

VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE
FAKULTA INFORMATIKY A STATISTIKY



DIPLOMOVÁ PRÁCE

VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE

FAKULTA INFORMATIKY A STATISTIKY



Analýza inflace v malé otevřené ekonomice

Autor: **Bc. Jiří Georgiev**

Katedra: **Katedra ekonometrie**

Obor: **Ekonometrie a operační výzkum**

Vedoucí práce: **Ing. Tomáš Formánek, Ph.D.**

Prohlášení:

Prohlašuji, že jsem diplomovou práci *Analýza inflace v malé otevřené ekonomice* zpracoval samostatně. Veškerou literaturu a další podkladové materiály uvádím v seznamu použité literatury.

V Praze dne

.....

Jiří Georgiev

Poděkování:

Rád bych na tomto místě poděkoval Ing. Tomáši Formánkovi, Phd za vedení diplomové práce, za cenné rady, věcné připomínky a vstřícnost při konzultacích a za podnětné návrhy, které ji obohatily.

Abstrakt

Název práce: Analýza inflace v malé otevřené ekonomice

Autor: Bc. Jiří Georgiev

Katedra: Katedra ekonometrie

Vedoucí práce: Ing. Tomáš Formánek, Ph.D.

Inflační predikce pomáhají ekonomickým subjektům při jejich rozhodování a tvorbě budoucích očekávání. Pro zlepšení inflačních predikcí by mohlo být potenciálně vhodné postihovat a reprodukovat nejen lineární, ale i nelineární vztahy mezi proměnnými. V této práci byly zkoumány predikční schopnosti vícerozměrných modelů na inflaci v malých otevřených ekonomikách, konkrétně na datech za Českou republiku, Slovensko, Maďarsko, Polsko a Rakousko. Porovnány byly výsledky lineárních VAR modelů oproti nelineárním TVAR modelům a feedforward neuronovým sítím. Neuronové sítě se v rámci této úlohy prokázaly jako nástroj rovnocenný VAR modelům, a to především díky regularizaci, zamezující přeučení modelu. Samotné zvýšení variability a povolení nelineárnosti nepřineslo výrazné zlepšení.

Klíčová slova: Inflace, VAR, TVAR, Neuronové sítě

Abstract

Title: Analysis of inflation in small open economies

Autor: Bc. Jiří Georgiev

Department: Department of Econometrics

Supervisor: Ing. Tomáš Formánek, Ph.D.

Inflation forecasts aid economical subjects in their decision making and creating of future expectations. To improve the forecast, it might be potentially appropriate to accommodate and reproduce not only linear but also nonlinear relationship among variables. This thesis explores forecast capabilities of multivariate models for inflation in small open economies, specifically of data of Czech Republic, Slovakia, Hungary, Poland and Austria. The results of linear VAR models are compared to nonlinear TVAR models and feedforward neural nets. Neural nets proved to be equal to VAR models in this task, mainly due to regularization, which prevents overfitting. The increase in variability itself and permitting non-linearity did not bring significant improvements.

Keywords: Inflation, VAR, TVAR, Neural nets

Obsah

Úvod	1
1 Časové řady	2
1.1 Stacionarita	2
1.1.1 Slabá závislost časových řad	3
1.1.2 Proces bílého šumu	4
1.1.3 Proces náhodné procházky	4
1.1.4 Testy jednotkového kořene	5
1.1.5 Kointegrované časové řady	8
1.2 Vektorové autoregresní modely	9
1.2.1 Diagnostické testy	12
1.3 Prahové vektorové autoregresní modely	14
1.3.1 Testování nelineárnosti	14
1.4 Neuronové sítě	16
1.4.1 Struktura feedforward neuronové sítě	17
1.4.2 Aktivační funkce	18
1.4.3 Normalizace dat	20
1.4.4 Učení neuronové sítě	21
1.4.5 TensorFlow a rozhraní Keras	22
1.5 Měření přesnosti předpovědí	23
1.5.1 Křížová validace časových řad	24
2 Inlace	26
2.0.1 Cílování inflace národními bankami	26
3 Inlace v malých otevřených ekonomikách	27
3.1 Data	27
3.1.1 Stacionarizace	28
3.2 Benchmark model	30
3.3 Odhad VAR modelu	32
3.3.1 Odhad Česká republika	33
3.3.2 Odhad Slovensko	34
3.3.3 Odhad Maďarsko	35
3.3.4 Odhad Polsko	36
3.3.5 Odhad Rakousko	37
3.4 Odhad Neuronové sítě	39
3.4.1 Odhad Česká republika	41

3.4.2	Odhad Slovensko	42
3.4.3	Odhad Maďarsko	43
3.4.4	Odhad Polsko	44
3.4.5	Odhad Rakousko	45
3.5	Odhad TVAR modelu	47
3.5.1	Odhad za Českou republiku	47
3.5.2	Odhad Slovensko	48
3.5.3	Odhad Maďarsko	48
3.5.4	Odhad Polsko	49
3.5.5	Odhad Rakousko	49
Závěr		51

A Grafy

B Skripty

B.1	Použité balíčky	
B.2	Benchmark	
B.3	Odhad neuronové sítě pomocí Keras(Tensorflow)	
B.4	Odhad TVAR modelů	
B.5	Odhad VAR modelů	
B.6	Prohledávání datasetu OECD	
B.7	Stahování požadovaných datasetů	
B.8	Příprava dat	
B.9	Podpůrné funkce	

Seznam obrázků

1.1	Příklad časových řad integrovaných řádu nula	3
1.2	Simulace náhodné procházky	5
1.3	Příklad kointegrace na simulovaných časových řadách	8
1.4	Vícevrstvá neuronová síť s l vrstvami (převzato: Smagt 1996 [25])	16
1.5	Lineární funkce	18
1.6	ReLU - rectified linear unit	18
1.7	Sigmoid funkce	19
1.8	Softmax funkce	19
1.9	Rozdělení časové řady na testovací a trénovací sadu	23
1.10	Křížová validace časových řad s různou délkou horizontu	24
1.11	Křížová validace časových řad se stejnou horizontu	25
3.1	Porovnání chyby pro modely se zpožděním dle různých informačních kritérií (omezeno pro $MSE_3 < 2$, $MSE < 2$)	32
3.2	Odhad VAR(4) cpi_q, gdp_q s predikcí na 10 období dopředu . . .	34
3.3	Odhad VAR(3) $cpi_q, int3_q, ulc_q, unemp_q$ s predikcí na 10 období dopředu	34
3.4	Odhad VAR(4) cpi_q, act_q s predikcí na 10 období dopředu . . .	35
3.5	Odhad VAR(7) cpi_q, imp_q s predikcí na 10 období dopředu . . .	35
3.6	Odhad VAR(7) $cpi_q, imp_q, unemp_q$ s predikcí na 10 období dopředu	36
3.7	Odhad VAR(1) $cpi_q, act_q, ulc_q, unemp_q$ s predikcí na 10 období dopředu	36
3.8	Odhad VAR(5) $cpi_q, ulc_q, unemp_q$ s predikcí na 10 období dopředu	37
3.9	Odhad VAR(2) $cpi_q, act_q, int3_q$ s predikcí na 10 období dopředu	37
3.10	Odhad VAR(3) $cpi_q, act_q, gdp_q, ulc_q$ s predikcí na 10 období dopředu	38
3.11	Odhad VAR(7) $cpi_q, act_q, imp_q, int3_q$ s predikcí na 10 období dopředu	38
3.12	Porovnání aktivačních funkcí a normalizačních přístupů	39
3.13	Porovnání optimalizačních metod ADAM a RMSprop	40
3.14	Odhad Neuronové sítě $cpi_q, imp_q, unemp_q$ s predikcí na 10 období dopředu	42
3.15	Odhad Neuronové sítě $cpi_q, imp_q, unemp_q$ s predikcí na 10 období dopředu	42
3.16	Odhad Neuronové sítě $cpi_q, imp_q, unemp_q$ s predikcí na 10 období dopředu	43

3.17	Odhad Neuronové sítě cpi_q , act_q s predikcí na 10 období dopředu	43
3.18	Odhad Neuronové sítě cpi_q , imp_q , $unemp_q$ s predikcí na 10 období dopředu	44
3.19	Odhad Neuronové sítě cpi_q , imp_q , $unemp_q$ s predikcí na 10 období dopředu	44
3.20	Odhad Neuronové sítě cpi_q , ulc_q s predikcí na 10 období dopředu	45
3.21	Odhad Neuronové sítě cpi_q , ulc_q s predikcí na 10 období dopředu	45
3.22	Odhad Neuronové sítě cpi_q , act_q , gdp_q , ulc_q s predikcí na 10 období dopředu	46
3.23	Odhad Neuronové sítě cpi_q , act_q , gdp_q , ulc_q s predikcí na 10 období dopředu	46
3.24	Odhad TVAR(2,3) cpi_q , imp_q , $int3_q$, ulc_q s predikcí na 10 období dopředu	47
3.25	Odhad TVAR(3,1) cpi_q , act_q , imp_q s predikcí na 10 období dopředu	48
3.26	Odhad TVAR(2,3) cpi_q , $m1_q$, $unemp_q$ s predikcí na 10 období dopředu	49
3.27	Odhad TVAR(2,1) cpi_q , $int3_q$, $unemp_q$ s predikcí na 10 období dopředu	49
3.28	Odhad TVAR(2,1) cpi_q , $unemp_q$ s predikcí na 10 období dopředu	50
A.1	Vývoj ukazatelů za Českou republiku	
A.2	Vývoj ukazatelů za Slovensko	
A.3	Vývoj ukazatelů za Polsko	
A.4	Vývoj ukazatelů za Maďarsko	
A.5	Vývoj ukazatelů za Rakousko	

Seznam tabulek

3.1	Ukazatele použité z OECD	28
3.2	Výsledky Rozšířeného Dickey-Fullerova testu	28
3.3	Odhady jednorozměrného AR modelu bez exogenní proměnné . .	30
3.4	Odhady AR modelu s exogenní proměnou	31
3.5	Odhad VAR model Česká republika - s exogenní proměnou EU28	33
3.6	Odhad VAR model Česká republika - bez exogenní proměnou EU28	33
3.7	Odhad VAR model pro Slovensko - s exogenní proměnnou EU28 .	34
3.8	Odhad VAR model Slovensko - bez exogenní proměnné EU28 . . .	35
3.9	Odhad VAR model pro Maďarsko - s exogenní proměnnou EU28 .	35
3.10	Odhad VAR model Maďarsko - bez exogenní proměnnou EU28 . .	36
3.11	Odhad VAR model pro Polsko - s exogenní proměnnou EU28 . . .	36
3.12	Odhad VAR model Polsko - bez exogenní proměnné EU28	37
3.13	Odhad VAR model pro Rakousko - s exogenní proměnnou EU28 .	38
3.14	Odhad VAR model Rakousko - bez exogenní proměnné EU28 . . .	38
3.15	Odhad NN model Česká republika - s exogenní proměnnou EU28	41
3.16	Odhad NN model Česká republika - bez exogenní proměnné EU28	41
3.17	Odhad nn model Slovensko - s exogenní proměnnou EU28	42
3.18	Odhad nn model Slovensko - bez exogenní proměnné EU28	43
3.19	Odhad nn model Maďarsko - s exogenní proměnnou EU28	43
3.20	Odhad nn model Maďarsko - bez exogenní proměnné EU28	44
3.21	Odhad nn model Polsko - s exogenní proměnnou EU28	44
3.22	Odhad nn model Polsko - bez exogenní proměnné EU28	45
3.23	Odhad nn model Rakousko - s exogenní proměnnou EU28	45
3.24	Odhad nn model Rakousko - bez exogenní proměnné EU28	46
3.25	Odhad TVAR modelu pro Českou republiku	47
3.26	Odhad TVAR modelu pro Slovensko	48
3.27	Odhad TVAR modelu pro Maďarsko	48
3.28	Odhad TVAR modelu pro Polsko	49
3.29	Odhad TVAR modelu pro Rakousko	50
B.1	Přehled použitých balíčků	

Úvod

Řada ekonomických a finančních rozhodnutí závisí na vývoji inflace. Inflační predikce umožňují ekonomickým subjektům dělat správná rozhodnutí a upravovat svá budoucí očekávání. Přesnost predikcí tedy není zájmem pouze národních bank, ale i soukromých podniků a bankovních institucí. Cílem predikčních modelů je co nejlépe extrahovat informace z minulého vývoje a přetvářet je na možný budoucí vývoj.

Donedávna analýze časových řad dominovali lineární modely. V posledních pár letech se však objevila řada studií poukazujících na vhodnost použití nelineárních modelů, a to zejména neuronových sítí. Například Marcellino [28] porovnává predikční schopnosti jednorozměrných lineárních modelů AR, náhodné procházky, exponenciálního vyrovnávání proti nelineárním neuronovým sítím na 480 makroekonomických časových řadách. Marcellino reportuje, že neuronové sítě dosahují přibližně v 30% případů lepších predikcí. K podobným výsledkům dochází i Nakamura [30] a Aiken [1] při porovnání AR a neuronových sítí na predikcích americké inflace. Další potvrzení výkonosti neuronových sítí v kontextu predikce inflace zemí OECD pro krátký horizont uvádí Choudhary[7], který je opět porovnává s AR modely. McNelis a McAdams [29] uvádějí možnost použití tzv. „thick (tlustých) neuronových sítí“, které jsou založeny na agregaci predikce z několika různých neuronových sítí. Takovýto model založený na Phillipsově křivce poskytuje lepší predikce pro USA, Japonsko a Eurozónu oproti modelům lineárním.

V této práci jsou zkoumány predikční schopnosti vektorových autoregresních modelů (VAR), prahových vektorových autoregresních modelů (TVAR) a feedforward neuronových sítí (NN) na inflaci v malých otevřených ekonomikách. Konkrétně jsou k modelování použity makroekonomické ukazatele za Českou republiku, Slovensko, Maďarsko, Polsko a Rakousko.

V teoretické části jsou nejdříve popsány základní koncepty analýzy časových řad, vícerozměrné modely (VAR, TVAR a NN) a možnosti použití křížové validace časových řad pro vyhodnocení predikčních chyb.

K výpočtům byl použit programovací jazyk R, knihovny Keras a Tensorflow. Všechny napsané skripty a použité balíčky jsou uvedeny v příloze B.

1. Časové řady

Časová řada je řadou hodnot uspořádaných v čase, a to směrem od minulosti do přítomnosti. V rámci analýzy časových řad je hlavním předmětem zájmu proces jejich generování. Tento proces může být deterministický případně stochastický. Stochastický proces lze chápat jako uspořádanou řadu náhodných veličin $X(s, t)$, $s \in \mathbf{S}$, $t \in \mathbf{T}$, kde $\mathbf{T}$ je indexní řada a $\mathbf{S}$ je prostor výběru. Časová řada je pak realizací tohoto stochastického procesu. [4]

1.1 Stacionarita

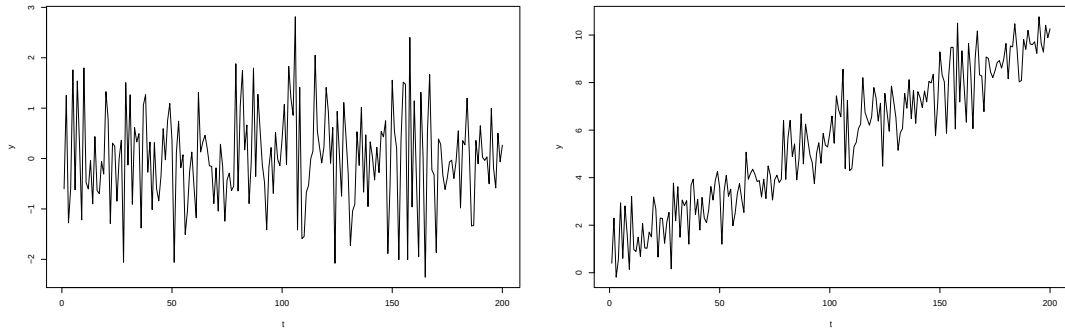
Striktně stacionární časová řada je taková řada, pro kterou platí, že pravděpodobností rozdělení každého pozorování $\{x_{t1}, x_{t2}, x_{t3}, \dots, x_{tk}\}$ je stejné jako u pozorování posunutých v čase $\{x_{t1+h}, x_{t2+h}, x_{t3+h}, \dots, x_{tk+h}\}$. Tedy spojitá pravděpodobnostní rozdělení platí

$$P\{x_{t1} \leq c_1, \dots, x_{tk} \leq c_k\} = P\{x_{t1+h} \leq c_1, \dots, x_{tk+h} \leq c_k\}$$

pro všechna $k = 1, 2, \dots$, pro všechny body v čase $t_1, t_2, \dots, t_k$, pro všechna $c_1, c_2, \dots, c_k$ a všechny posuny v čase $h = 0, \pm 1, \pm 2, \pm 3, \dots$. Pravděpodobností rozdělení dvou pozorování v čase tedy závisí pouze na vzdálenosti h mezi nimi a ne na čase t . Předpoklad striktní stacionarity je pro mnoho časových řad příliš restriktivní. Používá se tedy předpoklad **slabé stacionarity** a často se jako stacionární označuje řada splňující právě předpoklady slabé stacionarity. Pro slabě stacionární (kovariančně stacionární) řadu platí, že míry střední hodnoty, variace a kovariance nejsou funkcí času a závisí pouze na vzdálenosti mezi pozorováními. Formálně časová řada je slabě stacionární platí-li:

$$\begin{aligned} E(x_t) &= \mu, & t &= 1, 2, \dots, T \\ \text{Var}(x_t) &= \sigma_x^2, & t &= 1, 2, \dots, T \\ \text{Cov}(x_t, x_{t+h}) &= \sigma_h, & t &= 1, 2, \dots, T \end{aligned}$$

Časová řada nesplňující tyto podmínky je řadou nestacionární. Porušení těchto podmínek může být způsobeno například přítomností trendu. Pro proces s deterministickým trendem $y_t = \alpha + \beta t + x_t$, kde β je koeficient tempa trendu a x_t je stacionární časová řada, se střední hodnota y_t lineárně mění v čase $E(y_t) = \alpha + \beta t$ za předpokladu, že $E(x_t) = 0$. Další příčinou může být existence zlomu, v kterém se výrazně mění parametry data generujícího procesu.



(a) Časová řada $I(0)$

(b) Časová řada $I(0)$ s trendem

Obrázek 1.1: Příklad časových řad integrovaných řádu nula

Nejčastější příčinou nestacionarity je přítomnost stochastického trendu. Stochastický trend je tvořen kumulací náhodné složky a příkladem procesu generujícího stochastický trend je například náhodná procházka (viz 1.1.3).

Nestacionární časovou řadu lze převést na stacionární očištěním o trend v případě trendové nestacionarity, případně u řady se stochastickým trendem pomocí diferencí. Nestacionární časová řada, která po použití prvních diferencí je stacionární se označuje jako řada integrovaná řádu jedna $I(1)$. Obecně časová řada, která potřebuje k převodu na řadu stacionární k diferencí, je řadou integrovanou řádu k $I(k)$. [42][35]

1.1.1 Slabá závislost časových řad

Dalším důležitým konceptem je **slabá závislost**, která klade omezení na sílu vztahu mezi náhodnými proměnnými x_t a x_{t+h} v závislosti na rostoucí vzdálenosti mezi nimi (h). Stacionární časová řada se dá označit za slabě závislou, pokud korelace mezi x_t a x_{t+h} se „dostatečně rychle“ blíží k nule s h rostoucím do nekonečna

$$Cor(x_t, x_{t+h}) \rightarrow 0, \quad h \rightarrow \infty$$

a náhodné veličiny jsou pak asymptoticky nekorelované. [43]

1.1.2 Proces bílého šumu

Bílý šum je sérii nekorelovaných náhodných hodnot ϵ_t s nulovou střední hodnotou a konstantním rozptylem. Jedná se o zvláštní případ stacionárního procesu, v kterém jednotlivá zpoždění nejsou korelována

$$\text{cov}(\epsilon_t, \epsilon_{t+h}) = 0, \quad t = 1, 2, \dots, T$$

Bílý šum často představuje náhodnou složku časové řady tvořenou nesystematickým náhodnými pohyby. Jako pro náhodnou složku je požadováno, aby bílý šum pocházel z nezávislého, identického rozdělení (independent and identically distributed, iid). Zejména je užitečné pokud se jedná o bílý šum s normálním rozdělením $\epsilon_t \sim N(0, \sigma_\epsilon^2)$ [35]

1.1.3 Proces náhodné procházky

Náhodná procházka je klasickým příkladem nestacionárního procesu. Jedná se o zvláštní případ autoregresního modelu

$$y_t = \rho y_{t-1} + \epsilon_t$$

pro který platí, že ρ je rovno 1 a tedy proces obsahuje jednotkový kořen. Název „náhodná procházka“ pochází z faktu, že hodnota v čase t je ovlivněna pouze hodnotou v čase $t-1$ a naprosto náhodným pohybem ϵ_t . Náhodné šoky působící na řadu mají dlouhodobý efekt a jejich dopady v krátkém období nemizí. Položíme-li $y_0 = 0$ proces lze přepsat jako kumulaci náhodné složky, která tvoří stochastický trend

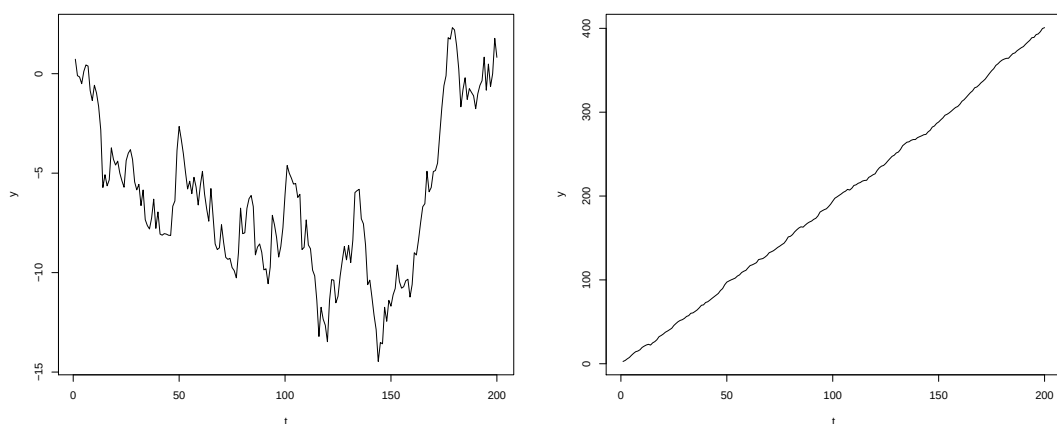
$$y_t = \sum_{t=1}^T \epsilon_t$$

Proces náhodné procházky rozšířený o úrovněnou konstantu se nazývá náhodná procházka s driftem a lze zapsat následovně

$$y_t = \alpha + \rho y_{t-1} + \epsilon_t$$

$$y_t = \alpha t + \sum_{t=1}^T \epsilon_t$$

Náhodná procházka s driftem tedy kromě stochastického trendu obsahuje i trend deterministický. Průběh obou procesů je ilustrován na obrázku 1.2.



(a) Náhodná procházka

(b) Náhodná procházka s driftem

Obrázek 1.2: Simulace náhodné procházky

1.1.4 Testy jednotkového kořene

Dickey-Fullerův test

Dickey-Fullerův test byl publikován roce 1979 [9]. Jedná se o test jednotkového kořene a je postaven na testování třech základních autoregresních modelů (AR)

$$y_t = \rho y_{t-1} + \epsilon_t \quad (1.1)$$

$$y_t = \alpha + \rho y_{t-1} + \epsilon_t \quad (1.2)$$

$$y_t = \alpha + \rho y_{t-1} + \beta_0 t + \epsilon_t \quad (1.3)$$

kde t je časový trend, α je úrovněová konstanta, ρ je koeficient zpožděného pozorování a ϵ_t je náhodná složka. Pro náhodnou složku se předpokládá, že není korelována a má nulovou střední hodnotu:

$$E(\epsilon_t) = 0, \text{Cov}(\epsilon_t, \epsilon_s) = 0 \quad \forall \quad t \neq s$$

Pokud by pro koeficient ρ v modelu model (1.1), platilo $|\rho| < 1$, pak časová řada generována tímto procesem je stacionární. Naopak, pokud by byl koeficient ρ roven 1 jednalo by se o proces náhodné procházky a řada byla nestacionární a její rozptyl by exponenciálně rostl s rostoucím t . Pro samotný test se modely

převádí na pomocné první difference:

$$\Delta y_t = (\rho - 1)y_{t-1} + u_t = \delta y_{t-1} + u_t \quad (1.4)$$

$$\Delta y_t = \alpha + (\rho - 1)y_{t-1} + u_t = \alpha + \delta y_{t-1} + u_t \quad (1.5)$$

$$\Delta y_t = \alpha + (\rho - 1)y_{t-1} + \beta_0 t + u_t = \alpha + \delta y_{t-1} + \beta_0 t + u_t \quad (1.6)$$

kde $\delta = (\rho - 1)$. Zde lze testovat pomocí dílčího t -testu nulovou hypotézu o $\delta = 0$, tedy hypotézu o ρ je rovno 1, a přítomnosti jednotkového kořene a nestacionaritě časové řady. Kritické hodnoty t -testu se získávají pomocí speciálního Dickey-Fullerova rozdělení. [9][14]

Rozšířený Dickey-Fullerův test

Rozšířený Dickey-Fullerův test (ADF) rozšiřuje modely o další zpoždění endogenní proměnné pomocné rovnice. Jednotlivé modely pak vypadají následovně:

$$\Delta y_t = \delta y_{t-1} + \gamma_1 \Delta y_{t-1} + \cdots + \gamma_{p-1} \Delta y_{t-p+1} + \epsilon_t \quad (1.7)$$

$$\Delta y_t = \alpha + \delta y_{t-1} + \gamma_1 \Delta y_{t-1} + \cdots + \gamma_{p-1} \Delta y_{t-p+1} + \epsilon_t \quad (1.8)$$

$$\Delta y_t = \alpha + \beta t + \delta y_{t-1} + \gamma_1 \Delta y_{t-1} + \cdots + \gamma_{p-1} \Delta y_{t-p+1} + \epsilon_t \quad (1.9)$$

kde p je počet přidanych zpoždění. Vhodný počet zpoždění lze určit například pomocí dílčího t -testu o hodnotách jednotlivých parametrů, případně pomocí hodnot informačních kritérií (Akaikeho informační kritérium, Bayesovo informační kritérium, Hannan-Quinnovo informační kritérium, ..., viz 1.2). Nulová hypotéza je opět o přítomnosti jednotkového kořene neboli $\delta = 0$. Alternativní hypotéza závisí na konkrétním vybraném modelu.

Phillips-Perronův test

Phillipsův a Perronův test [32] jednotkového kořene na rozdíl od ADF testu povoluje přítomnost heteroskedasticity a sériové korelace v chybové složce a nepožaduje specifikaci délky zpoždění pro testovou regresi. Testová regrese pro Phillips-Perronův test (PP test) vypadá následovně

$$\Delta y_t = \beta' D_t + \pi y_{t-1} + u_t \quad (1.10)$$

kde u_t je $I(0)$, β je vektor koeficientů a D_t je matice deterministických členů. PP test používá následně tyto upravené testové statistiky $t_{\pi=0}$ a $T_{\hat{\pi}}$

$$Z_t = \left(\frac{\hat{\sigma}^2}{\hat{\lambda}^2} \right)^{(1/2)} \cdot t_{\pi=0} - \frac{1}{2} \left(\frac{\hat{\lambda}^2 - \hat{\sigma}^2}{\hat{\lambda}^2} \right) \cdot \left(\frac{T \cdot SE(\hat{\pi})}{\hat{\sigma}^2} \right) \quad (1.11)$$

$$Z_{\pi} = T\hat{\pi} - \frac{1}{2} \frac{T^2 \cdot SE(\hat{\pi})}{\hat{\sigma}^2} (\hat{\lambda}^2 - \hat{\sigma}^2) \quad (1.12)$$

kde $\hat{\sigma}^2$ a $\hat{\lambda}^2$ jsou konzistentní odhady

$$\sigma^2 = \lim_{T \rightarrow \infty} T^{-1} \sum_{t=1}^T E(u_t^2), \quad \lambda^2 = \lim_{T \rightarrow \infty} \sum_{t=1}^T E(T^{-1} \sum_{t=1}^T u_t)$$

Při platnosti nulové hypotézy je π rovno 0 a řada je nestacionární s jednotkovým kořenem. [44]

KPSS test

KPSS (Kwiatkowski, Phillips, Schmidt, Shin) [26] test patří mezi jedny z nejpoužívanějších testů jednotkového kořene a je na rozdíl od ADF a PP testu založen na testování nulové hypotézy o stacionaritě řady. Výchozí model testu vypadá následovně[44]

$$y_t = \beta' D_t + \mu_t + u_t \quad (1.13)$$

$$\mu_t = \mu_{t-1} + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma_{\epsilon}^2) \quad (1.14)$$

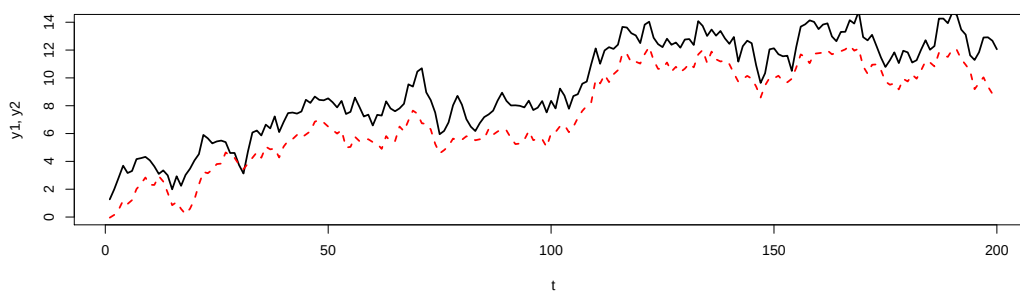
kde μ_t je proces náhodné procházky, u_t je stacionární náhodná složka, u které není nutné, aby splňovala předpoklad homoskedasticity a D_t je matice deterministických členů (například trend, konstanta, ...). Nulová hypotéza je o stacionaritě y_t , $y_t \sim I(0)$, a konkrétně je formulována o rozptylu náhodné složky μ_t

$$H_0 : \sigma_{\epsilon}^2 = 0$$

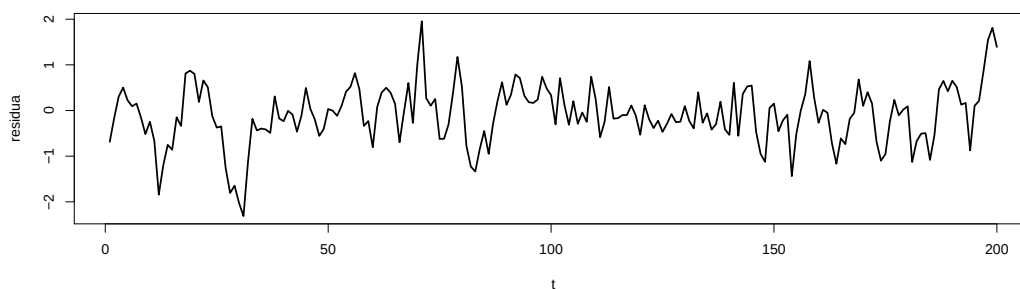
Při platnosti nulové hypotézy je proces μ_t konstantní a y_t neobsahuje jednotkový kořen. Testovací kritérium je LM:

$$\text{KPSS} = \left(T^{-2} \sum_{t=1}^T \sum_{j=1}^t \hat{u}_j \hat{u}_t \right) / \hat{\lambda}^2 \quad (1.15)$$

kde $\hat{u}_j, \hat{u}_t$ jsou rezidua regrese y_t na D_t a $\hat{\lambda}^2$ je odhad dlouhodobého rozptylu náhodné složky pomocí reziduí $\hat{u}_t$. Přesný tvar oboru kritických hodnot závisí na konkrétním tvaru deterministických členů.



(a) Časové řady $I(1)$



(b) Kointegrovaná rezidua

Obrázek 1.3: Příklad kointegrace na simulovaných časových řadách

1.1.5 Kointegrované časové řady

Mějme vektor nestacionárních časových řad $I(1)$ s jednotkovým kořenem $\mathbf{Y}_t$ o rozměru $m \times 1$. Pokud existuje taková lineární kombinace s nenulovým vektorem $\boldsymbol{\beta} = (\beta_1, \dots, \beta_m)'$, která je stacionární

$$\boldsymbol{\beta}\mathbf{Y}_t \sim I(0)$$

pak jsou tyto časové řady kointegrované (viz 1.3). Kointegrace znamená, že i přes veškeré šoky s trvalým dopadem existuje dlouhodobý rovnovážný stav, ke kterému jsou jednotlivé řady přitahovány. U kointegrovaných řad lze tedy na rozdíl od stacionárních řad pozorovat i dlouhodobé vztahy.

Pokud lineární kombinace řad s jednotkovým kořenem není stacionární, jedná se o takzvanou **zdánlivou regresi**. Tyto řady pak nemají společný stochastický trend a ve skutečnosti mají každá jiný směr vývoje. I přesto, že v případě zdánlivé regrese jsou pravé hodnoty regresních koeficientů nulové, jejich odhady metodou nejmenších čtverců vycházejí různé od nuly a jsou statisticky významné. Dílčí t -testy a koeficient determinace R^2 jsou tedy nadhodnoceny a nesprávně indikují vhodnost modelu. Regresní analýza nestacionárních řad má smysl pouze jsou-li řady kointegrované. Detailněji popsáno například Arlt [3].

1.2 Vektorové autoregresní modely

Vektorový autoregresní model (VAR) je jeden z nejpoužívanějších nástrojů pro analýzu vícerozměrných časových řad. VAR model je rozšířením autoregresního modelu o více proměnných, které do modelu vstupují rovnocenně a jsou považovány za endogenní. Vývoj proměnných modelu je vysvětlován jejich zpožděnými hodnotami a zpožděnými hodnotami ostatních proměnných. Obdobně jako u AR i u VAR modelu se předpokládá stacionarita. Pokud řady nejsou stacionární je nutné je stacionarizovat, případně pokud mezi nimi existuje kointegrační vztah je vhodné použít model korekce chyby. [20]

VAR model se prokázal jako zejména vhodný pro popis dynamického chování ekonomických a finančních časových řad a jejich predikce. Často poskytuje lepší predikce oproti jednorozměrným modelům a modelům simultánních rovnic [44].

Nechť $\mathbf{Y}_t = (y_{1t}, y_{2t}, \dots, y_{mt})'$ je $m \times 1$ rozměrný vektor časových řad. Pak základní vektorový autoregresní model s p zpožděními lze zapsat

$$\mathbf{Y}_t = \mathbf{c} + \Pi_1 \mathbf{Y}_{t-1} + \Pi_2 \mathbf{Y}_{t-2} + \dots + \Pi_p \mathbf{Y}_{t-p} + \boldsymbol{\epsilon}_t \quad (1.16)$$

$$t = 1, 2, \dots, T$$

kde Π_i je $m \times m$ rozměrná matice koeficientů, $\mathbf{c}$ je vektor úrovnových konstant a $\boldsymbol{\epsilon}_t$ je $m \times 1$ rozměrný vektor bílého šumu. Například dvojrozměrný VAR model s dvěma zpožděními lze zapsat jako

$$\begin{pmatrix} y_{1t} \\ y_{2t} \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} \pi_{11}^1 & \pi_{12}^1 \\ \pi_{21}^1 & \pi_{22}^1 \end{pmatrix} \begin{pmatrix} y_{1t-1} \\ y_{2t-1} \end{pmatrix} + \begin{pmatrix} \pi_{11}^2 & \pi_{12}^2 \\ \pi_{21}^2 & \pi_{22}^2 \end{pmatrix} \begin{pmatrix} y_{1t-2} \\ y_{2t-2} \end{pmatrix} + \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{pmatrix} \quad (1.17)$$

VAR model je tedy zdánlivě nezávislou soustavou modelů se společnými regresory ve tvaru zpožděných proměnných a deterministických členů. V tomto tvaru lze modely odhadnout zvláště pomocí metody nejmenších čtverců.

Základní VAR(p) model může být pro popis vývoje časových řad příliš restriktivní a často je vhodné rozšíření o exogenní proměnné. Nejčastěji se může jednat o pomocné proměnné deterministického trendu, případně o umělé sezónní proměnné, ale také o další stochastické proměnné. Rozšířený VAR(p) model lze zapsat

$$\mathbf{Y}_t = \mathbf{c} + \Pi_1 \mathbf{Y}_{t-1} + \Pi_2 \mathbf{Y}_{t-2} + \dots + \Pi_p \mathbf{Y}_{t-p} + \Phi \mathbf{D}_t + \mathbf{G} \mathbf{X}_t + \boldsymbol{\epsilon}_t \quad (1.18)$$

kde $\mathbf{D}_t$ je matice deterministických proměnných, $\mathbf{X}_t$ je matice stacionárních exogenních proměnných a $\Phi, \mathbf{G}$ jsou matice koeficientů. [44]

Výběr počtu zpoždění

Pro správnou specifikaci VAR(p) modelu je zásadní výběr vhodného počtu zpoždění p . Obecný přístup je odhadnout sadu modelů s různými zpožděními $p = 0, 1, \dots, p_{max}$ a následně volit takové zpoždění, které minimalizuje vybraná kritéria. Většinou se používají informační kritéria založená na analýze reziduální kovarianční matice $\bar{\Sigma}(p)$, která je definována jako

$$\bar{\Sigma}(p) = \frac{1}{T} \sum_{t=1}^T \hat{\epsilon}_t \hat{\epsilon}_t'$$

Kritéria se od sebe liší různým přístupem k penalizaci složitosti modelu (počtu zpoždění p a počtu rovnic n) vůči počtu pozorování T . Mezi nej-používanější patří Akaikeho informační kritérium (1.19) (AIC), Schwarz-Bayesovo informační kritérium (1.20) (BIC) a Hannan-Quinnovo (1.21) (HQ). [44]

$$AIC(p) = \ln |\bar{\Sigma}(p)| + \frac{2}{T}pn^2 \quad (1.19)$$

$$BIC(p) = \ln |\bar{\Sigma}(p)| + \frac{\ln T}{T}pn^2 \quad (1.20)$$

$$HQ(p) = \ln |\bar{\Sigma}(p)| + \frac{2 \ln \ln T}{T}pn^2 \quad (1.21)$$

Dle Lütkepohl [27] pro delší časové řady platí, že AIC určuje stejný nebo vyšší počet zpoždění než HQ a BIC. Dále pro všechny délky řad HQ určuje stejné nebo vyšší počet zpoždění než BIC.

$$\hat{p}(BIC) \leq \hat{p}(AIC) \quad \text{pro } T \geq 8$$

$$\hat{p}(BIC) \leq \hat{p}(HQ) \quad \forall T$$

$$\hat{p}(HQ) \leq \hat{p}(AIC) \quad \text{pro } T \geq 16$$

Podobných výsledků dosáhl i Gredenhoff [13], který dále podotýká, že AIC a HQ odhadují správnou délku zpoždění častěji než BIC. Dále u větších modelů má HQ tendenci vybírat správnou délku častěji než AIC a pro oba platí, že s rostoucí velikostí vzorku roste jejich přesnost rychleji než u BIC. BIC má tedy tendenci podhodnocovat a není vhodný pro výběr zpoždění u modelů určených k statistické inferenci, naopak může být velice vhodný pro predikční modely. U predikčních modelů totiž vychýlení způsobené podparametrizováním může být vyrovnáno snížením variability odhadu.

Grangerova kauzalita

Grangerova kauzalita pojednává o vazbách mezi stacionárními proměnnými ve vztahu k jejich predikčním schopnostem. Přesněji řečeno, pokud proměnná y_1 je nápomocná při tvorbě předpovědi jiné proměnné y_2 , pak lze říci, že y_1 působí ve smyslu Grangerovy kauzality na proměnnou y_2 .

Naopak y_2 nepůsobí ve smyslu Grangerovy kauzality, pokud střední čtvercová chyba (MSE) (viz (1.54)) předpovědi $y_{2,t+s}$, založené na zpožděních y_2 ($y_{2,t}, y_{2,t-1}, \dots$), je stejná jako MSE předpovědi $y_{2,t+s}$ založené na zpožděních ($y_{2,t}, y_{2,t-1}, \dots$) a zpožděních ($y_{1,t}, y_{1,t-1}, \dots$) [44].

$$\begin{aligned} MSE[\hat{E}(y_{2,t+s} | (y_{2,t}, y_{2,t-1}, \dots))] &= \\ &= MSE[\hat{E}(y_{2,t+s} | (y_{2,t}, y_{2,t-2}, \dots, y_{1,t}, y_{1,t-1}, \dots))] \end{aligned} \quad (1.22)$$

K testování Grangerovy kauzality dvou proměnných je možné použít porovnání dvou autoregresních modelů odhadnutých MNČ. První model je neomezený a uvažuje konkrétní délku zpoždění p

$$y_{1,t} = c_1 + \alpha_1 y_{1,t-1} + \dots + \alpha_p y_{1,t-p} + \beta_1 y_{2,t-1} + \dots + \beta_p y_{2,t-p} + u_t \quad (1.23)$$

kde u_t je bílý šum s normálním rozdělením. Druhý model je omezený a obsahuje pouze zpoždění první proměnné

$$y_{1,t} = c_1 + \alpha_1 y_{1,t-1} + \dots + \alpha_p y_{1,t-p} + e_t \quad (1.24)$$

Z těchto modelů jsou získány součty čtverců reziduí (1.25) a je proveden F test s nulovou hypotézou o neexistenci Grangerově kauzality mezi y_1 a y_2 , neboli o nevýznamnosti parametrů zpožděných hodnot y_2 v neomezeném modelu $H_0 : \beta_1 = \beta_2 = \dots = \beta_p = 0$

$$RSS_1 = \sum_{t=1}^T \hat{u}_t^2, \quad RSS_0 = \sum_{t=1}^T \hat{e}_t^2 \quad (1.25)$$

Testové kritérium vypadá následovně[16]

$$F = \frac{(RSS_0 - RSS_1)/p}{RSS_1/(T - 2p - 1)} \sim F(p, T - 2p - 1) \quad (1.26)$$

Tento test je možné zobecnit ve vícerozměrném kontextu pro dvě skupiny proměnných v rámci VAR modelu.

1.2.1 Diagnostické testy

Po odhadu VAR modelu je nutné ověřit předpoklady kladené na chybovou složku. Splnění předpokladů zaručuje, že se modelu podařilo odčerpát systematickou složku procesu. Na druhou stranu z hlediska kvality předpovědí nemusí být splnění všech předpokladů nutné, a to zejména z důvodů tzv. „bias-variance trade-off“.

Pomocí diagnostických testů lze zkoumat vlastnosti náhodné složky na reziduech modelu. Rezidua by neměla vykazovat přítomnost autokorelace, měla by mít homogenní rozptyl (homoskedasticita) a měla by mít přibližně normální rozdělení. [31]

Breusch-Godfrey

Pro testování autokorelace lze použít například Breusch-Godfreyův test. Test používá LM statistiky a je založen na pomocné regresi reziduí, která pro vícerozměrnou verzi testu vypadá následovně

$$\hat{\epsilon}_t = A_1 \mathbf{y}_{t-1} + \dots + A_p \mathbf{y}_{t-p} + C D_t + B_1 \hat{\epsilon}_{t-1} + \dots + B_h \hat{\epsilon}_{t-h} + u_t \quad (1.27)$$

kde C je matice koeficientů deterministických proměnných D_t . Nulová hypotéza je $H_0 : B_1 = B_2 = \dots = B_h = 0$ a alternativní hypotéza je, že alespoň jeden koeficient je statisticky významný. Testová statistika je definována jako

$$LM_h = T(n - stopa(\tilde{\sigma}_R^{-1} \tilde{\sigma}_e)) \sim \chi^2(hn^2) \quad (1.28)$$

kde $\tilde{\sigma}_R, \tilde{\sigma}_e$ jsou kovariační reziduální matice omezeného a neomezeného modelu. Alternativou je například populární Portmanteauův test. Dále je možné použít korelogram autokorelační funkce na rezidua každé rovnice. [31]

ARCH test

ARCH test konstantního rozptylu náhodné složky je založen na posouzení, zdali složka obsahuje tzv. ARCH efekt. Existuje jak v jednorozměrné variantě, která lze použít na rezidua jednotlivých rovnic, tak ve vícerozměrné variantě[31]. Jednorozměrný test je založený na pomocné regresi s druhými mocninami zpožděných reziduí

$$\epsilon_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \dots + \alpha_q \epsilon_{t-q}^2 + u_t \quad (1.29)$$

Při platnosti nulové hypotézy o homoskedasticitě náhodné složky platí, že $\alpha_1 = \alpha_2 = \dots = \alpha_q = 0$ a pak testová statistika je

$$LM = TR^2 \sim \chi^2(q) \quad (1.30)$$

kde T je délkou časové řady a R^2 je koeficient determinace z pomocné regrese. Odvození testu uvádí například Zivot[44].

Jarque-Bera test

Jednorozměrná verze Jarque-Berova testu normality může být použita zvláště na rezidua každé jednotlivé rovnice. Případně lze použít vícerozměrnou verzi testu, pro kterou je nejdříve nutné upravit a standardizovat rezidua [31]. Jarque-Berova (JB) statistika vychází z výběrové šikmosti a špičatosti a z předpokladu, že normálně rozdělená veličina má koeficient šikmosti roven nule a koeficient špičatosti přibližně roven 3. Pro jednorozměrný test je JB statistika definována následovně

$$JB = \frac{T}{6} \left(\widehat{Šikmost} + \frac{(\widehat{Špičatost} - 3)^2}{4} \right) \quad (1.31)$$

Při platnosti nulové hypotézy o normálním rozdělení náhodné složky $H_0 : \epsilon_t \sim N(\cdot)$ má JB statistika

$$JB \sim \chi^2(2)$$

Alternativní hypotéza říká, že náhodná složky nepocházejí z normálního rozdělení. Další možnosti pro testování normality je Shapiro-Wilkův test a případně grafické nástroje (histogram, Q-Q graf, ...).[44]

1.3 Prahové vektorové autoregresní modely

Prahové autoregresní modely, poprvé představeny Tongem [39] v roce 1978, jsou po částech lineární modely s měnícími se režimy. Skládají se z několika $AR(p)$ procesů mezi kterými se přepíná dle vybrané exogenní proměnné a jejích prahů. Prahové autoregresní modely se označují jako $TAR(k, p)$, kde k je počet režimů a p je počet zpoždění v každém AR procesu. Speciálním případem TAR je tzv. „self-exciting“ SETAR, kde prahovou proměnnou je určité zpoždění samotné modelované řady. Prahový autoregresní model umožňuje postihnout možné nelineárnosti, například nesymetrické reakce na šoky, případně existenci více ekvilibrií.[15]

Prahový vícerozměrný autoregresní model (TVAR) s dvěma režimy je dán následovně (viz například [21]):

$$\mathbf{y}_t = \sum_{i=1}^q (\boldsymbol{\mu}_i + \sum_{j=1}^p \boldsymbol{\Phi}_{ij} \mathbf{y}_{t-j} + \boldsymbol{\Gamma}_i \mathbf{x}_t + \boldsymbol{\epsilon}_{it}) 1(c_{i-1} < s_{t-d} \leq c_i) \quad (1.32)$$

kde $\mathbf{y}_t$ je náhodný vektor ($m \times 1$), $\boldsymbol{\Phi}_1$ jsou matice ($m \times m$) parametrů a $1(.)$ je indikátorová funkce určující režimy dle náhodné stacionární proměnné s_{t-d} , kde d je zpoždění prahové proměnné. $\boldsymbol{\epsilon}_t$ je náhodná složka s $E(\boldsymbol{\epsilon}_t) = 0$, $Var(\boldsymbol{\epsilon}_t) = \boldsymbol{\Sigma}$, kde $\boldsymbol{\Sigma}$ je pozitivně definitní matice s plnou hodnotostí. Pro prahovou proměnnou s_t se předpokládá, že pochází ze spojitého rozdělení.[21]

Pro tento model považujeme s_t za známou proměnnou a počet režimů q , prahy c_i a zpoždění prahové proměnné d za neznáme. Nejdříve však musíme určit, zdali vhodné použít model TVAR a tedy počet režimů je větší než jedna. Pokud je počet režimů roven jedné, jedná se o lineární model. Další podrobnější popis TVAR modelů uvádí například Tong[39], Tsay[40].

1.3.1 Testování nelineárnosti

Testování nulové hypotézy o vícerozměrném lineárním modelu oproti alternativní hypotéze o vícerozměrném prahovému modelu je náročné, a to zejména z toho důvodu, že za platnosti nulové hypotézy nejsou prahy c_i definovány.

Například Hansen[17] popisuje metody testování nulové hypotézy o linearitě jako porovnání dvou modelů SETAR. K porovnání $SETAR(j)$ oproti $SETAR(k)$ pro $(k > j)$ lze použít testovou statistiku

$$F_{jk} = n \left(\frac{S_j - S_k}{S_k} \right) \quad (1.33)$$

kde S_j, S_k jsou součty čtverců reziduí. Pokud rezidua jsou nezávislá z $N(0, \sigma^2)$

pak se jedná o likelihood ratio test a zároveň F test. Důležité je omezit prahy tak, aby každý j -tý režim obsahoval dostatek pozorování n_j

$$n_j = \sum_{t=1}^n I_{kt}(c_{i-1} < y_{t-d} \leq c_i) \quad (1.34)$$

Hanssen navrhuje, aby počet pozorování pro prahy byl, s n rostoucím do nekonečna, alespoň $(n_j/n) * n \geq n\tau$ pro $\tau > 0$. Například při porovnání SETAR(1) oproti SETAR(2), kde SETAR(1) je lineární autoregresní model ve tvaru

$$y_t = \mu + \sum_{j=1}^p \Phi_j y_{t-j} + \epsilon_t \quad (1.35)$$

a SETAR(2) obsahuje pouze dva režimy,

$$y_t = \mu + \sum_{j=1}^q \Phi_{1j} y_{t-j} I_{1t}(c_1 < y_{t-d} \leq c_2) + \sum_{j=1}^q \Phi_{2j} y_{t-j} I_{2t}(c_2 < y_{t-d} \leq c_3) + \epsilon_t \quad (1.36)$$

lze získat součet čtverců pro první model jako $S_1 = \hat{\epsilon}'\hat{\epsilon}$ a pro druhý model součet čtverců S_2 je funkcí parametrů c a d . Přesněji $S_2(c, d) = (\mathbf{y} - \hat{\mathbf{y}})'(\mathbf{y} - \hat{\mathbf{y}}) = \hat{\epsilon}'\hat{\epsilon} - f_2(c, d) = S_1 - f_2(c, d)$. Testové kritérium pak vypadá následovně

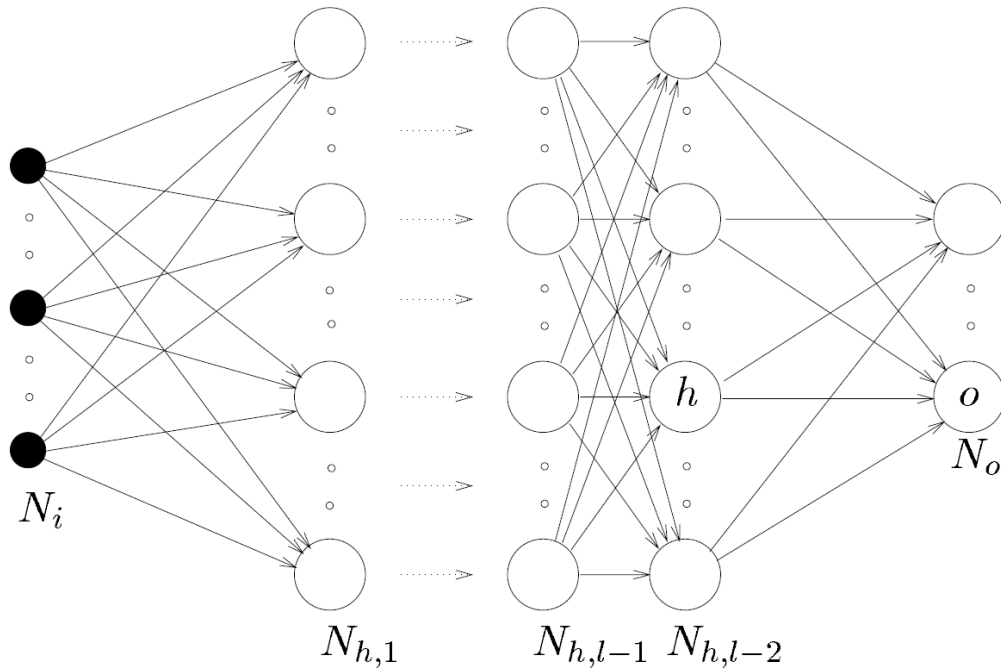
$$F_{12} = n \left(\frac{f_2(\hat{c}, \hat{d})}{S_1 - f_2(\hat{c}, \hat{d})} \right) \quad (1.37)$$

kde parametry $\hat{c}, \hat{d}$ maximalizují funkci f_2 a tedy zároveň minimalizují S_2 . Vysoké hodnoty testového kritéria by měli vést k zamítnutí nulové hypotézy, ale není možné určit kritický obor pokud není známo rozdělení F_{12} za platnosti nulové hypotézy. Hansen navrhuje použití techniky podobné bootstrapu k odhadu asymptotického rozdělení testové statistiky. Tvar tohoto rozdělení do značné míry závisí na konkrétním vztahu mezi regresory a prahovou proměnnou, z tohoto důvodu není možné rozdělení tabelizovat a je nutné jej odhadovat pro každou konkrétní aplikaci. Přesný postup je popsán v Hansen 1999 [17].

1.4 Neuronové sítě

Počátek neuronových sítí se datuje k roku 1943, kdy McCulloch a Pitts představili první neuronovou síť, pomocí které simulovali fungování neuronů v mozku. Dále v roce 1957 Rosenblatt představuje dvouvrstvou síť zvanou Perceptron, která po vhodném nastavení vah byla schopná jednoduché klasifikace. I tato relativně pokročilá síť měla řadu omezení a například nebyla schopná řešit XOR ¹ vztahy. Tyto limitace vedly k úpadku zájmu o neuronové sítě, trvajícím téměř dvě desetiletí. V 80. tých letech 19. století se téma oživilo, a to hned několika inovacemi. Kohonen vytvořil nový typ samoorganizační sítě tzv. Kohonenovy mapy. Dále v roce 1986 byl objeven **zpětně propagační** algoritmus, který umožňuje učení sítě pomocí zpětného šíření chyb.[11]

V dnešní době díky nárůstu výpočetní kapacity a zlepšení optimalizačních algoritmů se neuronové sítě znovu dostávají do popředí zájmu. Neuronové sítě jsou výkonný nástroj pro klasifikační a predikční úlohy. [25]



Obrázek 1.4: Vícevrstvá neuronová síť s l vrstvami (převzato: Smagt 1996 [25])

¹XOR neboli exkluzivní disjunkce, která nabývá hodnoty TRUE („pravda“) pokud se její vstupy liší

1.4.1 Struktura feedforward neuronové sítě

Feedforward neuronová síť neboli také vícevrstvý perceptron je čistý model hlubokého učení. Cílem feedforward sítě je aproximace dané funkce f . Například pro klasifikační síť je cílem namapování funkce $y = f(x)$ od vstupu x do kategorií y . Tento typ sítě se nazývá feedforward, protože informace jím jakoby protéká jedním směrem od vstupu do výstupu a neexistují žádná zpětná spojení. Síť obsahující zpětná spojení se nazývá rekurentní neuronová síť.[11]

Feedforward neuronovou síť si lze představit jako síť neuronů uspořádaných do vrstev. Vrstvy jsou navzájem propojeny a předávají si informace. Každá vrstva obsahuje neurony neboli jednotky. Každá jednotka přijímá vstupy z jednotek, které jsou ve vrstvách pod nimi a zároveň odesílají výstupy jednotkám ve vrstvách nad nimi. V rámci jedné vrstvy mezi jednotkami nejsou žádné vztahy. Jednotky z **vstupní vrstvy** N_i posílají informace do první skryté vrstvy $N_{h,1}$, následně pak přes všechny skryté vrstvy $N_{h,l}$ až do **výstupní vrstvy** N_0 (viz. 1.4). Mezi skrytými vrstvami jsou vstupy přenásobeny vahami a je na ně aplikována **aktivační funkce** F_i . [25] Aktivační funkce může být libovolná diferencovatelná funkce. Výstupní jednotka $\mathbf{x}^{(k)}$ k -té vrstvy pak vypadá následovně:

$$\mathbf{x}^{(k)} = F(\mathbf{a}^{(k)}) \quad (1.38)$$

kde $\mathbf{a}^{(k)}$ je vážený součet výstupu z jednotky předchozí

$$\mathbf{a}^{(k)} = \mathbf{b}^{(k)} + \mathbf{W}^{(k)}\mathbf{x}^{(k-1)} \quad (1.39)$$

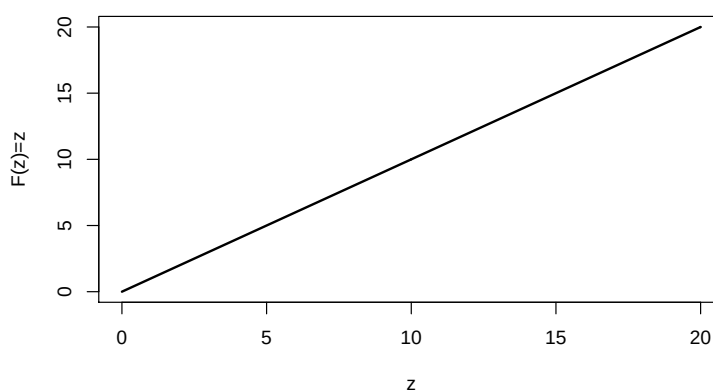
kde $\mathbf{W}^{(k)}$ je matice vah k -té vrstvy a $\mathbf{b}^{(k)}$ vektor přidáných konstant. [11] Váhy neuronové sítě jsou odhadnuty iteračně, pomocí dopředné a zpětné propagace (viz strana 21) vůči zvolené nákladové funkci. Po odhadu vah se používá již pouze dopředná propagace. I přestože samotné učení sítě může být časově náročné, získávání výsledků z již naučené sítě je velice rychlé. [10]

1.4.2 Aktivační funkce

Existuje mnoho různých aktivačních funkcí s různými vlastnostmi. Hlavní účel aktivační funkce je transformace vstupů. Aktivační funkce pro neuronovou síť se zpětnou propagací by měla mít několik důležitých charakteristik. Měla by být spojitá, diferencovatelná a monotónní neklesající. [10] Mezi nejpoužívanější aktivační funkce patří:

- Lineární funkce

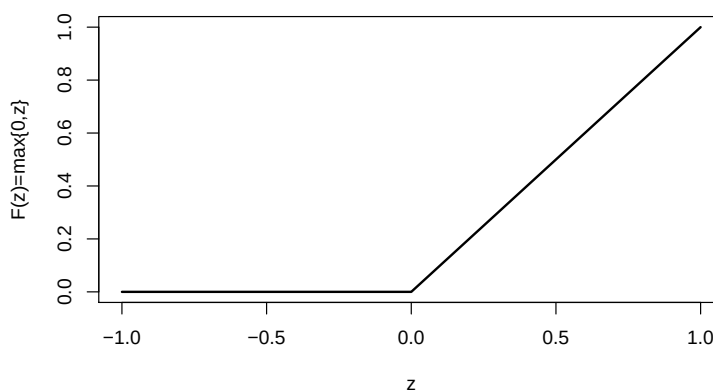
$$F(z) = z \quad (1.40)$$



Obrázek 1.5: Lineární funkce

- ReLU - rectified linear unit

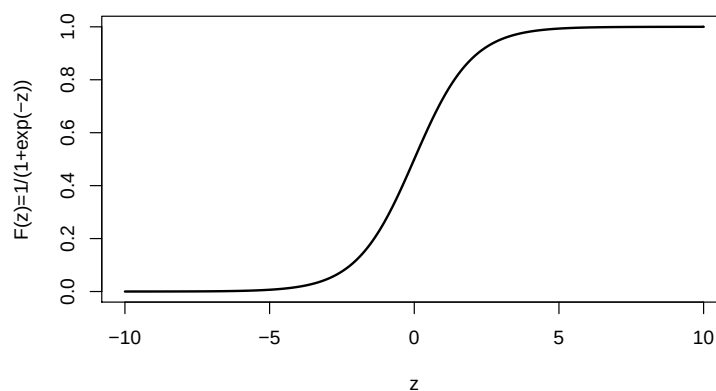
$$F(z) = \max\{0, z\} \quad (1.41)$$



Obrázek 1.6: ReLU - rectified linear unit

- Sigmoid funkce

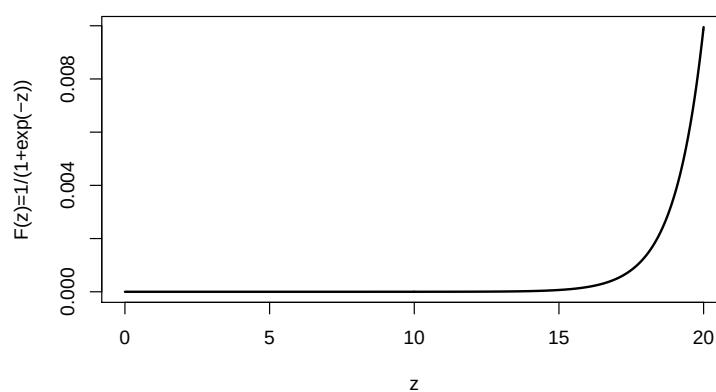
$$F(z) = \frac{1}{1 + e^{-z}} \quad (1.42)$$



Obrázek 1.7: Sigmoid funkce

- Softmax funkce

$$F(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (1.43)$$



Obrázek 1.8: Softmax funkce

1.4.3 Normalizace dat

Nelineární aktivační funkce často omezují nebo stlačují výstup jednotek do rozmezí (0,1), případně (-1,1). Z toho důvodu je vhodné data před trénováním sítě normalizovat. Pokud síť obsahuje nelineární aktivační funkce je nutné, aby hodnoty požadovaného výstupu byly ve stejném rozmezí jako výstupy sítě. I pokud síť obsahuje pouze lineární funkce může být vhodné data nejdříve normalizovat, a to zejména pro splnění požadavků učících algoritmů. Mezi nejpoužívanější normalizační přístupy patří:

- Normování dat

$$\hat{x} = \frac{x_i - \bar{x}}{\sigma} \quad (1.44)$$

- Prosté normování

$$\hat{x} = \frac{x_i}{x_{max}} \quad (1.45)$$

Případně pro některé aplikace může být vhodné data pouze lineárně transformovat například dle:

- Lineární transformace do rozmezí (0,1)

$$\hat{x} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (1.46)$$

- Lineární transformace do rozmezí (a, b)

$$\hat{x} = \frac{(b - a)(x_i - x_{min})}{x_{max} - x_{min}} \quad (1.47)$$

Dále normalizace dat předchází možným výpočetním problémům jako je například numerické přetékání ²

Data v rámci jedné neuronové sítě by měla být normalizována stejným způsobem, aby nebyla narušena jejich struktura a vazby, tedy nedocházelo k ztrátě informace.

²K přetékání dochází, pokud absolutní hodnota výsledku aritmetické operace je větší než jaké může být uložena v paměti počítače. Naopak pokud je absolutní hodnota příliš malá dochází k podtékání a hodnota je většinou nahrazena nulou.

1.4.4 Učení neuronové sítě

Na počátku výpočtu jsou váhy w_{jk} stanoveny náhodně a výstup sítě y_0 je spočítán takzvanou **dopřednou propagací** (na rozdíl od vyrovnaných hodnot). Vstupy jsou propagovány přes jednotlivé skryté vrstvy dle (1.48) až do výstupní vrstvy, kde produkují odhad $\hat{\mathbf{y}}$. Algoritmus dopředné propagace pro zjednodušený příklad vícevrstvé plně propojené sítě s pouze jedním vstupem a jedním výstupem lze zapsat následovně:

$$\begin{aligned}
 &\mathbf{h}^{(0)} = \mathbf{x} \\
 &\text{for } k = 1, \dots, l \text{ do} \\
 &\quad \mathbf{a}^{(k)} = \mathbf{b}^{(k)} + \mathbf{W}^{(k)} \mathbf{h}^{(k-1)} \\
 &\quad \mathbf{h}^{(k)} = F(\mathbf{a}^{(k)}) \\
 &\text{end for} \\
 &\hat{\mathbf{y}} = \mathbf{h}^{(l)} \\
 &J = L(\hat{\mathbf{y}}, \mathbf{y}) + \lambda \Omega(\theta)
 \end{aligned} \tag{1.48}$$

kde l le hloubka sítě (počet vrstev), $\mathbf{W}^{(k)}$, $k \in 1, \dots, l$ je matice vah, $\mathbf{b}^k$, $k \in 1, \dots, l$ je vektor konstant, $\mathbf{x}$ je vstup, $\mathbf{y}$ je cílový výstup, tedy skutečná nepozorovaná hodnota a $\hat{\mathbf{y}}$ je jejím odhadem.[11] Celková nákladová funkce J je složena z nákladové funkce $L(\hat{\mathbf{y}}, \mathbf{y})$ a takzvaného regularizačního členu $\Omega(\theta)$. Příkladem nákladové funkce může být například čtvercová chyba

$$J(\theta) = \frac{1}{2} \sum_{o=1}^{N_o} (d_o - y_o)^2 \tag{1.49}$$

kde θ jsou parametry modelu, y_o je výstup sítě a d_o je požadovaný výstup.

Regularizační člen penalizuje velikost odhadnutých parametrů θ , tedy všech parametrů (vah $\mathbf{W}$ i konstant $\mathbf{b}$). Síla regularizace je ovlivněna velikostí koeficientu λ , pokud je koeficient roven nule celková nákladová funkce neobsahuje regularizační člen. Vhodně zvolená velikost λ by měla zabránovat přeučení sítě. Dopředná propagace tedy mapuje parametry modelu k nákladové funkci.

Další fáze je zvaná **zpětná propagace** a jejím cílem je minimalizace této chyby. Zpětná propagace je založena na sledování toku informací zpět od nákladové funkce ke vstupům a následném výpočtu gradientu $\nabla_w J(w)$.

Algoritmus zpětné propagace, opět pro síť s jedním vstupem a jedním výstupem

například Godfellow[11] zapisuje následovně :

$$\begin{aligned}
& \mathbf{g} \leftarrow \nabla_{\hat{y}} J = \nabla_{\hat{y}} L(\hat{y}, y) \\
& \textbf{for } k = l, l-1, \dots, 1 \textbf{ do} \\
& \quad \mathbf{g} \leftarrow \nabla_{\mathbf{a}^{(k)}} J = \mathbf{g} \odot f'(\mathbf{a}^{(k)}) \\
& \quad \nabla_{\mathbf{b}} J = \mathbf{g} + \lambda \nabla_{\mathbf{b}^{(k)}} \omega(\theta) \\
& \quad \nabla_{\mathbf{W}} J = \mathbf{g} \mathbf{h}^{(k-1)T} + \lambda \nabla_{\mathbf{W}^{(k)}} \omega(\theta) \\
& \quad \mathbf{g} \leftarrow \nabla_{\mathbf{h}^{(k-1)}} J = \mathbf{W}^{(k)T} \mathbf{g} \\
& \textbf{end for}
\end{aligned} \tag{1.50}$$

kde $\odot$ značí násobení po složkách neboli tkz. Hadamardův produkt. Tento výpočet postupně získává gradienty aktivačních funkcí každé k vrstvy, tedy od výstupu přes všechny skryté vrstvy. Z takto získaných gradientů lze získat gradienty parametrů každé vrstvy a ty lze použít v některé z optimalizačních metod. [25]

1.4.5 TensorFlow a rozhraní Keras

TensorFlow je open source knihovna určená k numerickým výpočtům. TensorFlow byla původně vyvíjena teamem Google Brain, v rámci Google Machine Intelligence Research pro výzkum strojového učení a neuronových sítí, a následně byla uvolněna jako open source. Hlavním účelem je umožnit pomocí flexibilní struktury výpočet napříč platformami a operačními systémy. Knihovna Tensorflow je napsána v programovacím jazyku Python a je dostupná pro Windows, Linux, macOS, ale v odlehčené verzi i pro mobilní systémy Android a iOS. Výpočet lze provádět jak pomocí CPU a GPU, tak i pomocí specializovaných TPU³. [38]

Keras je vysokoúrovňové API⁴ pro odhad neuronových sítí. Umožňuje práci s knihovnami TensorFlow, CNTK⁵ a Theano⁶. Keras byl vyvinut pro usnadnění experimentování a výstavby neuronových sítí. Keras umožňuje práci s knihovnami i v rámci programovacího jazyku R. V rámci Keras lze vystavět síť s libovolnou strukturou, s libovolným počtem vstupů a výstupů, se sdílením vrstev a sdílením modelů, atd. Keras lze tedy v zásadě použít k výstavbě jakéhokoliv modelu hlubokého učení. [2]

³Tensor processing unit vyvinutých Googlem

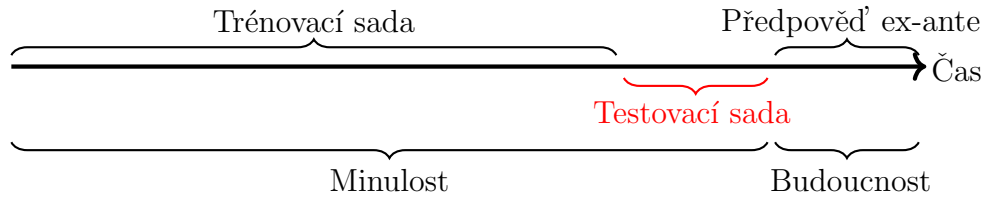
⁴Application Programming Interface

⁵Microsoft Cognitive Toolkit - <https://www.microsoft.com/en-us/cognitive-toolkit/>

⁶Theano <http://deeplearning.net/software/theano/>

1.5 Měření přesnosti předpovědí

Pro správnost vyhodnocení předpovědní přesnosti je nutné vyhodnocovat „skutečné“ předpovědi. Z toho důvodu je vhodné dostupná pozorování $(y_1, \dots, y_T)$ rozdělit na trénovací sadu $(y_1, \dots, y_N)$ a testovací sadu $(y_{N+1}, \dots, y_T)$. Kde na trénovací sadě je model odhadnut a následně je odhadnutý model použit ke generování tzv. Testovacích ex-post predikcí, které lze porovnávat s pozorováními z testovací sady. Běžně se například jako trénovací sada používá prvních 80% pozorování (viz 1.9). Případně se délka trénovací sady volí tak, aby délka testovací sady odpovídala alespoň délce požadovaného předpovědního horizontu h . [23]



Obrázek 1.9: Rozdělení časové řady na testovací a trénovací sadu

Predikční chyba je pak rozdílem mezi skutečnou hodnotou z testovací sady a predikcí podmíněna model odhadnutým na trénovací sadě, případně pozorováními exogenních regresorů z testovací sady

$$e_t = y_t - \hat{y}_{t|N}, t = N + 1, \dots, T \quad (1.51)$$

Míry predikčních chyb se dají dělit na absolutní, procentuální a relativní. Mezi míry používající absolutní chybu patří například Mean Absolute Error (MAE), Mean Square Error (MSE) a Root Mean Square Error (RMSE)

$$MAE_h = \frac{1}{h} \sum_{i=1}^h |e_i| \quad (1.52)$$

$$MSE_h = \frac{1}{h} \sum_{i=1}^h (e_i)^2 \quad (1.53)$$

$$RMSE_h = \sqrt{\frac{1}{h} \sum_{i=1}^h (e_i)^2} \quad (1.54)$$

kde h je predikční horizont. Nevýhodou těchto měr je jejich závislost na velikosti jednotek řady a jejich snadné vychýlení odlehlými pozorováními.

Další možností jsou procentuální chyby Mean Absolute Percentage Error (MAPE), Root Mean Square Percentage Error (RMSPE)

$$MAPE_h = \frac{1}{h} \sum_{i=1}^h 100 \cdot \frac{|e_i|}{y_i} \quad (1.55)$$

$$RMSPE_h = \sqrt{\frac{1}{h} \sum_{i=1}^h (100 \cdot \frac{|e_i|}{y_i})^2} \quad (1.56)$$

Procentuální chyby není vhodné používat, pokud některá pozorování jsou rovny nule. Dále nemusejí poskytovat nevychýlený odhad chyby. Relativní chyby jsou založeny na vyhodnocování vůči chybě referenčního modelu

$$r_t = \frac{|e_t|}{y_t - \hat{y}^*} \quad (1.57)$$

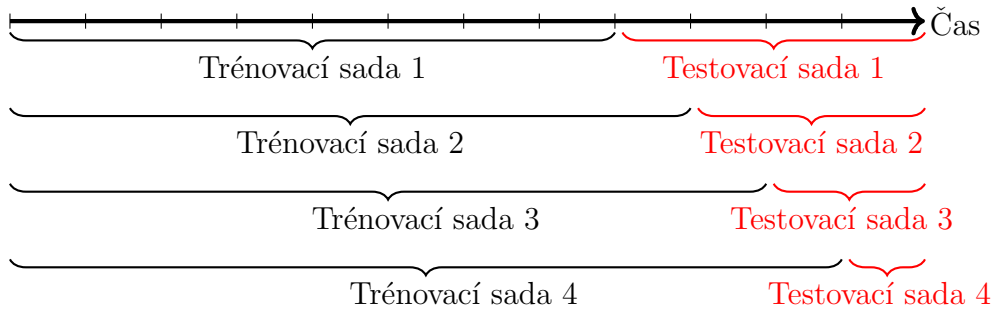
kde $\hat{y}_t^*$ je predikce získána pomocí referenčního modelu. Do této skupiny patří například Mean Relative Absolute Error (MRAE)

$$MRAE_h = \frac{1}{h} \sum_{i=1}^h |r_i| \quad (1.58)$$

Pro tyto míry může být problematické, pokud chyba referenčního modelu je rovna nule.[12][41]

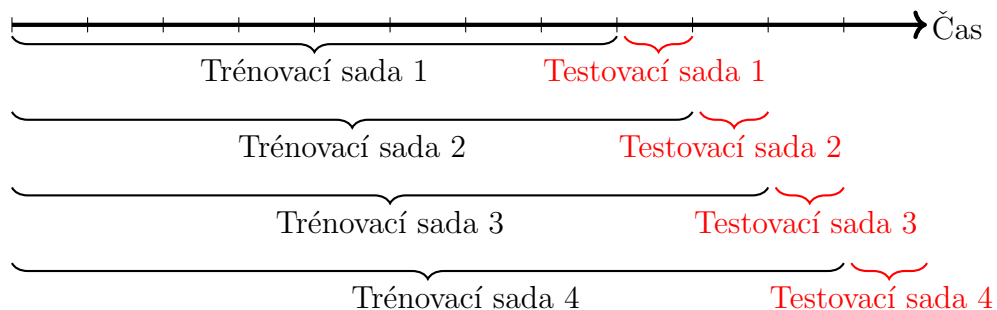
1.5.1 Křížová validace časových řad

Pokud testovací sada není dostatečně velká měření predikční chyby nemusí být příliš přesné a spolehlivé. Jedním z možných řešení je použití určité verze křížové validace pro časové řady. V tomto přístupu se používá mnoho různých trénovacích sad, kde každá obsahuje o jedno pozorování více než předchozí. Postup je zobrazený na obrázku 1.10. Následně je vyhodnocena predikční chyba pro všechny testovací sady a výsledky jsou zprůměrovány napříč sadami. [23]



Obrázek 1.10: Křížová validace časových řad s různou délkou horizontu

Alternativou může být pouze jeden predikční horizont pro každou testovací sadu. Tento postup je zobrazen na obrázku 1.11, kde každá testovací sada obsahuje pouze jedno pozorování. Pak míra odpovídá chybě předpovědi jednoho kroku.



Obrázek 1.11: Křížová validace časových řad se stejnou horizontu

Tento postup se často také nazývá „rolling forecasting origin“ a lze zobecnit i pro více krokovou predikci. Postup křížové validace pak dle Hyndman[22] vypadá následovně

1. Výběr pozorování v čas $k + h + i - 1$, kde k potřebný počet pozorování pro odhad modelu a h je predikční horizont. Použití pozorování z času $1, 2, \dots, k + i - 1$ pro odhad predikčního modelu. Vyhodnocení chyby predikce.
2. Opakování kroku 1. pro $i = 1, 2, \dots, T - k - h + 1$, kde T celkový počet pozorování
3. Výpočet chyby předpovědí založené na získaných chybách.

2. Inlace

Inlace je růst všeobecné cenové úrovně v ekonomice. Tento růst přesahuje růst poptávky po penězích a tím způsobuje pokles kupní síly peněz. Opakem je pokles cenové úrovně zvaný deflace. Ke sledování všeobecné úrovně cen v ekonomice, cenové hladiny, slouží cenové indexy. Nejčastěji používaný cenový index je index spotřebitelských cen (CPI). Cenové indexy poměří úroveň cen vybraného koše reprezentativních výrobků a služeb ve dvou srovnávaných obdobích, přičemž jednotliví reprezentanti jsou váženi jejich podílem na spotřebě domácností. Tento podíl zůstává v rámci cenového indexu konstantní a při jeho revizi je index přepočítán. [36] Kromě CPI je inflace také měřena pomocí indexu cen výrobců. Tento index sleduje změny cen zemědělské produkce, průmyslové produkce, stavebních prací atd. Další možnosti pro zjištění míry inflace je deflátor HDP. Deflátor HDP je definován jako poměr nominálního produktu a reálného produktu v daném roce. Deflátor obsahuje oproti CPI širší koš statků a jeho složení se mění.

CPI vydávané Českým statistickým úřadem je přepočítáváno na harmonizovaný index spotřebitelských cen HICP pro potřeby Eurostatu.

2.0.1 Cílování inflace národními bankami

Cílování inflace se stalo běžnou částí měnové politiky centrálních bank. Jako první jej začala využívat novozélandská Rezervní banka v roce 1990[20]. Záhy se cílování prokázalo jako dobrý nástroj pro stabilizaci jak inflace, tak ekonomiky země [37]. Následně se tedy k Novému Zélandu přidalo více než 20 zemí.

Cílování inflace je charakterizováno oznámením číselného inflačního cíle, implementací měnové politiky s významnou rolí inflačních předpovědí a vysokou mírou transparentnosti. Dále pro úspěšnost je důležitá nezávislost centrální banky s dostatečným mandátem pro stabilizování cen [37]. Neméně důležitá pro plnění inflačního cíle je i důvěryhodnost centrální banky[19].

Inflační cíle se liší mezi státy a to jak v konkrétním cíli, tak i v přípustném intervalu [19]. Například Polská centrální banka cílí na 2.5% s tolerancí ± 1 procentní bod [33]. Česká národní banka na 2% s tolerancí ± 1 procentní bod[6], Maďarská národní banka na 3% s tolerancí ± 1 [5]. Pokud se centrální banka snaží pouze přiblížit inflaci svému cíli jedná se o tzv. striktní cílování inflace[20]. V praxi se ale setkáváme pouze s flexibilním cílováním, kdy se centrální banka kromě stabilizování inflace snaží stabilizovat i ekonomický vývoj, měnový kurz, úrokové míry[37].

3. Inlace v malých otevřených ekonomikách

Malá otevřená ekonomika je taková ekonomika, která je dostatečně malá v porovnání se světovými trhy a to tak, že změny v jejím hospodaření nemění světové ceny ani příjmy. Země tudíž přejímá ceny jiných světových trhů [8]. Malé otevřené ekonomiky jsou silně provázány se svým okolím a vývoj jejich makroekonomických ukazatelů je svázán s vývoje ukazatelů okolních zemí. Toto je třeba zohlednit v rámci modelu přidáním vnějších vlivů jako může být například světová cena ropy, výkon silné sousední země, případně inflace na celo evropské úrovni. Jako představitelé malých otevřených ekonomik byly vybrány Česká republika, Slovensko, Maďarsko, Polsko a Rakousko. Na makroekonomický ukazatelích těchto zemí budou zkoumány predikční vlastnosti VAR, TVAR modelů a neuronových sítí. Nejdříve jsou popsána použitá data a následně je představen jednoduchý autoregresní model představující benchmark pro zbylé modely.

K výpočtům byl použit programovací jazyk R. Všechny napsané skripty a použité balíčky jsou uvedeny v příloze B.

3.1 Data

Použitá data byla získána z databáze OECD ¹. Jedná se o čtvrtletní data za Českou republiku, Maďarsko, Polsko, Slovensko a Rakousko do posledního čtvrtletí roku 2017. Z důvodu chybějících pozorování jsou řady různě dlouhé a s různými počátečními obdobími. Ze stejného důvodu nejsou všechny ukazatele použity pro všechny země. Přehled použitých ukazatelů s kódy databází je uveden v tabulce 3.1. Dále byla v modelech použita časová řada Indexu spotřebitelských cen agregovaných za 28 členských států Evropské unie. Grafy průběhu jednotlivých ukazatelů jsou uvedeny v příloze A.1.

¹Databáze je přístupná na <https://data.oecd.org/>.

Tabulka 3.1: Ukazatele použité z OECD

Proměnná	Databáze	Popis	ID
act_q	STLABOUR	Aktivní populace, starší 15 let, všechny osoby	LFACITTTT
cpi_q	MEI	Index spotřebitelských cen, všechny položky	CPALTT01
int3_q	MEI	Krátkodobá úroková sazba, 3 měsíční	IR3TIB01
ulc_q	MEI	Náklady práce	ULQEUL01
gdp_q	QNA	Hrubý domácí produkt	B1_GS1
imp_q	MEI	Import zboží	XTIMVA01
unemp_q	MEI	Nezaměstnanost, starší 15 let	LFUNTTTT
m1_q	MEI	Peněžní zásoba, M1	MANMM101

3.1.1 Stacionarizace

Všechny použité časové řady jsou nestacionární. Nejdříve je tedy nutné posoudit původ této nestacionarity, přesněji zdali je způsobena pouze deterministickým trendem, případně zdali řada obsahuje jednotkový kořen. Za tímto účelem byl použit Rozšířený Dickey-Fullerův test jednotkového kořene (viz sekce 1.1.4). Výsledky testů pro ukazatele jednotlivých zemí jsou zobrazeny v tabulce 3.2, kde sloupec Odhadnutý model odpovídá modelům (1.7), (1.8) a (1.9). Uvedená kritická hodnota je stanovena pro 5% hladinu významnosti. Žádná z použitých řad není integrována vyššího řádu než 1 a tedy pro stacionarizaci stačí první diference. Stacionarita diferencovaných řad byla opět ověřena pomocí ADF testu a žádná z diferencovaných řad již jednotkový kořen neobsahuje.

Tabulka 3.2: Výsledky Rozšířeného Dickey-Fullerova testu

Země	Ukazatel	Výsledek	Odhadnutý model	Testová statistika	Kritická hodnota
AUT	cpi_q	I(1) s driftem	model2	tau2= 1.36	-2.89
AUT	act_q	I(1) s driftem	model2	tau2= 0.63	-2.89
AUT	emp_q	I(0) s trendem	model3	tau3= -4.66	-3.45
AUT	exp_q	I(1)	model2	phi1= 4.23	4.71
AUT	gdp_q	I(0) s trendem	model3	tau3= -6.34	-3.45
AUT	imp_q	I(1)	model2	phi1= 3.18	4.71
AUT	int3_q	I(1)	model2	phi1= 1.61	4.71
AUT	net_q	I(1)	model2	phi1= 4.24	4.71
AUT	ulc_q	I(0) s trendem	model3	tau3= -4.19	-3.45
AUT	unemp_q	I(1)	model2	phi1= 2.09	4.71
CZE	cpi_q	I(1) s driftem	model2	tau2= -1.1	-2.89
CZE	act_q	I(1)	model2	phi1= 1.33	4.71
CZE	emp_q	I(1)	model2	phi1= 1.47	4.71
CZE	exp_q	I(1)	model2	phi1= 2.11	4.71
CZE	gdp_q	I(1)	model2	phi1= 3.6	4.71
CZE	imp_q	I(1)	model2	phi1= 1.81	4.71

CZE	int3_q	I(0) s trendem	model3	tau3= -7.18	-3.45
CZE	m1_q	I(1) s trendem a driftem	model3	phi2= 20.87	4.88
CZE	m3_q	I(1) s driftem	model2	tau2= 2.12	-2.89
CZE	net_q	I(1)	model2	phi1= 1.53	4.71
CZE	ulc_q	I(0) s trendem	model3	tau3= -4.92	-3.45
CZE	unemp_q	I(1)	model2	phi1= 0.31	4.71
EU28	cpi_q	I(1) s trendem a driftem	model3	phi2= 26.03	4.88
HUN	cpi_q	I(1) s driftem	model2	tau2= -2.17	-2.89
HUN	act_q	I(1) s driftem	model2	tau2= 1.32	-2.89
HUN	emp_q	I(1)	model2	phi1= 1.54	4.71
HUN	exp_q	I(1)	model2	phi1= 1.36	4.71
HUN	gdp_q	I(0) s trendem	model3	tau3= -4.83	-3.45
HUN	imp_q	I(1)	model2	phi1= 2.34	4.71
HUN	m1_q	I(0) s trendem	model3	tau3= 4.04	-3.45
HUN	m3_q	I(1) s driftem	model2	tau2= -0.28	-2.89
HUN	net_q	I(1)	model2	phi1= 1.47	4.71
HUN	ulc_q	I(0) s trendem	model3	tau3= -4.54	-3.45
HUN	unemp_q	I(1)	model2	phi1= 0.24	4.71
POL	cpi_q	I(1) s driftem	model2	tau2= -1.13	-2.89
POL	act_q	I(1)	model2	phi1= 1.41	4.71
POL	emp_q	I(1)	model2	phi1= 1.71	4.71
POL	exp_q	I(1) s driftem	model2	tau2= -1.98	-2.89
POL	gdp_q	I(0) s trendem	model3	tau3= -6.62	-3.45
POL	imp_q	I(1)	model2	phi1= 3.42	4.71
POL	int3_q	I(0) s trendem	model3	tau3= -4.35	-3.45
POL	m1_q	I(0)	model2	tau2= 2.98	-2.89
POL	m3_q	I(1) s trendem a driftem	model3	phi2= 17.9	4.88
POL	net_q	I(1)	model2	phi1= 1.51	4.71
POL	ulc_q	I(0) s trendem	model3	tau3= -4.38	-3.45
POL	unemp_q	I(1)	model2	phi1= 3.22	4.71
SVK	cpi_q	I(1) s driftem	model2	tau2= -2.86	-2.89
SVK	act_q	I(1) s driftem	model2	tau2= -1.57	-2.89
SVK	emp_q	I(1)	model2	phi1= 1.08	4.71
SVK	exp_q	I(1)	model2	phi1= 1.48	4.71
SVK	gdp_q	I(0) s trendem	model3	tau3= -4.62	-3.45
SVK	imp_q	I(1)	model2	phi1= 1.32	4.71
SVK	net_q	I(0) s trendem	model3	tau3= -3.92	-3.45
SVK	ulc_q	I(0) s trendem	model3	tau3= -5.91	-3.45
SVK	unemp_q	I(1)	model2	phi1= 0.51	4.71

Pro časové řady bez jednotkového kořene s deterministickým trendem byl do modelů přidán tento trend. Tímto způsobem je nestacionarita dostatečně očištěna a lze řady považovat za stacionarizované.

3.2 Benchmark model

Velikost predikční chyby vyjádřené střední čtvercovou chybou je závislá na jednotkách měření a na rozptylu zkoumané časové řady. K porovnání predikční chyby jednotlivých modelů a k vyhodnocení jejich kvality je třeba její úroveň vztáhnout k predikční chybě základního modelu. K tomuto složí odhad benchmark modelu, který byl zvolen jako jednorozměrný autoregresní model ve tvaru:

$$y_t = \alpha + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \cdots + \beta_p y_{t-p} + \epsilon_t \quad (3.1)$$

kde α, β jsou koeficienty a p je počet zpoždění. Počet zpoždění byl zvolen dle Akaikeho informačního kritéria((1.19)).

Pro vyhodnocení chyby předpovědi cpi_q daným modelem byla použita střední čtvercová chyba pro predikci s horizontem 10 kroků (MSE10) a chyba s klouzavým horizontem 3 kroky (MSE3) (obdobný postup je zobrazen na obrázku 1.11). Počáteční horizont MSE3 byl volen tak, aby se výpočet zopakoval 10 krát a výsledná chyba je tedy průměrem těchto opakování. Na rozdíl od postupu uvedeného v sekci „Křížová validace časových řad“ (1.5.1) se s horizontem posouvá i počátek trénovací sady. Odhad je tedy vždy tvořen pomocí stejného počtu pozorování a u výsledné chyby není nutné zohledňovat rozdíly v délkách trénovacích sad. Výsledky jednorozměrných AR modelů, dle (3.1) jsou i s p -hodnotami diagnostických testů uvedeny v tabulce 3.3.

Tabulka 3.3: Odhady jednorozměrného AR modelu bez exogenní proměnné

Země	Lag	BG (pval)	ARCH (pval)	JB (pval)	MSE	MSE10	MSE3
CZE	8	0.4195	0.1501	0.0008	0.307	0.237	0.266
SVK	6	0.9519	0.6847	0.0000*	0.655	0.628	0.406
HUN	3	0.6947	0.4701	0.3684	0.325	0.486	0.679
POL	5	0.8728	0.6526	0.1854	0.286	0.359	0.419
AUT	5	0.2379	0.7632	0.5415	0.108	0.103	0.108

Dále byl odhadnut druhý benchmark AR model s exogenní proměnnou ve tvaru

$$y_t = \alpha + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \cdots + \beta_p y_{t-p} + \gamma_1 x_t + \epsilon_t \quad (3.2)$$

,kde α, β, γ jsou koeficienty, x_t je exogenní proměnná CPI za 28 zemí EU. Výsledky druhého modelu a p -hodnoty jsou uvedeny v tabulce 3.4.

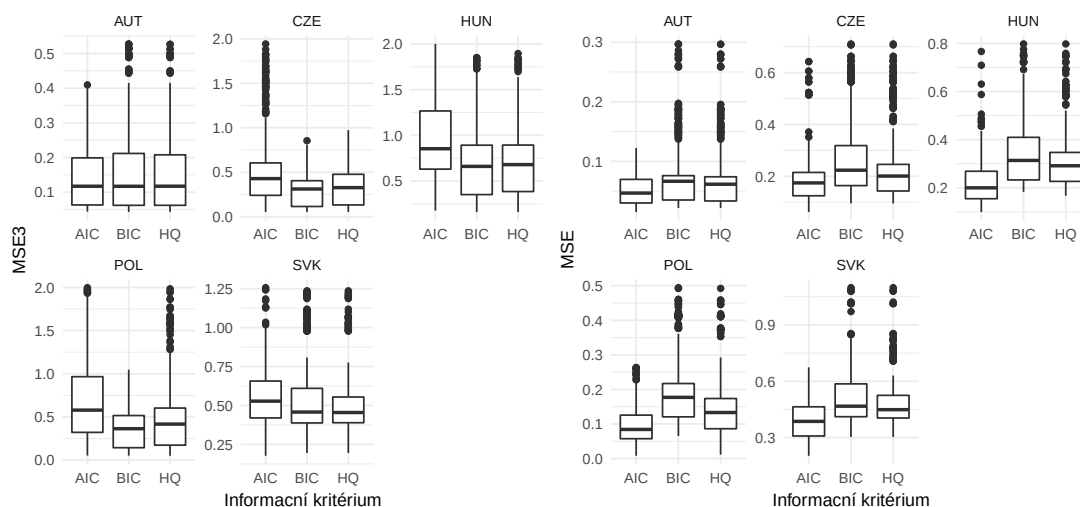
Tabulka 3.4: Odhady AR modelu s exogenní proměnnou

Země	Lag	BG (pval)	ARCH (pval)	JB (pval)	MSE	MSE10	MSE3
CZE	8	0.82658	0.8557	0.00004	0.239	0.380	0.083
SVK	6	0.97138	0.5582	0.00000*	0.575	0.149	0.243
HUN	3	0.58692	0.5506	0.51319	0.391	0.398	0.501
POL	5	0.60677	0.7967	0.38826	0.146	0.662	0.131
AUT	5	0.77117	0.0001	0.00001	0.070	0.049	0.046

3.3 Odhad VAR modelu

VAR modely byly odhadnuty pro všechny možné kombinace indexu spotřebitelských cen s dalšími proměnnými. Konkrétně kombinace byly tvořeny tak, aby model obsahoval dvě, tři a čtyři časové řady. Například pro Českou republiku tak bylo odhadnuto 63 kombinací. Pro každou kombinaci byla odhadnuta sada modelů s různým počtem zpoždění a s různými deterministickými členy. Počet zpoždění byl vybrán dle informačních kritérií, AIC(1.19), BIC(1.20) a HQ (1.21). Pro každé vybrané zpoždění byl odhadnut samostatný model. Protože jinak stejné modely jsou odhadnuty pro všechna doporučená zpoždění je možné, alespoň orientačně, porovnat jejich střední čtvercovou chybu (MSE) a jejich predikční chyby (MSE3) (viz obrázek 3.1). Z porovnání se potvrzuje předpoklad (viz sekce 1.2), že AIC a HQ mají tendenci vést k nižší chybě odhadnutého modelu, a naopak BIC má tendenci vést k nižší predikční chybě.

Dále do modelů byla přidána umělá proměnná nabývající hodnoty jedna v roce 2008 a nula pro zbylá období. Tato proměnná by měla pomoci očistit odhady od výkyvu způsobeného celosvětovou ekonomickou krizí.



Obrázek 3.1: Porovnání chyby pro modely se zpožděním dle různých informačních kritérií (omezeno pro $MSE3 < 2$, $MSE < 2$)

Jako další deterministické členy byly do modelu přidány umělé sezónní proměnné a trendová složka. Všechny modely byly odhadnuty ve verzi s deterministickými členy a ve verzi bez nich. Jako exogenní proměnná byla do modelů přidána úroveň Indexu spotřebitelských cen za 28 zemí Evropské unie. Pro tvorbu skutečných předpovědí by bylo nutné tuto proměnnou nahradit za některou z profesionálních předpovědí. Případně jí odhadovat samostatným modelem. Celkově pro všech pět zemí bylo odhadnuto 8073 VAR modelů.

3.3.1 Odhad Česká republika

Pro Českou republiku bylo celkově odhadnuto 1984 VAR modelů. Pět modelů s nejnižší hodnotou MSE3 je uvedeno v tabulkách 3.5 a 3.6². Chyby modelů s exogenní proměnnou jsou výrazně nižší než chyby modelů bez ní. Dokonce jako nejlepší se dle MSE3 jeví VAR(4) model (viz obrázek 3.2) obsahující pouze proměnné *cpi_q* a *gdp_q*. Tento model vyhovuje i dle diagnostických testů, která na pěti procentní hladině významnosti nezamítají nulové hypotézy o neautokorelaci, homoskedasticitě náhodné složky a normalitě jejího rozdělení. K testování jsou použity vícerozměrné verze Brausch-Godfrey testu, ARCH testu a Jarque-Bera testu. V tabulkách jsou uvedeny *p*-hodnoty těchto testů. Všechny zmíněné model dosahují nižší chyby MSE3 oproti benchmark modelům.

Tabulka 3.5: Odhad VAR model Česká republika - s exogenní proměnou EU28

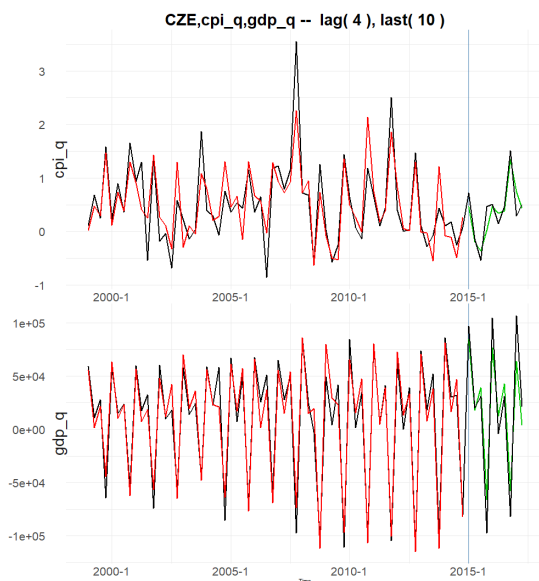
	Model	Seas. dummy	Cons.	Trend	Lag	BG (pval)	ARCH (pval)	JB (pval)	MSE	MSE10	MSE3	IC
1	<i>cpi_q,gdp_q</i>	Ano	Ano	Ano	4	0.0614	0.5889	0.0516	0.145	0.058	0.054	HQ,BIC
2	<i>cpi_q,imp_q,int3_q,unemp_q</i>	Ne	Ne	Ne	2	0.0512	0.3893	0.0000*	0.140	0.057	0.055	HQ
6	<i>cpi_q,act_q,imp_q,ulc_q</i>	Ne	Ano	Ano	1	0.0088	0.8228	0.0000*	0.162	0.070	0.057	AIC,HQ
7	<i>cpi_q,imp_q</i>	Ne	Ne	Ne	5	0.1295	0.2691	0.2032	0.131	0.052	0.057	AIC,HQ
8	<i>cpi_q,imp_q,int3_q</i>	Ne	Ne	Ne	1	0.1252	0.1256	0.0000*	0.176	0.063	0.057	BIC

Model s nejnižší chybou MSE3 bez exogenní proměnné obsahuje *cpi_q*, *int_3*, *ulc_q*, *unemp_q* (viz obrázek 3.3). Tento model by mohl být vhodný pro tvorbu predikcí i přestože nesplňuje předpoklady neautokorelovanosti chybové složky a její normality. Alternativou by mohl být VAR(5) model s proměnnými *cpi_q*, *int_3*, *unemp_q*, který je dle diagnostických testů validní a jeho predikční chyba není výrazně vyšší. Na obrázcích 3.2 a 3.3 jsou zobrazeny vyrovnané hodnoty modelů (červená barva) a predikce na 10 období dopředu (zelená barva).

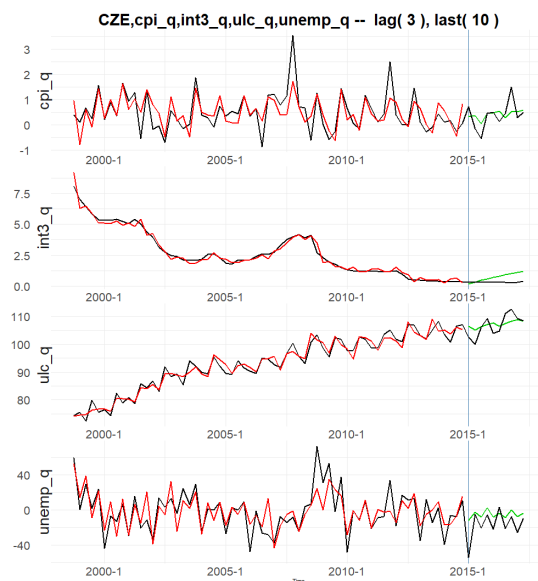
Tabulka 3.6: Odhad VAR model Česká republika - bez exogenní proměnou EU28

	Model	Seas. dummy	Cons.	Trend	Lag	BG (pval)	ARCH (pval)	JB (pval)	MSE	MSE10	MSE3	IC
1	<i>cpi_q,int3_q,ulc_q,unemp_q</i>	Ano	Ne	Ne	3	0.0000*	0.6802	0.0000*	0.245	0.196	0.206	BIC
3	<i>cpi_q,imp_q,int3_q,unemp_q</i>	Ano	Ne	Ne	5	0.0001	0.5462	0.1931	0.242	0.321	0.210	AIC
4	<i>cpi_q,imp_q,ulc_q,unemp_q</i>	Ano	Ano	Ne	3	0.0001	0.9516	0.0000	0.250	0.198	0.221	BIC
7	<i>cpi_q,unemp_q</i>	Ano	Ne	Ne	5	0.0050	0.1752	0.2583	0.344	0.444	0.233	AIC,HQ,BIC
8	<i>cpi_q,int3_q,unemp_q</i>	Ano	Ne	Ne	5	0.0882	0.3817	0.8404	0.271	0.362	0.237	AIC,HQ

²Pro každou kombinaci proměnných je uvedena pouze ta s nejnižší hodnotu MSE3. Tak aby tabulka obsahovala pouze unikátní kombinace proměnných. Pořadí odpovídající skutečnému je uvedeno v prvním sloupci tabulky.



Obrázek 3.2: Odhad VAR(4) cpi_q, gdp_q s predikcí na 10 období dopředu



Obrázek 3.3: Odhad VAR(3) $cpi_q, int3_q, ulc_q, unemp_q$ s predikcí na 10 období dopředu

3.3.2 Odhad Slovensko

Pro Slovensko bylo celkově odhadnuto 768 modelů. Model s nejnižší MSE3 s exogenní proměnou je VAR(5) model (viz obrázek 3.4) obsahující pouze cpi_q a $unemp_q$. Jarque-Berův test zamítá nulovou hypotézu o normalitě náhodné složky a model dle tohoto testu nesplňuje předpoklad normality. Tento test je však relativně citlivý na odlehlá pozorování. Pomocí Grangerova testu lze ověřit, že existuje kauzalita v Grangerově smyslu mezi proměnnými $unemp_q$ a cpi_q . V opačném směru již tato kauzalita není. Informace v cpi_q tedy nezlepšuje předpovědi $unemp_q$. Predikční chyby modelu je výrazně nižší než u benchmark modelu s EU28.

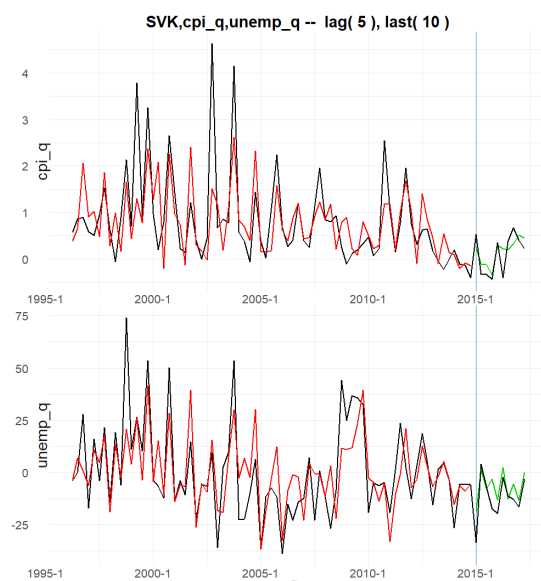
Tabulka 3.7: Odhad VAR model pro Slovensko - s exogenní proměnnou EU28

Model	Seas. dummy	Cons.	Trend	Lag	BG (pval)	ARCH (pval)	JB (pval)	MSE	MSE10	MSE3	IC
1 $cpi_q, unemp_q$	Ano	Ne	Ano	5	0.1230	0.1618	0.0000*	0.486	0.079	0.066	HQ,BIC
3 cpi_q, act_q, gdp_q	Ne	Ne	Ne	5	0.1645	0.0689	0.0000*	0.420	0.080	0.067	AIC,HQ,BIC
4 $cpi_q, act_q, unemp_q$	Ano	Ne	Ne	5	0.2981	0.7175	0.0000*	0.454	0.068	0.067	HQ,BIC
7 $cpi_q, ulc_q, unemp_q$	Ano	Ne	Ne	8	0.0027	0.0031	0.0000*	0.350	0.100	0.074	AIC
8 $cpi_q, act_q, gdp_q, ulc_q$	Ano	Ano	Ne	5	0.0002	0.0167	0.0000*	0.400	0.054	0.075	AIC,HQ,BIC

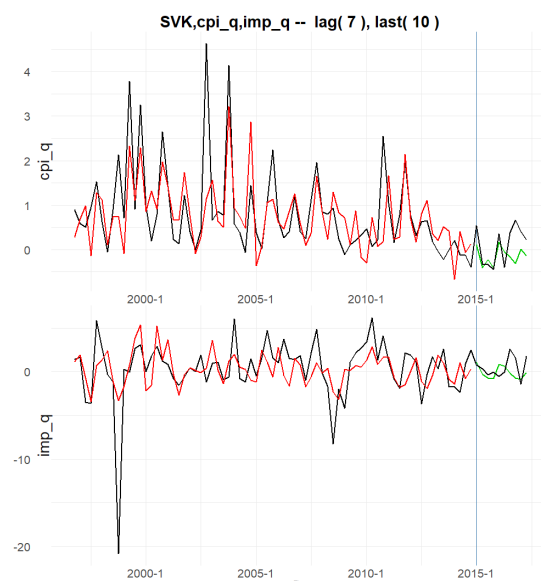
Model s nejnižší MSE3 bez exogenní proměnné je VAR(7) model (viz obrázek 3.5) obsahující pouze cpi_q a imp_q . U tohoto modelu opět vícerozměrná verze Jarque-Bera testu zamítá nulovou hypotézu o normalitě náhodné složky. Zbylé modely však nesplňují ani podmínku nekorelovanosti náhodné složky.

Tabulka 3.8: Odhad VAR model Slovensko - bez exogenní proměnné EU28

	Model	Seas. dummy	Cons.	Trend	Lag	BG (pval)	ARCH (pval)	JB (pval)	MSE	MSE10	MSE3	IC
1	cpi_q,imp_q	Ano	Ne	Ne	7	0.5886	0.9999	0.0000*	0.520	0.191	0.177	AIC
3	cpi_q,act_q,imp_q	Ano	Ne	Ne	1	0.0015	0.9308	0.0000*	1.083	0.186	0.195	HQ,BIC
5	cpi_q,act_q	Ano	Ne	Ne	1	0.0001	0.4967	0.0000*	1.098	0.192	0.200	HQ,BIC
6	cpi_q,act_q,imp_q,unemp_q	Ano	Ne	Ne	1	0.0001	0.5231	0.0000*	1.079	0.187	0.205	HQ,BIC
7	cpi_q,imp_q,unemp_q	Ano	Ne	Ne	1	0.0000	0.0708	0.0000*	1.079	0.187	0.206	HQ,BIC



Obrázek 3.4: Odhad VAR(4) cpi_q, act_q s predikcí na 10 období dopředu



Obrázek 3.5: Odhad VAR(7) cpi_q, imp_q s predikcí na 10 období dopředu

3.3.3 Odhad Maďarsko

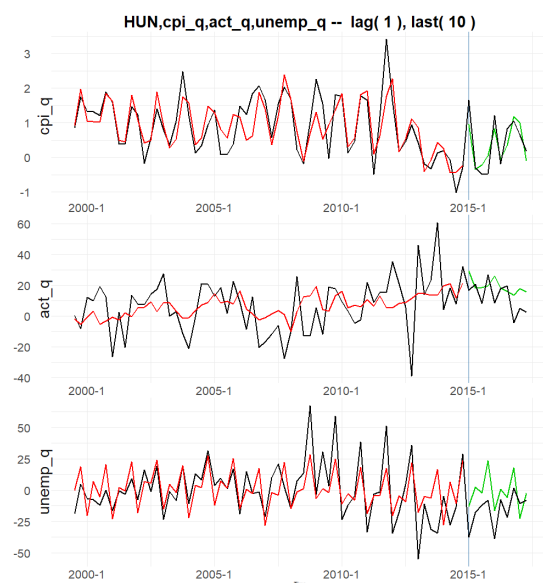
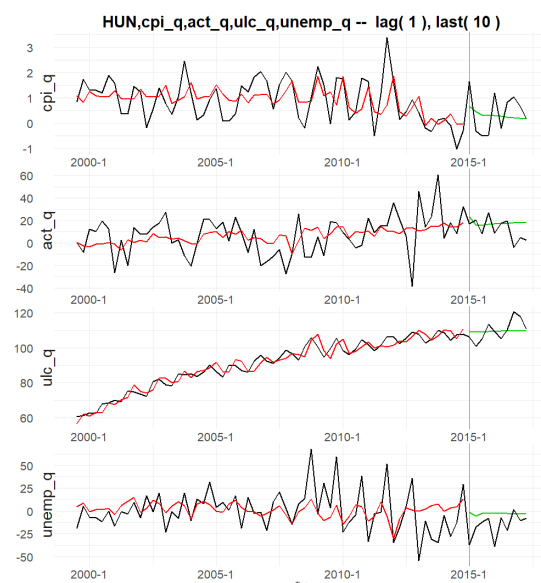
Pro Maďarsko bylo celkem odhadnuto 1408 modelů. Z nich nejnižší predikční chyby dosáhl VAR(1) model s proměnnými $cpi_q, act_q, unemp_q$ (viz obrázek 3.6). Výrazný rozdíl mezi predikčními chybami s proměnou $EU28$ a bez ní ukazuje, že maďarská inflace je silně závislá na té evropské. Ale i modelům bez exogenní proměnné se daří dosáhnout nižší predikční chyby oproti benchmark modelu bez $EU28$.

Tabulka 3.9: Odhad VAR model pro Maďarsko - s exogenní proměnnou EU28

	Model	Seas. dummy	Cons.	Trend	Lag	BG (pval)	ARCH (pval)	JB (pval)	MSE	MSE10	MSE3	IC
1	cpi_q,act_q,unemp_q	Ne	Ano	Ano	1	0.1104	0.0020	0.0005	0.220	0.144	0.161	HQ,BIC
2	cpi_q,act_q,ulc_q,unemp_q	Ne	Ano	Ne	1	0.0052	0.2506	0.0029	0.220	0.155	0.162	BIC
9	cpi_q,act_q,m1_q,unemp_q	Ne	Ano	Ne	1	0.0360	0.1683	0.0000*	0.222	0.174	0.175	HQ,BIC
10	cpi_q,m1_q,unemp_q	Ne	Ano	Ne	1	0.2749	0.2075	0.0000*	0.225	0.184	0.176	HQ,BIC
11	cpi_q,act_q,imp_q,unemp_q	Ne	Ne	Ne	1	0.1224	0.0887	0.2094	0.222	0.166	0.178	AIC,HQ,BIC

Tabulka 3.10: Odhad VAR model Maďarsko - bez exogenní proměnnou EU28

	Model	Seas. dummy	Cons.	Trend	Lag	BG (pval)	ARCH (pval)	JB (pval)	MSE	MSE10	MSE3	IC
1	cpi_q,act_q,ulc_q,unemp_q	Ano	Ne	Ano	1	0.0000*	0.1709	0.0062	0.483	0.506	0.545	BIC
2	cpi_q,imp_q,ulc_q,unemp_q	Ano	Ne	Ano	1	0.0000*	0.3244	0.6326	0.451	0.538	0.562	BIC
3	cpi_q,act_q,unemp_q	Ano	Ano	Ano	1	0.0001	0.0510	0.0026	0.489	0.496	0.562	HQ,BIC
4	cpi_q,act_q,imp_q,unemp_q	Ano	Ano	Ano	1	0.0010	0.1132	0.1209	0.456	0.531	0.573	AIC,HQ,BIC
5	cpi_q,imp_q,unemp_q	Ano	Ano	Ano	1	0.0005	0.1661	0.4866	0.461	0.525	0.575	HQ,BIC


 Obrázek 3.6: Odhad VAR(7) cpi_q , imp_q , $unemp_q$ s predikcí na 10 období dopředu

 Obrázek 3.7: Odhad VAR(1) cpi_q , act_q , ulc_q , $unemp_q$ s predikcí na 10 období dopředu

3.3.4 Odhad Polsko

Pro Polsko bylo odhadnuto 2480 modelů. Obdobně jako u ostatních zemí i pro Polsko predikce tvořené modely s exogenní proměnnou *EU28* mají výrazně nižší predikční chyby MSE10, MSE3. Model s nejnižší chybou je VAR(5) s proměnnými $cpi_q, ulc_q, unemp_q$ (viz obrázek 3.8). Tento model nesplňuje podmínku nekorelovanosti náhodné složky. Naopak model VAR(5) cpi_q, ulc_q je dle všech diagnostických testů validní a jeho predikční chyba MSE3 se příliš neliší.

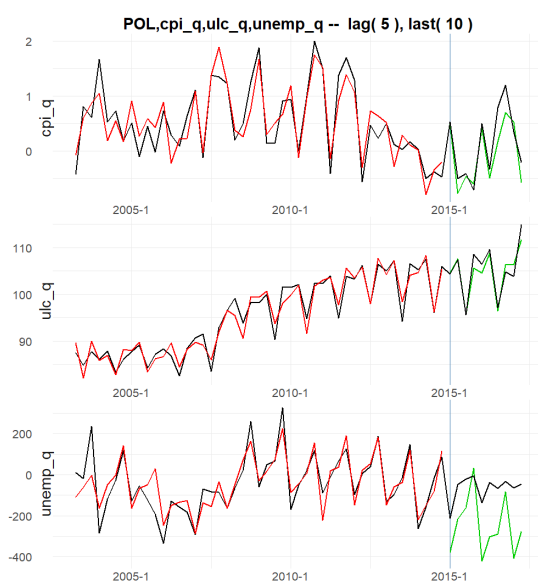
Tabulka 3.11: Odhad VAR model pro Polsko - s exogenní proměnnou EU28

	Model	Seas. dummy	Cons.	Trend	Lag	BG (pval)	ARCH (pval)	JB (pval)	MSE	MSE10	MSE3	IC
1	cpi_q,ulc_q,unemp_q	Ano	Ne	Ano	5	0.0065	0.6532	0.5406	0.079	0.092	0.050	HQ
2	cpi_q,gdp_q,imp_q,ulc_q	Ano	Ne	Ano	4	0.0000*	0.4135	0.8139	0.074	0.027	0.052	BIC
3	cpi_q,gdp_q,ulc_q,unemp_q	Ano	Ne	Ano	4	0.0000*	0.4790	0.2485	0.081	0.061	0.053	BIC
4	cpi_q,ulc_q	Ano	Ne	Ano	5	0.0791	0.8426	0.3376	0.092	0.123	0.055	AIC,HQ,BIC
5	cpi_q,int3_q,m1_q	Ne	Ano	Ne	2	0.2305	0.3536	0.0000*	0.135	0.090	0.055	HQ

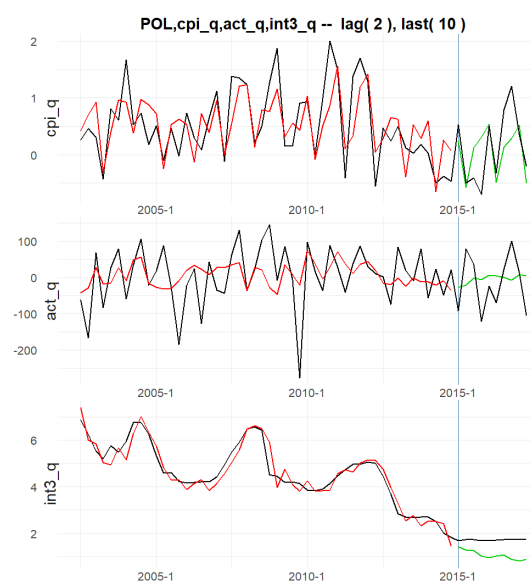
V rámci modelů bez exogenního členu je nejlepší model VAR(2) bez sezonních proměnných obsahující pouze proměnné $cpi_q, act_q, int3_q$. Dle obrázků 3.8 a 3.9 se oběma modelům daří vhodně predikovat tvar cpi_q v horizontu 10 kroků. V případě proměnných $unemp_q$ a $int3_q$ se však v roce 2015 výrazně změnit jejich průběh a predikci se nedaří tuto změnu postihnout.

Tabulka 3.12: Odhad VAR model Polsko - bez exogenní proměnné EU28

	Model	Seas. dummy	Cons.	Trend	Lag	BG (pval)	ARCH (pval)	JB (pval)	MSE	MSE10	MSE3	IC
1	$cpi_q, act_q, int3_q$	Ne	Ne	Ne	2	0.0001	0.0512	0.0000*	0.211	0.282	0.289	HQ, BIC
2	$cpi_q, act_q, imp_q, int3_q$	Ne	Ne	Ne	2	0.0017	0.4120	0.0000*	0.202	0.272	0.299	HQ
4	$cpi_q, act_q, int3_q, unemp_q$	Ne	Ne	Ne	1	0.0092	0.3412	0.0000*	0.215	0.282	0.308	BIC
6	$cpi_q, imp_q, int3_q$	Ne	Ne	Ne	2	0.0266	0.0588	0.0001	0.203	0.272	0.315	HQ
7	$cpi_q, imp_q, m1_q, unemp_q$	Ne	Ne	Ano	4	0.0161	0.3184	0.5655	0.132	0.326	0.315	HQ



Obrázek 3.8: Odhad VAR(5) $cpi_q, ulc_q, unemp_q$ s predikcí na 10 období dopředu



Obrázek 3.9: Odhad VAR(2) $cpi_q, act_q, int3_q$ s predikcí na 10 období dopředu

3.3.5 Odhad Rakousko

Pro Rakousko bylo odhadnuto 1433 VAR modelů. Všechny uvedené modely s exogenní proměnou dosahují výrazně nižší predikční chyby MSE3 oproti benchmark modelům. U modelů bez exogenní proměnné vykazuje lepší hodnoty pouze model s proměnnými $cpi_q, act_q, imp_q, int3_q$ (viz obrázek 3.11). Rezi-dua tohoto modelu jsou autokorelovaná a nemají normální rozdělení a dle testu Grangerovy kauzality ani jedna z těchto proměnných nepůsobí na cpi_q ve smyslu Grangerovy kauzality. I přesto tyto výsledky tento model dosahuje v rámci VAR modelů nejnižší předpovědní chyby MSE3. Pro VAR(5) model s proměnnými $cpi_q, gdp_q, imp_q, ulc_q$, mezi kterými již existují vzájemné vztahy ve smyslu

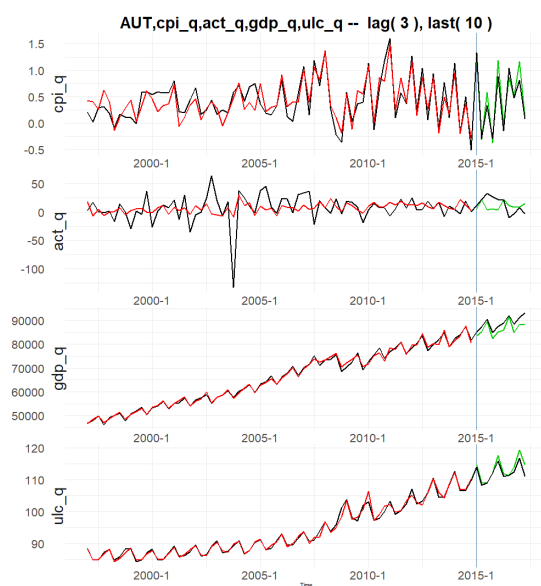
Grangerovy kauzality, a který je dle diagnostických testů validní, by MSE3 vycházel 0.129. Chyba by tedy byla výrazně vyšší nežli u benchmark modelu bez EU28.

Tabulka 3.13: Odhad VAR model pro Rakousko - s exogenní proměnnou EU28

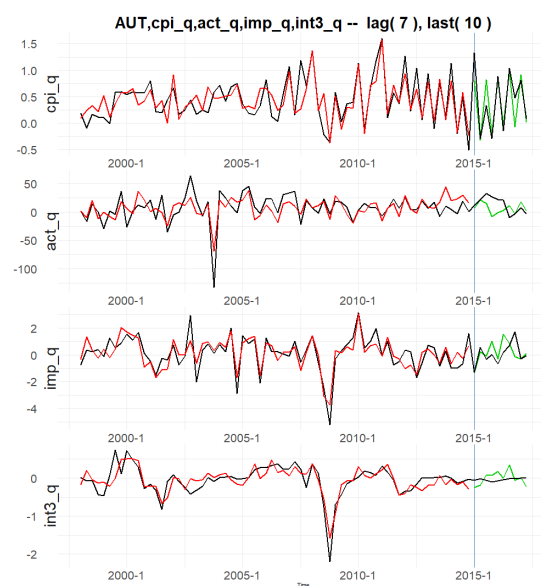
	Model	Seas. dummy	Cons.	Trend	Lag	BG (pval)	ARCH (pval)	JB (pval)	MSE	MSE10	MSE3	IC
1	cpi_q,act_q,gdp_q,ulc_q	Ano	Ne	Ano	3	0.0000	0.2459	0.0000*	0.039	0.033	0.043	BIC
2	cpi_q,act_q,unemp_q	Ne	Ne	Ano	4	0.0017	0.3511	0.0000*	0.040	0.033	0.043	HQ
3	cpi_q,act_q,gdp_q,unemp_q	Ano	Ano	Ano	5	0.0038	0.3763	0.0000*	0.023	0.050	0.043	HQ,BIC
6	cpi_q,act_q,gdp_q,imp_q	Ano	Ano	Ano	7	0.0040	0.4760	0.0000*	0.020	0.074	0.044	AIC
10	cpi_q,gdp_q,unemp_q	Ne	Ano	Ano	5	0.0026	0.3632	0.7064	0.025	0.055	0.045	HQ,BIC

Tabulka 3.14: Odhad VAR model Rakousko - bez exogenní proměnné EU28

	Model	Seas. dummy	Cons.	Trend	Lag	BG (pval)	ARCH (pval)	JB (pval)	MSE	MSE10	MSE3	IC
1	cpi_q,act_q,imp_q,int3_q	Ano	Ano	Ne	7	0.0000*	0.0957	0.0000*	0.060	0.085	0.096	AIC
5	cpi_q,act_q,unemp_q	Ano	Ne	Ne	7	0.0077	0.8869	0.0000*	0.099	0.181	0.117	AIC
6	cpi_q,act_q	Ne	Ne	Ne	5	0.2098	0.7289	0.0000*	0.101	0.204	0.117	AIC,HQ
7	cpi_q,imp_q	Ne	Ano	Ne	6	0.0428	0.3851	0.0495	0.076	0.104	0.121	AIC
12	cpi_q,act_q,imp_q	Ne	Ano	Ne	4	0.0030	0.4881	0.0000	0.084	0.089	0.123	AIC,BIC



Obrázek 3.10: Odhad VAR(3) cpi_q , act_q , gdp_q , ulc_q s predikcí na 10 období dopředu

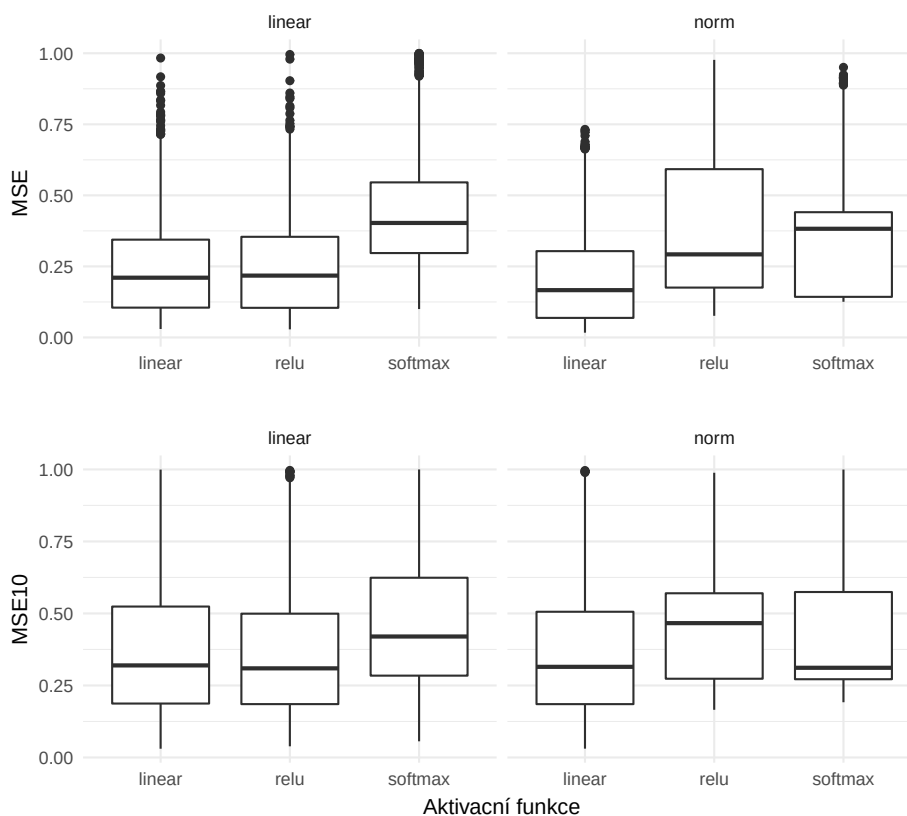


Obrázek 3.11: Odhad VAR(7) cpi_q , act_q , imp_q , $int3_q$ s predikcí na 10 období dopředu

3.4 Odhad Neuronové sítě

K odhadu vícevrstvých feedforward neuronových sítí byl použit Keras s knihovnou TensorFlow (viz sekce 1.4.5). Odhad neuronových sítí již nevychází ze všech možných kombinací proměnných, ale pouze z těch, které se prokázaly jako vhodné v rámci VAR modelů, případně existuje-li mezi proměnnými Grangerova kauzalita. I přestože bylo použito méně kombinací, celkově bylo pro všechny země odhadnuto 16849 modelů. Pro každou kombinaci byly odhadnuty verze s různým normalizačním přístupem, s různou aktivační funkcí, s různým počtem zpoždění a různými deterministickými členy.

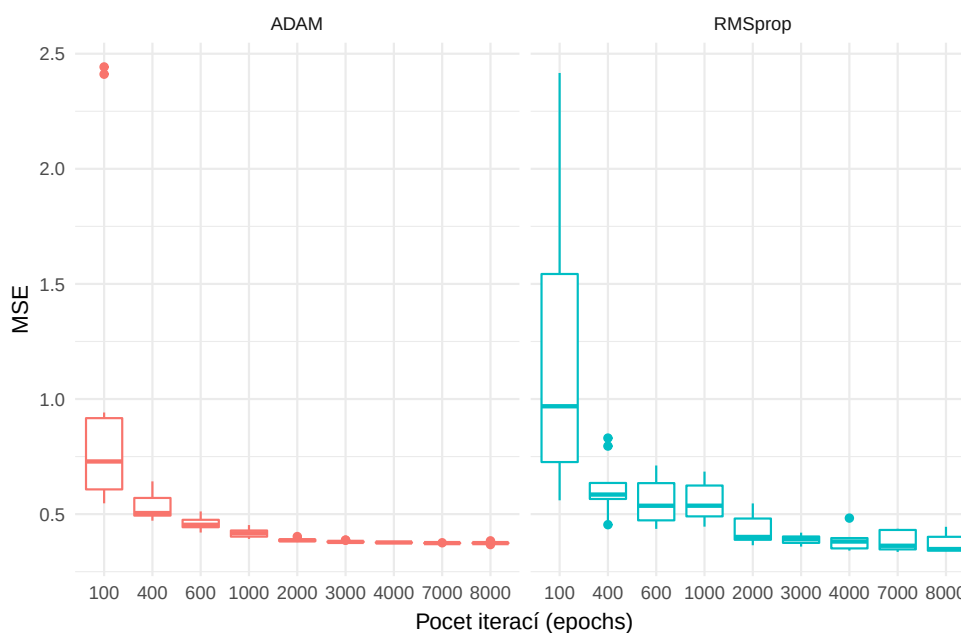
K normalizaci dat byla použita lineární transformace (1.46) a normování (1.44). Jako aktivační funkce byly použity softmax (1.43), relu (1.41) a lineární funkce (1.40). Dále byla použita regularizace vah a regularizace aktivačních funkcí. Oba tyto přístupy by měli zamezit přeučení modelů. Regularizační konstanta λ byla nastavena na hodnotu 0.01. Takto hodnota byla získána vyhodnocením řady různých hodnot v rozmezí od 0.5 do 0.001 na několika modelech za všechny země. Vybrána byla taková hodnota, která pro nejvíce modelů minimalizovala střední čtvercovou chybu pro predikční horizonty 3, 5, 10. Tuto hodnotu však nelze brát jako optimální a pro další zlepšení predikčních vlastností by měla být zvolena pro každý odhadovaný model samostatně.



Obrázek 3.12: Porovnání aktivačních funkcí a normalizačních přístupů

Na obrázku 3.12 je zobrazeno porovnání aktivačních funkcí a normalizačních přístupů. Pro tuto úlohu lineární funkce, bez ohledu na normalizační přístup, umožňuje v průměru lépe minimalizovat chybu sítě. To však nemusí znamenat, že je vhodná pro všechny použité modely.

K optimalizaci byla použita metoda ADAM [24], která v rámci této úlohy na rozdíl od metod RSMprop[18] a SGD³ poskytuje stabilnější výsledky. Neboli chyba sítě s rostoucím počtem iterací dostatečně rychle klesá. Odhad neuronové sítě začíná náhodnými vahami, které jsou následně iteračně upravovány pro minimalizaci nákladové funkce. Pokud je tedy síť odhadována opakovaně a počet iterací není dostatečný pro nalezení absolutního minima výsledná chyba se bude lišit v závislosti na počátečních vahách. Vhodné je pak použít takovou metodu a takový počet iterací, aby chyba sítě měla minimální rozptyl. Na grafu 3.13 je zobrazen opakovaná optimalizace pomocí metod ADAM a RMSprop. Dosažení absolutního minima je značně výpočetně, časově náročné, a to zejména pro zpomalení výpočtu v okolí optima. Pro odhad všech modelů bylo použito 4000 iterací, které dle provedených experimentů s modely dosahují dostatečně nízkého rozptylu. Z důvodu výpočetní náročnosti byly modely nejdříve vyhodnoceny pouze dle MSE10 a následně 50 modelů s nejnižšími hodnotami MSE10 bylo odhadnuto i pro MSE3. Výpočet MSE3 totiž vyžaduje, aby každý model byl znovu 10krát odhadnut. Tento postup může vést k opomenutí některých kvalitních modelů, ale MSE10 se na těchto datech ukázalo jako dostatečný indikátor predikčního potenciálu.



Obrázek 3.13: Porovnání optimalizačních metod ADAM a RMSprop

³SGD - Stochastic gradient descent optimizer

3.4.1 Odhad Česká republika

Pro Českou republiku bylo odhadnuto 3456 modelů. V tabulkách 3.15 a 3.16 jsou opět zobrazeny modely s nejnižší predikční chybou MSE3 a unikátní kombinací proměnných. Všechny vybrané modely používají lineární transformaci a oba model s minimální MSE3 jako aktivační funkci používají softmax a mají tři skryté vrstvy. Dále všechny modely s exogenní proměnnou obsahují regularizaci vah. Model odhadnutý neuronovou sítí s proměnnými cpi_q , imp_q , $unemp_q$ dosahuje podobné predikční chyby jako VAR(4) model. I přestože na neuronovou síť nejsou kladeny žádné předpoklady je možné zkoumat neautokorelovanost a normalitu náhodné složky. Jednorozměrný Braushe-Paganův test nezamítá na p -hodnotě 0.6296 nulovou hypotézu o neautokorelovanosti náhodné složky pro odhad cpi_q . Jednorozměrná verze Jarque-Bera test však zamítá na p -hodnotě 0.0006 hypotézu o normálním rozdělení náhodné složky.

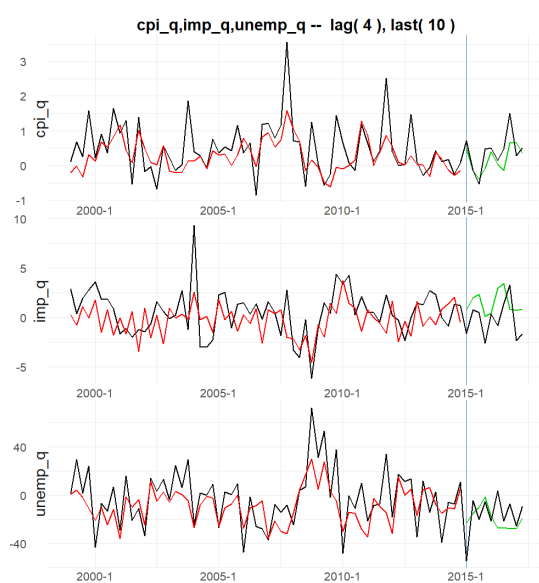
Tabulka 3.15: Odhad NN model Česká republika - s exogenní proměnnou EU28

	Model	Norm.	Seas.	Regul. (act. fun)	Regul. (weights)	Lags	Activ. Fun.	Hidden Units	Hidden Layers	MSE	MSE10	MSE3
1	$cpi_q, imp_q, unemp_q$	linear	Ne	Ne	Ano	4	softmax	15	3	0.379	0.125	0.058
2	$cpi_q, imp_q, int3_q, unemp_q$	linear	Ne	Ne	Ano	4	softmax	30	1	0.602	0.132	0.059
3	cpi_q, gdp_q	linear	Ne	Ne	Ano	6	linear	20	3	0.170	0.126	0.162
4	$cpi_q, gdp_q, imp_q, ulc_q$	linear	Ne	Ne	Ano	6	linear	15	3	0.141	0.115	0.184
8	$cpi_q, gdp_q, ulc_q, unemp_q$	linear	Ne	Ne	Ano	4	linear	15	3	0.171	0.109	0.233

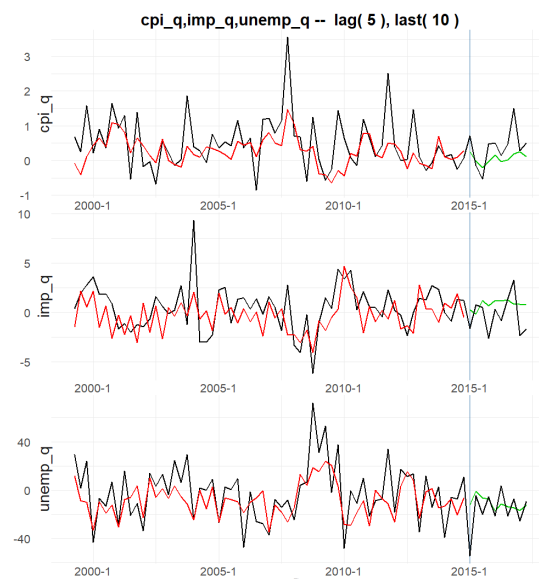
Model bez exogenní proměnné obsahuje regularizaci vah a stejné proměnné jako model s exogenní proměnnou, ale liší se v počtu zpoždění a v počtu skrytých jednotek. Predikční chyba MSE3 je u toho modelu nižší než u VAR modelu. Celkově predikční chyba modelů České republiky bez *EU28* je nižší než u VAR modelů.

Tabulka 3.16: Odhad NN model Česká republika - bez exogenní proměnné EU28

	Model	Norm.	Seas.	Regul. (act. fun)	Regul. (weights)	Lags	Activ. Fun.	Hidden Units	Hidden Layers	MSE	MSE10	MSE3
1	$cpi_q, imp_q, unemp_q$	linear	Ne	Ne	Ano	5	softmax	30	3	0.427	0.163	0.156
2	$cpi_q, int3_q, ulc_q, unemp_q$	linear	Ano	Ne	Ano	4	linear	20	1	0.318	0.154	0.339
4	$cpi_q, gdp_q, ulc_q, unemp_q$	linear	Ne	Ne	Ano	6	linear	20	1	0.283	0.136	0.557
5	$cpi_q, gdp_q, imp_q, ulc_q$	linear	Ne	Ne	Ne	6	linear	20	3	0.256	0.136	0.561
7	$cpi_q, gdp_q, ulc_q, unemp_q$	linear	Ne	Ne	Ano	5	relu	15	1	0.216	0.132	0.622



Obrázek 3.14: Odhad Neuronové sítě cpi_q , imp_q , $unemp_q$ s predikcí na 10 období dopředu



Obrázek 3.15: Odhad Neuronové sítě cpi_q , imp_q , $unemp_q$ s predikcí na 10 období dopředu

3.4.2 Odhad Slovensko

Predikční chyba MSE3 u modelů s exogenní proměnnou je opět vyšší oproti VAR modelům. Naopak podobně jako pro data za Českou republiku predikční chyba modelů bez exogenní proměnné je nižší u neuronových sítí. Všechny modely s exogenní proměnnou používají lineární transformaci a oba vybrané modely používají regularizaci vah. Z obrázku 3.16 je patrné, že zvolenému modelu se příliš dobře nedaří vyrovnávat hodnoty časové řady a dosahuje i relativně vysoké chyby MSE. To je způsobeno převážně tím, že model je vybírán na základě MSE3 a nikoliv na základě MSE. Modelu se tedy sice nedaří vysvětlit podstatnou část variability původní řady, ale relativně se mu dobře daří zachytit tu část variability, která udává budoucí chování.

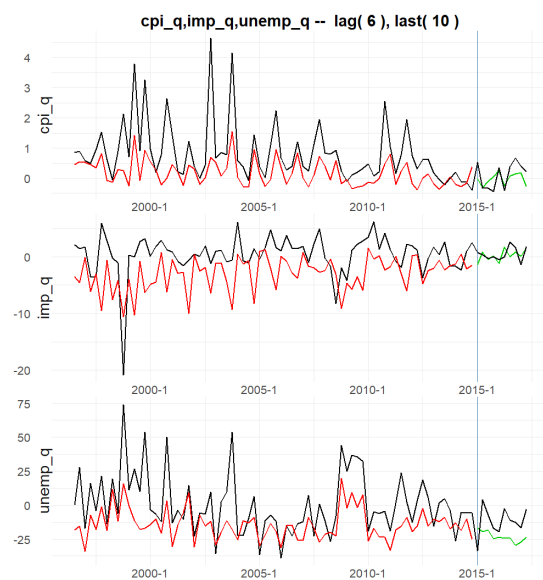
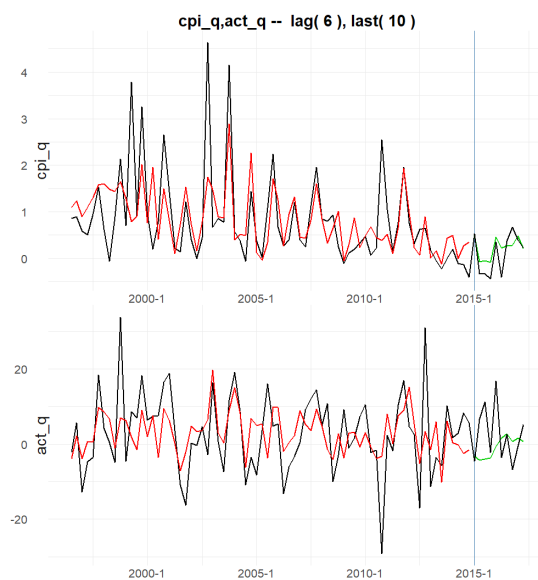
Tabulka 3.17: Odhad nn model Slovensko - s exogenní proměnnou EU28

	Model	Norm.	Seas.	Regul. (act. fun)	Regul. (weights)	Lags	Activ. Fun.	Hidden Units	Hidden Layers	MSE	MSE10	MSE3
1	cpi_q,imp_q,unemp_q	linear	Ne	Ne	Ano	6	softmax	30	3	0.769	0.120	0.072
4	cpi_q,ulc_q,unemp_q	linear	Ne	Ne	Ano	5	relu	30	3	0.456	0.111	0.158
8	cpi_q,gdp_q,ulc_q,unemp_q	linear	Ne	Ne	Ano	5	relu	15	3	0.457	0.102	0.182
6	cpi_q,act_q	linear	Ne	Ne	Ano	6	relu	15	3	0.571	0.123	0.206
12	cpi_q,act_q,imp_q,unemp_q	linear	Ne	Ne	Ne	5	linear	15	3	0.565	0.111	0.260

V tabulce 3.18 s výsledky modelů bez exogenní proměnné jsou uvedeny pouze tři modely. V celkových výsledcích do 16. místa se objevovali pouze tyto tři kombinace proměnných s různým rozměrem a nastavením sítě.

Tabulka 3.18: Odhad nn model Slovensko - bez exogenní proměnné EU28

	Model	Norm.	Seas.	Regul. (act. fun)	Regul. (weights)	Lags	Activ. Fun.	Hidden Units	Hidden Layers	MSE	MSE10	MSE3
1	cpi_q,act_q	norm	Ne	Ne	Ano	6	linear	15	3	0.582	0.069	0.174
11	cpi_q,gdp_q,imp_q,ulc_q	norm	Ne	Ne	Ano	6	linear	15	3	0.441	0.122	0.282
16	cpi_q,act_q,imp_q,unemp_q	norm	Ne	Ne	Ne	6	linear	20	3	0.366	0.124	0.383


 Obrázek 3.16: Odhad Neuronové sítě *cpi_q*, *imp_q*, *unemp_q* s predikcí na 10 období dopředu

 Obrázek 3.17: Odhad Neuronové sítě *cpi_q*, *act_q*, s predikcí na 10 období dopředu

3.4.3 Odhad Maďarsko

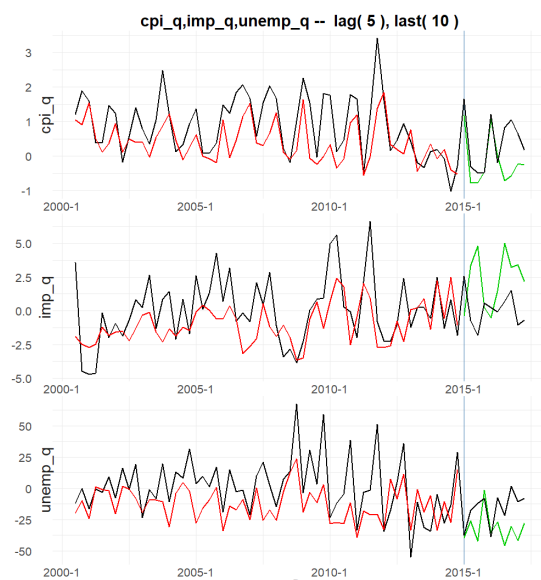
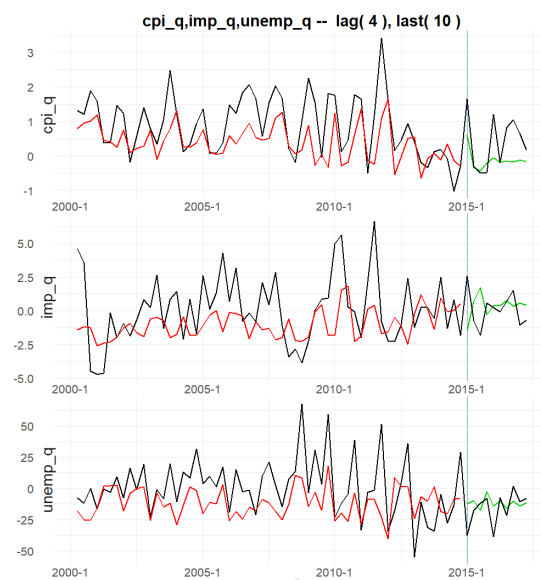
Celkově pro Maďarsko bylo odhadnuto 3888 modelů. Výsledek neuronových sítí je velice podobný výsledku VAR modelů. Obdobně jako pro Českou republiku oba modely s nejnižší chybou používají lineární transformaci a aktivizační funkci softmax. Pro model s exogenní proměnou neuronové sítě dosahují lehce nižší chyby. Naopak u modelů bez exogenních se predikční chyba blíží chybě předpovědi pomocí aritmetického průměru. I přesto jsou predikce stále lepší než u benchmark modelu bez *EU28*.

Tabulka 3.19: Odhad nn model Maďarsko - s exogenní proměnnou EU28

	Model	Norm.	Seas.	Regul. (act. fun)	Regul. (weights)	Lags	Activ. Fun.	Hidden Units	Hidden Layers	MSE	MSE10	MSE3
1	cpi_q,imp_q,unemp_q	linear	Ne	Ne	Ne	5	softmax	15	3	0.566	0.294	0.155
2	cpi_q,act_q,unemp_q	linear	Ne	Ne	Ano	5	softmax	20	3	0.523	0.262	0.312
11	cpi_q,ulc_q,unemp_q	norm	Ne	Ne	Ano	6	linear	30	3	0.159	0.273	0.373
14	cpi_q,gdp_q,ulc_q,unemp_q	norm	Ne	Ne	Ano	5	linear	20	3	0.175	0.314	0.418
15	cpi_q,m1_q,unemp_q	linear	Ne	Ne	Ne	4	softmax	15	1	0.315	0.287	0.451

Tabulka 3.20: Odhad nn model Maďarsko - bez exogenní proměnné EU28

	Model	Norm.	Seas.	Regul. (act. fun)	Regul. (weights)	Lags	Activ. Fun.	Hidden Units	Hidden Layers	MSE	MSE10	MSE3
1	cpi_q,imp_q,unemp_q	linear	Ne	Ne	Ne	5	softmax	20	1	0.522	0.214	0.641
5	cpi_q,gdp_q,imp_q,ulc_q	linear	Ne	Ne	Ne	5	linear	15	1	0.321	0.287	0.728
6	cpi_q,m1_q,unemp_q	norm	Ne	Ne	Ne	6	linear	20	3	0.310	0.333	1.083
8	cpi_q,act_q,unemp_q	norm	Ne	Ne	Ne	6	linear	30	3	0.300	0.322	1.458


 Obrázek 3.18: Odhad Neuronové sítě *cpi_q*, *imp_q*, *unemp_q* s predikcí na 10 období dopředu

 Obrázek 3.19: Odhad Neuronové sítě *cpi_q*, *imp_q*, *unemp_q* s predikcí na 10 období dopředu

3.4.4 Odhad Polsko

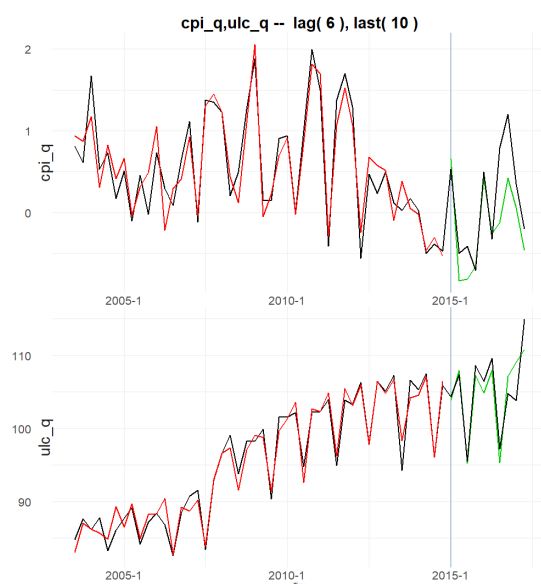
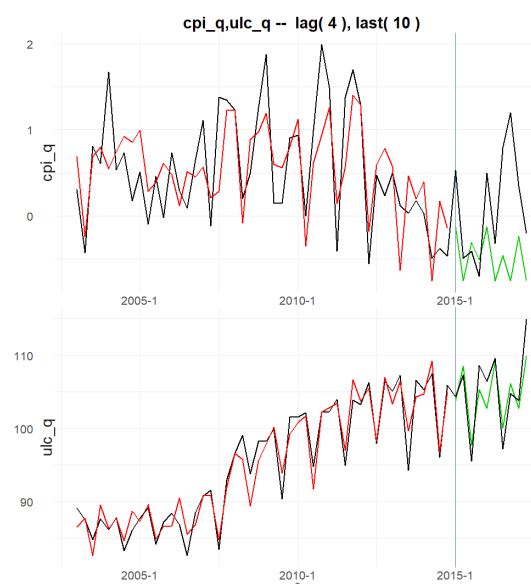
Všechny modely s exogenní proměnou používají lineární aktivační funkce. Model s nejnižší chybou obsahuje proměnné *cpi_q*, *ulc_q* a tři skryté vrstvy po 30 jednotkách a data jsou normována. Predikční chyba tohoto modelu je lehce vyšší než u všech vybraných VAR modelů, ale stále výrazně nižší oproti benchmark modelům.

Tabulka 3.21: Odhad nn model Polsko - s exogenní proměnnou EU28

	Model	Norm.	Seas.	Regul. (act. fun)	Regul. (weights)	Lags	Activ. Fun.	Hidden Units	Hidden Layers	MSE	MSE10	MSE3
1	cpi_q,ulc_q	norm	Ne	Ne	Ne	6	linear	30	3	0.041	0.127	0.068
2	cpi_q,ulc_q,unemp_q	linear	Ne	Ne	Ne	5	linear	30	3	0.038	0.169	0.082
3	cpi_q,gdp_q,ulc_q,unemp_q	linear	Ne	Ne	Ano	6	linear	30	1	0.045	0.161	0.084
8	cpi_q,gdp_q,imp_q,ulc_q	norm	Ne	Ne	Ano	6	linear	20	3	0.021	0.179	0.119

Tabulka 3.22: Odhad nn model Polsko - bez exogenní proměnné EU28

	Model	Norm.	Seas.	Regul. (act. fun)	Regul. (weights)	Lags	Activ. Fun.	Hidden Units	Hidden Layers	MSE	MSE10	MSE3
1	cpi_q,ulc_q	linear	Ne	Ne	Ano	4	relu	20	1	0.187	0.236	0.283
2	cpi_q,gdp_q,imp_q,ulc_q	linear	Ne	Ne	Ano	5	linear	15	3	0.133	0.231	0.335
4	cpi_q,ulc_q,unemp_q	linear	Ne	Ne	Ano	6	relu	30	1	0.125	0.229	0.376
17	cpi_q,imp_q,unemp_q	linear	Ne	Ne	Ano	6	linear	30	3	0.130	0.223	0.468


 Obrázek 3.20: Odhad Neuronové sítě *cpi_q*, *ulc_q* s predikcí na 10 období dopředu

 Obrázek 3.21: Odhad Neuronové sítě *cpi_q*, *ulc_q* s predikcí na 10 období dopředu

3.4.5 Odhad Rakousko

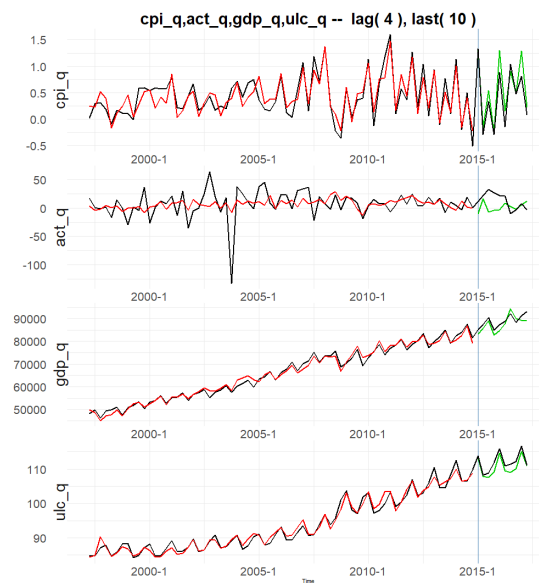
Neuronové sítě pro Rakousko mají nižší predikční chybu oproti VAR modelům pro modely s exogenní proměnnou. Oba modely, podobně jako u ostatních zemí, obsahují stejné proměnné *cpi_q*, *act_q*, *gdp_q*, *ulc_q* a liší se zejména v počtu zpoždění, skrytých vrstev a skrytých jednotek. Obdobně jako v případě VAR modelu je predikční chyba modelu bez exogenní proměnné vysoká a v tomto případě dokonce přesahuje chybu benchmark modelu. Naopak NN model s exogenní proměnnou překonává predikce jak benchmark modelů, tak VAR modelů.

Tabulka 3.23: Odhad nn model Rakousko - s exogenní proměnnou EU28

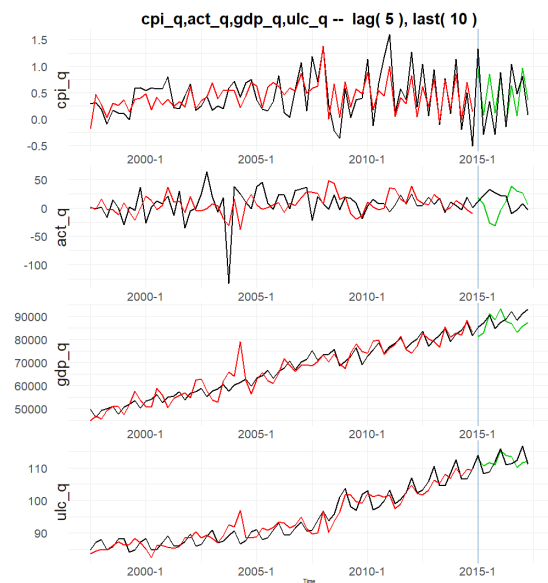
	Model	Norm.	Seas.	Regul. (act. fun)	Regul. (weights)	Lags	Activ. Fun.	Hidden Units	Hidden Layers	MSE	MSE10	MSE3
1	cpi_q,act_q,gdp_q,ulc_q	linear	Ne	Ne	Ano	4	linear	30	3	0.027	0.030	0.037
2	cpi_q,gdp_q,ulc_q,unemp_q	norm	Ne	Ne	Ne	4	linear	20	3	0.023	0.031	0.041
3	cpi_q,gdp_q,imp_q,ulc_q	linear	Ne	Ne	Ne	4	linear	30	3	0.026	0.040	0.048
4	cpi_q,ulc_q,unemp_q	linear	Ne	Ne	Ano	4	relu	30	1	0.035	0.042	0.129

Tabulka 3.24: Odhad nn model Rakousko - bez exogenní proměnné EU28

	Model	Norm.	Seas.	Regul. (act. fun)	Regul. (weights)	Lags	Activ. Fun.	Hidden Units	Hidden Layers	MSE	MSE10	MSE3
1	cpi_q,act_q,gdp_q,ulc_q	linear	Ne	Ne	Ne	5	linear	15	1	0.156	0.046	0.157
2	cpi_q,ulc_q,unemp_q	linear	Ne	Ne	Ne	6	linear	15	1	0.092	0.055	0.166
3	cpi_q,act_q,unemp_q	norm	Ne	Ne	Ne	5	linear	20	1	0.099	0.059	0.217
4	cpi_q,imp_q,int3_q,ulc_q	norm	Ne	Ne	Ano	4	linear	30	1	0.072	0.056	0.359



Obrázek 3.22: Odhad Neuronové sítě *cpi_q*, *act_q*, *gdp_q*, *ulc_q* s predikcí na 10 období dopředu



Obrázek 3.23: Odhad Neuronové sítě *cpi_q*, *act_q*, *gdp_q*, *ulc_q* s predikcí na 10 období dopředu

3.5 Odhad TVAR modelu

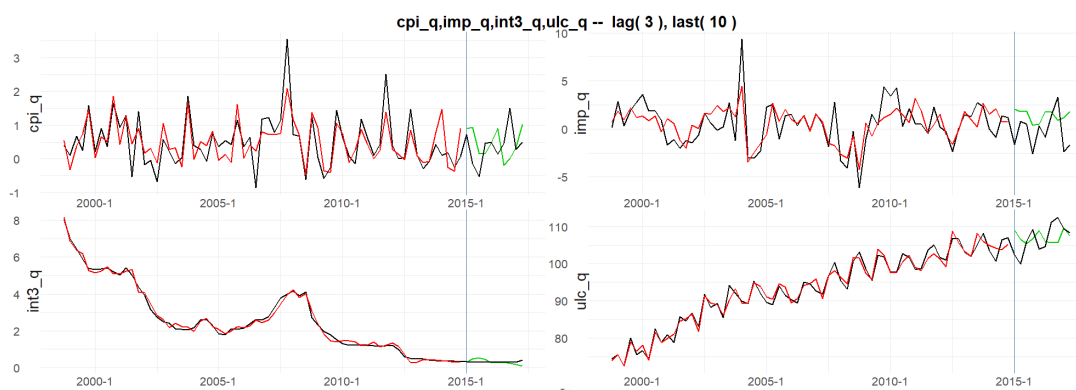
Obdobně jako u VAR modelu byly TVAR modely odhadnuty na všech možných kombinacích proměnných. Na rozdíl od předchozích dvou modelovacích přístupů implementace TVAR modelů v R balíčku `tsDyn`⁴ neumožňuje vložení exogenních proměnných. Z tohoto důvodu byly pro TVAR odhadnuty pouze modely bez Indexu spotřebitelských cen za 28 zemí EU. Počty zpoždění byly opět voleny dle informačních kritérií AIC, HQ, BIC a pro každé byl odhadnut samostatný model. Dále všechny kombinace zpoždění a proměnných byly odhadnuty pro jeden a dva prahy. Predikční chyba byla opět vyhodnocena pomocí chyby s klouzavým horizontem MSE3 a chyby s horizontem 10 kroků MSE10. Celkově bylo odhadnuto 1758 TVAR modelů.

3.5.1 Odhad za Českou republiku

Model TVAR s nejnižší predikční chybou MSE3 pro Českou republiku má dva režimy a tři zpoždění obsahuje proměnné *cpi-q*, *imp-q*, *int3-q*, *ulc-q* (viz obrázek 3.24). Predikční chyba tohoto modelu je nižší pouze oproti benchmark modelu, který dosahuje MSE3 0.266 a výrazně vyšší oproti VAR modelu a neuronovým sítím.

Tabulka 3.25: Odhad TVAR modelu pro Českou republiku

Model	Cons.	Trend	Num. Thresholds	Lag	MSE	MSE10	MSE3
<i>cpi-q,imp-q,int3-q,ulc-q</i>	Ano	Ne	1	3	0.167	0.532	0.236
<i>cpi-q,act-q,int3-q,ulc-q</i>	Ano	Ano	1	1	0.385	0.279	0.239
<i>cpi-q,int3-q,m1-q</i>	Ne	Ne	1	4	0.272	0.433	0.258
<i>cpi-q,m1-q</i>	Ne	Ne	1	4	0.390	0.290	0.274
<i>cpi-q,int3-q,m1-q,ulc-q</i>	Ano	Ano	1	1	0.419	0.291	0.280



Obrázek 3.24: Odhad TVAR(2,3) *cpi-q*, *imp-q*, *int3-q*, *ulc-q* s predikcí na 10 období dopředu

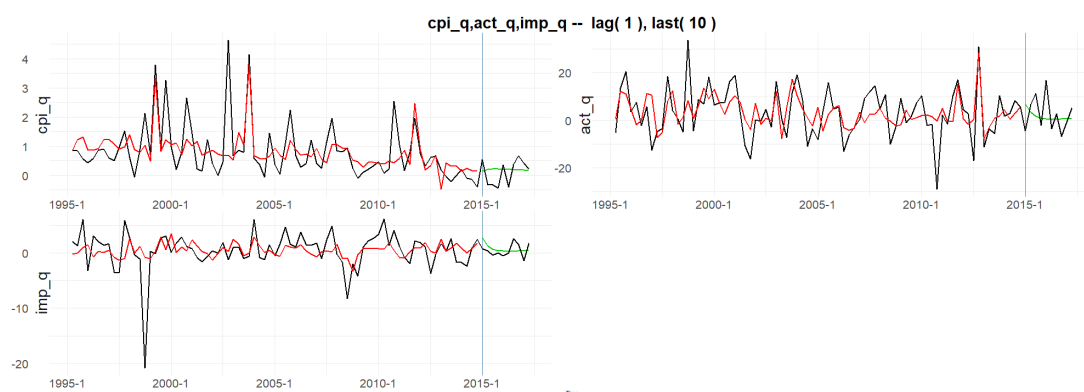
⁴Balíček `tsDyn` je dostupný z CRAN <https://cran.r-project.org/web/packages/tsDyn/>

3.5.2 Odhad Slovensko

TVAR model pro Slovensko s proměnnými cpi_q , act_q , imp_q obsahuje tři režimy a pouze jedno zpoždění. Dosahuje výrazně nižší predikční chyby pouze oproti benchmark modelu.

Tabulka 3.26: Odhad TVAR modelu pro Slovensko

Model	Cons.	Trend	Num. Thresholds	Lag	MSE	MSE10	MSE3
cpi_q, act_q, imp_q	Ano	Ano	2	1	0.533	0.186	0.242
cpi_q, act_q	Ano	Ano	1	1	0.702	0.189	0.244
cpi_q, imp_q	Ano	Ne	2	4	0.449	5.486	0.272
$cpi_q, act_q, gdp_q, imp_q$	Ne	Ano	2	1	0.672	1.487	0.294



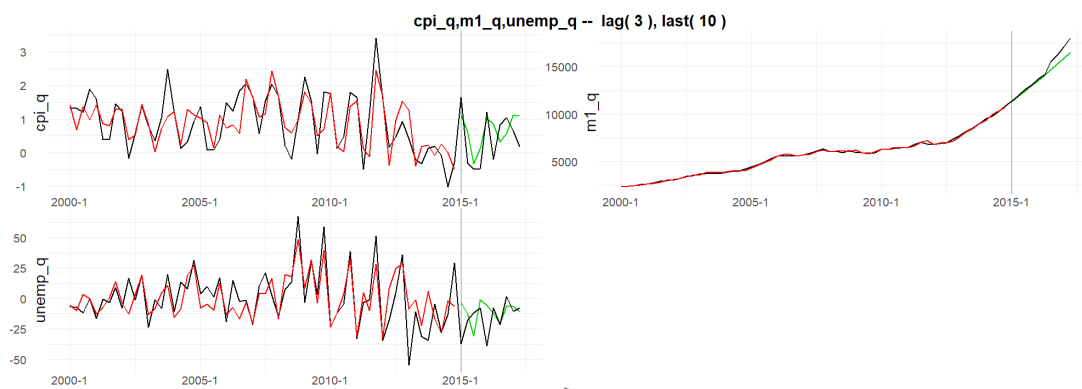
Obrázek 3.25: Odhad TVAR(3,1) cpi_q , act_q , imp_q s predikcí na 10 období dopředu

3.5.3 Odhad Maďarsko

Model s nejnižší predikční chybou pro Maďarsko je TVAR(2,3) s proměnnými cpi_q , $m1_q$, $unemp_q$. Predikční chyby tohoto modelu je nižší než u benchmark modelu.

Tabulka 3.27: Odhad TVAR modelu pro Maďarsko

Model	Cons.	Trend	Num. Thresholds	Lag	MSE	MSE10	MSE3
$cpi_q, m1_q, unemp_q$	Ne	Ano	1	3	0.373	0.414	0.660
cpi_q, act_q	Ano	Ne	1	4	0.371	0.348	0.666
$cpi_q, m1_q, unemp_q$	Ano	Ne	1	1	0.494	0.601	0.687
$cpi_q, m1_q$	Ne	Ne	1	4	0.400	0.592	0.743
cpi_q, act_q, imp_q	Ano	Ano	1	1	0.526	0.643	0.757



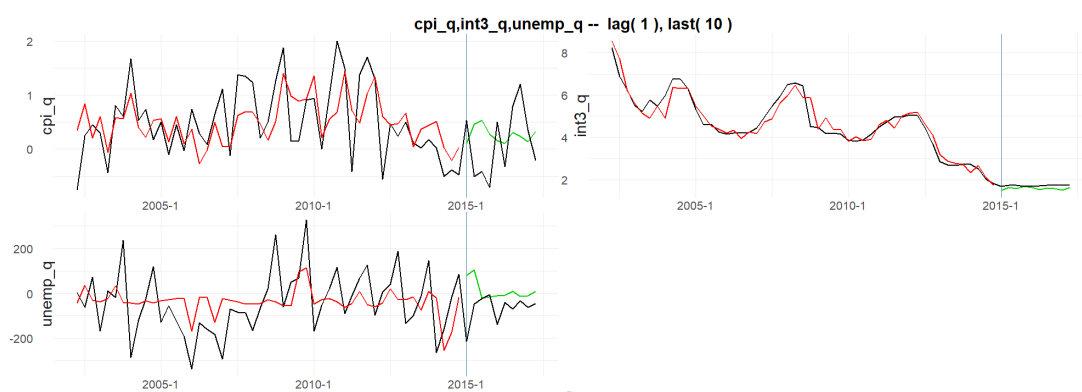
Obrázek 3.26: Odhad TVAR(2,3) cpi_q , $m1_q$, $unemp_q$ s predikcí na 10 období dopředu

3.5.4 Odhad Polsko

Pro Polskou inflaci má nejnižší MSE3 model TVAR(2,1) s dvěma režimy a proměnnými cpi_q , $int3_q$, $unemp_q$.

Tabulka 3.28: Odhad TVAR modelu pro Polsko

Model	Cons.	Trend	Num. Tresholds	Lag	MSE	MSE10	MSE3
$cpi_q, int3_q, unemp_q$	Ne	Ano	1	1	0.295	0.477	0.377
$cpi_q, imp_q, int3_q$	Ne	Ne	1	2	0.245	0.326	0.384
$cpi_q, imp_q, int3_q$	Ne	Ne	1	1	0.415	0.375	0.387
cpi_q, act_q	Ano	Ne	2	1	0.360	0.530	0.403
cpi_q, imp_q	Ne	Ne	1	1	0.491	0.392	0.404



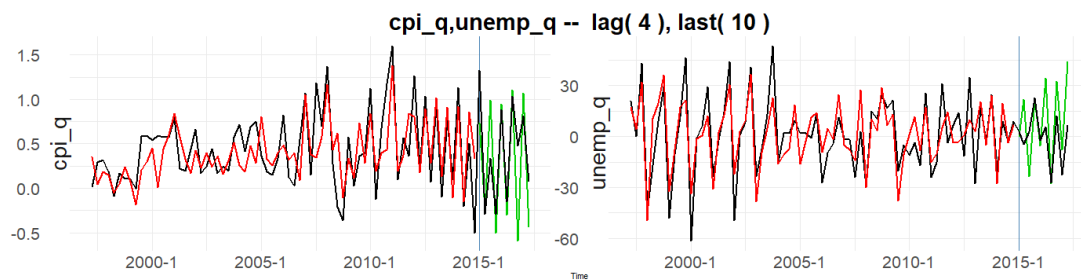
Obrázek 3.27: Odhad TVAR(2,1) cpi_q , $int3_q$, $unemp_q$ s predikcí na 10 období dopředu

3.5.5 Odhad Rakousko

Pro Rakousko model TVAR(2,1) pouze s proměnnými cpi_q , $unemp_q$. Chyba MSE3 tohoto modelu je vyšší než u benchmark a VAR modelu, ale je nižší oproti NN.

Tabulka 3.29: Odhad TVAR modelu pro Rakousko

Model	Cons.	Trend	Num. Tresholds	Lag	MSE	MSE10	MSE3
cpi_q,unemp_q	Ne	Ano	1	4	0.101	0.235	0.154
cpi_q,int3_q	Ano	Ne	1	4	0.107	0.293	0.166
cpi_q,ulc_q	Ano	Ano	1	5	0.062	0.732	0.171
cpi_q,act_q	Ne	Ne	1	5	0.106	0.201	0.179
cpi_q,act_q,int3_q	Ano	Ne	1	4	0.099	0.212	0.180



Obrázek 3.28: Odhad TVAR(2,1) cpi_q , $unemp_q$ s predikcí na 10 období dopředu

Závěr

Pro pět malých otevřených ekonomik byly použity vícerozměrné modely VAR, TVAR a neuronové sítě pro predikci inflace vyjádřené spotřebitelským indexem cen. Modely VAR a neuronové sítě byly odhadnuty ve verzi s exogenní proměnnou a ve verzi bez ní. TVAR modely byly odhadnuty pouze bez exogenní proměnné. Pro vyhodnocení predikční chyby byla použita křížová validace ve formě klouzavého horizontu s predikcí na 3 kroky dopředu a pro nastavení dolní příčky byly použity benchmark modely v podobě jednoduchých AR modelů. Všem vícerozměrným modelům se podařilo překonat predikční chyby odpovídajících benchmark modelů s výjimkou predikcí pro Rakousko bez exogenní proměnné, kde uspěl pouze VAR model.

V modelech bez exogenní proměnné pro Českou republiku, Slovensko a Polsko poskytly nejlepší predikce neuronové sítě. Pro Maďarsko a Rakousko byly predikce získané VAR modely lepší. Predikce TVAR modelů byly pro všechny země překonány jak neuronovými sítěmi, tak VAR modely. V modelech s exogenní proměnnou jsou výsledky přesně obráceně. Nejlepší predikce pro Českou republiku, Slovensko a Polsko poskytují VAR modely a neuronové sítě lépe predikují pro Maďarsko a Rakousko. Z porovnání VAR a neuronových sítí nevychází žádný vítěz. V rámci této úlohy se neuronové sítě ukázaly jako rovnocenný nástroj pro modelování vícerozměrných predikčních modelů.

Neuronové sítě však oproti VAR modelům neumožňují snadnou interpretaci. Výběr vhodné struktury neuronové sítě hraje velmi důležitou roli. V práci bylo odhadnuto množství různých kombinací počtů skrytých vrstev a jednotek s různými normalizačními přístupy a aktivačními funkcemi. Žádná z těchto kombinací však nedominovala nad ostatními a nelze tedy jednoznačně určit optimální strukturu sítě, která by mohla být aplikovatelná na všechny země. V rámci neuronových sítí existuje řada dalších nástrojů, které v této práci nejsou použity, ale mohly by vést ke zlepšení predikčních schopností. Jedním z takových přístupů je například tzv. „brzké zastavení“, pro které jsou pozorování rozdělena na trénovací, testovací a validační sadu. Model je klasicky odhadován na trénovací sadě a výpočet je zastaven v momentě, kdy chyba na validační sadě přestane klesat. Metodu „Early Stopping“ podrobně popisuje například Prechtl[34]. Podobný postup by bylo vhodné aplikovat i pro výběr regularizační konstanty λ , která se projevila jako velmi důležitá část modelu.

Na příkladu TVAR modelů lze vidět, že samotné zvýšení variability a povolení nelineárnosti nepřináší v rámci predikcí kýžený efekt. Nejspíše je nutné kombinovat právě toto zvýšení s metodami zamezujícími přeučení modelu.

Dále je důležité zmínit, že dle výsledků modely s nízkou predikční chybou nemusí zároveň dobře popisovat samotný generující proces. Naopak, jak lze vidět na výsledných modelech, méně vysvětlené variability původních data často koreluje s nižší predikční chybou. Z toho důvodu je vhodné volit predikční model na základě chyby z testovací sady. Tento postup však přináší riziko, že zvolená specifikace modelu je ovlivněna pouze konkrétními pozorováními z testovací sady, které nemusejí odrážet budoucí vývoj. Toto riziko lze snížit použitím robustnějšího výpočtu predikční chyby, případně pro delší časové řady rozdělením pozorování na trénovací, testovací a validační sadu.

Literatura

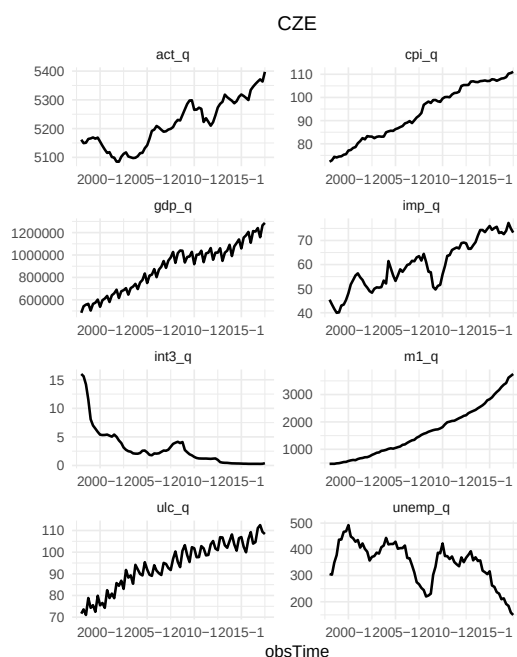
- [1] Milam Aiken. „Using a neural network to forecast inflation“. In: *Industrial Managment & Data Systems* (1999).
- [2] JJ Allaire a François Chollet. *R interface to Keras*. [vid. 20.4.2018]. URL: <https://keras.rstudio.com/>.
- [3] Josef Arlt. *Moderní metody modelování ekonomických časových řad*. Grada Publishing, 1999.
- [4] Josef Arlt a Markéta Arltová. *Ekonomické časové řady: [vlastnosti, metody modelování, příklady a aplikace]*. Praha: Grada, 2007. ISBN: 9788024713199.
- [5] Magyar Nemzeti Bank. *Inflation Targeting*. [vid. 24.4.2018]. URL: <https://www.mnb.hu/en/monetary-policy/monetary-policy-framework/inflation-targeting>.
- [6] Česká národní banka. *Inflation targeting in the Czech Republic*. [vid. 24.4.2018]. URL: https://www.cnb.cz/en/monetary_policy/inflation_targeting.html.
- [7] M. Ali Choudhary a Adnan Haider. „Neural network models for inflation forecasting: an appraisal“. In: *Applied Economics* 44.20 (2012), s. 2631–2635. DOI: 10.1080/00036846.2011.566190.
- [8] Alan V. Deardoff. *Terms of Trade: Glossary of International Economics*. World Scientific Pub Co Inc, 2006. ISBN: 9812566287.
- [9] David A. Dickey a Wayne A. Fuller. „Distribution of the Estimators for Autoregressive Time Series With a Unit Root“. In: *Journal of the American Statistical Association* 74.366 (1979), s. 427. DOI: 10.2307/2286348.
- [10] Laurene V. Fausett. *Fundamentals of Neural Networks: Architectures, Algorithms And Applications*. Pearson, 1993. ISBN: 0133341860.
- [11] Ian Goodfellow, Yoshua Bengio a Aaron Courville. *Deep Learning*. The MIT Press, 3. led. 2017. 800 **pagetotals**. ISBN: 0262035618.
- [12] Jan G. De Gooijer a Rob J. Hyndman. „25 years of time series forecasting“. In: *International Journal of Forecasting* 22.3 (2006), s. 443–473.
- [13] Mikael Gredenhoff a Sune Karlssony. „Lag-length Selection in VAR-models Using Equal and Unequal Lag-Length Procedures“. In: *Working Paper Series in Economics and Finance* (1997).
- [14] William H. Greene. *Econometric Analysis (7th Edition)*. Pearson, 2011. ISBN: 978-0131395381.

- [15] Galyna Grynkviv a Lars Stentoft. „Stationary Threshold Vector Autoregressive Models“. In: (2014).
- [16] James Douglas Hamilton. *Time Series Analysis*. Princeton University Press, 1994. ISBN: 978-0691042893.
- [17] Bruce Hansen. „Testing for Linearity“. In: *Journal of Economic Surveys* 13.5 (1999), s. 551–576. DOI: 10.1111/1467-6419.00098.
- [18] Geoffrey Hinton. *Neural Networks for Machine Learning - rmsprop*. [vid: 10.2.2018]. URL: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.
- [19] Roman Horváth a Jakub Matějů. „How are Inflation Targets Set?“ In: *CNB WORKING PAPER SERIES* (2011).
- [20] Roman Hušek. *Aplikovaná ekonometrie - Teorie a praxe*. 1. vyd. Oeconomica, 2009. ISBN: 978-80-245-1623-3.
- [21] Kristin Hubrich a Timo Terasvirta. „Thresholds and Smooth Transitions in Vector Autoregressive Models“. In: *CREATES Research Paper* (2013).
- [22] Rob J. Hyndman. „Measuring forecast accuracy“. In: (2014).
- [23] Rob J. Hyndman a George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2013. ISBN: 0987507109.
- [24] Diederik P. Kingma a Jimmy Ba. „Adam: A Method for Stochastic Optimization“. In: (22. pros. 2014). arXiv: 1412.6980v9 [cs.LG].
- [25] Ben Krose a Patrik Smagt. *An introduction to Neural Networks*. The University of Amsterdam, 1996.
- [26] Denis Kwiatkowski et al. „Testing the null hypothesis of stationarity against the alternative of a unit root“. In: *Journal of Econometrics* 54.1-3 (1992), s. 159–178. DOI: 10.1016/0304-4076(92)90104-y.
- [27] Helmut Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Springer, 2005.
- [28] Massimiliano Marcellino. „Instability and non-linearity in the EMU“. In: *IEP-Università Bocconi, IGER and CEPR* (2002).
- [29] Paul McNelis a Peter McAdam. „Forecasting inflation with thick models and neural networks“. In: *European Central Bank* (2004).
- [30] Emi Nakamura. „Inflation forecasting using a neural network“. In: *Economics letters* (2004).
- [31] Bernhard Pfaff. *Analysis of Integrated and Cointegrated Time Series with R (Use R!)* Springer, 2008. ISBN: 978-0-387-75967-8.

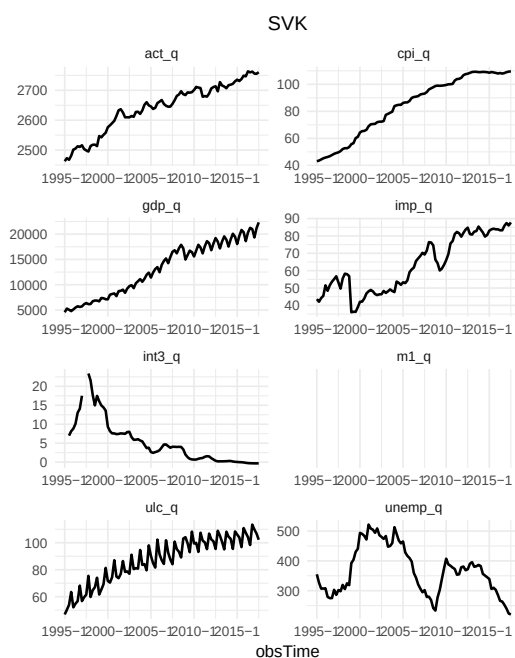
- [32] Peter C. B. Phillips a Piere Perron. „Testing for a unit root in time series regression“. In: *Biometrika* 75.2 (1988), s. 335–346. DOI: 10.1093/biomet/75.2.335.
- [33] Narodowy Bank Polski. *Monetary Policy*. [vid. 26.4.2018]. URL: http://www.nbp.pl/homen.aspx?f=/en/onbp/polityka_pieniezna.html.
- [34] Lutz Prechelt. *Neural Networks: Tricks of the Trade (Lecture Notes in Computer Science)*. Springer, 2012. ISBN: 978-3-642-35289-8.
- [35] Robert H. Shumway a David S. Stoffer. *Time Series Analysis and Its Applications: With R Examples (Springer Texts in Statistics)*. Springer, 2010. ISBN: 9781441978646.
- [36] Jindřich Soukup a kol. Pošta Vít. *Makroekonomie*. Managment Press, 2010.
- [37] Lars E. O. Svensson. „Inflation Targeting“. In: *NBER WORKING PAPER SERIES* (2010).
- [38] TensorFlow. *TensorFlow*. [vid 26.4.2018]. URL: <https://www.tensorflow.org/>.
- [39] Howell Tong. „On a Threshold Model in Pattern Recognition and Signal Processing“. In: (led. 1978).
- [40] Ruey S. Tsay. „Testing and Modeling Multivariate Threshold Models“. In: *Journal of the American Statistical Association* (1998).
- [41] Maxim Vladimirovich, Shcherbakov a Col. *A Survey of Forecast Error Measures*. 2013.
- [42] Jeffrey M. Wooldridge. *Econometric Analysis of Cross Section and Panel Data*. The MIT Press, 2001. ISBN: 0-262-23219-7.
- [43] Jeffrey M. Wooldridge. *Introductory Econometrics: A Modern Approach*. South-Western College Pub, 2002. ISBN: 0-324-11364-1.
- [44] Eric Zivot a Jiahui Wang. *Modeling Financial Time Series with S-PLUS®*. Springer New York, 10. říj. 2007.

Přílohy

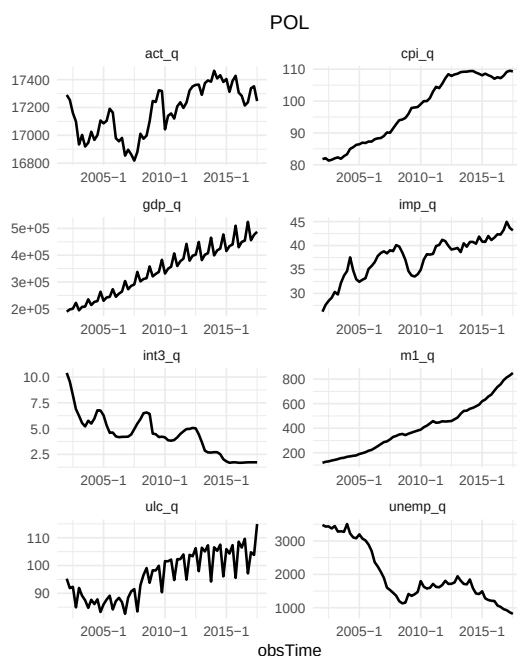
A. Grafy



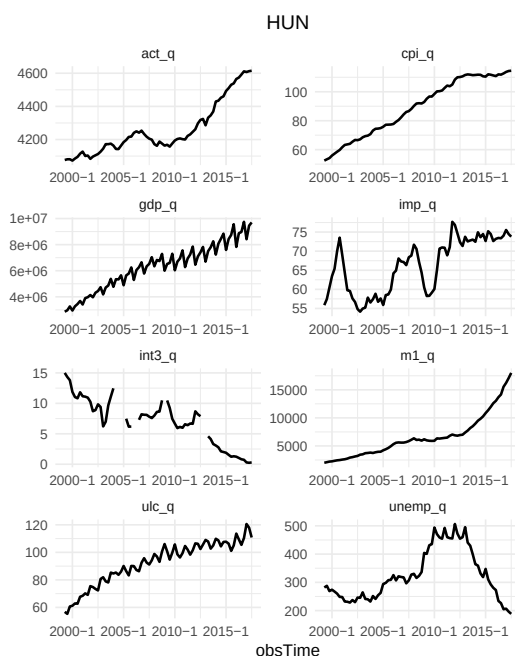
Obrázek A.1: Vývoj ukazatelů za Českou republiku



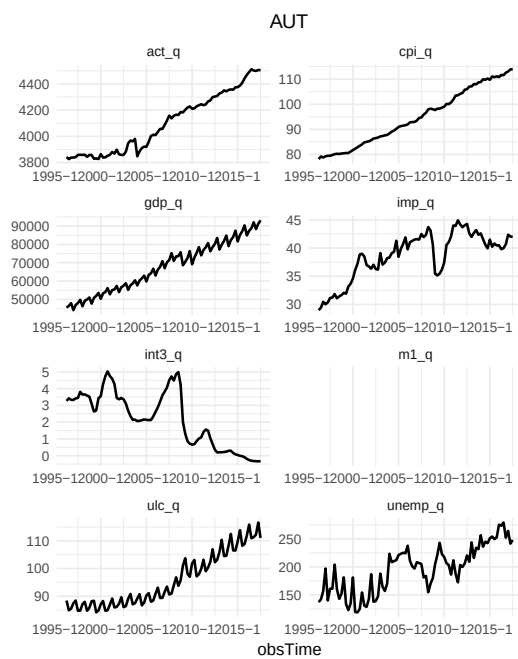
Obrázek A.2: Vývoj ukazatelů za Slovensko



Obrázek A.3: Vývoj ukazatelů za Polsko



Obrázek A.4: Vývoj ukazatelů za Maďarsko



Obrázek A.5: Vývoj ukazatelů za Rakousko

B. Skripty

B.1 Použité balíčky

Tabulka B.1: Přehled použitých balíčků

Balíček	Popis	Verze
dplyr	A Grammar of Data Manipulation	0.7.4
forecast	Forecasting Functions for Time Series and Linear Models	8,2
ggplot2	Create Elegant Data Visualisations Using the Grammar of Graphics	2.2.1
keras	R Interface to 'Keras'	2.1.5
lmtest	Testing Linear Regression Models	0.9-35
OECD	Search and Extract Data from the OECD	0.2.2
openxlsx	Read, Write and Edit XLSX Files	4.0.17
reshape2	Flexibly Reshape Data: A Reboot of the Reshape Package	1.4.2
tsDyn	Nonlinear Time Series Models with Regime Switching	0.9-46
tseries	Time Series Analysis and Computational Finance	0.10-44
urca	Unit Root and Cointegration Tests for Time Series Data	1.3-0
vars	VAR Modelling	1.5-2
stats	The R Stats Package	3.4.3

B.2 Benchmark

```
require(dplyr)
library(urca)
require(tseries)
#-----
rm(list=ls())
gc()
dev.off(dev.list()["RStudioGD"])
#-----
list_of_files <- dir("dta")[grepl("*.q.dif", dir("dta"))][-3]
list_of_country <- sub(".q.dif.RDS", "", list_of_files)
names(list_of_files) <- list_of_country
df_models_all <- c()
df_models_tvar_all <- c()
df_models_all <- c()
df_models_nn_all <- c()
country <- "CZE"
#-----
#BENCHMARK
#-----
for(country in list_of_country){
  gc()
  #-----
  #creating plot folder
  #-----
  print_plot <- FALSE
  #dir.create(paste0("dta\\plots\\",country))
  #-----
  #load data
  #-----
  ts.df <- readRDS(file.path("dta", list_of_files[country]))
  ts.df1 <- ts.df[,colnames(ts.df) %in% c("cpi_q","gdp_q","imp_q" ,"int3_q" ,"m1_q","act_q","ulc_
    ↪ q","unemp_q")]
  ts.df1 <- na.omit(ts.df1)
  ts.EU28 <- readRDS("dta/EU28.q.dif.RDS")
  ts.EU28 <- ts.EU28[time(ts.EU28) %in% time(ts.df1)]
  write.csv(ts.df[, "cpi_q"], paste0("c:/git/",country,".csv"))
}
#-----
#List of models
#-----
list_of_models0 <- combn(colnames(ts.df1), 2, simplify = FALSE)
help0 <- t(combn(colnames(ts.df1), 2, simplify = TRUE))
list_of_models0 <- list_of_models0[help0[,1]=="cpi_q"]

list_of_models <- combn(colnames(ts.df1), 3, simplify = FALSE)
help <- t(combn(colnames(ts.df1), 3, simplify = TRUE))
list_of_models <- list_of_models[help[,1]=="cpi_q"]

list_of_models2 <- combn(colnames(ts.df1), 4, simplify = FALSE)
help2 <- t(combn(colnames(ts.df1), 4, simplify = TRUE))
list_of_models2 <- list_of_models2[help2[,1]=="cpi_q"]

list_of_models <- c(list_of_models0,list_of_models, list_of_models2)
#-----
#VAR
#-----
```

```

source("model_var.r")
df_models_all <- rbind(df_models_all, df_models)
#-----
#TVAR
# #-----
source("model_tvar.r")
df_models_tvar_all <- rbind(df_models_tvar_all, df_models_tvar)
#-----
#NN
#-----
list_of_lags <- unique(df_models$lag)
list_of_lags <- c(4,5,6)
list_of_n_hidden <- c(1,3)
epochs <- 4000
list_of_hidden <- c(15, 20, 30)
act_fce <- c("linear","relu","softmax")
list_std <- c("linear","norm")
optimizer2 <- "adam"
trend = TRUE
seas = FALSE
dum_yes = TRUE
source("model_nn_tensorflow.R")
df_models_nn_all <- rbind(df_models_nn_all, df_models_nn)
#-----
rm(list = ls()[!ls() %in% c(
  "df_models_all",
  "df_models",
  "df_models_nn_all",
  "df_models_nn",
  "df_models_tvar_all",
  "df_models_tvar",
  "list_of_models",
  "list_of_country",
  "ts.df1",
  "ts.df",
  "country",
  "list_of_files"
)])
gc()
print(country)
#-----
}
#-----
saveRDS(df_models_all,paste0("dta/results/df_models_all2",Sys.Date(),".RDS"))
saveRDS(df_models_nn_all,paste0("dta/results/df_models_nn_template",Sys.Date(),".RDS"))
saveRDS(df_models_tvar_all,paste0("dta/results/df_models_tvar_all2",Sys.Date(),".RDS"))
#-----

```

B.3 Odhad neuronové sítě pomocí Keras(Tensorflow)

```
#-----
ts.org <- ts.df
ts <- data.frame(ts.org)
#-----
# Normalization functions
#-----
if (std == "norm") {
  standardize <- function(z) {
    means <- apply(z, 2, mean)
    names(means) <- colnames(z)
    vars <- apply(z, 2, var)
    names(vars) <- colnames(z)
    z <- sweep(z, 2, means, "-")
    z <- sweep(z, 2, vars, "/")
    return(list(
      z = z,
      means = means,
      vars = vars
    ))
  }
  unstandardize <- function(z, mean, var) {
    z <- sweep(z, 2, var, "*")
    z <- sweep(z, 2, mean, "+")
    return(z)
  }
  unstandardize.vector <- function(z, mean, var) {
    z <- z * var
    z <- z + mean
    return(z)
  }
}else if (std == "linear") {
  standardize <- function(z) {
    max <- apply(z, 2, max)
    names(max) <- colnames(z)
    min <- apply(z, 2, min)
    names(min) <- colnames(z)
    z <- sweep(z, 2, min, "-")
    z <- sweep(z, 2, (max - min), "/")
    return(list(
      z = z,
      means = max,
      vars = min
    ))
  }
  unstandardize <- function(z, max, min) {
    z <- sweep(z, 2, (max - min), "*")
    z <- sweep(z, 2, min, "+")
    return(z)
  }
  unstandardize.vector <- function(z, max, min) {
    z <- z * (max - min)
    z <- z + min
    return(z)
  }
}
```



```

}
#-----
#Data selection
#-----
model_vars <- unlist(model_vars)
if(template$EU[i] == TRUE){
  ts <- ts.org[, c(model_vars,"eu")]
}else{
  ts <- ts.org[, model_vars]
}
#-----
#Data normalization
#-----
means <- standardize(ts)$means
vars <- standardize(ts)$vars
ts <- standardize(ts)$z
ts <- data.frame(ts)
#-----
#Add lags
#-----
ts.lag <- data.frame(matrix(ncol = lags * ncol(ts), nrow = dim(ts)[1]))
col <- 0
for (il in 1:lags) {
  for (j in colnames(ts)) {
    col <- col + 1
    ts.lag[, col] <- dplyr::lag(ts[, j], il)
    colnames(ts.lag)[col] <- c(paste0(j, "_l", il))
  }
}
ts <- cbind(ts, ts.lag)
if(template$EU[i] == TRUE){
  ts1 <- ts[, !colnames(ts) %in% c("eu",paste0("eu_l",seq(1,lags,1)))]
  ts2 <- ts[, colnames(ts) %in% c("eu",paste0("eu_l",seq(1,lags,1)))]
  ts <- cbind(ts1, ts2)}
#-----
#Add trend
#-----
if (trend == TRUE) {
  ts$t <- 1:nrow(ts)
}
#-----
#Add seasonal dummy (s1,s2,S3,s4)
#-----
if (seas == TRUE) {
  ts$s2 <- rep(c(0, 1, 0, 0), (nrow(ts) + 4) / 4)[1:nrow(ts)]
  ts$s3 <- rep(c(0, 0, 1, 0), (nrow(ts) + 4) / 4)[1:nrow(ts)]
  ts$s4 <- rep(c(0, 0, 0, 1), (nrow(ts) + 4) / 4)[1:nrow(ts)]
}
#-----
#Add dummy (year = 2008)
#-----
if (dum_yes == TRUE) {
  ts$dummy <- rep(0, dim(ts)[1])
  ts$dummy[time(ts.org) == "2008"] <- 1
}
ts$c <- 1
ts <- na.omit(ts)
#-----
#Cross validation - dataset split (training, testing)

```

```

#-----
size <- nrow(ts) - 10
ts.test <- ts[(size + 1):nrow(ts), ]
ts.train <- ts[1:size, ]
#-----
y_train <- ts.train[, model_vars]
y_test <- ts.test[, model_vars]
x_train <- ts.train[, !colnames(ts.train) %in% model_vars]
x_test <- ts.test[, !colnames(ts.test) %in% model_vars]
#-----
x_train <- as.matrix(x_train)
y_train <- as.matrix(y_train)
#-----
x_test <- as.matrix(x_test)
y_test <- as.matrix(y_test)
#-----
#Model estimation
#-----
batch_size <- dim(ts)[1]
num_variables <- length(model_vars)
number_of_hidden_layers <- n_hidden

clip_norm <- 1.0
hidden_units <- hidden
#-----
#Model estimation
#-----
model <- keras_model_sequential()
model %>%
  layer_dense(units = hidden_units, input_shape = dim(x_train)[2])
#-----
for (n in 1:number_of_hidden_layers) {
  model %>% layer_dense(units = hidden_units)
}
#-----

if (reg_def1== TRUE) {
  model %>%
    layer_dense(units = num_variables,
                 activation = actf,
                 use_bias = TRUE,
                 kernel_regularizer = regularizer_l2(0.01))
}else{
  model %>%
    layer_dense(units = num_variables,
                 activation = actf,
                 use_bias = TRUE)
}
#-----
if (reg_def) {
  model %>% layer_activity_regularization(l1 = 0.0001, l2 = 0.0001)
}
#summary(model)
#-----
#sgd <- optimizer_sgd(lr=0.01, decay=1e-6, momentum=0.9, nesterov=TRUE)
model %>% compile(loss = 'mean_squared_error',
                 optimizer = optimizer2)
#-----

```

```

history <- model %>% fit(
  x_train,
  y_train,
  verbose = getOption("keras.fit_verbose", default = 0),
  epochs = epochs,
  batch_size = batch_size
)
#-----
#plot(history)
print(paste("beep_nn-_",paste(model_vars,collapse = "-"), as.numeric(Sys.time()-time.flag)))
#-----
#Prediction function
#-----
nn_predict <- function(y_train, x_train, last, model_vars) {
  dta <- c()
  for (i in model_vars) {
    j = which(i == model_vars)
    #-----
    #PREDICTIONS
    #-----
    yhat.est.m <- c()
    yhat.est <- y_train[nrow(y_train),]
    x.est <- x_train[nrow(x_train),!colnames(x_train) %in% c("t", "s2", "s3", "s4", "dummy", "c",
      ↪ c("eu",paste0("eu_1",seq(1,lags,1)))))]
    names_of_columns <- colnames(x_train)
    for (k in 1:last) {
      test_dummy <- x_train[k, names_of_columns[names_of_columns %in% c("t", "s2", "s3", "s4", "
      ↪ dummy", "c", c("eu",paste0("eu_1",seq(1,lags,1)))))]
      last_obs <- c(yhat.est, x.est[0:(length(x.est) - length(yhat.est))], test_dummy)
      names(last_obs) <- names_of_columns
      yhat.est <- model %>% predict(t(last_obs))
      yhat.est.m <- rbind(yhat.est.m, yhat.est)
      x.est <-
        last_obs[!names(last_obs) %in% c("t", "s2", "s3", "s4", "dummy","c",c("eu",paste0("eu_1",
          ↪ seq(1,lags,1)))))]
    }
    yhat.est <-
      unstandardize.vector(yhat.est.m[, j], means[j], vars[j])

    dta <- cbind(dta, yhat.est)
  }
  colnames(dta) <- model_vars
  return(dta)
}
#-----
#Prediction
#-----
yhat <- (model %>% predict(x_train))[, 1]
yhat <- unstandardize.vector(yhat, means[1], vars[1])
yhat.est <- nn_predict(y_train, x_train, 10, model_vars)[, 1]
y <- ts.org[(lags + 1):(nrow(ts.org) - 10), "cpi_q"]
y.p <- ts.org[(nrow(ts.org) + 1 - 10):nrow(ts.org), "cpi_q"]
yhat2 <- c()
for (l in 1:dim(y_train)[2]) {
  yhat2 <-
    cbind(yhat2,
      unstandardize.vector((model %>% predict(x_train))[, 1],
        means[1],
        vars[1]))
}

```

```

}
#-----
# Error Measure
#-----
Rsqr <- (sum((yhat - y) ^ 2)) / sum((y - mean(y)) ^ 2)
MSE = mean((yhat - y) ^ 2)
MSE.est = mean((yhat.est - y.p) ^ 2)
res <- yhat - y
#-----
tseries::jarque.bera.test(res)
acf(res)
lmtest::bgtest(res~1)
summary(lm(res~lag(res)))
#-----
#Saving Plots
#-----
if(print_plot == TRUE){
  png(file.path("dta","plots","fin_plots2",paste0(tolower(country),irina,"nn",".png")),
      width=1000,
      height = 1050
  )
  print(plot_fcst(list(nn_predict(y_train, x_train, 10, model_vars),
                          yhat2),
                  as.data.frame(ts.org[, model_vars]),
                  lags,
                  10,
                  0,
                  model_vars,
                  nn = TRUE))
  dev.off()
}
#-----
#Saving results
#-----
#result <-
# data.frame(
# index = country,
# model = paste(model_vars, collapse = ","),
# std = std,
# trend = trend,
# seas = seas,
# dum_yes = dum_yes,
# EU = EU,
# reg_def = reg_def,
# reg_def1 = reg_def1,
# lags = lags,
# actf = actf,
# hidden = hidden,
# n_hidden = n_hidden,
# MSE = MSE,
# MSE.est = MSE.est,
# Rsqr = Rsqr,
# time = as.numeric(Sys.time()-time.flag)
# )
df_models_nn <- rbind(df_models_nn, result)
#-----
rm("model")
gc()
#-----

```

B.4 Odhad TVAR modelu

```
library(urca)
library(vars)
library(forecast)
library(dplyr)
require(reshape2)
require(ggplot2)
require(OECD)
require(stats)
require(tsDyn)
source("scripts.r")

#-----
df_models_tvar <-
  data.frame(
    index = c(),
    model = c(),
    n_tresh = c(),
    EU = c(),
    lag = c(),
    BG = c(),
    ARCH = c(),
    JB = c(),
    tvarlin.p = c(),
    cpi.r = c(),
    cpi.rs.adj = c(),
    AICv = c(),
    MSE = c(),
    RMSE = c(),
    MSE10 = c(),
    RMSE10 = c(),
    MSE3 = c(),
    MAE3 = c()
  )

#-----
#Model selection
#-----
for(model in list_of_models){
  for(ntresh in 1:3){
    for(EU in c(TRUE, FALSE)){
      dummies = 1
      tryCatch({
        #-----
        #Data selection
        #-----
        ts.model <- ts.df1[,unlist(model)]
        #-----
        #Lag selection
        #-----
        AIC_def <- VARselect(ts.model, lag.max = 8, type = "both")$selection[1]
        HQ_def <- VARselect(ts.model, lag.max = 8, type = "both")$selection[2]
        SC_def <- VARselect(ts.model, lag.max = 8, type = "both")$selection[3]
        #-----
        #Optimal lag search
        #-----
        for(AIC in c(AIC_def, HQ_def, SC_def)){
          #-----
          tvar <- TVAR(data=ts.model, include= c("both"), nthresh = ntresh, lag=AIC, plot=FALSE)
```

```

#-----
tvarlin.p <- TVAR.LRtest(ts.model, lag = 3, series = "cpi_q", thDelay=1:2, trend=TRUE, test="lvs"
  ↪ )$Pvalueboot[1]
#-----
#Model validation tests
#-----
BG <- 0
ARCH <- 0
JB <- 0
#-----
cpi.r <- 0
cpi.rs.adj <- 0
#-----
#Error measurement
#-----
AICv<-AIC(tvar)
#-----
fit <- fitted(tvar)[,"cpi_q"]
y <- ts.model[(AIC+1):length(ts.model[, "cpi_q"]), "cpi_q"]
MSE <- round(mean((y-fit)^2),4)
RMSE <- round(sqrt(MSE),4)
#-----
#Cross validation error
#-----
ts.model.est <- ts(ts.df1[1:(dim(ts.df1)[1]-10),unlist(model)], start=c(1998,2),frequency = 4)
dummy_est <- dummy[1:dim(ts.model.est)[1]]
#-----
tvar.est <- TVAR(data=ts.model.est, include= c("both"), lag=AIC)
#-----
fcst <- predict(tvar.est, n.ahead = 10)
#-----
fcst.value<- as.numeric(fcst[,1])
y.value <- ts.model[(length(ts.model[, "cpi_q"])-9):length(ts.model[, "cpi_q"]), "cpi_q"]
MSE.test<- round(mean((y.value-fcst.value)^2),4)
RMSE.test <- round(sqrt(MSE.test),4)
#-----
#Rolling window
#-----
mse3 <- c()
mae3 <- c()
rmse3 <- c()
for(i in 1:10){
  ts.model.est3 <- ts(ts.df1[i:(dim(ts.df1)[1]-(13-i)),unlist(model)], frequency = 4)
  dummy3 <- dummy[i:(dim(ts.df1)[1]-(13-i))]
  var.est3 <- TVAR(data=ts.model.est, include= c("both"), lag=AIC)
  y3 <- ts.df1[(dim(ts.df1)[1]-(13-i)+1):(dim(ts.df1)[1]-(13-i)+3)]
  fcst3 <- predict(var.est3, n.ahead = 3)[,1]
  mse3 <- c(mse3, mean((y3-fcst3)^2))
  rmse3 <- c(rmse3, sqrt(mean((y3-fcst3)^2)))
  mae3 <- c(mae3, mean(abs(y3-fcst3)))
}
MSE3 <- mean(mse3)
RMSE3 <- mean(rmse3)
MAE3 <- mean(mae3)
#-----
#Results
#-----
result <-
  data.frame(

```

```

    index = country,
    model = paste(model, collapse = ","),
    n_tresh = ntresh,
    EU = EU,
    lag = AIC,
    BG = BG,
    ARCH = ARCH,
    JB = JB,
    tvarlin.p = tvarlin.p,
    cpi.r = length(cpi.r),
    cpi.rs.adj = length(cpi.rs.adj),
    AICv = AICv,
    MSE = round(MSE,3),
    RMSE = round(RMSE,3),
    MSE10 = MSE.test,
    RMSE10 = RMSE.test,
    MSE3 = MSE3,
    MAE3 = MAE3

)

df_models_tvar <- rbind(df_models_tvar, result)
#-----
#Plot
#-----
if(print_plot == TRUE){
png(file.path("dta","plots",country,paste0("tvar_",country,"_",paste(unlist(model),collapse=","),
  ↪ "ntr=", ntresh, "lag=", AIC,".png")))
plot_fcst(tvar.est, ts.model, AIC, 10, model, tvar=TRUE)
dev.off()
}
#-----
}}, error=function(e){cat("ERROR_:",conditionMessage(e), "\n")})
}}}
```

B.5 Odhad VAR modelu

```
library(urca)
library(vars)
library(forecast)
library(dplyr)
require(reshape2)
require(ggplot2)
require(stats)
source("scripts.r")
#-----
df_models <-
  data.frame(
    index = c(),
    model =c(),
    seas =c(),
    dettype = c(),
    EU=c(),
    lag = c(),
    BG = c(),
    ARCH = c(),
    JB = c(),
    cpi.r = c(),
    cpi.rs.adj = c(),
    MSE = c(),
    RMSE = c(),
    AICv = c(),
    MSE10 = c(),
    RMSE10 = c(),
    MSE3 = c(),
    MAE3 = c(),
    AIC_def = c(),
    HQ_def = c(),
    SC_def = c()
  )
#-----
#Model selection
#-----
for(model in list_of_models){
  for(seas in c(1,0)){
    for(dettype in c("none","trend","const","both")){
      for(dumyes in c(1)){
        for(EU in c(TRUE,FALSE)){
          #-----
          #Data selection
          #-----
          ts.model <- ts.df1[,unlist(model)]
          #-----
          if(country == "SVK"){
            ts.EU282 <- c(rep(mean(ts.EU28),5), ts.EU28)
          }

          dummy <- rep(0, dim(ts.model)[1])
          if(dumyes == 1){
            dummy[time(ts.model) == "2008"] <- 1
          }
          if(EU == TRUE){
            dummy <- cbind(dummy, ts.EU282)
```



```

}
#-----
#Lag selection (AIC, HQ, SC)
#-----
AIC_def <- VARselect(ts.model, lag.max = 8, type = "both")$selection[1]
HQ_def <- VARselect(ts.model, lag.max = 8, type = "both")$selection[2]
SC_def <- VARselect(ts.model, lag.max = 8, type = "both")$selection[3]
#-----
#Optimal lag search
#-----
for(AIC in unique(c(AIC_def, HQ_def, SC_def))){
#-----
if(seas == 1){
var <- VAR(ts.model, p=AIC, type=dettype, exogen= cbind(dummy))
}else{
var <- VAR(ts.model, p=AIC, type=dettype, season = 4, exogen= cbind(dummy))
}
#-----
#Model validation tests
#-----
BG <- serial.test(var, lags.bg=5, type="BG")$serial$p.value
ARCH <- arch.test(var)$arch.mul$p.value
JB <- normality.test(var)$jb.mul$JB$p.value
#-----
#Error measurement
#-----
AICv<-AIC(var)
#-----
cpi.r <- round(as.numeric(var$varresult$cpi_q$r.squared),4)
cpi.rs.adj <- round(as.numeric(var$varresult$cpi_q$adj.r.squared),4)
#-----
fit <- fitted(var)[,"cpi_q"]
y <- ts.model[(AIC+1):length(ts.model[, "cpi_q"]), "cpi_q"]
MSE <- round(mean((y-fit)^2),4)
RMSE <- round(sqrt(MSE),4)
#-----
#Cross validation error
#-----
ts.model.est <- ts(ts.df1[1:(dim(ts.df1)[1]-10),unlist(model)], frequency = 4)
if(EU == TRUE){
dummy_est <- dummy[1:dim(ts.model.est)[1],]
dummy_est2 <- dummy[(dim(ts.df1)[1]-9):dim(ts.df1)[1],]
}else{
dummy_est <- dummy[1:dim(ts.model.est)[1]]
dummy_est2 <- dummy[(dim(ts.df1)[1]-9):dim(ts.df1)[1]]
}
#-----
if(seas == 1){
var.est <- VAR(ts.model.est, p=AIC, type=dettype, exogen= cbind(dummy_est))
}else{
var.est <- VAR(ts.model.est, p=AIC, type=dettype, season = 4, exogen= cbind(dummy_est))
}
#-----
fcst <- predict(var.est, n.ahead = 10, dumvar=cbind(dummy_est=dummy_est2))
fcst.value<- as.numeric(fcst$fcst$cpi_q[,1])
y.value <- ts.model[(length(ts.model[, "cpi_q"])-9):length(ts.model[, "cpi_q"]), "cpi_q"]
MSE.test<- round(mean((y.value-fcst.value)^2),4)
RMSE.test <- round(sqrt(MSE.test),4)
#-----

```

```

#Rolling window
#-----
mse3 <- c()
mae3 <- c()
rmse3 <- c()

for(i in 1:10){
  ts.model.est3 <- ts(ts.df1[i:(dim(ts.df1)[1]-(13-i)),unlist(model)], frequency = 4)
  if(EU == TRUE){
    dummy3 <- dummy[i:(dim(ts.df1)[1]-(13-i)),]
    dummy4 <- dummy[(dim(ts.df1)[1]-(13-i)+1):(dim(ts.df1)[1]-(13-i)+3),]
  }else{
    dummy3 <- dummy[i:(dim(ts.df1)[1]-(13-i))]
    dummy4 <- dummy[(dim(ts.df1)[1]-(13-i)+1):(dim(ts.df1)[1]-(13-i)+3)]
  }
  #-----
  if(seas == 1){
    var.est3 <- VAR(ts.model.est3, p=AIC, type=dettype, exogen= cbind(dummy3))
  }else{
    var.est3 <- VAR(ts.model.est3, p=AIC, type=dettype, season = 4, exogen= cbind(dummy3))
  }
  #-----
  y3 <- ts.df1[(dim(ts.df1)[1]-(13-i)+1):(dim(ts.df1)[1]-(13-i)+3)]
  fcst3 <- predict(var.est3, n.ahead = 3, dumvar=cbind(dummy3=dummy4))$fcst$cpi_q[,1]
  #-----
  mse3 <- c(mse3, mean((y3-fcst3)^2))
  rmse3 <- c(rmse3, sqrt(mean((y3-fcst3)^2)))
  mae3 <- c(mae3, mean(abs(y3-fcst3)))

}
#-----
MSE3 <- mean(mse3)
RMSE3 <- mean(rmse3)
MAE3 <- mean(mae3)
#-----
#Results
#-----
result <-
  data.frame(
    index = country,
    model = paste(paste(model, collapse = ",")),
    seas = paste(seas,dumyes,dettype),
    dettype = dettype,
    EU = EU, lag = AIC,
    BG = BG, ARCH = ARCH,
    JB = JB, cpi.r = length(cpi.r),
    cpi.rs.adj = length(cpi.rs.adj),
    MSE = MSE, RMSE = RMSE,
    AICv = AICv, MSE10 = MSE.test,
    RMSE10 = RMSE.test, MSE3 = MSE3,
    MAE3 = MAE3, AIC_def = AIC_def,
    HQ_def = HQ_def, SC_def = SC_def
  )
df_models <- rbind(df_models, result)
#-----
#Plot
#-----
if(print_plot == TRUE){
  print(paste("beep", paste(model, collapse = ",")))
  png(file.path("dta","plots",country,paste0("var_",country,"_",paste(unlist(model),collapse=",")),

```

```
    ↪ seas,dumyes, dettype,"eu=",EU,"lag=",AIC,".png"))
plot_fcst(var.est, ts.model, AIC, 10, dummy_est2, c(country,model), tvar=FALSE)
dev.off()
}
#-----
}}}}
  print(paste("beep", paste(model, collapse = ",")))
}
```

B.6 Prohledávání datasetu OECD

```
require(dplyr)
require(reshape2)
require(ggplot2)
require(OECD)

#-----
#http://stats.oecd.org/#
#-----
# cpi - customer price index
# exc - exchange rate
# gpd - gross domestic product
# unemp - unemployment rate
# emp - employment rate
# int - interest rate
#-----

dta.selected = data.frame(id=c(),dataset=c(),label=c(),LOCATION=c(),
                          SUBJECT=c(),MEASURE=c(), FREQUENCY_M=c(),
                          FREQUENCY_Q=c(),TIME_FORMAT=c(), UNIT=c(),
                          POWERCODE=c(), obsTime=c()
                          )

count=0
dtasets <- "MEI"
dtasets <- c("MEI","MEI_PRICES","QNA","ULC_EEQ","STLABOUR","KEI","MEI_FIN")
#-----
for (dtaset in dtasets){
  ds <- get_data_structure(dtaset)
  str(ds, max.level = 1)
  ds$VAR_DESC
  selected <- ds$SUBJECT
  for (i in selected$id) {
    try({
      count = count+1
      print(paste(i,round(count/length(selected$id)*100,2),"%","dataset:",dtaset))
      query <- get_dataset(dtaset, filter = list(LOCATION="CZE", SUBJECT=i))
      dta.selected <- rbind(dta.selected,data.frame(id = i,
                                                    dataset=dtaset,
                                                    label=selected$label[selected$id==i],
                                                    LOCATION=paste(unique(query$LOCATION),collapse="_"),
                                                    SUBJECT=paste(unique(query$SUBJECT),collapse="_"),
                                                    MEASURE=paste(unique(query$MEASURE),collapse="_"),
                                                    FREQUENCY_M=any((unique(query$FREQUENCY) %in% c("M")
                                                    ↪ )),
                                                    FREQUENCY_Q=any(unique(query$FREQUENCY) %in% c("Q"))
                                                    ↪ ,
                                                    TIME_FORMAT=paste(unique(query$TIME_FORMAT),collapse
                                                    ↪ ="_"),
                                                    UNIT=paste(unique(query$UNIT),collapse="_"),
                                                    POWERCODE=paste(unique(query$POWERCODE),collapse="_"
                                                    ↪ ),
                                                    obsTime=query$obsTime[1]
                                                    ))
    })
  }
}
#-----
saveRDS(dta.selected,"./dta/dta_selected2.RDS")
write.csv(dta.selected,"./dta/dta_selected2.csv")
```

B.7 Stahování požadovaných datasetů

```
require(dplyr)
require(reshape2)
require(ggplot2)
require(OECD)

#-----
#http://stats.oecd.org/#
#-----

#ds$MEASURE
#IXOB Index 2010=100
#IXOBSA Index 2010=100, s.a.
#STSA Level, rate or national currency, s.a.
#ST Level, rate or national currency
#-----

#load instructions
#-----

download <- openxlsx::read.xlsx("./dta/download.xlsx")
#-----

country <- download$LOCATION
#-----

for (j in country) {
  download$LOCATION <- j
  count = 0
  for (i in 1:length(download$id)) {
    dta <- download[i,]
    count = count + 1
    try({
      print(paste(
        j,
        dta$SUBJECT,
        round(count / length(download$id) * 100, 2),
        "%",
        "dataset:",
        dta$dataset
      ))
      filter <-
        list(
          LOCATION = dta$LOCATION,
          SUBJECT = dta$SUBJECT,
          MEASURE = dta$MEASURE,
          FREQUENCY = dta$FREQUENCY
        )
      #filter = list(LOCATION="CZE", SUBJECT=dta$SUBJECT)
      query <- get_dataset(dta$dataset, filter = filter)
      saveRDS(query, file.path("dta", "oecd_new", j, paste0(dta$file_name, ".RDS")))
    })
  }
}
```

B.8 Příprava dat

```
require(dplyr)
require(reshape2)
#-----
#Data merge
#-----
for(j in c("CZE", "POL", "SVK", "AUT", "HUN")){
#-----
#Load files (folder: dta)
#-----
list_of_files<-dir(file.path("dta","oecd_new",j))
list_of_files <- list_of_files[grepl("*_q.RDS",list_of_files)]
list_of_files <- list_of_files[-2] #exclude cpi
#-----
#Load initial file
#-----
dta <- readRDS(file.path("dta","oecd_new",j,"cpi_q.RDS"))
dta <- dta %>% dplyr::select(LOCATION, obsTime, obsValue)
colnames(dta)[3] <- "cpi_q"
#-----
for (i in list_of_files) {
  query <- readRDS(file.path("dta","oecd_new",j,i))
  name <-gsub(".RDS","",i)
  query <- query %>% dplyr::select(LOCATION, obsTime, obsValue)
  colnames(query)[3] <- name
  dta <- dta %>% left_join(query, by=c("LOCATION", "obsTime"))
}
#-----
#Save merged data
#-----
saveRDS(dta, paste0("dta/",j,".dta_oecd_new_q.RDS"))
}

require(dplyr)
library(urca)
require(tseries)
#-----
source("scripts.r")
#-----
list_of_files <- dir("dta")[grepl("*.dta_oecd_new_q.RDS",dir("dta"))]
list_of_country <- sub(".dta_oecd_new_q.RDS","",list_of_files)
#-----
dif_all <- c()
for(i in 1:length(list_of_files)){
dta <- readRDS(file.path("dta",list_of_files[i]))
#-----
country <- list_of_country[i]
#-----
start_date <- c(AUT = "1996-Q1",CZE="1997-Q4",EU28 = "1996-Q1", HUN = "1999-Q1",POL="2001-Q4",SVK
  ↪="1994-Q4")
#-----
dta <- dta %>% filter(obsTime > start_date[country], obsTime < "2017-Q4")
if(country == "EU28"){
  dta <- dplyr::select(dta, cpi_q)
}else{
dta <- dplyr::select(dta, -LOCATION, - obsTime, -int10_q)
}
#-----
```

```

if(country %in% c("SVK","HUN")){
  dta <- dplyr::select(dta, -int3_q)
}
ts.dta <- ts(dta, start = as.numeric(unlist(strsplit(start_date[country],split="-Q"))), frequency
  ↪ = 4)
#-----
dif <- c()
for(j in names(dta)){
  adft <-df_test(ts.dta[,j])

  dif <- rbind(dif,data.frame(country,
    index = j,
    df1 = adft$res,
    df2 = df_test(diff(ts.dta[,j]))$res,
    dif =adft$dif,
    model = adft$model,
    stat = adft$test_stat_val,
    st = adft$test_stat))

  print(j)
  plot_acf(ts.dta[,j], j)
}
dif_all <- rbind(dif_all,dif)
ts.dif <- ts.dta
for(index in colnames(dta)){
  if(dif[dif$index == index,"dif"]==1){
    ts.dif[,index] <- c(NA,diff(ts.dif[,index]))
  }else if (dif[dif$index == index,"dif"]==2){
    ts.dif[,index] <- c(NA,NA,diff(diff(ts.dif[,index])))
  }else if (dif[dif$index == index,"dif"]==3){
    ts.dif[,index] <- c(NA,NA,NA,diff(diff(diff(ts.dif[,index])))
  }else if (dif[dif$index == index,"dif"]==0){
    ts.dif[,index] <- ts.dif[,index]
  }
}
#-----
ts.df <-ts.dif
ts.df <- na.omit(ts.dif)
#-----
#plot_acf(ts.df[, "cpi_q"], "cpi_q")
#-----
#saveRDS(ts.df,paste0("dta/",country,".q.dif.RDS"))
}
#-----

require(dplyr)
require(reshape2)
#-----
#Data merge
#-----
for(j in c("CZE","POL", "SVK", "AUT", "HUN")){
  #-----
  #Load files (folder: dta)
  #-----
  list_of_files<-dir(file.path("dta","oecd_new",j))
  list_of_files <- list_of_files[grepl("*_q.RDS",list_of_files)]
  list_of_files <- list_of_files[-2] #exclude cpi
  #-----

```

```

#Load initial file
#-----
dta <- readRDS(file.path("dta","oecd_new",j,"cpi_q.RDS"))
dta <- dta %>% dplyr::select(LOCATION, obsTime, obsValue)
colnames(dta)[3] <- "cpi_q"
#-----
for (i in list_of_files) {
  query <- readRDS(file.path("dta","oecd_new",j,i))
  name <- gsub(".RDS","",i)
  query <- query %>% dplyr::select(LOCATION, obsTime, obsValue)
  colnames(query)[3] <- name
  dta <- dta %>% left_join(query, by=c("LOCATION", "obsTime"))
}
#-----
#Save merged data
#-----
saveRDS(dta, paste0("dta/",j,".dta_oecd_new_q.RDS"))
}

```

B.9 Podpůrné funkce

```

#-----
#Transform prediction from VAR
#-----
get_fcst <- function(predict.var){
  dta <- c()
  for(i in names(predict.var$fcst)){
    dta <- cbind(dta, predict.var$fcst[[i]][,1])
  }
  colnames(dta) <- names(predict.var$fcst)
  return(dta)
}
#-----
#Transform original data
#-----
get_y <- function(dta, last){
  y <- dta[(dim(dta)[1]-(last-1)):dim(dta)[1],]
  return(y)
}
#-----
#Plot forecast
#-----
plot_fcst <- function(var.est, dta, lag, last, dummy, model_vars, tvar=FALSE, nn =FALSE){
  if(nn == TRUE){
    yhat.fcst <- var.est[[1]]
    yhat <- var.est[[2]]
    yhat <- rbind(yhat,matrix(rep(NA,dim(dta)[2]*last),nrow=last))
    colnames(yhat) <- model_vars
    y.fcst <- get_y(dta, last)
    colnames(y.fcst) <- model_vars

    start_point <- time(ts.org)[1]

  }else{

    if(tvar == FALSE){
      predict.var <- predict(var.est, n.ahead = last, dumvar=cbind(dummy_est=dummy))
    }
  }
}

```



```

    yhat.fcst <- get_fcst(predict.var)
  }else{
    yhat.fcst <- predict(var.est, n.ahead=last)
  }
  y.fcst <- get_y(dta, last)
  yhat <- rbind(fitted(var.est),matrix(rep(NA,dim(dta)[2]*last),nrow=last))
  start_point <- time(ts.model)[1]
}

ts.yhat <- ts(yhat, start=c(start_point+lag/4),frequency = 4)
ts.y <- ts(dta[(lag+1):dim(dta)[1],],start=c(start_point+lag/4),frequency = 4)

ts.yhat.fcst <- ts(yhat.fcst, start=c(2017.5-last/4),frequency = 4)
ts.y.fcst <- ts(y.fcst, start=c(2017.5-last/4),frequency = 4)

# par(mfrow=c(dim(ts.y)[2],1),mar = rep(2, 4))
# for(i in colnames(ts.y)){
# plot(ts.y[,i],main=paste(paste(model_vars,collapse=","),"--",i," lag(",lag,"), last(",last,")
#   ↪ ")
# lines(ts.yhat[,i], col="red")
# abline(v=(2017.5-last/4), col="blue")
# lines(ts.yhat.fcst[,i], col="green",type="l", lwd=1)
# lines(ts.y.fcst[,i], col="blue",type="l", lwd=1)
# }

library(zoo)
library(ggplot2)

ts.y_ <- as.data.frame(ts.y)
ts.y_$time <- as.yearqtr(time(ts.y), format = "%Y-Q%q")
m.ts.y <- reshape2::melt(ts.y_ , "time")

ts.yhat_ <- as.data.frame(ts.yhat)
ts.yhat_$time <- as.yearqtr(time(ts.yhat), format = "%Y-Q%q")
m.ts.yhat <- reshape2::melt(ts.yhat_ , "time")

ts.yhat.fcst_ <- as.data.frame(ts.yhat.fcst)
ts.yhat.fcst_$time <- as.yearqtr(time(ts.yhat.fcst), format = "%Y-Q%q")
m.ts.yhat.fcst <- reshape2::melt(ts.yhat.fcst_ , "time")

ts.y.fcst_ <- as.data.frame(ts.y.fcst)
ts.y.fcst_$time <- as.yearqtr(time(ts.y.fcst), format = "%Y-Q%q")
m.ts.y.fcst <- reshape2::melt(ts.y.fcst_ , "time")

ggplot() +
  geom_line(data = m.ts.y, aes(x = time, y = value, group = variable),size=0.8, show.legend = F
  ↪ ) +
  geom_line(data = m.ts.yhat, aes(x = time, y = value, group = variable),size=0.8, color = "red
  ↪ ", show.legend = F) +
  geom_vline(xintercept = (2017.5-last/4), color = "steelblue") +
  geom_line(data = m.ts.yhat.fcst, aes(x = time, y = value, group = variable),size=0.8, color =
  ↪ "green3", show.legend = F) +
  geom_line(data = m.ts.y.fcst, aes(x = time, y = value, group = variable),size=0.8, color = "
  ↪ black", show.legend = F) +
  facet_wrap(~variable, scales = 'free', ncol = 1, switch = "y") +
  labs(x = 'Time', y = "", title = paste(paste(model_vars,collapse=","),"--","_lag(",lag,"),_

```

```

    ↪ last(",last,")")) +
theme_minimal() +
theme(plot.title = element_text(hjust = 0.5, size = 29, face = "bold"),
      strip.text.y = element_text(size=29),
      axis.text = element_text(size = 21)) +
scale_x_yearqtr()

}

#-----
#ADF test
#-----
df_test <- function(series) {
  library(urca)
  require(tseries)
  # /test static/ > critical = H0 rejected
  #phi3 null - no det trend, I(1)
  #phi2 null - no det trend, no drift, I(1)
  #phi1 null - no drift, I(1)
  #tau - I(1)
  df1 <- summary(ur.df(series, type = "trend", selectlags = "AIC"))
  stat <- df1@teststat[3]
  phi3 <- df1@cval[3, 2]
  if (abs(stat) > abs(phi3)) {
    #rejected - there is a trend
    stat <- df1@teststat[1]
    tau3 <- df1@cval[1, 2]
    if (abs(stat) > abs(tau3)) {
      #reject - there is no unit root
      res <- "I(0)_t"
      dif <- 0
      test_stat = "tau3"
      test_stat_val = stat
      test_stat = paste("tau3_=",tau3)
    } else{
      #accept - unit root
      stat <- df1@teststat[2]
      phi2 <- df1@cval[2, 2]
      if (abs(stat) > abs(phi2)) {
        res <- "I(1)_trend_+drift"
        dif <- 1
        test_stat = "phi2"
        test_stat_val = stat
        test_stat = paste("phi2_=",phi2)
      } else{
        res <- "I(1)"
        dif <- 1
        test_stat = "phi2"
        test_stat_val = stat
        test_stat = paste("phi2_=",phi2)
      }
    }
  }
  model = "model3"
} else{
  #accept - no trend
  df2 <- summary(ur.df(series, type = "drift", selectlags = "AIC"))
  stat <- df2@teststat[2]
  phi1 <- df2@cval[2, 2]
  if (abs(stat) > abs(phi1)) {

```

```

stat <- df2@teststat[1]
tau2 <- df2@cval[1, 2]
if (abs(stat) > abs(tau2)) {
  res <- "I(0)"
  dif <- 0
  test_stat = paste("tau2_=", tau2)
  test_stat_val = stat
} else{
  res <- "I(1)_drift"
  dif <- 1
  test_stat = paste("tau2_=", tau2)
  test_stat_val = stat
}
} else{
  res <- "I(1)"
  dif <- 1
  test_stat = paste("phi1_=", phi1)
  test_stat_val = stat
}
model = "model2"
}
return(list(res=res, dif=dif, model= model, test_stat=test_stat, test_stat_val=test_stat_val))
}
#-----
#Plot ACF
#-----
plot_acf <- function(series, name){
  par(mfrow=c(3,3), mar= c(2,2,2,2))
  plot(series, main = name)
  acf(series)
  pacf(series)
  plot(diff(series), main = paste("Prvni_diference_=", name))
  acf(diff(series))
  pacf(diff(series))
  plot(diff(series), main = paste("Druhe_diference_=", name))
  acf(diff(diff(series)))
  pacf(diff(diff(series)))
}

```