. & Song, G.: Optimization of Enterprise Production Based on Carbon Emission Credits Trading. *Proceedings of the 2nd International Conference on Green Communications and Networks*. 2012.
- [7] Zapletal, F. & Moravcová, E.: Analysis of current conditions within the ETS and its impacts on firms in the Czech Republic. In: *Proceedings of the 10th International Conference on Strategic*

- Management and its Support by Information Systems. Ostrava: VŠB – Technical University of Ostrava, Faculty of Economics, pp. 235-255, 2013.
- [8] Zapletal, F. & Němec, R.: The Usage of Linear Programming to Construction of Ecologic-economical Model for Industrial Company Profit Optimization. In: Proceedings of 30th International Conference Mathematical Methods in Economics. Karviná, pp. 1010-1015, 2012.
- [9] Zapletal, F.: Mean-risk Model Optimizing the Heavy Industrial Company's Profit with Respect to Environmental Aspects. Proceedings of 32th International Conference Mathematical Methods in Economics. Olomouc, 2014.
- [10] Zhang, B. & Xu, L.: Multi-item production planning with carbon cap and trade mechanism. International Journal of Production Economics, 144(1), pp. 118-127, 2013.

The Architecture of Semantically Partitioned Complex Event Processing

Filip Nguyen, Tomáš Pitner

Masarykova univerzita, Fakulta informatiky
Botanická 68a, 602 00 Brno
xnguyen@fi.muni.cz
tomp@fi.muni.cz

Abstract

This paper describes the architecture of Semantically Partitioned Peer to Peer Complex Event Processing. Advantages and disadvantages of the architecture are estimated and followed by a list and description of chosen technologies used to implement the architecture itself. In the end, the paper describes the experiments and experimental data and metrics that will be used to verify the architecture's implementation.

Abstrakt

Průspěvek popisuje architekturu takzvaného Sémanticky děleného komplexního zpracování událostí. Jsou diskutovány výhody a nevýhody architektury. Dále vypíšeme technologie, které používáme při implementaci této architektury s jejich popisem. Konec článku diskutuje experiment, data a metriky plánované k vyhodnocení implementaci architektury.

Keywords

Distributed Complex Event Processing, Java

Klíčová slova

Distribuované komplexní zpracování událostí, Java

1 Introduction

2 Semantically Partitioned Complex Event Processing Architecture

Complex Event Processing (CEP) is a borderline technological and conceptual area of computer science. It is both a theoretical and technological tool to be used for processing large amounts of data that flow in a stream of so called Events, which are introduced more formally in the following definition.

Definition 1. An *Event* is a record of an activity in a system. The event has two aspects: the content (carries static data) and the time stamp. Formally, an event E is a tuple $E(t, p)$ where t is the time stamp and p is a the set of key-value properties (k, v) of an arbitrary type.

An example of an event stream could be a set of login attempts to a web server. Another example might be sell/buy actions on a stock market. These examples show that the concrete mapping of the event from Definition 1 into computing is rather free and one can choose to look at a system as a collection of very low level events (TCP packets) or high level events (stock manipulations).

First and foremost, CEP provides theoretical tools to analyze event streams. The goal of CEP is for the analysis to have the following properties:

1. It's temporally oriented - as we can see from Definition 1, the time stamp is an important part of an event. It allows for the analysis to compare times at which the event is produced along

with time correlations to other events (e.g. a high volume of certain stocks were sold in the time frame of 5 minutes).

2. It is real time and high performance oriented - no data should be stored for future analysis as the analysis is carried out using streaming data. Thanks to this aspect, it should be possible to use CEP in environments with large volumes of data.
3. It produces events - by analyzing the stream of events, CEP is allowed to generate deduced events called Complex Events. This is very often called *aggregation*. These events can be further analyzed as any other events.

CEP was introduced in [1]. Apart from theory, the book also defines specific implementation and the authors share their experiences stemming from the use of the implemented CEP system called Rapide. This is an interesting observation. The area of CEP is seldom purely theoretical, and research in the area is usually tied to a specific CEP engine implementation.

2.1 Distributed Event Processing

Much research attention has been allotted to distributed CEP (DCEP). The main goal of DCEP is to improve the second feature of CEP abstraction: the performance. Distributed CEP is a very vague notion, therefore we use the following definition for our purposes:

Definition 2. Distributed Complex Event Processing is the collection and processing of events on several computing nodes divided by a computer network. The addition of new processing nodes is carried out with the goal to increase the performance of the whole system.

A system that implements DCEP according to Definition 2 will scale horizontally. Such DCEP implementation is very similar to middleware solutions on Java platforms today (the area of Enterprise Systems).

Well known existing CEP systems such as Esper [4] or Drools Fusion [11] do not yet implement DCEP. It is implemented mostly in an ad hoc fashion without the support of any specific tool. One of the main goals of our research is to introduce comprehensive theoretical abstraction for DCEP, and also to implement an engine that allows for the use of abstraction.

Figure 1 DCEP Notation shows how we aim to model the DCEP system as a graph. A node in the graph is a so called *peer* and represents an event producer. Such a producer will typically be a server computer or sensor - the producer of events. Edges in the graph represent event flows which should be oriented. The dashed lines on the figure express the possibility of a new connection between P1, P3 and P3, P4. The connections in our model are created and destroyed very quickly in an automated fashion to improve performance and simultaneously not lose any expressive power (the loss of a Complex Event detection).

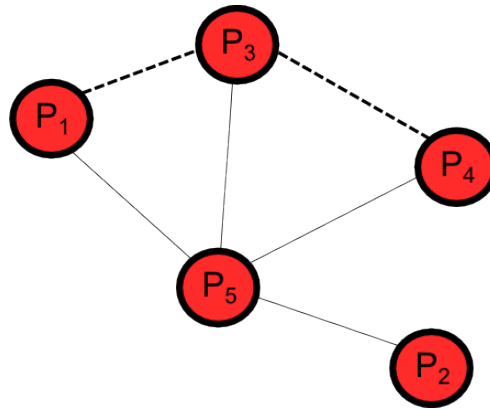


Figure 9 DCEP Notation

Following the short introduction to DCEP and our notation, we would like to briefly discuss state of the art research with the same aim.

The distribution expressiveness trade-off has been recently studied by Tariq [12,13,14,15] in the area of publish subscribe. The authors use spectral clustering to group subscribers and exploit the similarities of events. Their method also uses peer to peer networking which resonates with our approach.

The research of Balis et. al [16] focuses on distributed monitoring of events and the work consists of DCEP implementation in accordance with Definition 2. The difference from our method is that Balis answers queries without precision loss precision and the method seems to be limited to very specific CEP scenarios. More on CEP engine precision can be found in Section 2.2 of this paper.

From the self-organization point of view, the Event Subscription Recommender (ESR) introduced in [17] is a very interesting work. The recommender dynamically produces new event subscriptions. It is a software component that is part of the PLAY platform as part of the Seventh Framework Programme.

2.2 Result Scalable Queries

In this paper, we introduce DCEP architecture that answers CEP queries.

Definition 3. A CEP Query is a template that matches a set of events. Formally, a CEP Query $Q(P, W)$ is a tuple where P is a pattern and W is a temporal length (a time frame, e.g. 10 min). The pattern P is a logical formula in the propositional calculus. Let $E0(a, b)$ and $E1(c, d)$ be events. A variable in the formula takes on the form of a comparison of two elements from a set: $\{a, c\} \cup b \cup d \cup \text{CONSTANTS}$, where CONSTANTS is a predefined set of constants.

The ability to execute CEP queries and thus uncover interesting patterns of events is part of every CEP engine implementation. The engines allow a user to define CEP queries and to process the results. We observe that CEP queries can be further categorized into two disjunctive categories according to the expectations of the user. The categories are: result binary, result scalable.

Definition 4. A CEP query is result binary if and only if the receiver of the query results requires the results to be deterministic with regards to the input events.

Queries are result scalable if and only if they are not result binary. From our research, we have concluded that there is large amount of problems that need to employ result binary queries - those that

need exact deterministic results every time. An example of such might be the computation of financial information based on high level business events.

However, by studying research done nowadays, it can be observed that many problems are good with result scalable queries. These queries might not uncover all Complex Events, but thanks to the allowance of this imprecision, we are able to boost CEP engine performance.

Our architecture is called Semantically Partitioned Peer to Peer Complex Event Processing (PCEP). The main implementation piece is the so called *peer* which is a process deployed to a computer that supports a Java execution environment. The peer is composed of the following components:

- CEP Engine - a peer bundles an existing CEP engine that has no DCEP capabilities. Examples of such engines are Esper for Java or Drools Fusion.
- Communication Module - this module is used to ensure communication between different peers. The module needs to handle decentralized peer-to-peer communication.
- Monitoring Module - this module is used to collect statistics about the CEP engine and the context in which the peer is deployed (e.g. CPU consumption on the node).
- High Speed Gateway - a component used by the communication module to disseminate Events.
- Peer Logical Module - a module that executes distributed algorithms to answer CEP queries.

These five components are bundled together to form a *peer*. Typical PCEP deployment consists of deploying similar peers across the network, one peer for one producer. The peers connect to each other, communicate, and answer CEP queries submitted by a user. There are two modes in which a peer may be deployed. This is shown on Figure 2 Modes of Deployment. The figure shows that in one instance the peer on the right is deployed directly on the same PC as the MSSQL database (the producer of events). The second mode of deployment is shown on the left. The peer is deployed on a computer that is in near proximity (latency wise) to the producer (Web Server) but is not on the same PC.

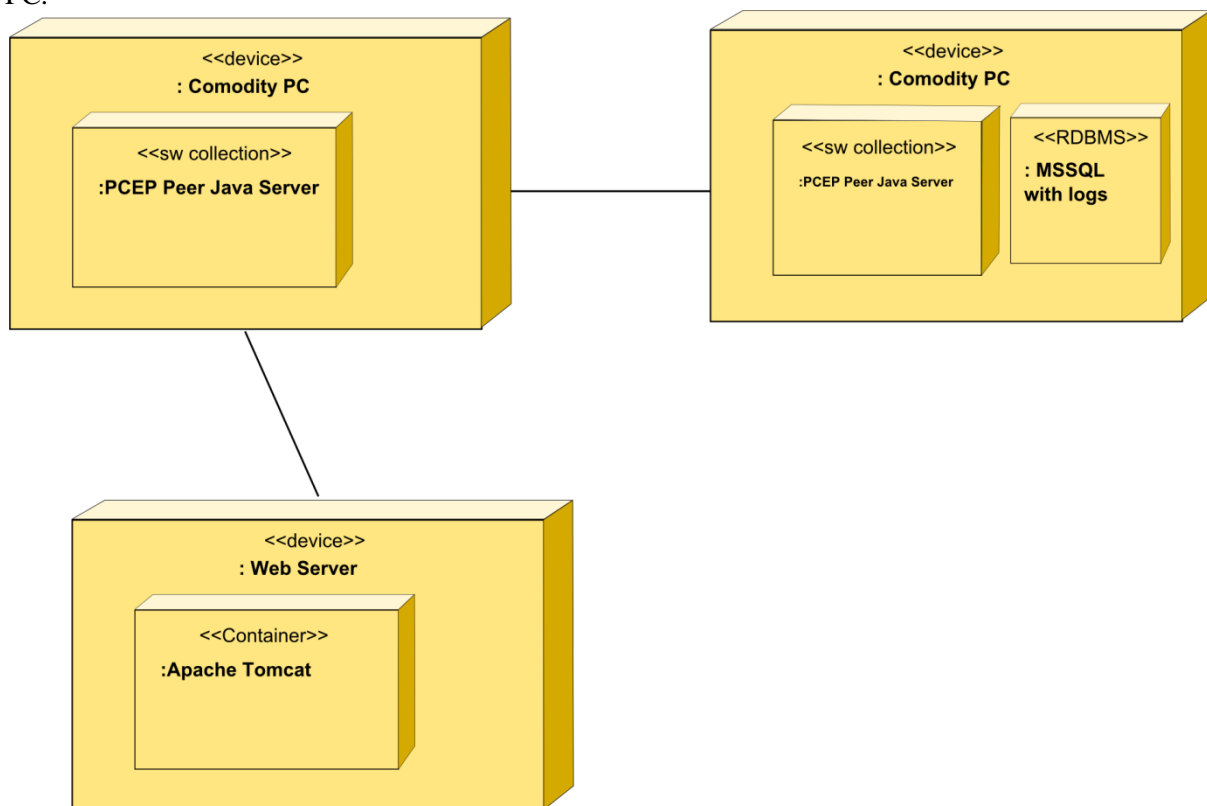


Figure 10 Modes of Deployment

2.3 Technologies

This section introduces technologies that are being used to implement the PCEP. The following table lists the components and along with the technologies that are used to implement them. The rest of the section will be dedicated to discussing the most important ones.

Component	Technology	Notes
CEP Engine	Esper for Java	
Communication Module	Apache Zookeeper	
Monitoring Module	JMX of Esper engine, Java	Technically pure Java EE
High Speed Gateway	Netty	As low level as possible
Peer Logical Module	Java language	Written in pure Java language
Visualization	GWT framework	
Generator	Java	Again pure Java. The implemenetation is distributed
Virtualization	Docker	

Esper for Java is a state of the art CEP engine. It uses query language to capture events in the stream of events. The engine has one important property that makes it a good technological choice for PCEP: the possibility to embed the engine in a Java process. This feature is also available with the Drools Fusion project.

For the communication module, we have chosen to use Apache Zookeeper. This framework is used to perform distributed algorithms which allows sharing of a tree-like datastructure between all computers that are part of the peer-to-peer network. On top of this functionality, it has several features that simplify distributed programming. Using this simple concept of distributed data structure, it is possible to implement even the most complicated distributed algorithms.

An important technology to be mentioned is the Docker, a virtualization tool. This tool is a para-virtualization solution that allows to set up a virtualized environment with low overhead compared to classical full virtualization solutions.

3 Experimental Data Sets and Metrics

The most important dataset that will be used to evaluate the performance of the algorithm is that from the assignment for the competition DEBS Grand Challenge 2014 [3]. The data set contains data collected from sensors: smart plugs. A smart plug is a device in between a power outlet on a wall and an appliance at home. The plug collects data about the consumption of electrical energy and periodically reports the information in the form of a temporal event. The structure of the dataset is a set of events of which each event is described using the following fields:

- id
- timestamp
- value
- property
- plug_id
- household_id
- house_id

The record encodes two types of events: *load* and *work*. The *load* is absolute energy consumption while *work* is cumulative. Events are reported every second.

The over 50GB of data contains events collected from 40 houses over a span of 30 days.

Thanks to the fact that the data set has already been used within the framework of an assignment for a state of the art academic challenge, we will be able to compare our results with the results of other research groups in the field.

The metrics to be used to evaluate PCEP will be the following:

- throughput compared to existing solutions
- per-query precision compared to existing solutions

4 References

- [1] David Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Professional, New York, 2002.
- [2] Filip Nguyen, Daniel Tovarňák, and Tomáš Pitner. Semantically partitioned peer to peer complex event processing. In Filip Zavoral, Jason J. Jung, and Costin Badica, editors, *Intelligent Distributed Computing VII*, volume 511 of *Studies in Computational Intelligence*, pages 55–65. Springer International Publishing, 2014.
- [3] DEBS Grand Challenge 2014 Assignment
27.1.2014 <http://www.cse.iitb.ac.in/debs2014/?page_id=42>
- [4] Esper Engine
29.10.2014 <<http://esper.codehaus.org/>>.
- [5] Internet Trace Archive
29.10.2014 <<http://ita.ee.lbl.gov/html/traces.html>>.
- [6] Eugene Wu, Yanlei Diao, and Shariq Rizvi. High-performance complex event processing over streams. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, SIGMOD '06, pages 407–418, New York, NY, USA, 2006. ACM.
- [7] Vojtech Krmicek and Jan Vykopal. Netflow based network protection. In Muttukrishnan Rajarajan, Fred Piper, Haining Wang, and George Kesidis, editors, *Security and Privacy in Communication Networks*, volume 96 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 543–546. Springer Berlin Heidelberg, 2012.
- [8] Pavel Minarik, Jan Vykopal, and Vojtech Krmicek. Improving host profiling with bidirectional flows. In *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 03*, CSE '09, pages 231–237, Washington, DC, USA, 2009. IEEE Computer Society
- [9] Daniel Tovarňák, Filip Nguyen, and Tomáš Pitner. Distributed event-driven model for intelligent monitoring of cloud datacenters. In Filip Zavoral, Jason J. Jung, and Costin Badica, editors, *Intelligent Distributed Computing VII*, volume 511 of *Studies in Computational Intelligence*, pages 87–92. Springer International Publishing, 2014.
- [10] Filip Nguyen and Tomáš Pitner. Scaling CEP to infinity. In *9th Summer School of Applied Informatics*, Brno, Czech Republic, 2012. Masaryk University.

- [11] Drools Fusion 29.10.2014 < <http://www.drools.org>>.
- [12] Muhammad Adnan Tariq, Boris Koldehofe, and Kurt Rothermel. Efficient content-based routing with network topology inference. In *Proceedings of the 7th ACM International Conference on Distributed Event-based Systems*, DEBS '13, pages 51–62, New York, NY, USA, 2013. ACM.
- [13] Muhammad Adnan Tariq, Boris Koldehofe, Gerald G. Koch, and Kurt Rothermel. Distributed spectral cluster management: A method for building dynamic publish/subscribe systems. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, DEBS '12, pages 213–224, New York, NY, USA, 2012. ACM.
- [14] Muhammad Adnan Tariq, Gerald G. Koch, Boris Koldehofe, Imran Khan, and Kurt Rothermel. Dynamic publish/subscribe to meet subscriber-defined delay and bandwidth constraints. In *Proceedings of the 16th International Euro-Par Conference on Parallel Processing: Part I*, EuroPar'10, pages 458–470, Berlin, Heidelberg, 2010. Springer-Verlag.
- [15] Muhammad Adnan Tariq, Boris Koldehofe, Gerald G. Koch, Imran Khan, and Kurt Rothermel. Meeting subscriber-defined qos constraints in publish/-subscribe systems. *Concurrency and Computation: Practice and Experience*, 23(17):2140–2153, 2011.
- [16] Bartosz Balis, Grzegorz Dyk, Marian Bubak . On-line grid monitoring based on distributed query processing. In *Proceedings of the 9th International Conference on Parallel Processing and Applied Mathematics - Volume Part II*, PPAM'11, pages 131–140, Berlin, Heidelberg, 2012. Springer-Verlag.
- [17] Yiannis Verginadis, Nikos Papageorgiou, Ioannis Patiniotakis, Dimitris Apostolou, and Gregoris Mentzas. A goal driven dynamic event subscription approach. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, DEBS '12, pages 81–84, New York, NY, USA, 2012. ACM.

Matematické základy optimalizace svozové trasy komunálního odpadu

Michal Petřík, Stanislav Bartoň

Mendlova univerzita v Brně
Zemědělská 1, 61300 Brno
michal.petrik@mendelu.cz

Abstrakt

Tento článek je zaměřen na přípravu matematických podkladů pro optimalizaci nakládání s odpady v rámci určité obce. Jako vstupní data jsou použity GPS souřadnice odpadních kontejnerů a map poskytovaných webovou stránkou www.mapy.cz/. Křižovatky jsou označeny a jejich GPS souřadnice jsou uloženy v souboru. Na začátku jsou cesty rozděleny do dvou skupin podle přítomnosti odpadních kontejnerů na místech a bez nich.

Tyto vstupy definují problém teorie grafů o nalezení nejkratší trajektorie spojující všechny body s odpadem, která je rovna nejkratšímu Eulerovskému tahu v grafu. Program Maple byl použit pro všechny výpočty a grafické vizualizace i k řešení minimálního párování na základě teorie množin. Ke snížení exponenciální složitosti problému byly použity tři podmínky a dodatečné okraje jsou zavedeny pouze v případě, že spojují sousední body. Nesousedící body jsou spojeny pouze tehdy, leží-li mezi nimi bod s lichou hodnotou. Body s lichou přilehlostí jsou ve vzestupném pořadí s ohledem na přilehlost. Minimální párování začíná s bodem nejnižší přilehlosti. Tento předpoklad umožňuje optimalizovat trajektorii sběrného vozidla komunálního odpadu v mnohem větších vesnicích, jak je uvedeno v následujícím článku.

Abstract

This article is focused on the preparation, with the mathematical bases, used for the optimisation of the waste disposal within the certain village. As an input data are used GPS coordinates of the waste containers and maps provided by the www.mapy.cz web page. Crossroads are indexed and their GPS coordinates are saved in a file. The other file contains connections of the crossroads as a list of ordered couples crossroads indexes. At the beginning, the roads are divided into two sets according to the waste containers presence places with and without them.

These inputs define problem of the Graph theory to find the shortest trajectory connecting all waste sources points, which is equal to find the shortest Eulerian line in the given graph. Program Maple is used for all computations and graph visualizations. Solution to the minimal matching problem in the Maple based on the set theory was used to solve this problem. To reduce exponential complexity of the problem free premises are used. Additional edges are introduced only if they connect adjacent points. Non-adjacent points are connected if only one point with even adjacency lies between them. Points with the odd adjacency are in ascending order with respect to adjacency. Minimal matching begins with the point of the lowest adjacency. This assumption enables to optimise trajectory of the municipal waste pick up trailer in much greater villages as it is shown in the following article.

Klíčová slova

Logistika, svoz odpadu, obchodní cestující, optimalizace, minimální párování, teorie grafů, matice sousednosti, Maple13

Key-Words

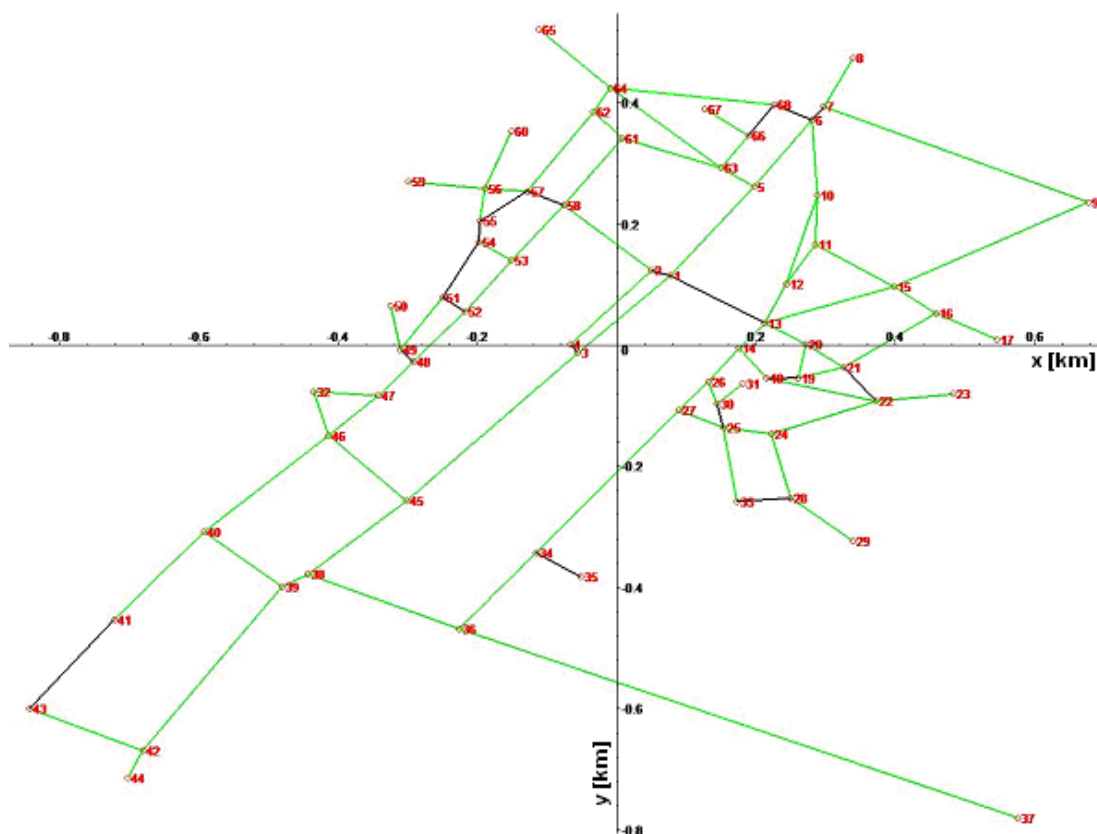
Logistics, waste disposal, travelling salesman, optimisation, minimum matching, graph theory, adjacency matrix, Maple13

1 Materiály a metodika

Jako zdroj základního vstupního materiálu posloužily mapy ze serveru www.mapy.cz/, odkud byla stažena mapa nejmenované obce. Dalším vstupním materiálem byly GPS lokace potřebných bodů a lokace nádob pro sběr odpadů. Zeměpisné souřadnice WGS-84 byly převedeny do soustavy pravoúhlých kartézských souřadnic. Následné spojení bodů tvoří hodnocený neorientovaný graf, jehož nejvyšší generalizace má vliv na výpočetní složitost. Graf je zjednodušen. Odstraněny z něj jsou cesty se slepými konci a cesty bez zdrojů odpadů (pokud se připojí uzly s lichým přilehlost). Na závěr je zjednodušený graf upraven minimálním párováním tak, že je možné implementovat eulerovský tah. Pro tuto případovou studii byla použita metodika teorie grafů - eulerovský tah, Hamilton kruhu, a minimální párování, viz [2]. Potřebné výpočty byly provedeny pomocí programu Maple, viz [4].

2 Příprava vstupních informací

Na začátek bylo potřeba převést mapu dané obce do podoby grafu. K tomu bylo potřeba využít GPS lokací křižovek ulic v obci, které budou v grafu tvořit jednotlivé uzly. GPS lokace byly převedeny pomocí referenčního elipsoidu WGS 84 (World Geodetic System 84) do kartézské soustavy souřadnic, čímž byly získány uzly grafu. Pro vytvoření hran grafu bylo zapotřebí sestavit matici sousednosti, která obsahuje vzdálenosti mezi jednotlivými křižovatkami – uzly grafu, které jsou spojeny komunikací. Mezi křižovatkami, které spojeny komunikací nejsou, je vzdálenost „-1“ pro jednodušší práci. V tomto konkrétním případě je matice sousednosti symetrická dle hlavní diagonály. Rozdíl by nastal u větších měst, ve kterých jsou komunikace s jednosměrným provozem.



Obrázek č.1: Základní graf obce

Dalším rozdílem mohou být komunikace o šířce, která neumožní sběr odpadu jedním průjezdem sběrného vozu, ale je třeba sbírat odpad na každé straně komunikace zvlášť, tedy dvě cesty přes jednu hranu. Těmito kroky byl vytvořen neorientovaný ohodnocený graf, který slouží jako základ pro další řešení. V základním grafu, který je odvozen od mapového podkladu, bylo potřeba odlišit od sebe komunikace, na kterých jsou umístěny sběrné nádoby pro odpad a na kterých ne (obr. č. 1). Je zřejmé, že komunikace se sběrnými nádobami musí sběrné vozidlo projet. Komunikace bez sběrných nádob není třeba projíždět, ale v grafu figurují jako potenciální pomocné cesty při minimálním párování.

Základní graf obsahuje 68 bodů, což je z hlediska výpočetní složitosti, která s počtem bodů exponenciálně roste, nelze provést „hrubou silou“ – porovnáním všech možností (jednalo by se totiž o 68!). Proto je třeba hledat vhodné zjednodušení stávajícího grafu s ohledem na funkci optimalizace. V prvním kroku lze z grafu odebrat uzly, do kterých vede pouze jedna hrana, [5]. Tedy body, do nichž a z nichž se lze dostat pouze po jedné komunikaci. Tím to krokem se graf zjednoduší na 52 bodů. Při pohledu na graf lze vidět, že prvním odebráním hran, v grafu vznikly nové uzly, které splňují podmínku, že do nich vede pouze jedna komunikace. Lze tedy první krok aplikovat opakovaně, dokud v grafu nezbudou uzly, do kterých vede dvě a více hran. Nyní má graf 51 uzlů.

Pro další zjednodušení bylo třeba prohledat množinu hran, na kterých nejsou sběrné nádoby. A zjistit zda jsou nápomocny optimalizaci řešení či nikoliv a můžeme je tedy vypustit. Podmínkou tohoto rozhodnutí je mocnost – počet cest do jednotlivých uzlů, které tyto hrany bez sběrných nádob spojují. Cílem je získat uzly o sudé mocnosti. Pokud odebráním hrany bez odpadu získáme dva uzly o sudé mocnosti, je tato hrana odebrána a dále s ní v grafu není pracováno. Naopak hrany, díky kterým jsou jejich koncové uzly sudé, v grafu zůstanou. Nyní již není brán rozdíl mezi hranou se sběrnými nádobami a bez nich. Úloha se zjednodušila na 49 uzlů a ubýlo 6 hran.

V dalším kroku jsou vyhledány všechny uzly, do kterých vedou právě dvě. V praxi jsou to křižovatky, přes které se projíždí a díky zjednodušení z nich již nelze odbočit jinak. Tyto uzly můžeme vynechat a hrany, které do těchto bodů vedly spojit v jednu hranu. Hrany v grafu si nejsou rovny měřítkem. Jedná se pouze o grafické znázornění. Jejich ohodnocení zůstává nezávisle na jejich délkách. Nyní v grafu zůstává jen 24 uzlů.

Nyní je graf složen z uzlů, jejichž mocnost je tři a více. Jsou zde i uzly s lichým stupněm, což znemožňuje provést Eulerovský tah. K jeho provedení je třeba, aby měly uzly sudý stupeň mocnosti, [3]. V současné podobě zjednodušeného grafu je 18 bodů, které mají lichý stupeň mocnosti.

3 Minimální párování

Nyní je třeba užít minimálního párování, tedy spojit liché body pomocnou hranou, která bude kopírovat stávající hrany (komunikace), a součet těchto hran bude nejmenší možný, [1]. Všechny uzly s lichou mocností spolu nesousedí, tedy budeme muset při minimálním párování uvažovat i s uzly ležícími ob uzel. Nikdy však pomocná hrana spojující dva uzly s lichou mocností nesmí protínat třetí uzel s lichou mocností, to by bylo neefektivní.

Možný počet kombinací všech párů z 18ti uzlů je 34 459 425. Řešení tohoto problému hrubou silou je z hlediska výpočtové složitosti neefektivní. Proto je nutné uvažovat tři zjednodušující podmínky.

- 1 Jako možné vložené hrany budou uvažovány pouze hrany, spojující přímo sousedící uzly.
- 2 Všechny uzly s lichou mocností spolu nesousedí, je tedy třeba uvažovat i o spojení s body ob uzel. Podmínkou spojení ob uzel je, že na spojnici dvou bodů s lichou mocností smí být pouze jeden další uzel a to pouze uzel se sudou mocností.
- 3 Vnitřní bod musí být lichý.

4 Výpočetní tabulka Maple

Soubor DZL obsahuje matici sousednosti DD a skupinu cest LL. Počet lichých bodů je $n_3=18$. Nejprve musíme nalézt všechny cesty spojující sousedící body a uložit je do skupiny K1.

```
> restart; with(LinearAlgebra): with(plots):
> read "DZL.sav": LL:=map(u->{u[]},{LL[]}):
> n3:=18: K1:={}: K2:={}:
> for i from 1 to n3 do; for j from i+1 to n3 do;
if has(LL,[{i,j}]) then K1:={K1[],{i,j}}; end if: end do: end do:
> for i from 1 to n3 do; for j from i+1 to n3 do;
mp:=subs(i=NULL,map(u->u[],select(has,LL,i))) intersect
subs(j=NULL,map(u->u[],select(has,LL,j)));
if mp<>{} then K2:={K2[],[i,mp[],j]} end if: end do: end do:
```

Nyní je důležité nalézt hrany, které splňují podmínky 2 a 3 a uložit je do souboru K2.

```
> k2:=map(u->'if'(nops(u)=3,u,NULL),K2): k2:=map(u->'if'(u[2]>n3,u,NULL),k2):
> K2:=map(u->'if'(nops(u)>3,u,NULL),K2):
```

```
> K2:=map(u->map(v->[u[1],v,u[-1]],u[2..-2])[],K2):
> K2:=map(u->'if'(u[2]>n3,u,NULL),K2): K2:={k2[],K2[]}:
```

Matice sousednosti D3, vytvořená pouze pro uzly s lichou mocností, respektující předchozí podmínky, vypovídá o vzdálenostech mezi jednotlivými a uzly a hlavně udává počet možností spojení jednotlivých uzlů.

```
> D3:=Matrix(n3,n3,fill=infinity,shape=symmetric):
> for k in K1 do; D3[k[]]:=DD[k[]]; end do;
> for k in K2 do; d3:=DD[k[1],k[2]]+DD[k[2],k[3]];
if D3[k[1],k[-1]]>d3 then D3[k[1],k[-1]]:=d3; end if; end do;
```

Každý z těchto uzlů má dle matice sousednosti daný počet možností spojení s jiným lichým uzlem. Tyto možnosti představují neuspořádané dvojice {počáteční bod, koncový bod}, tedy ke každému uzlu jsou přiřazeny neuspořádané dvojice dle možností jeho spojení. Seřazením vektoru 18 uzlů od uzlu s nejnižším počtem spojů po uzel s nejvyšším počtem spojů, snížíme výpočetní složitost, viz obrázek č. 3. Všechna možná spojení budou uložena v proměnné K.

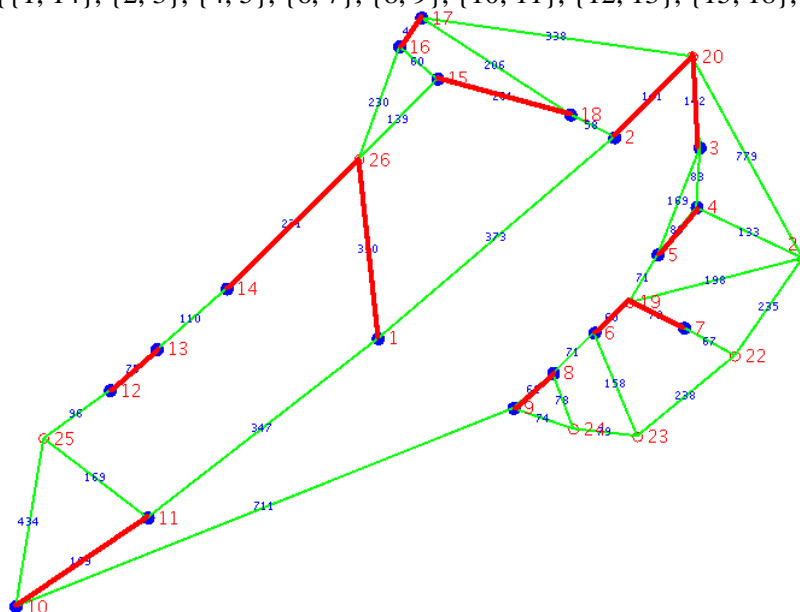
```
> KN:=map(u->{lhs(u)},op(2,subs(infinity=0,D3))): ND:=[1..n3]:
> MN:=Matrix(map(u->[u,nops(select(has,KN,u))],ND)): Go:=true:
> while Go do; Go:=false; for i from 2 to n3 do;
if MN[i,2]<MN[i-1,2] then MN:=RowOperation(MN,[i,i-1]);
Go:=true; end if; end do; end do;
> ND:=convert(Column(MN,1),list): NN:=convert(Column(MN,2),list):
> K:=map(u->[select(has,KN,u)[]],ND):
K := [[{3, 4}], [{4, 5}], [{5, 7}], {6, 7}], [{6, 8}], {8, 9}], [{8, 9}], {9, 10}], [{12, 13}], {13, 14}], . . . ]
```

Nyní z tohoto vektoru vezmeme první uzel „4“, ke kterému náleží dvě možnosti spojení – {3,4} a {4,5}. Vznikly dvě varianty prvního výběru – dvě větve řešení. Pro každou větev nyní postupujeme samostatně. Nejprve ze všech zbývajících uspořádaných dvojic vyřadíme ty, které obsahují použité body (zvláště pro každou větev, tedy pro první větev „3“ a „4“, pro druhou „4“ a „5“). Nyní se přidá ke každé větvi další neuspořádané dvojice pro další uzel v pořadí a celý proces se opakuje. Pokud druhý uzel má také dvě možnosti spojení dostaneme tak čtyři větve. Možností – větví přibývá, ale s odebráním již použitých uzlů se stává spousta větví slepými – bez možnosti dokončení a dochází k redukci větví. Celý postup se opakuje až do vyčerpání nepoužitých uzlů. Všechny možné kombinace párů jsou uloženy v proměnné **pe**.

```
> PE:=[]: pe:=map(u->{u},K[1]); nu:=1:
> while nu<n3/2 do;
nn:=map(u->{map(v->v[],u)[]},pe);
su:=map(u->map(v->v=NULL,u),nn); KS:=map(u->subs(u,{ }=NULL,[],=NULL,K),su);
ks:=map(u->u[],map(u->u[],KS)); ks:={map(u->'if'(nops(u)=1,u,NULL),ks)[]};
su:=map(u->u=NULL,ks); KS:=subs(su,[],=NULL,KS);
idel:=[seq('if'(KS[i]=[],i,NULL),i=1..nops(KS))]; su:=map(u->u=NULL,idel);
KS:=subsop(su[],KS); pe:=subsop(su[],pe); ks:=map(u->u[1],KS);
pe:=zip((u,v)->map(w->{u[],w},v)[],pe,ks); nu:=nu+1; PE:=[PE[],nops(pe)];
end do;
```

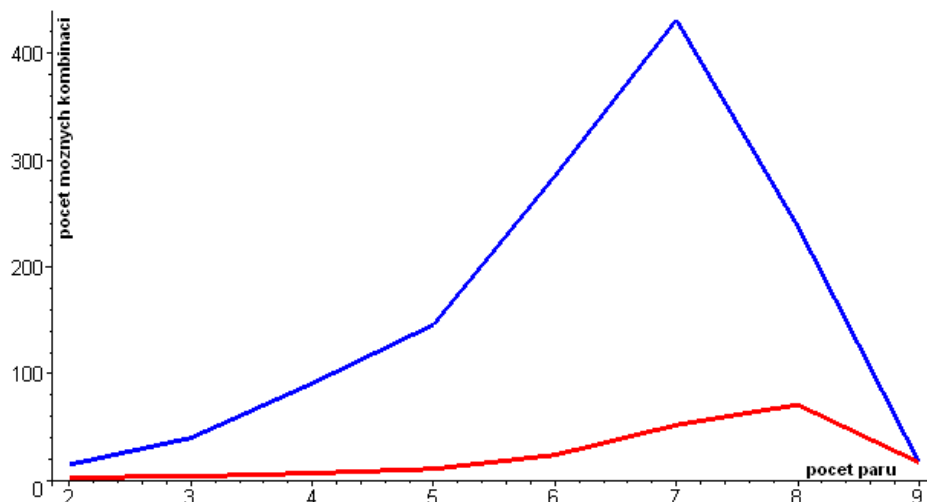
V řešeném případě je výsledkem 17 možností minimálního párování, což pro „hrubou sílu“ – porovnání všech možností pomocí matice sousednosti, nepředstavuje nijak náročný úkol. Nalezení minimálního párování je nyní snadné.

```
> SE:=map(u->add(w,w=map(v->D3[v[]],u)),pe); Smin:=min(SE):
> MMatching:=zip((u,v)->'if'(u=Smin,v=NULL),SE,pe)[];
MMatching := {{1, 14}, {2, 3}, {4, 5}, {6, 7}, {8, 9}, {10, 11}, {12, 13}, {15, 18}, {16, 17}}
```



Obrázek č. 2: Zjednodušený graf a minimální párování - tučně

V průběhu výpočtu bylo prošetřováno maximálně 75 nezávislých větví výpočtu, na rozdíl od obráceného postupu, kdy se vektor uzlů seřadil od uzlu s nejvyšším počtem spojů. V tomto případě bylo sledováno maximálně 475 nezávislých větví výpočtu, viz obrázek 2, řazeno od minima – dolní linie, řazeno od maxima – horní linie. Výsledek – minimální párování je zobrazeno na obrázku č. 3.



Obrázek č. 3: Výpočetní složitost

5 Shrnutí

Při optimalizaci svozové trasy je nutné nejprve převést topografické podklady do digitální podoby. V tomto článku bylo využito GPS souřadnic a systému WGS – 84 pomocí, kterých byl mapový podklad převeden do podoby neorientovaného ohodnoceného grafu. Ve druhém kroku proběhlo snížení složitosti grafu. Třetím krokem bylo vytvoření Eulerovského tahu, k němuž se dospělo pomocí minimálního párování. Použití zjednodušujících předpokladů vedlo k velmi výraznému snížení výpočetní složitosti problému.

6 Poděkování

Výzkum byl podpořen v rámci projektu TP 4/2014 Analýza degradace procesů moderních materiálů používaných v zemědělské technice financován IGA AF MENDELU.

7 Literatura

- [1] Berge, C.: The Theory of graphs. Courier Dover Publications, 2001, ISBN 978-0-486419-75-6
- [2] Cook, W., J.: Po stopách obchodního cestujícího. Dokořán a Argo, 2012, ISBN 978-8-073634-12-4
- [3] Demel, J.: Grafy a jejich aplikace. Academia, 2002, ISBN 978-8-020009-90-6
- [4] Maple User Manual Maplesoft, 2011, Waterloo Canada, ISBN 978-1-926902-07-4
- [5] Matoušek, J., Nešetřil, J.: Kapitoly z diskrétní matematiky. ISBN 978-8-024617-40-4

**11th Summer School of Applied Informatics
Proceedings**

**11. letní škola aplikované informatiky
Sborník příspěvků**

Bedřichov, 12.–14. září 2014

Editoři/Editors:

prof. RNDr. Jiří Hřebíček, CSc.

ing. Jan Ministr, Ph.D.

doc. RNDr. Tomáš Pítner, Ph.D.

Vydal: Karel Kovařík - Littera

1. vydání

Tisk: Tiskárna Didot, Trnkova 119, 628 00 Brno

ISBN 978-80-85763-87-4